Click Here

Let's start adding style. There are hundreds and hundreds of different properties at our disposal to change the look and feel of our web pages. In this section we'll start at the beginning and introduce some of the more basic properties to get you started. From there, future sections will build upon this. Before we Dive in I don't believe it would be productive for us to go through every single property available to you. Instead, what I am going to do is introduce you to some of the more frequently used properties to give you a feel for how they work. Once you are comfortable with them, picking up new ones is not that difficult. We'll start with CSS properties at CSS Level 1. CSS Levels 2 and 3 mostly just introduce new properties (some of them are a bit tricky to understand the intricacies of but as your skill with CSS grow they won't be that scary). The official CSS Level 1 standard may be found here: If you scroll down a bit on the document you will see they have provided a nice table of contents. Properties start from item 5. Most of the properties are named so that you can easily understand what they do. This makes it easy to find the properties you need. If you click on a property it will give you the full outline of it. This includes: Values - what values the property accepts. A description - outlining what it does and any interesting aspects of its behaviour. Sample code - to give you an idea of how it is implemented. There are other things that will be listed but these will be the most useful to you. I wouldn't try to remember all the properties but instead keep this document open when you are writing your code so you can easily refer back to it. Some sample HTML In the rest of this section we'll demonstrate CSS properties. We will apply them all to the following HTML. our_page.html Hobbies Hobbies On the weekend I got all my watches and attached them together into a belt. It turned out to be a waist of time. Here is a small sample of things you can change with text. Font Family The font-family property allows us to change the particular font we are using. You may select any font which is installed on the clients computer with this property. Some of the more commonly used fonts include: Times New Roman Arial Verdana It is possible to specify and include non standard fonts but this is a little more complex. We'll look at this later. style.css h1 { font-family: Times New Roman; } p { font-family: Verdana; } Our Funky HTML Page On the weekend I got all my watches and attached them together into a belt. It turned out to be a waist of time. Font-size may be specified using a few different types but the easiest to work with is pixels (px). style.css h1 { font-size: 30px; } p { font-size: 16px; } Our Funky HTML Page On the weekend I got all my watches and attached them together into a belt. It turned out to be a waist of time. The font-weight css property allows us to specify how thick the lines of the characters are. It may be one of the following values: lighter normal bold bolder Some font families only have normal and bold weights. In this case if you specify lighter it will default back to normal and if you specify bolder it will default back to bold. style.css h1 { font-weight: normal; } p { font-weight: bold; } Our Funky HTML Page On the weekend I got all my watches and attached them together into a belt. It turned out to be a waist of time. Colour is a reasonably complex thing to define. There are several ways in which we can do this. We will look at colour in much more detail in a later section. Here we will just work with it in it's simplest form which is with Hexadecimal numbers. A colour is specified by a hash (#) followed by 6 hexadeximal numbers. The first 2 are the amount of red, the second 2 the amount of green and the final 2 the amount of blue. Note also that in CSS we use the American spelling for colour so we drop the u, eg color. If you happen to be American then this won't be a problem but for others it is something to keep in mind. Colour The color property allows us to specify the colour of the text. style.css h1 { color: #9ACA42; } p { color: #357180; } Our Funky HTML Page On the weekend I got all my watches and attached them together into a belt. It turned out to be a waist of time. The background-color property allows us to specify the background colour for the element. style.css h1 { background-color: #9ACA42; } p { background-color: #A0D2DC; } Our Funky HTML Page On the weekend I got all my watches and attached them together into a belt. It turned out to be a waist of time. Picking good colours for your content can have a huge impact on how it looks. Spacing Properties Spacing is another area which is easy to implement but hard to master. There are a lot of subtle little details to understand once we start creating complex layouts. We'll cover the topic in more detail when we look at the box model in a later section. Don't let this deter you from playing about and experimenting with the properties below, however, in the mean time. Similar to font-size, they may be specified in a variety of units but we'll stick to just pixels for now to keep it simple. Padding padding refers to the space around out content. We've included the background-color property here as well just to make it a bit easier to see what is happening. We may specify the padding in 3 methods: A single value - in which case it will be applied to every side. As two values - in which case the first unit specifies the top and bottom padding and the second unit the left and right padding. As four values - in which case the first applies to the top, the second to the right, the third to the bottom and the fourth to the left. style.css h1 { padding: 5px; background-color: #9ACA42; } p { padding: 5px 10px 8px 30px; background-color: #A0D2DC; } Our Funky HTML Page On the weekend I got all my watches and attached them together into a belt. It turned out to be a waist of time. margin is similar to padding in that is affects the spacing around the content. It is a second area outside the padding however. Compare the example below to the one above for padding and note where the background colour ends. Similar to padding, you may specify a single value, or two or four. style.css h1 { margin: 10px; background-color: #9ACA42; } p { margin: 5px 10px 8px 30px; background-color: #A0D2DC; } Our Funky HTML Page On the weekend I got all my watches and attached them together into a belt. It turned out to be a waist of time. The border is inbetween the padding and margin. I'll outline basic usage here but there are various properties available to control the look of the border with greater detail so I would recommend having a look at the Border Properties in the CSS specification. The various aspects of the border which you may control are: It's width. It's colour. It's style (none, dotted, dashed, solid, double, groove, ridge, inset, outset). style.css h1 { border: 2px solid #E9B82B; } p { border: 1px dashed #4D8353; } Our Funky HTML Page On the weekend I got all my watches and attached them together into a belt. It turned out to be a waist of time. Now let's put all the above properties together and see what effect it has. They may have seemed a little boring when looked at individually but as you'll see in the example below, even with just simple tweaks, we can really make our content shine. style.css h1 { font-family: Arial; font-size: 24px; font-weight: normal; color: #efefef; background-color: #9ACA42; margin: 2px 0px 20px 0px; padding: 5px; border-bottom: 2px solid #4D8353; } p { margin: 10px 0px 10px 30px; color: #357180; font-size: 16px; } Our Funky HTML Page On the weekend I got all my watches and attached them together into a belt. It turned out to be a waist of time. font-family Set the font for the given item. font-size Set the font for the given item. font-weight Set the weight for the font (eg, bold or normal). color Set the font colour for the given item. background-color Set the background colour for the given item. padding Set a padding around the given item. margin Set a margin around the given item. border Set a border around the given item. I know we haven't looked at many rules yet so for these activities use some of the rules we have seen in the examples. As you change the code in your files you may need to refresh the page to see the changes (rather than just reloading it), this is because the browser will cache the page by default. Take your page and link it to an external style sheet. Now play about with some of the properties we have introduced above. Visit the specification for CSS Level 1 and experiment with some of the other properties you find there. A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Properties are what we apply to elements when they are selected, such as a border, background color, or font, among many others that youll find on this page. The accent-color property in CSS is capable of re-tinting the accented color of form controls provided by the browsers default [].element { accent-color: #f8a100; }Continue ReadingThealign-contentproperty is a sub-property of theFlexible Box Layout module. It helps to align a flex containers lines within it when [].element { align-content: space-around; }Continue ReadingThe align-items property is related to CSS layout. It effects how elements are aligned both in Flexbox and Grid layouts. [].element { align-items: flex-start; }Continue ReadingThe align-self property is a sub-property of the Flexible Box Layout module. It makes possible to override the align-items value [].box { align-self: flex-end; }Continue ReadingThe all property in CSS resets all of the selected elements properties, except the direction and unicode-bidi properties that control []button { all: unset; }Continue Reading Declaring the CSS anchor-name property on an element registers it as an anchor that we can use to position as [].anchor { anchor-name: --my-anchor; }Continue Reading The anchor-scope is a property included in the CSS Anchor Positioning module that scopes an anchor element to a particular [].element { anchor-scope: --anchor-one, --anchor-two; }Continue ReadingThe animation property in CSS can be used to animate many other CSS properties such as color, background-color, height, or [].element { animation: pulse 5s infinite; }Continue Reading The animation-timeline CSS property controls the progress of animations using timelines beyond the traditional document timeline. Typically, CSS animations progress [].element { animation-timeline: --imageTimeline; }Continue ReadingThe appearance property is used to display an element using a platform-native styling based on the users operating systems theme. []input { appearance: none; }Continue ReadingThe CSS property aspect-ratio lets you create boxes that maintain proportional dimensions where the height and width of a box [].element { aspect-ratio: 2 / 1; }Continue ReadingThe backdrop-filter property in CSS is used to apply filter effects (grayscale, contrast, blur, etc) to the background/backdrop of an [].element { backdrop-filter: blur(10px); }Continue ReadingThe backface-visibility property relates to 3D transforms. With 3D transforms, you can manage to rotate an element so what we [].element { backface-visibility: hidden; }Continue ReadingThe background property in CSS allows you to control the background of any element (what paints underneath the content in [].element { background: url(texture.svg) top center / 200px 200px no-repeat fixed #f8a100; }Continue ReadingThe background-attachment property in CSS specifies how to move the background relative to the viewport. There are three values: scroll, [].hero { background-attachment: fixed; }Continue ReadingThe background-blend-mode property defines how an elements background-image should blend with its background-color: In the demo above, the default background-image [].element { background-blend-mode: screen; }Continue Readingbackground-clip lets you control how far a background image or color extends beyond an elements padding or content. Values Demos [].element { background-clip: padding-box; }Continue ReadingThe background-color property in CSS applies solid colors as background on an element. Heres an example: The example used above [].element { background-color: #ff7a18; }Continue ReadingThe background-image property in CSS applies a graphic (e.g. PNG, SVG, JPG, GIF, WEBP) or gradient to the background of [].element { background: url(texture.svg); }Continue ReadingThe background-origin property defines where to paint the background: across the whole element, inside the border, or inside the padding. [].element { background-origin: border-box; }Continue ReadingThe background-position property in CSS allows you to move a background image (or gradient) around within its container. It has [].element { background-position: 100px 5px; }Continue ReadingIf a background-image property is specified, the background-repeat property in CSS defines if (and how) it will repeat. Values The [].element { background-repeat: repeat-x; }Continue ReadingThe background-size property in CSS is one of the most useful and most complex of the background properties. [].element { background-size: 300px 100px; }Continue ReadingThe bleed property in CSS is for specifying space outside of the page box boundary when determining the size of []@page :top { bleed: 5cm; }Continue ReadingThe block-overflow property truncates text and indicates more content follows by inserting an ellipsis or custom string after a number [].element { block-overflow: clip; }Continue Readingblock-size is a CSS logical property that defines the height of an element when the writing-mode is horizontal, or the [].element { block-size: 700px; }Continue ReadingThe border property is a shorthand syntax in CSS that accepts multiple values for drawing a line around the element [].element { border: 3px solid #f8a100; }Continue Readingborder-block is a CSS logical shorthand property that combines border-block-color, border-block-style, and border-block-width into a single declaration. border-block is the [].element { border-block: 5px solid #f8a100; }Continue ReadingThe border-boundary property in CSS set constraints to the bounds of an element that affects how the elements borders behave. [].element { border-boundary: display; }Continue ReadingTheborder-collapseproperty is for use onelements (or elements made to behave like a table throughdisplay: tableordisplay: inline-table)..element { border-collapse: separate; }Continue Readingborder-image is a shorthand property that lets you use an image or CSS gradient as the border of an element. [].element { border-image: url(border.png) 25 25 round; }Continue ReadingThe CSSborder-image-outsetproperty sets distance between an elementsborder-imagearea and how far it goes beyond the elementsborder-box..element { border-image-outset: 2; }Continue ReadingThe CSS border-image-repeat property controls how a border image is repeated to fill the outer edges of an element. It [].element { border-image-repeat: repeat; }Continue ReadingThe CSS border-image-slice tells the browser where to slice an image being used to create the pieces of an elements [].element { border-image-slice: 30%; }Continue ReadingThe border-image-source CSS property creates borders around elements using an image file or CSS gradient as the source..element { border-image-source: url('path/to/image.webp'); }Continue ReadingThe CSS border-image-width property is used to determine the width of the borders image file, as provided by the CSS [].element { border-image-width: 4rem; }Continue Readingborder-inline is a CSS logical shorthand property that combines border-inline-color, border-inline-style, and border-inline-width into a single declaration, styling an elements [].element { border-inline: 5px solid #f8a100; }Continue ReadingYou can give any element rounded corners by applying a border-radius through CSS. Youll only notice if there is a [].element { border-radius: 10px; }Continue ReadingTheborder-spacingproperty in CSS controls how far apart cells are in aelement (or an element that is made to be table-like [].element { border-spacing: 5px; }Continue ReadingThe bottom property in CSS goes hand in hand with positioning. By default, elements are static positioned in which the []Continue ReadingThe CSS box-decoration-break property determines what happens to a box when it wraps to multiple lines..element { box-decoration-break: clone; }Continue ReadingThe box-shadow property in CSS is for putting shadows on elements (sometimes referred to as drop shadows, ala Photoshop/Figma). That [].element { box-shadow: 0 3px 10px rgb(0 0 0 / 0.2); }Continue ReadingThe box-sizing property in CSS controls how the box model is handled for the element it applies to. One of [].element { box-sizing: border-box; }Continue ReadingThe CSS break-after property is neat in that it allows, forces, or prevents breaks in paged media, multi-column layouts, and [].element { page-break: always; }Continue ReadingThe CSS break-before property is neat in that it allows, forces, or prevents breaks in paged media, multi-column layouts, and [].element { break-before: column; }Continue ReadingColumns do a great job of flowing and balancing content. Unfortunately, not all elements flow gracefully. Sometimes elements get stuck [].element { break-inside: avoid; }Continue ReadingThe caption-side property in CSS allows you to define where to position HTMLs element, used on HTML table headers. []table { caption-side: top; }Continue Readingcaret in CSS is a shorthand property that combines the caret-color and caret-shape properties into a single declaration. So, we [].element { caret: #ff7a18 underscore; }Continue ReadingThe caret-color property in CSS changes the color of the cursor (caret) in inputs, texareas, or really any element that [].element { caret-color: red; }Continue ReadingThe caret-shape property in CSS changes the shape of the text cursor inside editable elements that indicates a user is [].element { caret-shape: block; }Continue ReadingThe clear property is directly related to floats. If the element can fit horizontally in the space next to another [].element { clear: both; }Continue ReadingThe clip-path property in CSS allows you to specify a specific region of an element to display, with the rest [].element { clip: rect(110px, 160px, 170px, 60px); }Continue ReadingThe color property in CSS sets the color of text and text decorations. Values The color property can accept any [].element { color: #f8a100; }Continue ReadingYou know how we have dark mode and light mode these days? Browsers also have dark and light color schemes []:root { color-scheme: light dark; }Continue ReadingIf you need an exact numbers of columns when designing a multi-column layout, use column-count. Given the number of columns, [].people-list { column-count: 3; }Continue ReadingWhen you add height to a multi-column element, you can control how the content fills the columns. The content can [].element { column-fill: balance; }Continue ReadingThe CSS column-gap property sets space (also called gutters) between between columns in CSS Grid, Flexbox, and CSS Columns layouts..example { column-gap: 25px; }Continue ReadingTo make columns distinct, you can add a vertical line between each column. The line sits in the center of [].element { column-rule: 1px solid black; }Continue ReadingThe column-rule-color CSS property determines the color of a line between columns in a CSS multi-column layout. The property cant [].element { column-rule-color: #f8a100; }Continue ReadingThe column-rule-style CSS property specifies type of line thats drawn between columns in a CSS multi-column layout. The property is [].element{ column-rule-style: solid; }Continue ReadingThe column-rule-width CSS property sets the thickness of the line that is drawn between columns by column-rule-style in a CSS [].element{ column-rule-width: 3px; }Continue ReadingIn a multi-column layout, you can make elements expand across the columns with column-span. Assign column-span to an element inside [].element { column-span: all; }Continue ReadingWhen you want to keep your columns at a specific width, use column-width. The browser will calculate how many columns [].element { column-width: 200px; }Continue ReadingWith just a few CSS rules, you can create a print-inspired layout that has the flexibility of the web. Its [].element { columns: 300px 2; }Continue ReadingThe contain property in CSS indicates to the browser that the element and its descendants are considered independent of the [].element { contain: layout paint; }Continue ReadingThe CSS container property is a shorthand that combines the container-name and container-type properties into a single declaration. Constituent properties [].element { container: cards-grid / inline-size; }Continue ReadingThe CSS container-name property is used to register an element as a container that applies styles to other elements based [].element { container-name: cards-grid; }Continue ReadingThe CSS container-type property is part of the Container Queries feature used to register an element as a container that [].element { container-type: inline-size; }Continue ReadingThe content property in CSS is used in conjunction with the pseudo-elements ::before and ::after. It is used to literally []div::before { content: url(image.jpg); }Continue ReadingThe content-visibility property in CSS indicates to the browser whether or not an elements contents should be rendered at initial [].element { content-visibility: hidden; }Continue ReadingOrdered lists arent the only elements that can be automatically numbered. Thanks to the various counter-related properties, any element can [].step { counter-increment: my-awesome-counter; }Continue ReadingThe counter-reset property allows for automatic numbering of elements. Like an ordered list () element. The counter-reset property is used []article { counter-reset: section; }Continue ReadingThe counter-set CSS property, true to its name, sets the starting value for a CSS counter. You know how ordered []h2:first-of-type::before { counter-set: chapter; }Continue ReadingThe cursor property in CSS controls what the mouse cursor will look like when it is located over the element [].element { cursor: pointer; }Continue ReadingThe direction property in CSS sets the direction of of content flow within a block-level element. This applies to text, [].element { direction: rtl; }Continue ReadingEvery element on a web page is a rectangular box. The display property in CSS determines just how that rectangular [].element { display: inline-block; }Continue ReadingThe empty-cells property in CSS selects empty table cells for the purpose of specifying whether or not to display borders []table { empty-cells: show; }Continue Reading The CSS field-sizing property allows certain form controls such as inputs, select, and textareas to automatically grow as []input, select, textarea { field-sizing: content; }Continue ReadingThe fill property in CSS is for filling in the color of a SVG shape. .eyeball { fill: red; } [].element { fill: red; }Continue ReadingCSS Filters are a powerful tool that authors can use to achieve varying visual effects (sort of like Photoshop filters [].element { filter: blur(3px); }Continue ReadingThe flex property is a sub-property of the Flexible Box Layout module. This is the shorthand for flex-grow, flex-shrink and [].element { flex: 1 1 100px; }Continue ReadingThe flex-basis property is a sub-property of the Flexible Box Layout module. It specifies the initial size of the flex [].element { flex-basis: 100px; }Continue ReadingThe flex-direction property is a sub-property of the Flexible Box Layout module. It establishes the main-axis, thus defining the direction [].element { flex-direction: column-reverse; }Continue ReadingThe flex-flow property is a sub-property of the Flexible Box Layout module.It is a shorthand for flex-direction and flex-wrap. Syntax [].element { flex-flow: row wrap; }Continue ReadingThe flex-grow property is a sub-property of the Flexible Box Layout module. It defines the ability for a flex item [].flex-item { flex-grow: 2; }Continue ReadingThe flex-shrink property is a sub-property of the Flexible Box Layout module. It specifies the flex shrink factor which determines [].element { flex-shrink: 2; }Continue ReadingThe flex-wrap property is a sub-property of the Flexible Box Layout Module. It defines whether the flex items are forced [].example { flex-wrap: wrap; }Continue ReadingThe float property in CSS is used for positioning and layout on web pages. A common usage might be floating [].element { float: right; }Continue ReadingThe font property in CSS is a shorthand property that combines all the following sub-properties in a single declaration. body [].element { font: normal small-caps normal 16px/1.4 Georgia; }Continue ReadingThe font-display property defines how font files are loaded and displayed by the browser. It is applied to the @font-face []@font-face { font-display: fallback; }Continue ReadingThe font-family property defines the font that is applied to the selected element. The font that is selected is not [].element { font-family: Arial, Helvetica, sans-serif; }Continue ReadingThis property gives us control over advanced typographic settings such as small caps, ligatures, and swashes. To activate them you [].element { font-feature-settings: "liga" on; }Continue ReadingThe CSS font-kerning property defines the way specific pairs of letters are spaced. You see, fonts often contain information about [].text { font-kerning: none; }Continue ReadingThe font-optical-sizing CSS property allows the browser to adjust the outline of font glyphs to make them more legible at [].element { font-optical-sizing: none; }Continue ReadingThefont-paletteproperty allows you to pick the color palette of a multi-color font.element { font-palette: --pink; }Continue ReadingThe font-size property specifies the size, or height, of the font. font-size affects not only the font to which it []p { font-size: 1.5rem; }Continue ReadingThis property gives developers the ability to set the font-size of an element depending on the size of its lowercase [].element { font-size-adjust: .5; }Continue ReadingThe font-stretch property, introduced in CSS3, selects a normal, condensed or expanded face from a font. In order to use [].element { font-stretch: ultra-condensed; }Continue ReadingThe font-style property allows you to make text appear italicized (i.e. sloped, or slanted). This property accepts one of three [].element { font-style: italic; }Continue ReadingThe font-systhesis property in CSS gives the browser instructions for how to handle font bolded and italicized character when the [].intro { font-synthesis: weight style; }Continue ReadingThe font-variant property allows you to change the targeted text to small caps. This property has been extended in CSS3. []{ p:first-letter: font-variant: small-caps; }Continue ReadingThe CSSfont-variant-alternatesproperty lets you apply font features picked by the@font-feature-valuesat-rule, such as style sets, ornaments, and swashes..element { font-variant-alternates: styleset(pigtail); }Continue ReadingThefont-variant-emojiproperty let us define how emoji code points should be displayed, either as an emoji or text..element { font-variant-emoji: emoji; }Continue ReadingThe font-variant-numeric property in CSS supports the OpenType font format by specifying which numeric glyphs to use on a class, [].fraction { font-variant-numeric: diagonal-fractions; }Continue ReadingThe font-weight property sets the weight, or thickness, of a font and is dependent either on available font faces within []span { font-weight: 800; }Continue ReadingThe gap property in CSS is a shorthand for row-gap and column-gap, specifying the size of gutters, which is the [].element { gap: 20px 30px; }Continue ReadingThe grid CSS property is a shorthand that allows you to set all the implicit and the explicit grid properties [].element { grid: auto-flow dense / repeat(5, 150px); }Continue ReadingThegrid-areaCSS property is a shorthand that specifies the location and the size of a grid item in a grid layout [].grid-item { grid-area: 2 / 4 / 4 / 6; }Continue ReadingThe grid-auto-columns CSS property is part of the CSS Grid Layout specification, specifying the size of the grid columns that [].element { grid-auto-columns: minmax(100px, 200px); }Continue ReadingThe grid-auto-flow CSS property is part of the CSS Grid Layout specification that allows us to control the flow of [].element { grid-auto-flow: dense; }Continue ReadingThe grid-auto-rows CSS property is part of the CSS Grid Layout specification, specifying the size of the grid rows that [].element { grid-auto-rows: minmax(1fr, 500px) 3fr; }Continue ReadingThe grid-column CSS property is a shorthand that specifies the column grid lines where a grid item starts and ends [].element { grid-column: 3 / 5; }Continue ReadingThe grid-column-end CSS property is part of the CSS Grid Layout specification, used to indicate the column grid line where [].element { grid-column-end: 4; }Continue ReadingThe grid-column-start CSS property is part of the CSS Grid Layout specification, used to indicate the column grid line where [].element { grid-column-start: 3; }Continue ReadingThe grid-row CSS property is a shorthand that specifies the row grid lines where a grid item starts and ends [].element { grid-row: 2 / 4; }Continue ReadingThe grid-row and grid-column properties define which row or column an element will be displayed on. Heres an explicit 3 [].element { grid-column: 3 / -1; }Continue ReadingThe grid-row-end CSS property is part of the CSS Grid Layout specification, used to indicate the row grid line where [].element { grid-row-end: 3; }Continue ReadingThe grid-row-start CSS property is part of the CSS Grid Layout specification, used to indicate the row grid line where [].element { grid-row-start: 2; }Continue ReadingThe grid-template CSS property is a shorthand that allows you to define grid columns, rows, and areas in a CSS [].element { grid-template: 100px 1fr / 300px 1fr; }Continue ReadingThe grid-template-areas CSS property allows you to define and name the cells (or areas) in a CSS grid container. So, [].element { grid-template-areas: "header header" "sidebar main"; }Continue ReadingThe grid-template-columns CSS property is part of the CSS Grid Layout specification, defining the columns of a grid container by [].element { grid-template-columns: 300px 1fr; }Continue ReadingThe grid-template-rows CSS property is part of the CSS Grid Layout specification, defining the rows of a grid container by [].element { grid-template-rows: minmax(auto, 1fr) 3fr; }Continue ReadingThe hanging-punctuation property aims at giving web designers a finer grained control over typography on the web. The idea behind []blockquote p { hanging-punctuation: first; }Continue ReadingThe height property in CSS defines specifies the content height of boxes and accepts any of the length values. The [].element { height: 500px; }Continue ReadingThe CSS hyphenate-character property in CSS adds a hyphen or custom string at the end of a text line that wraps to the next line.p { hyphenate-character: "-"; }Continue ReadingThe hyphenate-limit-chars property in CSS tells the browser how many characters a word should be before its hyphenated, as well [].element { hyphenate-limit-chars: 10 3 4; }Continue ReadingThe hyphens property controls hyphenation of text in block level elements. You can prevent hyphenation from happening at all, allow [].element { hyphens: auto; }Continue ReadingThe image-rendering property defines how the browser should render an image if it is scaled up or down from its []img { image-rendering: pixelated; }Continue Readinginitial-letter is a CSS property that selects the first letter of the element where it is applied and specifies the [].element { initial-letter: 2; }Continue Readinginline-size is a logical property that defines the width of an element when the writing-mode is horizontal, or the height [].element { inline-size: 700px; }Continue ReadingThe inset property in CSS is a shorthand for the four inset properties, top, right, bottom and left in one [].box { inset: 10px 20px 30px 40px; position: relative; }Continue Readinginset-block is a shorthand logical CSS property that sets the length that an element is offset in the block direction [].element { inset-block: 50px 15%; }Continue Readinginset-block-end is a logical CSS property that sets the length that an element is offset in the block direction from [].element { inset-block-start: 50px; }Continue Readinginset-block-start is a logical CSS property that sets the length that an element is offset in the block direction from [].element {inset-block-start: 1.5rem; }Continue Readinginset-inline is a shorthand logical CSS property that sets the length that an element is offset in the inline direction [].element { inset-inline: 50px 15%; }Continue Readinginset-inline-end is a CSS property that sets the length that an element is offset in the starting inline direction. Its [].element { inset-inline-end: 50px; }Continue Readinginset-inline-start is a CSS property that sets the length that an element is offset in the starting inline direction. Its [].element { inset-inline-start: 80px; }Continue Reading The interpolate-size property in CSS allows us to transition an element from a certain length to its intrinsic size. Without []:root { interpolate-size: allow-keywords; }Continue ReadingThe isolation property in CSS is used to prevent elements from blending with their backdrops. It is most commonly used [].element { isolation: isolate; }Continue ReadingThe justify-content property is a sub-property of the Flexible Box Layout module. It defines the alignment along the main axis. [].element { justify-content: center; }Continue ReadingThe justify-items property is a sub-property of the CSS Box Alignment Module which basically controls the alignment of grid items [].element { justify-items: center; }Continue ReadingThe justify-self property in CSS sets the justification of an element within its containing block. Its behavior depends on the [].element { justify-self: stretch; }Continue ReadingThe left property in CSS goes hand in hand with positioning. By default, elements are static positioned in which the []Continue ReadingThe letter-spacing property controls the amount of space between each letter in a given element or block of text. Values [].element { letter-spacing: .0625em; }Continue ReadingThe CSS line-break property defines how strictly to enforce rules for wrapping text wrapping on new lines, particularly when working [].element { line-break: strict; }Continue ReadingThe line-clamp property truncates text at a specific number of lines. The spec for it is currently an Editors Draft, [].element { -webkit-line-clamp: 3; }Continue ReadingThe line-height property defines the amount of space above and below inline elements. That is, elements that are set to [].element { line-height: 1.35; }Continue ReadingThe list-style property is a shorthand property that sets values for three different list-related properties in one declaration: Heres an []ul { list-style: square outside none; }Continue ReadingThe margin property defines the outermost portion of the box model, creating space around an element, outside of any defined [].element { margin: 50px 2rem; }Continue Readingmargin-block is a shorthand property in CSS that sets an elements margin-block-start and margin-block-end values, both of which are logical [].element { margin-block: 30px 60px; }Continue ReadingThe margin-block-end property in CSS defines the amount of space along the outer ending edge of an element in the block direction. It's included in the CSS Logical Properties Level 1 specification, which is currently in Working Draft..element { margin-block-end: 25%; }Continue ReadingThe margin-block-start property in CSS defines the amount of space along the outer starting edge of an element in the block direction. Its included in the CSS Logical Properties Level 1 specification, which is currently in Working Draft..element { margin-block-start: 25%; }Continue Readingmargin-inline is a shorthand property in CSS that sets an elements margin-inline-start and margin-inline-end values, both of which are logical [].element { margin-inline: 60px auto; }Continue ReadingThe margin-inline-end property in CSS defines the amount of space along the outer ending edge of an element in the [].element { margin-inline-end: 3rem; }Continue ReadingThe margin-inline-start property in CSS defines the amount of space along the outer starting edge of an element in the [].element { margin-inline-start: 25%; }Continue ReadingThe margin-trim property in CSS removes the additional margin along the edge of a container when the last child element [].element { margin-trim: block-end; }Continue ReadingThe mask property is a shorthand to specify all mask-* properties. mask can hide part of the element is applied [].element { mask: url(mask.png) right bottom / 100px repeat-y; }Continue ReadingThe mask-border CSS property sets a border image to an elements masked border area. It is s shorthand for setting the mask-border-source, mask-border-slice, mask-border-width, mask-border-outset, mask-border-repeat, and mask-border-mode properties in a single declaration..element { mask-border: url('mask.png') 100 / 50px round; }Continue ReadingThe mask-clip CSS property is part of the CSS Masking Module Level 1 specification, and sets the mask painting area. [].element { mask-clip: padding-box; }Continue ReadingThe mask-composite CSS property allows us to combine a mask layer image with the mask layers below it. When there [].element { mask-composite: subtract; }Continue ReadingThe mask property in CSS allows you to hide parts of an element. For example, if you have a black and white image, you can apply that as a mask and the black parts will force the element to be transparent on that elements. Masks can also match the alpha transparency of the mask image. .element { mask-image: url(star.svg); }Continue ReadingThe mode CSS property indicates whether the CSS mask layer image is treated as an alpha mask or a luminance [].element { mask-mode: alpha; }Continue ReadingThe mask-origin specifies the mask position area of a mask layer image. In other words, it defines where the origin [].element { mask-origin: content-box; }Continue ReadingIn CSS, the mask-position property specifies the initial position of a mask layer image relative to the mask position area. [].element { mask-position: 20px center; }Continue ReadingIn CSS, the mask-repeat property tells a mask if it should repeated or not and in which directions. Its a [].element { mask-repeat: repeat-y; }Continue ReadingIn CSS, the mask-size property specifies the size of a mask layer image. In many ways, it works very much [].element { mask-size: 200px 100px; }Continue ReadingThe mask-type CSS property indicates whether the SVG element is treated as an alpha mask or a luminance mask. [].element { mask-type: alpha; }Continue ReadingThe max-height property in CSS is used to set the maximum height of a specified element. Authors may use any [].element { max-height: 3rem; }Continue Readingmax-inline-size is a logical property in CSS that defines the maximum width of an element when the writing-mode is horizontal, [].element { max-inline-size: 500px; }Continue ReadingThe max-width property in CSS is used to set the maximum width of a specified element. The max-width property overrides [].element { max-width: 100%; }Continue ReadingThe min-height property in CSS is used to set the minimum height of a specified element. The min-height property always [].element { min-height: 100vh; }Continue ReadingThe min-width property in CSS is used to set the minimum width of a specified element. The min-width property always [].element { min-width: 300px; }Continue ReadingThe mix-blend-mode property defines how an elements content should blend with its background. For example, the text of a [].blend { mix-blend-mode: exclusion; }Continue ReadingThe object-fit property defines how an element responds to the height and width of its content box. Its intended for [].cover { object-fit: cover; }Continue ReadingThe CSS property object-position is used to align any selected replaced element (e.g., an) within the box that contains []img { object-position: right bottom; }Continue ReadingThe offset-anchor property defines a point within the box it is applied as the anchor that moves along the offset-path. [].element { offset-anchor: right top; }Continue ReadingThis property began life as motion-offset. This, and all other related motion-* properties, are being renamed offset-* in the spec. [].element { offset-distance: 50%; }Continue ReadingThis property began life as motion-path. This, and all other related motion-* properties, are being renamed offset-* in the spec. [].element { offset-path: path("M 5 5 m -4, 0 a 4,4 0 1,0 8,0 a 4,4 0 1,0 -8,0"); }Continue ReadingThis property began life as motion-rotation, then it became offset-rotation, now its offset-rotate. Its still a Working Draft experimental property [].element { offset-rotate: 30deg; }Continue ReadingThe opacity property in CSS specifies how transparent an element is. Basic use: Opacity has a default initial value of [].element { opacity: 0.5; }Continue ReadingThe order property is a sub-property of the Flexible Box Layout module.Flex items are displayed in the same order as [].elememt { order: 3; }Continue ReadingIn typography terms, an orphan is the first line of a paragraph that is left behind on the old page [].element { orphans: 3; }Continue ReadingThe outline property in CSS draws a line around the outside of an element. Its similar to border except that: [].element { outline: 1px dashed #f8a100; }Continue ReadingThe outline-color CSS property specifies the color of an elements outline. Whats an outline? An outline is a line that [].element { outline-color: #f8a100; }Continue ReadingThe outline-offset property in CSS offsets a defined outline from an elements border edge by a specified amount. An outline, [].element { outline-offset: 10px; }Continue ReadingThe outline-style CSS property specifies the style of an elements outline. Whats an outline? An outline is a line that []button { outline-style: dashed; }Continue ReadingThe outline-width CSS property specifies the thickness of an elements outline. Whats an outline? An outline is a line that [].element { outline-width: 5px; }Continue ReadingThe overflow property controls what happens to content that breaks outside of its bounds: imagine a div in which youve [].element { overflow: hidden; }Continue ReadingThe overflow-anchor property enables us to opt out of Scroll Anchoring, which is a browser feature intended to allow content [].element { overflow-anchor: none; }Continue ReadingThe CSS overflow-clip-margin property determines how far the overflow of an element can go beyond the elements box before being [].element { overflow-clip-margin: 20px; }Continue ReadingThe overflow-wrap property in CSS allows you to specify that the browser can break a line of text inside the [].element { overflow-wrap: break-word; }Continue Reading When an element is set in the top layer (e.g. a dialog or an element using the Popover API), the []dialog {transition: opacity 500ms ease-out, display 500ms allow-discrete, overlay 500ms allow-discrete;}Continue ReadingThe overscroll-behavior CSS property controls whether an element will use scroll chaining or not. You have likely experienced this behavior [].element { overscroll-behavior: contain; }Continue ReadingThe padding property in CSS defines the innermost portion of the box model, creating space around an elements content, inside [].element { padding: 35px auto; }Continue Readingpadding-block is a CSS logical shorthand property that combines the padding-block-start and padding-block-end properties into a single declaration, creating space [].element { padding-block: 30px 60px; }Continue Readingpadding-inline is a CSS logical shorthand property that combines the padding-inline-start and padding-inline-end properties into a single declaration, creating space [].element { padding-inline: 30px 60px; }Continue ReadingThere isnt an actual page-break property in CSS. It is actually a set of 3 properties: page-break-before, page-break-after and page-break-inside. [].element { page-break-before: always; }Continue ReadingThe CSS paint-order property sets the order that SVG shapes and text are drawn, including the fill, stroke, and any [].element { paint-order: stroke; }Continue ReadingThe perspective CSS property gives an element a 3D-space by affecting the distance between the Z plane and the user. [].element { perspective: 1000px; }Continue ReadingThe perspective-origin property determines the origin for the perspective property. Think of it as the vanishing point of the current [].element { perspective-origin: 25% 75%; }Continue ReadingThe place-content property in CSS is shorthand for the align-content and justify-content properties, combining them into a single declaration in [].element { place-content: start space-evenly; }Continue ReadingThe place-items property in CSS is shorthand for the align-items and justify-items properties, combining them into a single declaration. A [].element { place-items: center; }Continue ReadingThe place-self property in CSS is shorthand for the align-self and justify-self properties, combining them into a single declaration in [].element { place-self: center start; }Continue ReadingThe pointer-events property allows for control over how HTML elements respond to mouse/touch events including CSS hover/active states, click/tap []button > .icon { pointer-events: none; }Continue ReadingThe position property can help you manipulate the location of an element, for example: Relative to its original position the [].header { position: fixed; }Continue Reading The position-anchor property links an absolutely positioned element to an anchor element. This will define the default anchor and is [].target { position-anchor: --my-anchor; }Continue ReadingThe position-area property creates an imaginary 33 grid surrounding an anchor element by declaring rows and columns and uses it [].target { position-area: bottom end; }Continue Reading The CSS position-try-fallbacks property accepts a list of different options for a target element to position itself relative to an [].target { position-try-fallbacks: flip-inline, bottom left; }Continue Reading The CSS position-try-order property sets the order of positions that a target element tries to use to display itself around [].element { position-try-order: most-width; }Continue Reading The position-visibility property provides certain conditions to hide an anchor-positioned element from the viewport. The position-visibility property wont have any [].target { position-visibility: no-overflow; }Continue ReadingThe print-color-adjust CSS property is sort of like a cost-cutting measure when it comes to printing a webpage. You know [].element { print-color-adjust: economy; }Continue ReadingThe quotes property in CSS allows you to specify which types of quotes are used when quotes are added via []q { quotes: "" "" "" ""; }Continue ReadingThe resize property controls if and how an element can be resized by the user by clicking and dragging the [].element { resize: both; }Continue ReadingThe right property in CSS goes hand in hand with positioning. By default, elements are static positioned in which the []Continue ReadingThe rotate property in CSS turns an element around one or more axes. Think of it like poking one or [].element { rotate: 45deg; }Continue ReadingThe row-gap property in CSS sets space (formally called gutters) between rows in CSS Grid, Flexbox, and CSS Columns layouts..element { row-gap: 2rem; }Continue ReadingThe scale property in CSS resizes an elements width and height in proportion. So, if we have an element thats [].element { scale: 2; }Continue ReadingThe scroll-behavior property in CSS allows us to define whether the scroll location of the browser jumps to a new []html { scroll-behavior: smooth; }Continue Readingscroll-margin is part of the CSS Scroll Snap Module. Scroll snapping refers to locking the position of the viewport to [].scroll-element { scroll-margin: 50px 0 0 50px; }Continue Reading Thescroll-marker-groupCSS property determines if the::scroll-marker-grouppseudo-element is generated within the scroll container that the property is set on, and where..scroll-container { scroll-marker-group: before; }Continue Readingscroll-padding is part of the CSS Scroll Snap Module. Scroll snapping refers to locking the position of the viewport to [].scroll-element{ scroll-padding: 50px 0 0 50px; }Continue Readingscroll-snap-align is part of the CSS Scroll Snap Module. Scroll snapping refers to locking the position of the viewport to [].element { scroll-snap-align: start; }Continue Readingscroll-snap-stop is part of the CSS Scroll Snap Module. Scroll snapping refers to locking the position of the viewport to [].element { scroll-snap-stop: always; }Continue ReadingThe scroll-snap-type property is part of the CSS Scroll Snap Module. Scroll snapping refers to locking the position of the [].scroll-element { scroll-snap-type: y mandatory; }Continue ReadingA brief history of styling scrollbars: It used to be a thing only Internet Explorer could do (ancient versions) with []body::-webkit-scrollbar { width: 10px; }Continue ReadingThe scrollbar-color property controls the two colors of a scrollbar: the thumb color and the track color. scrollbar-color is part [].element { scrollbar-color: red yellow; }Continue ReadingThe scrollbar-gutter property in CSS provides flexibility to determine how the space the browser uses to display a scrollbar that [].scroll-element { scrollbar-gutter: stable both-edges; }Continue ReadingThescrollbar-width property in CSS controls the width or thickness of a scrollbar. scrollbar-width is part of the CSS Working Groups [].scroll-element { scrollbar-width: thin; }Continue ReadingThe CSS shape-image-threshold property sets which pixels are included in the shape of an image when shape-outside is used to [].logo { shape-image-threshold: .6; }Continue ReadingThe CSS shape-margin property adds margin to CSS shapes that are created with the shape-outside property. It acts sorta like [].element { shape-margin: calc(100% - 2vw); }Continue ReadingThe shape-outside property controls how content will wrap around a floated elements bounding-box. Typically this is so that text can [].element { shape-outside: circle(50%); }Continue ReadingThe speak property in CSS is for specifying if a browser should speak the content it reads, such as through [].element { speak: never; }Continue ReadingThe stroke property in CSS is for adding a border to SVG shapes. Remember: Values The stroke property can accept [].module { stroke: black;}Continue ReadingThe stroke-dasharray property in CSS sets the length of dashes in the stroke of SVG shapes. More specifically, it sets [].element { stroke-dasharray: 5 10; }Continue ReadingThe stroke-dashoffset property in CSS defines the location along an SVG path where the dash of a stroke will begin. [].element { stroke-dashoffset: 100; }Continue ReadingThe stroke-linecap property in CSS is for setting the starting and ending points of a border on SVG shapes. Remember: [].element { stroke-linecap: round; }Continue Readingstroke-linejoin is a CSS property that defines how SVG paths should appear at the point where two lines are joined [].element { stroke-linejoin: miter; }Continue ReadingThe stroke-width property in CSS is for setting the width of a border on SVG shapes. Remember: Values The stroke-width [].element { stroke-width: 2; }Continue ReadingThe tab-size property in CSS is used to adjust the amount of spaces that display for the tab character. Have [].element { tab-size: 2; }Continue ReadingThe table-layout property defines what algorithm the browser should use to lay out table rows, cells, and columns.table { table-layout: fixed; }Continue ReadingThe text-align property in CSS is used for aligning the inner content of a block element. These are the traditional [].element { text-align: center; }Continue Readingtext-align-last lets you control the alignment of the last (or only) line of text right before a forced line break [].element { text-align-last: center; }Continue Reading A shorthand for thetext-box-trimandtext-box-edgeproperties, allowing us to choose where and how much whitespace is trimmed from text..element { text-box: trim-both cap alphabetic; }Continue Reading Thetext-box-edgeproperty let us trim the whitespace around text content. Its used alongside thetext-box-trimproperty to pick whether to remove whitespace above [].element { text-box-edge: cap alphabetic; }Continue Reading Thetext-box-trimproperty let us specify whether to trim the whitespace above or/and below text.element { text-box-trim: trim-both; }Continue ReadingThe text-combine-upright CSS property combines characters into the space of one character. The spec calls this horizontal-in-vertical composition which is []span { text-combine-upright: all; }Continue ReadingThe text-decoration property adds an underline, overline, line-through, or a combination of lines to selected text. Values The blink value [].element { text-decoration: underline; }Continue ReadingThe text-decoration-color property sets the color of the underline, overline, or line-through on text with the text-decoration property applied. It [].element { text-decoration-color: orange; }Continue ReadingThe text-decoration-line property decorates text with an underline, overline, line-through, or a combination of those lines. Values The value blink [].element { text-decoration-line: underline; }Continue ReadingThe text-decoration-skip property specifies where a text underline, overline, or strike-through should break. This improves legibility of decorated text and [].element { text-decoration-skip: ink; }Continue Readingtext-decoration-skip-ink is a property that controls the behavior of underline and overline (but not line-through) when the line passes through [].element {text-decoration-skip-ink: none; }Continue ReadingThe text-decoration-style property sets the style of the underline on links and the underline, overline, or line-through on any text [].element { text-decoration-style: wavy; }Continue ReadingThe text-decoration-thickness property in CSS sets the stroke thickness of the decoration line that is used on text in an [].element { text-decoration-thickness: 2px; }Continue ReadingThe text-emphasis property in CSS, as the name implies, is used to emphasize text in CSS. It does it by [].element { text-emphasis: circle red; }Continue ReadingThe text-indent property specifies how much horizontal space text should be moved before the beginning of the first line of [].element { text-indent: 1.5em; }Continue ReadingThe text-justify property in CSS is a companion to the text-align property that is used to set the justification method [].element { text-justify: inter-word; }Continue ReadingThe text-orientation property in CSS aligns text in a line when working with a vertical writing-mode. Basically, it rotates either []element { text-orientation: mixed; }Continue ReadingThe text-overflow property in CSS deals with situations where text is clipped when it overflows the elements box. It can [].element{ text-overflow: ellipsis; white-space: nowrap; overflow: hidden; }Continue ReadingThe text-rendering property in CSS allows you to choose quality of text over speed (or vice versa) allowing you to [].element { text-rendering: optimizeLegibility; }Continue ReadingYou can apply multiple text shadows by comma separating The first two values specify the length of the shadow offset. []p { text-shadow: 1px 1px 1px #000; }Continue Readingtext-stroke is an experimental property that provides text decoration options similar to those found in Adobe Illustrator or other vector [].element { text-stroke: 2px #f8a100; }Continue ReadingThe text-transform property in CSS controls text case and capitalization. Text-Transform Values The demo below shows lowercase, uppercase, and capitalize [].element { text-transform: lowercase; }Continue ReadingThe text-underline-offset property in CSS sets the distance of text underlines from their initial position. Once you apply an underline [].element { text-underline-offset: 0.5em; }Continue ReadingThe text-underline-position property sets the placement of the underline on links or on text with text-decoration: underline; applied. Values These [].element { text-underline-position: under; }Continue ReadingThetext-wrap-modeproperty lets us control whether text should wrap into new lines or stay (and possibly overflow) on the same line. []p { text-wrap-mode: nowrap; }Continue Reading Thetext-wrap-styleproperty lets us control how text lines should wrap.h1 { text-wrap-style: balance; }Continue ReadingThe top, bottom, left, and right properties are used with position to set the placement of an element. They only [].element { top: calc(100cqi / 2);)Continue ReadingThe touch-action property in CSS gives you control over the effect of touchscreen interactions with an element, similar to the [].element { touch-action: pan-right pinch-zoom; }Continue ReadingThe transform property allows you to visually manipulate an element by skewing, rotating, translating, or scaling:.element { transform: scale(2); }Continue ReadingThe transform-origin property is used in conjunction with CSS transforms, letting you change the point of origin of a transform. [].element { transform-origin: top left; }Continue ReadingThe transform-style property, when applied to an element, determines if that elements children are positioned in 3D space, or flattened. [].element { transform-style: preserve-3d; }Continue ReadingThe transition property is a shorthand property used to represent up to four transition-related longhand properties: Syntax Demo transition properties [].element { transition: background-color 0.5s ease; }Continue ReadingThe transition-behavior property allows us to make transitions in-between discrete properties, such as display or visibility.dialog { transition-behavior: allow-discrete; }Continue ReadingThe transition-delay property, normally used as part of transition shorthand, is used to define a length of time to delay []li:nth-child(3) { transition-delay: calc(var(--delay) * 3); }Continue ReadingThe transition-duration property, normally used as part of transition shorthand, is used to define the duration of a specified transition. [].element { transition-duration: 3s; }Continue ReadingThe transition-property property, normally used as part of transition shorthand, is used to define what property, or properties, you want [].element { transition-property: color; }Continue ReadingThe transition-timing-function property, normally used as part of transition shorthand, is used to define a function that describes how a [].element { transition-timing-function: ease-out; }Continue ReadingThe translate CSS property allows you to transfer an element from one place to another along the X (horizontal) axis, [].element { translate: 100px; }Continue ReadingThe unicode-bidi property is one of two CSS properties that handle the rendering of bi-directional text in HTML and similar [].element { unicode-bidi: embed; }Continue ReadingThe unicode-range property in CSS is used by the @font-face to define the characters that are supported by the font []@font-face { unicode-range: U+00-FF; }Continue ReadingThe user-select property in CSS controls how the text in an element is allowed to be selected. For example, it [].element { user-select: none; }Continue ReadingThe vertical-align property in CSS controls how elements set next to each other on a line are lined up. In []td { vertical-align: top; }Continue Reading The CSS view-timeline-name property allows us to identify and reference an element we want to animate based on its scroll [].element { view-timeline-name: --image-zoom; }Continue Reading The CSS view-transition-name property uniquely identifies an element that can be used in a view transition. We give the element [].element { view-transition-name: image-zoom; }Continue ReadingThevisibilityproperty in CSS has two different functions. It hides rows and columns of a table, and it also hides an [].element { visibility: hidden; }Continue ReadingThe CSS white-space property controls how text is handled on the element it is applied to. Lets say you have [].element { white-space: pre; }Continue ReadingIn typography terms, a widow is the last line of a paragraph that is left alone on a new page [].element { widows: 3; }Continue ReadingThe width property in CSS specifies the width of the elements content area. This content area is the portion inside [].elment { width: 50%; }Continue ReadingThe will-change property in CSS optimizes animations by letting the browser know which properties and elements are just about to [].element { will-change: transform; }Continue ReadingThe word-break property in CSS can be used to change when line breaks ought to occur. Normally, line breaks in [].element { word-break: break-all; }Continue ReadingThe word-spacing property is similar to letter-spacing, though naturally its use governs the amount of space between the words in [].element { word-spacing: .75em; }Continue ReadingThe writing-mode property changes the alignment of the text so that it can be read from top to bottom or [].element { writing-mode: vertical-rl; }Continue ReadingThe z-index property in CSS controls the vertical stacking order of elements that overlap. As in, which one appears as [].menu { z-index: 1; }Continue ReadingThe zoom property in CSS allows you to scale your content. It is non-standard, and was originally implemented only in [].element { zoom: 150%; }Continue Reading When the first CSS specification was published, all of CSS was contained in one document that defined CSS Level 1. CSS Level 2 was defined also by a single, multi-chapter document. However for CSS beyond Level 2, the CSS Working Group chose to adopt a modular approach, where each module defines a part of CSS, rather than to define a single monolithic specification. This breaks the specification into more manageable chunks and allows more immediate, incremental improvement to CSS. Since different CSS modules are at different levels of stability, the CSS Working Group has chosen to publish this profile to define the current scope and state of Cascading Style Sheets as of 2024. Cascading Style Sheets (CSS) CSS is a language for writing style sheets, and is designed to describe the rendering of structured documents (such as HTML and XML) on a variety of media. CSS is used to describe the presentation of a source document, and usually does not change the underlying semantics expressed by its document language. Style sheet A set of rules that specify the presentation of a document. Style sheets are written by an Author, and interpreted by a User Agent, to present the document to the User. Source document The document to which one or more style sheets apply. A source documents structure and semantics are encoded using a document language (e.g., HTML, XHTML, or SVG). Author An author is a person who writes documents and associated style sheets. An authoring tool is a User Agent that generates style sheets. User A user is a person who interacts with a user agent to view, hear, or otherwise use the document. User Agent (UA) A user agent is any program that interprets a document and its associated style sheets on behalf of a user. A user agent may display a document, read it aloud, cause it to be printed, convert it to another format, etc. For the purposes of the CSS specifications, a User Agent is one that supports and interprets Cascading Style Sheets as defined in these specifications. This section is non-normative. In the W3C Process, a Recommendation-track document passes through three levels of stability, summarized below: Working Draft (WD) This is the design phase of a W3C spec. The WG iterates the spec in response to internal and external feedback. The first official Working Draft is designated the First Public Working Draft (FPWD). In the CSSWG, publishing FPWD indicates that the Working Group as a whole has agreed to work on the module, roughly as scoped out and proposed in the editors draft. The transition to the next stage is sometimes called Last Call Working Draft (LCWD) phase. The CSSWG transitions Working Drafts once we have resolved all known issues, and can make no further progress without feedback from building tests and implementations. This Last Call for Comments sets a deadline for reporting any outstanding issues, and requires the WG to specially track and address incoming feedback. The comment-tracking document is the Disposition of Comments (DoC). It is submitted along with an updated draft for the Directors approval, to demonstrate wide review and acceptance. Candidate Recommendation (CR) This is the testing phase of a W3C spec. Notably, this phase is about using tests and implementations to test the specification: it is not about testing the implementations. This process often reveals more problems with the spec, and so a Candidate Recommendation will morph over time in response to implementation and testing feedback, though usually less so than during the design phase (WD). Demonstration of two correct, independent implementations of each feature is required to exit CR, so in this phase the WG builds a test suite and generates implementation reports. The transition to the next stage is Proposed Recommendation (PR). During this phase the W3C Advisory Committee must approve the transition to REC. Recommendation (REC) This is the completed state of a W3C spec and represents a maintenance phase. At this point the WG only maintains an errata document and occasionally publishes an updated edition that incorporates the errata back into the spec. An Editors Draft is effectively a live copy of the editors own working copy. It may or may not reflect Working Group consensus, and can at times be in a self-inconsistent state. (Because the publishing process at W3C is time-consuming and onerous, the Editors Draft is usually the best (most up-to-date) reference for a spec. Efforts are currently underway to reduce the friction of publishing, so that official drafts will be regularly up-to-date and Editors Drafts can return to their original function as scratch space.) A list of all CSS modules, stable and in-progress, and their statuses can be found at the CSS Current Work page. This profile includes only specifications that we consider stable and for which we have enough implementation experience that we are sure of that stability. Note: This is not intended to be a CSS Desktop Browser Profile: inclusion in this profile is based on feature stability only and not on expected use or Web browser adoption. This profile defines CSS in its most complete form. As of 2024, Cascading Style Sheets (CSS) is defined by the following specifications. CSS Level 2, latest revision (including errata) [CSS2] This defines the core of CSS, parts of which are overridden by later specifications. We recommend in particular reading Chapter 2, which introduces some of the basic concepts of CSS and its design principles. CSS Syntax Level 3 [CSS-SYNTAX-3] Replaces CSS24.1, CSS24.2, CSS24.4, and CSS2G, redefining how CSS is parsed. CSS Style Attributes [CSS-STYLE-ATTR] Defines how CSS declarations can be embedded in markup attributes. Media Queries Level 3 [CSS3-MEDIAQUERIES] Replaces CSS27.3 and expands on the syntax for media-specific styles. CSS Conditional Rules Level 3 [CSS-CONDITIONAL-3] Extends and supersedes CSS27.2, updating the definition of @media rules to allow nesting and introducing the @supports rule for feature-support queries. Selectors Level 3 [SELECTORS-3] Replaces CSS25 and CSS26.4.3, defining an extended range of selectors. CSS Namespaces [CSS3-NAMESPACE] Introduces an @namespace rule to allow namespace-prefixed selectors. CSS Cascading and Inheritance Level 4 [CSS-CASCADE-4] Extends and supersedes CSS21.4.3 and CSS26, as well as [CSS-CASCADE-3]. Describes how to collate style rules and assign values to all properties on all elements. By way of cascading and inheritance, values are propagated for all properties on all elements. CSS Values and Units Level 3 [CSS-VALUES-3] Extends and supersedes CSS21.4.2.1, CSS24.3, and CSS2A.2.13, defining CSSs property definition syntax and expanding its set of units. CSS Custom Properties for Cascading Variables Module Level 1 [CSS-VARIABLES-1] Introduces cascading variables as a new primitive value type that is accepted by all CSS properties, and custom properties for defining them. CSS Box Model Level 3 [CSS-BOX-3] Replaces CSS28.1, 8.2, 8.3 (but not 8.3.1), and 8.4. CSS Color Level 4 [CSS-COLOR-4] Extends and supersedes CSS24.3.6, CSS214.1, and CSS218.2, also extends and supersedes [CSS-COLOR-3], introducing an extended range of color spaces beyond sRGB, extended color values, and CSS Object Model extensions for color. Also defines the opacity property. CSS Backgrounds and Borders Level 3 [CSS-BACKGROUNDS-3] Extends and supersedes CSS28.5 and CSS214.2, providing more control of backgrounds and borders, including layered background images, image borders, and drop shadows. CSS Images Level 3 [CSS-IMAGES-3] Redefines and incorporates the external 2D image value type, introduces native 2D gradients, and adds additional controls for replaced element sizing and rendering. CSS Fonts Level 3 [CSS-FONTS-3] Extends and supersedes CSS215 and provides more control over font choice and feature selection. CSS Writing Modes Level 3 [CSS-WRITING-MODES-3] Defines CSS support for various international writing modes, such as left-to-right (e.g. Latin or Indic), right-to-left (e.g. Hebrew or Arabic), bidirectional (e.g. mixed Latin and Arabic) and vertical (e.g. Asian scripts). Replaces and extends CSS28.6 and 9.10. CSS Multi-column Layout Level 1 [CSS-MULTICOL-1] Introduces multi-column flows to CSS layout. CSS Flexible Box Module Level 1 [CSS-FLEXBOX-1] Introduces a flexible linear layout model for CSS. CSS Basic User Interface Module Level 3 [CSS-UI-3] Extends and supersedes CSS218.1 and CSS218.4, defining cursor, outline, and several new CSS features that also enhance the user interface. CSS Containment Module Level 1 [CSS-CONTAIN-1] Introduces the contain property, which enforces the independent CSS processing of an elements subtree in order to enable heavy optimizations by user agents when used well. CSS Transforms Level 1 [CSS-TRANSFORMS-1] Introduces coordinate-based graphical transformations to CSS. CSS Compositing and Blending Level 1 [COMPOSITING] Defines the compositing and blending of overlaid content and introduces features to control their modes. CSS Easing Functions Level 1 [CSS-EASING-1]. Describes a way for authors to define a transformation that controls the rate of change of some value. Applied to animations, such transformations can be used to produce animations that mimic physical phenomena such as momentum or to cause the animation to move in discrete steps producing robot-like movement. CSS Counter Styles Level 3 [CSS-COUNTER-STYLES-3] Introduces the @counter-style rule, which allows authors to define their own custom counter styles for use with CSS list-marker and generated-content counters [CSS-LISTS-3]. It also predefines a set of common counter styles, including the ones present in CSS2 and CSS2.1. Note: Although we dont anticipate significant changes to the specifications that form this snapshot, their inclusion does not mean they are frozen. The Working Group will continue to address problems as they are found in these specs. Implementers should monitor www-style and/or the CSS Working Group Blog for any resulting changes, corrections, or clarifications. The following specifications are considered to be in a reliable state, meaning they have largely stable implementations and specifications, but are not yet at the Recommendation level due to minor issues or the need for additional implementation reports. Media Queries Level 4 [MEDIAQUERIES-4] Extends and supersedes [CSS3-MEDIAQUERIES], expanding the syntax, deprecating most media types, and introducing new media features. CSS Scroll Snap Module Level 1 [CSS-SCROLL-SNAP-1] Contains features to control panning and scrolling behavior with snap positions. CSS Scrollbars Styling Module Level 1 [CSS-SCROLLBARS-1] Defines properties to influence the visual styling of scrollbars, introducing controls for their color and width. CSS Grid Layout Module Level 1 [CSS-GRID-1] Introduces a two-dimensional grid-based layout system, optimized for user interface design. In the grid layout model, the children of a grid container can be positioned into arbitrary slots in a predefined flexible or fixed-size layout grid. CSS Grid Layout Module Level 2 [CSS-GRID-2] Extends and supersedes [CSS-GRID-1], introducing subgrids for managing nested markup in a shared grid framework. The following modules have completed design work, and are fairly stable, but have not received much testing and implementation experience yet. We hope to incorporate them into the official definition of CSS in a future snapshot. Media Queries Level 4 [MEDIAQUERIES-4] Extends and supersedes [CSS3-MEDIAQUERIES], expanding the syntax, deprecating most media types, and introducing new media features. CSS Display Module Level 3 [CSS-DISPLAY-3] Replaces CSS29.1.2, 9.2.1 (but not 9.2.1.1), 9.2.2 (but not 9.2.2.1), 9.2.3, and 9.2.4 (and lays the foundations for replacing 9.7), defining how the CSS formatting box tree is generated from the document element tree and defining the display property that controls it. CSS Writing Modes Level 4 [CSS-WRITING-MODES-4] Extends and supersedes [CSS-WRITING-MODES-3], adding more options for vertical writing. CSS Fragmentation Module Level 3 [CSS-BREAK-3] Describes the fragmentation model that partitions a flow into pages, columns, or regions and defines properties that control it. Extends and supersedes CSS213.3. CSS Box Alignment Module Level 3 [CSS-ALIGN-3] Introduces properties to control the alignment of boxes within their containers in the various CSS box layout models: block layout, table layout, flex layout, and grid layout. CSS Shapes Module Level 1 [CSS-SHAPES-1] Extends floats (CSS29.5) to effect non-rectangular wrapping shapes. CSS Text Module Level 3 [CSS-TEXT-3] Extends and supersedes CSS216 excepting 16.3, defining properties for text manipulation and specifying their processing model. It covers line breaking, justification and alignment, white space handling, and text transformation. CSS Text Decoration Module Level 3 [CSS-TEXT-DECOR-3] Extends and supersedes CSS216.3, providing more control over text decoration lines and adding the ability to specify text emphasis marks and text shadows. CSS Masking Module Level 1 [CSS-MASKING-1] Replaces CSS211.1.2 and introduces more powerful ways of clipping and masking content. CSS Scroll Snap Module Level 1 [CSS-SCROLL-SNAP-1] Contains features to control panning and scrolling behavior with snap positions. CSS Speech Module Level 1 [CSS-SPEECH-1] Replaces CSS2A, overhauling the (non-normative) speech rendering chapter. CSS View Transitions Module Level 1 [CSS-VIEW-TRANSITIONS-1] Defines the View Transition API, along with associated properties and pseudo-elements, which allows developers to create animated visual transitions representing changes in the document state. Although the following modules have been widely deployed with rough interoperability, their details are not fully worked out or sufficiently well-specified and they need more testing and bugfixing. We hope to incorporate them into the official definition of CSS in a future snapshot. CSS Transitions Level 1 [CSS-TRANSITIONS-1] and CSS Animations Level 1 [CSS-ANIMATIONS-1]. Introduces mechanisms for transitioning the computed values of CSS properties over time. CSS Will Change Level 1 [CSS-WILL-CHANGE-1] Introduces a performance hint property called will-change. Filter Effects Module Level 1 [FILTER-EFFECTS-1] Introduces filter effects as a way of processing an elements rendering before it is displayed in the document. CSS Font Loading Module Level 3 [CSS-FONT-LOADING-3] Introduces events and interfaces used for dynamically loading font resources. CSS Box Sizing Level 3 [CSS-SIZING-3] Overlays and extends CSS10., expanding the value set of the sizing properties, introducing more precise sizing terminology, and defining with more precision and detail various automatic sizing concepts only vaguely defined in CSS2. CSS Transforms Level 2 [CSS-TRANSFORMS-2] Builds upon [CSS-TRANSFORMS-1] to add new transform functions and properties for three-dimensional transforms, and convenience functions for simple transforms. CSS Lists and Counters Module Level 3 [CSS-LISTS-3] Contains CSS features related to list counters: styling them, positioning them, and manipulating their value. CSS Logical Properties and Values Level 1 [CSS-LOGICAL-1] Introduces logical properties and values that provide the author with the ability to control layout through logical, rather than physical, direction and dimension mappings. Also defines logical properties and values for the features defined in [CSS2]. These properties are writing-mode relative equivalents of their corresponding physical properties. CSS Positioned Layout Module Level 3 [CSS-POSITION-3] Contains defines coordinate-based positioning and offsetting schemes of CSS: relative positioning, sticky positioning, absolute positioning, and fixed positioning. Resize Observer [RESIZE-OBSERVER-1] This specification describes an API for observing changes to elements principal boxs size. Web Animations [WEB-ANIMATIONS-1] Defines a model for synchronization and timing of changes to the presentation of a Web page. Also defines an application programming interface for interacting with this model. CSS Fonts Module Level 4 [CSS-FONTS-4] Extends and supersedes CSS Fonts 3 and provides more control over font choice and feature selection, including support for OpenType variations. CSS Color Adjustment Module Level 1 [CSS-COLOR-ADJUST-1] This module introduces a model and controls over automatic color adjustment by the user agent to handle user preferences and device output optimizations. CSS Conditional Rules Module Level 4 [CSS-CONDITIONAL-4] Extends CSS Conditional 3 to allow testing for supported selectors. CSS Cascading and Inheritance Level 5 [CSS-CASCADE-5] Extends CSS Cascade 4 to add cascade layers. Motion Path Module Level 1 [MOTION-1] This module allows authors to position any graphical object and animate it along an author specified path. CSS Scroll Anchoring Module Level 1 [CSS-SCROLL-ANCHORING-1] This module aims to minimize content shifts by locking the scroll position of a scroll container to a particular anchor element. CSS Object Model (CSSOM) [CSSOM-1] This module defines APIs for parsing, serializing, and manipulating CSS, Media Queries, and Selectors. CSS Color Module Level 5 [CSS-COLOR-5] Extends CSS Color 4 to add color spaces and color modification functions. Selectors Level 4 [SELECTORS-4] Extends Selectors Level 3 by introducing new pseudo-classes, pseudo-elements, and combinators, enhancing the ability to select elements based on more complex criteria and states. Cascading Style Sheets does not have versions in the traditional sense; instead it has levels. Each level of CSS builds on the previous, refining definitions and adding features. The feature set of each higher level is a superset of any lower level, and the behavior allowed for a given feature in a higher level is a subset of that allowed in the lower levels. A user agent conforming to a higher level of CSS is thus also conformant to all lower levels. CSS Level 1 The CSS Working Group considers the CSS1 specification to be obsolete. CSS Level 1 is defined as all the features defined in the CSS1 specification (properties, values, at-rules, etc), but using the syntax and definitions in the CSS2.1 specification. CSS Style Attributes defines its inclusion in element-specific style attributes. CSS Level 2 Although the CSS2 specification is technically a W3C Recommendation, it passed into the Recommendation stage before the W3C had defined the Candidate Recommendation stage. Over time implementation experience and further review has brought to light many problems in the CSS2 specification, so instead of expanding an already unwieldy errata list, the CSS Working Group chose to define CSS Level 2 Revision 1 (CSS2.1). In case of any conflict between the two specs CSS2.1 contains the definitive definition. Once CSS2.1 became Candidate Recommendationeffectively though not officially the same level of stability as CSS2obsoleted the CSS2 Recommendation. Features in CSS2 that were dropped from CSS2.1 should be considered to be at the Candidate Recommendation stage, but note that many of these have been or will be pulled into a CSS Level 3 working draft, in which case that specification will, once it reaches CR, obsolete the definitions in CSS2. The CSS2.1 specification defines CSS Level 2 and the CSS Style Attributes specification defines its inclusion in element-specific style attributes. CSS Level 3 CSS Level 3 builds on CSS Level 2 module by module, using the CSS2.1 specification as its core. Each module adds functionality and/or replaces part of the CSS2.1 specification. The CSS Working Group intends that the new CSS modules will not contradict the CSS2.1 specification: only that they will add functionality and refine definitions. As each module is completed, it will be plugged in to the existing system of CSS2.1 plus previously-completed modules. From this level on modules are levelled independently: for example Selectors Level 4 may well be completed before CSS Line Module Level 3. Modules with no CSS Level 2 equivalent start at Level 1; modules that update features that existed in CSS Level 2 start at Level 3. CSS Level 4 and beyond There is no CSS Level 4. Independent modules can reach level 4 or beyond, but CSS the language no longer has levels. ("CSS Level 3" as a term is used only to differentiate it from the previous monolithic versions.) Not all implementations will implement all functionality defined in CSS. In the past, the Working Group published a few Profiles, which were meant to define the minimal subset of CSS that various classes of user agents were expected to support. This effort has been discontinued, as the Working Group was not finding it effective or useful, and the profiles previously defined are now unmaintained. Note: Partial implementations of CSS, even if that subset is an official profile, must follow the forward-compatible parsing rules for partial implementations. The following sections define several conformance requirements for implementing CSS responsibly, in a way that promotes interoperability in the present and future. So that authors can exploit the forward-compatible parsing rules to assign fallback values, CSS renderers must treat as invalid (and ignore as appropriate) any at-rules, properties, property values, keywords, and other syntactic constructs for which they have no usable level of support. In particular, user agents must not selectively ignore unsupported property values and honor supported values in a single multi-value property declaration: if any value is considered invalid (as unsupported values must be), CSS requires that the entire declaration be ignored. To avoid clashes with future stable CSS features, the CSSWG recommends the following best practices for the implementation of unstable features and proprietary extensions to CSS: Implementations of unstable features that are described in W3C specifications but are not interoperable should not be released broadly for general use; but may be released for limited, experimental use in controlled environments. Why? We want to allow both authors and implementors to experiment with the feature and give feedback, but prevent authors from relying on them in production websites and thereby accidentally "locking in" (through content dependence) certain syntax or behavior that might change later. For example, a UA could release an unstable features for experimentation through beta or other testing-stage builds; behind a hidden configuration flag; behind a switch enabled only for specific testing partners; or through some other means of limiting dependent use. A CSS feature is considered unstable until its specification has reached the Candidate Recommendation (CR) stage in the W3C process. In exceptional cases, the CSSWG may additionally, by an officially-recorded resolution, add pre-CR features to the set that are considered safe to release for broad use. See 4 Safe to Release pre-CR Exceptions. Note: Vendors should consult the WG explicitly and not make assumptions on this point, as a pre-CR spec that hasnt changed in awhile is usually more out-of-date than stable. To avoid clashes with future CSS features, the CSS2.1 specification reserves a prefixed syntax [CSS2] for proprietary and experimental extensions to CSS. A CSS feature is a proprietary extension if it is meant for use in a closed environment accessible only to a single vendors user agent(s). A UA should support such proprietary extensions only through a vendor-prefixed syntax and not expose them to open (multi-UA) environments such as the World Wide Web. Why? The prefixing requirement allows shipping specialized features in closed environments without conflicting with future additions to standard CSS. The restriction on exposure to open systems is to prevent accidentally causing the public CSS environment to depend on an unstandardized proprietary extensions. For example, Firefoxs XUL-based UI, Apples iTunes UI, and Microsofts Universal Windows Platform app use extensions to CSS implemented by their respective UAs. So long as these UAs do not allow Web content to access these features, they do not provide an opportunity for such content to become dependent on their proprietary extensions. Even if a feature is intended to eventually be used in the Web, if it hasnt yet been standardized it should still not be exposed to the Web. If a feature is unstable (i.e. the spec has not yet stabilized), but at least three UAs implement the feature(or a UA has broken the other rules and shipped for broad usean unstable or otherwise non-standard feature in a production release), and the implementations have rough interoperability, and the CSS Working Group has recorded consensusthat this feature should exist and be released, implementers may ship that feature unprefixed in broad-release builds. Rough interoperability is satisfied by a subjective judgment that even though there may be differences, the implementations are sufficiently similar to be used in production websites for a substantial number of use cases. Note that the CSSWG must still be consulted to ensure coordination across vendors and to ensure coherency review by the CSS experts from each vendor. Note also that rough interoperability still usually means painful lack of interop in edge (or not-so-edge) cases, particularly because details have not been ironed out through the standards review process. Why? If a feature is sufficiently popular that three or more browsers have implemented it before its finished standardization, this clause allows releasing the pressure to ship. Also, if a feature has already escaped into the wild and sites have started depending on it, pretending its still experimental doesnt help anyone. Allowing others to ship unprefixed recognizes that the feature is now de facto standardized and encourages authors to write cross-platform code. When exposing such a standards-track unstable feature to the Web in a production release, implementations should support both vendor-prefixed and unprefixed syntaxes for the feature. Once the feature has stabilized and the implementation is updated to match interoperable behavior, support for the vendor-prefixed syntax should be removed. Why? This is recommended so that authors can use the unprefixed syntax to target all implementations, but when necessary, can target specific implementations to work around incompatibilities among implementations as they get ironed out through the standards/bugfixing process. The lack of a phase where only the prefixed syntax is supported greatly reduces the risk of stylesheets being written with only the vendor-prefixed syntax. This in turn allows UA vendors to retire their prefixed syntax once the feature is stable, with a lower risk of breaking existing content. It also reduces the need occasionally felt by some vendors to support a feature with the prefix of another vendor, due to content depending on that syntax. Anyone promoting unstable features to authors should document them using their standard unprefixed syntax, and avoid encouraging the use of the vendor-prefixed syntax for any purpose other than working around implementation differences. In order to preserve the open nature of CSS as a technology, vendors should make it possible for other implementors to freely implement any features that they do ship. To this end, they should provide spec-editing and testing resources to complete standardization of such features, and avoid other obstacles (e.g., platform dependency, licensing restrictions) to their competitors shipping the feature. Once a specification reaches the Candidate Recommendation stage, implementers should release an unprefixed implementation of any CR-level feature they can demonstrate to be correctly implemented according to spec, and should avoid exposing a prefixed variant of that feature. To establish and maintain the interoperability of CSS across implementations, the CSS Working Group requests that non-experimental CSS renderers submit an implementation report (and, if necessary, the testcases used for that implementation report) to the W3C before releasing an unprefixed implementation of any CSS features. Testcases submitted to W3C are subject to review and correction by the CSS Working Group. Further information on submitting testcases and implementation reports can be found from on the CSS Working Groups website at . Questions should be directed to the public-css-testsuite@w3.org mailing list. The following features have been explicitly and proactively cleared by the CSS Working Group for broad release prior to the spec reaching Candidate Recommendation. See 3.2.1 Experimentation and Unstable Features. The following features have been explicitly and retroactively cleared by the CSS Working Group for broad release prior to the spec reaching Candidate Recommendation: Everything in CSS Animations Level 1 and CSS Transitions Level 1. These sections are non-normative. = ~= 1st 2d matrix 2nd 3rd 4th absolute length absolute length unit absolutely positioned element abstract dimensions :active activeborder activecaption active duration active (pseudo-class) actual value in css-cascade-4 in css21 additive tuple adjoining margins advance measure :after after after-change style aliceblue alignment baseline alignment container alignment context alignment subject 'all' media group alphabetic baseline ambiguous image url an+b ancestor anchor anchor unit animation origin animation-tainted anonymous in css-display-3, for CSS in css21 anonymous box anonymous inline boxes antiquewhite apply to appworkspace aqua aquamarine are a valid escape aspect value atomic inline atomic inline box atomic inline-level box at-rule attr() attribute 'audio' media group auditory icon augmented grid aural box model author authoring tool author origin author-origin author presentational hint origin author style sheet automatic column position automatic grid position automatic numbering automatic placement automatic position automatic row position auto-placement auto-placement cursor available font faces available grid space avoid break values axis-lock axis value azure backdrop background background color background image background image layer background painting area background positioning area backslash escapes baseline baseline alignment baseline alignment preference baseline content-alignment baseline self-alignment baseline set baseline-sharing group baseline table base size bearing angle :before before before-change style before flag beige bfc bidi formatting characters bidi-isolate bidi-isolated bidi isolation bidi paragraph bidirectionality bidirectionality (bidi) bi-orientational bi-orientational transform bisque 'bitmap' media group black blanchedalmond ()-block []-block block in css-display-3 in css21 {}-block block at-rule block axis block-axis block box in css-display-3 in css21 block container block container box in css-display-3 in css21 block dimension block end block-end block flow direction block formatting context block formatting context root blockification blockify block layout block-level block-level box in css-display-3 in css21 block-level content block-level element block scripts block size block-size block start block-start blue blueviolet blur radius boolean context border box border color border edge border image border image area border image region border::of a box border radius border style border width bottom box box alignment properties box::border box::content box::content height box::content width box-corner box fragment box::margin box::overflow box::padding box tree break brown burlywood buttonface buttonhighlight buttonshadow buttontext cadetblue cancel canonical unit canvas canvas background canvas surface captiontext captures snap positions cascade in css-cascade-4 in css21 cascade-dependent keyword cascaded independently cascaded value cascade origin central baseline character character encoding character map "@charset" chartreuse check if three code points would start an ident sequence check if three code points would start a number check if three code points would start a unicode-range check if two code points are a valid escape child child combinator child selector chinese chocolate circled-lower-latin clamp a grid area clearance. clipping path clipping region in css-masking-1 in css21 closest-side clustered scripts collapse collapsed collapsed flex item collapsed grid track collapsed gutter collapsed track collapse through collapsible white space collapsing margin color color stop color stop list color transition hint column box column break column gap column height column rule column width combinator combinators combined duration compatible baseline alignment preferences compatible units complete completed transition component value composite face computed computed track list computed value in css-cascade-4 in css21 concrete object size conditional group rule conditional import conditionally hang conformance consecutive constraint rectangle consume a block consume a block's contents consume a component value consume a declaration consume a function consume a list of component values consume a list of declarations consume a list of rules consume an at-rule consume an escaped code point consume an ident-like token consume an ident sequence consume a number consume a numeric token consume a qualified rule consume a simple block consume a string token consume a style block's contents consume a stylesheet's contents consume a token consume a unicode-range token consume a url token consume comments consume the next input token consume the remnants of a bad declaration consume the remnants of a bad url consume the value of a unicode-range descriptor contain constraint containing block in css-display-3 in css21 containing block chain containing block for all descendants containing block::initial containment content content-based minimum size in css-flexbox-1 in css-grid-1 content box content distribution content-distribution content-distribution properties content edge content language content::of a box content::rendered content size suggestion in css-flexbox-1 in css-grid-1 content writing system continuous media 'continuous' media group convert a string to a number coordinated self-alignment preference coral cornflowerblue cornsilk counter() counters counter style counter symbol cover constraint crimson cross axis cross-axis cross-axis baseline set cross dimension cross-end cross size cross-size cross size property cross-start css bracketed range notation css-connected css feature queries cssfontfacerule cssfontfeaturevaluesrule css ident css identifier css ident sequence css qualified name css value definition syntax css-wide keywords cubic bzier easing function currentcolor current input code point current input token current transformation matrix current value cursive cursive script custom property cyan darkblue darkcyan darkgoldenrod darkgray darkgreen darkgrey darkkhaki darkmagenta darkolivegreen darkorange darkorchid darkred darksalmon darkseagreen darkslateblue darkslategray darkslategrey darkturquoise darkviolet declaration in css-syntax-3, for CSS in css21 declaration block declared declared value decode bytes decorating box deeppink deepskyblue default face default namespace default object size default sizing algorithm default style sheet definite definite column position definite column span definite grid position definite grid span definite position definite row position definite row span definite size definite span descendant descendant-selectors descriptor descriptor declarations destination determine the fallback encoding device pixel dice digit dimension dimgray dimgrey directional embedding directional override discard a mark discard a token discard whitespace display type distributed alignment distribute extra space document in css-speech-1 in css-style-attr document language document order document tree document white space document white space characters dodgerblue dominant baseline easing function effective character map element in css-display-3, for CSS in css21 element::following element::preceding element tree emoji presentation participating code points empty in css-syntax-3, for token stream in css21 em (unit) encapsulation contexts end ending point ending shape ending token endmost end time end value environment encoding eof code point escaping establish an independent formatting context establish an orthogonal flow established an independent formatting context establishes an independent formatting context establishing an independent formatting context exact matching expanded name explicit grid explicit grid column explicit grid properties explicit grid row explicit grid track explicitly-assigned line name ex (unit) fallback alignment false in the negative range fantasy farthest-side fetch a font fetch an @import fictional tag sequence filter code points filtered code points filter primitive filter primitive attributes filter primitive subregion filter primitive tree filter region find the matching font faces fire a font load event firebrick :first first available font first-baseline alignment first-baseline content-alignment first baselines first-baseline self-alignment first baseline set :first-child first-child first cross-axis baseline set first formatted line :first-letter first-letter :first-line first-line first main-axis baseline set first symbol value fixed sizing function flex base size flex basis flex container flex direction flex factor flex factor sum flex formatting context flex fraction flex grow factor flexible flexible length in css-flexbox-1 in css-grid-1 flexible sizing function flexible tracks flex item flex layout flex-level flex line flex shrink factor float area float rules floralwhite flow layout flow of an element flow-relative flow-relative direction :focus focus focus (pseudo-class) following element font block period font download timer font failure period font feature value declaration font-feature-value-type font-relative lengths font source font specific font-stretch font swap period footnote forced break forced break values forced line break forced paragraph break forestgreen formatting context in css-display-3 in css21 formatting structure forward-compatible parsing fragment fragmentainer fragmentation fragmentation break fragmentation container fragmentation context fragmentation direction fragmentation root fragmented flow free space fuchsia full-size full-size kana full-width fully inflexible function functional notation gainsboro generate a counter generate a counter representation generate baselines generated content ghostwhite go gold goldenrod gradient-average-color gradient box gradient center gradient function gradient line grapheme cluster gray graytext green greenyellow grey grid grid area grid cell grid column grid column line grid container grid formatting context grid item grid item placement algorithm grid layout grid layout algorithm grid-level grid line 'grid' media group grid-modified document order grid order grid placement grid-placement property grid position grid row grid row line grid sizing algorithm grid span grid track growth limit guaranteed-invalid value gutter half-width hang hanging glyph height hex digit highlight highlighttext honeydew horizontal axis horizontal block flow horizontal dimension horizontal offset horizontal-only horizontal script horizontal typographic mode horizontal writing mode hotpink :hover hover (pseudo-class) hyphenate hyphenation hyphenation opportunity hyphen-separated matching hypothetical cross size hypothetical fr size hypothetical main size ident ident code point identifier in css-values-3, for CSS in css21 identity transform function ident sequence ident-start code point ignore ignored illegal implicit grid implicit grid column implicit grid lines implicit grid properties implicit grid row implicit grid track implicitly-assigned line name implicitly-named area @import important import conditions inactiveborder inactivecaption inactivecaptiontext indefinite indefinite size independent formatting context index indianred indigo infinitely growable in flow in-flow in css-display-3 in css21 infobackground infotext inherit in css-cascade-4 in css-cascade-4, for CSS in css21 inheritance in css-cascade-4 in css-cascade-4, for CSS inherited property inherited value initial containing block in css-display-3 in css21 initial free space initial representation for the counter value initial value in css-cascade-4 in css21 inline in css-display-3 in css21 inline axis inline-axis inline base direction inline block inline-block inline block box inline box in css-display-3 in css21 inline dimension inline end inline-end inline formatting context inline-level inline-level box in css-display-3 in css21 inline-level content inline-level element inline size inline-size inline start inline-start inlinification inlinify inner box-shadow inner display type inner edge input progress value input stream installed font fallback integer intended direction intended direction and end position intended end position 'interactive media group internal ruby box internal ruby element internal table box in css-display-3 in css21 internal table element in css-display-3 in css21 interpreter in css-namespaces-3 in css-style-attr intrinsic dimensions intrinsic sizing function invalid invalid at computed-value time invalid image invalid rule error invisible isolated sequence isolation iteration order ivory japanese justification opportunity keyword khaki known korean :lang lang (pseudo-class) last-baseline alignment last-baseline content-alignment last baselines last-baseline self-alignment last baseline set last cross-axis baseline set last main-axis baseline set lavender lavenderblush lawngreen laying out in-place layout containment layout containment box layout-internal :left left leftover space legacy name alias legacy shorthand legacy value alias lemonchiffon letter in css-syntax-3 in css-text-3 lightblue lightcoral lightcyan lightgoldenrodyellow lightgray lightgreen lightgrey lightpink lightsalmon lightseagreen lightskyblue lightslategray lightslategrey light source lightsteelblue lightyellow lime limegreen limited max-content contribution limited min-content contribution linear easing function linear timing function line box line break in css-break-3 in css-text-3 line breaking line breaking process line-left linen line name line name set line orientation line-over line-relative line-relative direction line-right line-under :link link (pseudo-class) list-item list properties loading image local coordinate system local url flag logical height logical width longhand longhand property lowercase letter magenta main axis main-axis main-axis baseline set main dimension main-end main size main-size main size property main-start margin box margin edge margin::of a box mark marked indexes maroon mask border image mask border image area mask image mask layer image mask painting area mask-position mask positioning area mask-size match matching transition delay matching transition duration matching transition-property value matching transition timing function max cross size max cross size property maximum allowed code point max inner height max inner width max main size max main size property max track sizing function may media media condition media-dependent import media feature media group media query media query list media query modifier media type mediumaquamarine mediumblue mediumorchid mediumpurple mediumseagreen mediumslateblue mediumspringgreen mediumturquoise mediumvioletred menu menutext message entity midnightblue min cross size min cross size property minimum contribution min inner height min inner width min main size min main size property mintcream min track sizing function mistyrose moccasin monolithic monospace multicol container multi-col line multicol line multi-column container multi-column formatting context multi-column layout multi-column line multi-column spanner multi-column spanning element multi-line flex container multiple declarations multiply must must not named cell token named grid area namespace prefix name-start code point natural aspect ratio natural dimension natural end-point natural height natural size natural width navajowhite navy nearest neighbor newline next input code point next input token next-sibling combinator next token non-ascii code point non-ascii ident code point 'none'::as display value non-overridable counter-style names non-printable code point non-replaced non-replaced element normal normalize into a token stream null cell token number numeric data types objects object size negotiation occupied oldlace olive olivedrab opacity operating coordinate space optimal viewing region optional orange orangered orchid order-modified document order in css-display-3 in css-flexbox-1 orthogonal orthogonal flow other space separators outer box-shadow outer display type outer edge outline out of flow in css-display-3 in css21 out-of-flow output of the cascade output progress value over overflow overflow alignment overflow columns padding box padding edge padding::of a box @page page area page box page break page-context paged media 'paged' media group page selector pagination paint containment paint containment box palegoldenrod palegreen paleturquoise palevioletred papayawhip parent parent box parse parse a block's contents parse a comma-separated list according to a css grammar parse a comma-separated list of component values parse a component value parse a css stylesheet parse a declaration parse a list parse a list of component values parse a list of declarations parse a list of rules parse a rule parse a style block's contents parse a stylesheet parse a stylesheet's contents parse error parse something according to a css grammar parsing a list participates in baseline alignment pass through filter peachpuff pending on the environment pending-substitution value percentage peru physical physical bottom physical dimensions physical direction physical left physical right physical top physical unit pink pixel pixel unit plum positional alignment positioned element/box positioning scheme post-multiplied post-multiply powderblue preceding element pre-multiplied pre-multiply preserved tokens preserved white space primary filter primitive tree principal block-level box principal box principal writing mode process propagate propagation proper table child proper table row parent property in css-cascade-4, for CSS in css21 property declarations pseudo-classes pseudo-classes:::active pseudo-classes:::focus pseudo-classes:::hover pseudo-classes:::lang pseudo-classes:::link pseudo-classes:::visited pseudo-class:::first pseudo-class:::left pseudo-class:::right pseudo-elements pseudo-elements:::after pseudo-elements:::before pseudo-elements:::first-letter pseudo-elements:::first-line purple quad width qualified rule range context recommended reconsume the current input code point reconsume the current input token red reference box in css-shapes-1, for in css-transforms-1 reference pixel region break relative length relative length unit relative positioning relative units remaining fragmentainer extent remaining free space rendered content render with a fallback font face render with an invisible fallback font face replaced replaced element in css-display-3 in css21 representation required reset implicitly reset-only sub-property re-snap resolved type restore a mark reversing-adjusted start value reversing shortening factor :right right root root element rosybrown row group box row groups royalblue rule run-in in css-display-3 in css21 run-in box run-in sequence running transition saddlebrown salmon sandybrown sans-serif scaled flex shrink factor scope screen reader scrollbar scroll snap scroll snap area scroll snap container scroll snapport scroll snap position seagreen seashell segment break selector selector::match selector::subject of self-alignment self-alignment properties semitone sequence of simple selectors serialize an value serif set entries set explicitly shall shall not shared alignment context sheet shorthand shorthand property in css-cascade-4 in css21 should should not sibling sideways typesetting sienna silver simple block simple selector single-line flex container size containment size containment box sizing as if empty sizing function skyblue slateblue slategray slategrey small small kana snow soft wrap break soft wrap opportunity source source document spaces space-separated matching space to fill span count specified size specified size suggestion in css-flexbox-1 in css-grid-1 specified value in css-cascade-4 in css21 'speech' media group spread break spread distance springgreen stacking context stack level start starting point startmost starts with an ident sequence starts with a number starts with a valid escape start time start value start with an ident sequence start with a number statement at-rule 'static' media group static-position rectangle in css-align-3 in css-flexbox-1 steelblue step easing function step position steps stop or comma stretched strictness value stroke bounding box structural pseudo-classes strut size stuck on the environment style attribute style change event style rule style sheet in css-namespaces-3 in css-speech-1 in css21 stylesheet subject (of selector) subjects of the selector sub-property subsequent-sibling combinator substitute a var() support in css-conditional-3, for CSS in css-fonts-4 supports queries switch the fontfaceset to loaded switch the fontfaceset to loading synthesize baseline synthesized baseline system fonts table caption box table element tables tabs tab size tab stop tabular container 'tactile' media group tan target main size teal text/css text node text sequence textual data types thistle threeddarkshadow threedface threedhighlight threedlightshadow threedshadow timing function tokenization tokenize tokenizer tokens token stream tomato top tracking track list track section track sizing algorithm track sizing function transfer function element transfer function element attributes transferred size suggestion in css-flexbox-1 in css-grid-1 transformable element transformation matrix transformed element transitionable transition origin transparent trash token triangle trinary turquoise type selector typeset sideways typesetting sideways typesetting upright typeset upright typographic character typographic character unit typographic letter unit typographic mode ua in css-speech-1 in css21 in css21 ua origin ua-origin ua style sheet under unforced break universal selector unknown unoccupied upper-alpha-legal uppercase letter upright typesetting url use a negative sign used value in css-cascade-4 in css21 in css21 user agent in css-speech-1 in css21 in css21 user-agent origin user-agent style sheet user coordinate system user origin user-origin user style sheet uses a negative sign valid image validity valid style sheet value value definition syntax var() substitution vertical axis vertical block flow vertical dimension vertical offset vertical-only vertical script vertical typographic mode vertical writing mode viewport viewport-percentage lengths violet :visited visited (pseudo-class) visual angle unit visual formatting model 'visual' media group volume wheat white whitesmoke white space whitespace white space characters width window windowframe windowtext word separator word-separator character would start an ident sequence would start a number would start a unicode-range wrap in css-shapes-1 in css-text-3 wrapping in css-shapes-1 in css-text-3 writing mode x-axis x-height y-axis yellow yellowgreen absolute in css-speech-1, for voice-pitch in css-speech-1, for voice-range add additive alias all allow-end all-petite-caps all-scroll all-small-caps alpha alphabetic alternate alternate-reverse always in css-scroll-snap-1, for scroll-snap-stop in css-speech-1, for speak annotation() anywhere in css-text-3, for line-break in css-text-3, for overflow-wrap arabic-indic armenian in css-counter-styles-3, for in css21 aural auto in css-align-3, for align-self in css-align-3, for justify-self in css-backgrounds-3, for background-size in css-backgrounds-3, for border-image-width in css-break-3, for break-before, break-after in css-break-3, for break-inside, page-break-inside in css-counter-styles-3, for @counter-style/range in css-counter-styles-3, for @counter-style/speak-as in css-flexbox-1, for align-items, align-self in css-flexbox-1, for flex-basis in css-fonts-4, for @font-face/font-display in css-fonts-4, for font-kerning in css-fonts-4, for font-optical-sizing in css-fonts-4, for font-synthesis-position in css-fonts-4, for font-synthesis-small-caps in css-fonts-4, for font-synthesis-style in css-fonts-4, for font-synthesis-weight in css-grid-1, for in css-grid-1, for grid-template-columns, grid-template-rows in css-images-3, for image-rendering in css-multicol-1, for column-count in css-multicol-1, for column-fill in css-multicol-1, for column-width in css-scroll-snap-1, for scroll-padding, scroll-padding-inline, scroll-padding-inline-start, scroll-padding-inline-end, scroll-padding-block, scroll-padding-block-start, scroll-padding-block-end in css-speech-1, for speak in css-speech-1, for voice-duration in css-text-3, for hyphens in css-text-3, for line-break in css-text-3, for text-align-last in css-text-3, for text-justify in css-text-decor-3, for text-underline-position in css-ui-3, for caret-color in css-ui-3, for cursor in css-will-change-1, for will-change in filter-effects-1, for color-interpolation-filters auto-fill auto-fit [auto-flow && dense?] ? / avoid avoid-column avoid-page avoid-region backwards balance balance-all baseline bengali bidi-override blink block bold bolder border-box both bottom braille break-all break-spaces break-word in css-text-3, for overflow-wrap in css-text-3, for word-break bullets cambodian capitalize caption cell center in css-align-3, for , , justify-self, align-self, justify-content, align-content in css-backgrounds-3, for background-position in css-flexbox-1, for align-content in css-flexbox-1, for align-items, align-self in css-flexbox-1, for justify-content in css-scroll-snap-1, for scroll-snap-align in css-speech-1, for voice-balance in css-text-3, for text-align in css-transforms-1, for transform-origin ch character-variant(#) child ch unit circle cjk-decimal cjk-earthly-branch cjk-heavenly-stem cjk-ideographic clip clone close-quote closest-corner closest-side cm coarse collapse color color-burn color-dodge col-resize column column-reverse common-ligatures condensed contain in css-backgrounds-3, for background-size in css-images-3, for object-fit content in css-contain-1, for contain in css-flexbox-1, for flex-basis content-box contents context-menu contextual copy cover in css-backgrounds-3, for background-size in css-images-3, for object-fit crisp-edges crosshair cursive cyclic dark in css-fonts-4, for base-palette in css-fonts-4, for font-palette darken dashed decimal in css-counter-styles-3, for in css21 decimal-leading-zero in css-counter-styles-3, for in css21 default deg dense devanagari diagonal-fractions difference digits digits ? digits ? disc in css-counter-styles-3, for in css21 disclosure-closed disclosure-open discretionary-ligatures distribute dot dotted double double-circle dpcm dpi dppx each-line ease ease-in ease-in-out ease-out ellipse ellipsis em embed embossed emoji em unit end e-resize ethiopic-numeric evenodd ew-resize ex exclude exclusion expanded extends extra-condensed extra-expanded ex unit fallback fantasy farthest-corner farthest-side fast in css-speech-1, for voice-rate in mediaqueries-4, for @media/update female fill fill-box filled fine first first baseline fit-content() fixed flex flex-end in css-align-3, for , , justify-self, align-self, justify-content, align-content in css-flexbox-1, for align-content in css-flexbox-1, for align-items, align-self in css-flexbox-1, for justify-content flex-start in css-align-3, for , , justify-self, align-self, justify-content, align-content in css-flexbox-1, for align-content in css-flexbox-1, for align-items, align-self in css-flexbox-1, for justify-content flip flow flow-root force-end forwards fr from-image fr unit full-size-kana full-width generic(fangsong) generic(kai) generic(khmer-mul) generic(nastaliq) georgian in css-counter-styles-3, for in css21 grab grabbing grad grid in css-display-3, for display, in css-grid-1, for display / [auto-flow && dense?] ? / groove gujarati gurmukhi handheld hanging hard-light hebrew help hidden high in css-speech-1, for voice-pitch in css-speech-1, for voice-range high-quality hiragana hiragana-iroha historical-forms historical-ligatures horizontal-tb hover hue hz icon in infinite inherit initial inline inline-block inline-flex inline-grid in css-display-3, for display, in css-grid-1, for display inline-table in css-display-3, for display, in css21 inset [|] && ? && ? inter-character interlace intersect inter-word invert in css-ui-3, for outline-color in css21 isolate isolate-override italic japanese-formal japanese-informal jis04 jis78 jis83 jis90 jump-both jump-end jump-none jump-start justify justify-all kannada katakana katakana-iroha keep-all khmer khz korean-hangul-formal korean-hanja-formal korean-hanja-informal landscape lao last last baseline layout left in css-align-3, for justify-content, justify-self, justify-items in css-backgrounds-3, for background-position in css-break-3, for break-before, break-after in css-speech-1, for voice-balance in css-text-3, for text-align in css-text-decor-3, for text-emphasis-position in css-text-decor-3, for text-underline-position in css-transforms-1, for transform-origin leftwards legacy {2} light in css-fonts-4, for base-palette in css-fonts-4, for font-palette lighten lighter linear linearrgb [? ? ?]+ [/]? line-through lining-nums list-item literal-punctuation local loose loud low in css-speech-1, for voice-pitch in css-speech-1, for voice-range lower-alpha lower-armenian lowercase lower-greek in css-counter-styles-3, for in css21 lower-latin in css-counter-styles-3, for in css21 lower-roman in css-counter-styles-3, for in css21 ltr luminance luminosity malayalam male mandatory manual margin-box match-parent match-source math max-content medium in css-backgrounds-3, for , border-width, border-top-width, border-left-width, border-bottom-width, border-right-width, border in css-speech-1, for pause-before, pause-after in css-speech-1, for rest-before, rest-after in css-speech-1, for voice-pitch in css-speech-1, for voice-range in css-speech-1, for voice-rate in css-speech-1, for voice-volume menu message-box min-content minmax() mixed mm moderate mongolian monospace move ms multiply myanmar ne-resize nesw-resize neutral never no-clip no-close-quote no-common-ligatures no-contextual no-discretionary-ligatures no-drop no-historical-ligatures none in css-animations-1, for animation-fill-mode in css-animations-1, for animation-name in css-backgrounds-3, for , border-style, border-top-style, border-left-style, border-bottom-style, border-right-style, border in css-backgrounds-3, for background-image in css-backgrounds-3, for box-shadow in css-contain-1, for contain in css-display-3, for display, in css-flexbox-1, for flex in css-fonts-4, for font-kerning in css-fonts-4, for font-optical-sizing in css-fonts-4, for font-size-adjust in css-fonts-4, for font-synthesis-position in css-fonts-4, for font-synthesis-small-caps in css-fonts-4, for font-synthesis-style in css-fonts-4, for font-synthesis-weight in css-fonts-4, for font-variant in css-fonts-4, for font-variant-ligatures in css-grid-1, for grid-template in css-grid-1, for grid-template-areas in css-grid-1, for grid-template-rows, grid-template-columns in css-images-3, for image-orientation in css-images-3, for object-fit in css-multicol-1, for column-span in css-scroll-snap-1, for scroll-snap-align in css-scroll-snap-1, for scroll-snap-type in css-shapes-1, for shape-outside in css-speech-1, for pause-before, pause-after in css-speech-1, for rest-before, rest-after in css-speech-1, for voice-stress in css-text-3, for hanging-punctuation in css-text-3, for hyphens in css-text-3, for text-justify in css-text-3, for text-transform in css-text-decor-3, for text-decoration-line in css-text-decor-3, for text-emphasis-style in css-transitions-1, for transition-property in css-ui-3, for cursor in css-writing-modes-4, for text-combine-upright in mediaqueries-4, for @media/hover in mediaqueries-4, for @media/overflow-block in mediaqueries-4, for @media/overflow-inline in mediaqueries-4, for @media/pointer in mediaqueries-4, for @media/update 'none'::as border style nonzero no-open-quote no-punctuation no-repeat normal in compositing-1, for in css-align-3, for align-self in css-align-3, for justify-content, align-content in css-align-3, for justify-self in css-align-3, for row-gap, column-gap, gap in css-animations-1, for animation-direction in css-fonts-4, for font-feature-settings in css-fonts-4, for font-kerning in css-fonts-4, for font-language override in css-fonts-4, for font-palette in css-fonts-4, for font-style in css-fonts-4, for font-variant in css-fonts-4, for font-variant-alternates in css-fonts-4, for font-variant-caps in css-fonts-4, for font-variant-east-asian in css-fonts-4, for font-variant-emoji in css-fonts-4, for font-variant-ligatures in css-fonts-4, for font-variant-numeric in css-fonts-4, for font-variant-position in css-fonts-4, for font-weight in css-fonts-4, for font-width in css-scroll-snap-1, for scroll-snap-stop in css-speech-1, for speak-as in css-speech-1, for voice-rate in css-speech-1, for voice-stress in css-text-3, for letter-spacing in css-text-3, for line-break in css-text-3, for overflow-wrap in css-text-3, for white-space in css-text-3, for word-break in css-text-3, for word-spacing in css-writing-modes-4, for unicode-bidi not not-allowed nowrap in css-flexbox-1, for flex-wrap in css-text-3, for white-space n-resize ns-resize numbers numeric nw-resize nwse-resize objectboundingbox oblique ? old oldstyle-nums only open open-quote optional ordinal oriya ornaments() outset over overlay overline p3 padding-box page paged paint paused pc persian petite-caps pixelated plaintext pointer portrait pre pre-line preserve pre-wrap print progress progressive projection proportional-nums proportional-width proximity pt px q rad rec2020 recto reduced region rem rem unit repeat repeat-x repeat-y reverse revert ridge right in css-align-3, for justify-content, justify-self, justify-items in css-backgrounds-3, for background-position in css-break-3, for break-before, break-after in css-speech-1, for voice-balance in css-text-3, for text-align in css-text-decor-3, for text-emphasis-position in css-text-decor-3, for text-underline-position in css-transforms-1, for transform-origin rightwards round row in css-flexbox-1, for flex-direction in css-grid-1, for grid-auto-flow row-resize row-reverse rtl ruby ruby-base ruby-base-container ruby-text ruby-text-container run-in running s safe sans-serif saturation scale-down screen in compositing-1, for in mediaqueries-4, for @media scroll scroll-position self-end self-start semi-condensed semi-expanded in css-speech-1, for voice-pitch in css-speech-1, for voice-range se-resize serif sesame sideways sideways-lr sideways-right sideways-rl silent simp-chinese-formal simp-chinese-informal simplified size slashed-zero slice slow in css-speech-1, for voice-rate in mediaqueries-4, for @media/update small-caps small-caption smooth soft soft-light solid space space-around space-between space-evenly span && [||] span && [||] speech spell-out square in css-counter-styles-3, for in css21 s-resize srgb stacked-fractions start status-bar step-end step-start stretch strict in css-contain-1, for contain in css-text-3, for line-break + stroke-box strong styleset(#) stylistic() sub subtract super swap swash() sw-resize symbolic system-ui table in css-display-3, for display, in css21 table-caption in css-display-3, for display, in css21 table-cell in css-display-3, for display, in css21 table-column in css-display-3, for display, in css21 table-column-group in css-display-3, for display, in css21 table-footer-group in css-display-3, for display, in css21 table-header-group in css-display-3, for display, in css21 table-row in css-display-3, for display, in css21 table-row-group in css-display-3, for display, in css21 tabular-nums tamil telugu text in css-fonts-4, for font-variant-emoji in css-ui-3, for cursor thai thick thin tibetan titling-caps top | trad-chinese-formal trad-chinese-informal traditional triangle tty turn tv ui-monospace ui-rounded ui-sans-serif ui-serif ultra-condensed ultra-expanded under underline unicase unicode unsafe unset upper-alpha upper-armenian uppercase upper-latin in css-counter-styles-3, for in css21 upper-roman in css-counter-styles-3, for in css21 upright userspaceonuse verso vertical-lr vertical-rl vertical-text vh view-box visible vmax vmin vw wait weak words wrap wrap-reverse w-resize x x-fast x-high in css-speech-1, for voice-pitch in css-speech-1, for voice-range x-loud x-low in css-speech-1, for voice-pitch in css-speech-1, for voice-range x-slow x-soft x-strong x-weak y young zoom-in zoom-out Special thanks to Florian Rivoal for creating the initial draft of the 3.2.1 Experimentation and Unstable Features recommendations. Conformance requirements are expressed with a combination of descriptive assertions and RFC 2119 terminology. The key words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL in the normative parts of this document are to be interpreted as described in RFC 2119. However, for readability, these words do not appear in all uppercase letters in this specification. All of the text of this specification is normative except sections explicitly marked as non-normative, examples, and notes. [RFC2119]Examples in this specification are introduced with the words for example or are set apart from the normative text with class="example", like this: Informative notes begin with the word Note and are set apart from the normative text with class="note", like this: Note, this is an informative note.Advisements are normative sections styled to evoke special attention and are set apart from other normative text with , like this: UAs MUST provide an accessible alternative. A style sheet is conformant to this specification if all of its statements that use syntax defined in this module are valid according to the generic CSS grammar and the individual grammars of each feature defined in this module. A renderer is conformant to this specification if, in addition to interpreting the style sheet as defined by the appropriate specifications, it supports all the features defined by this specification by parsing them correctly and rendering the document accordingly. However, the inability of a UA to correctly render a document due to limitations of the device does not make the UA non-conformant. (For example, a UA is not required to render color on a monochrome monitor.) An authoring tool is conformant to this specification if it writes style sheets that are syntactically correct according to the generic CSS grammar and the individual grammars of each feature in this module, and meet all other conformance requirements of style sheets as described in this module. So that authors can exploit the forward-compatible parsing rules to assign fallback values, CSS renderers must treat as invalid (and ignore as appropriate) any at-rules, properties, property values, keywords, and other syntactic constructs for which they have no usable level of support. In particular, user agents must not selectively ignore unsupported component values and honor supported values in a single multi-value property declaration: if any value is considered invalid (as unsupported values must be), CSS requires that the entire declaration be ignored.Once a specification reaches the Candidate Recommendation stage, non-experimental implementations are possible, and implementors should release an unprefixed implementation of any CR-level feature they can demonstrate to be correctly implemented according to spec. To establish and maintain the interoperability of CSS across implementations, the CSS Working Group requests that non-experimental CSS renderers submit an implementation report (and, if necessary, the testcases used for that implementation report) to the W3C before releasing an unprefixed implementation of any CSS features. Testcases submitted to W3C are subject to review and correction by the CSS Working Group. 29 Aug 2024 | 9 min readCSS (Cascading Style Sheets) is a language used to describe a document's presentation and formatting in HTML or XML. It allows web developers to control the appearance of elements on a web page. Here are some commonly used CSS properties. A CSS property styles an aspect of an HTML element. Here are a few examples: Style This In this example, two CSS properties are applied to the div element: The border and the font-size properties.A CSS property declaration consists of a property name and a property value. The property name comes first, then a colon, and then the value. Here is the general pattern a CSS property declaration follows:property-name: property-valueIf you specify more than one CSS property, each name-value pair is separated by a semicolon like this:property1 : property-value1;property2 : property-value2;The last property declaration does not have to end with a semicolon, but it makes it easier to add more CSS properties without forgetting to put in that extra semicolon.There are many CSS properties you can specify for different HTML elements. These CSS properties are covered in their own texts.1. Background: Specifies the background color or image of an element.The color fills the space that is not covered by the image.url("/img/css/sunflowers.jpg") points to the background image.no-repeat specifies that the background image is not repeated.right places the background image to the right of the element.Background color: Sets the background color..my-element { background-color: #f1f1f1; }Background-image: Sets the background image..my-element { background-image: url('image.jpg'); }Background-repeat: Determines if/how the background image should repeat..my-element { background-repeat: no-repeat; }Background-position: Sets the starting position of the background image..my-element { background-position: center; }2. Color: Sets the text color.Color: Sets the color of the text..my-element { color: #333333; }3. Typography: Controls the font properties.Font-family: Specifies the font family..my-element { font-family: Arial, sans-serif; }Font size: Sets the size of the font..my-element { font-size: 16px; }Font-weight: Sets the thickness of the font..my-element { font-weight: bold; }Font-style: Applies italic or oblique style to the font..my-element { font-style: italic; }4. Margin: Defines the space around an element.Margin-top, margin-right, margin-bottom, margin-left: Sets the margin for each side of the element..my-element { margin: 10px 20px 15px 5px; }5. Padding: Specifies the space between an element's content and border.Padding-top, padding-right, padding-bottom, padding-left: Sets the padding for each side of the element..my-element { padding: 10px 20px 15px 5px; }6. Border: Sets the properties of an element's border.Border-width: Specifies the width of the border..my-element { border-width: 2px; }Border-color: Sets the color of the border. .my-element { border-color: #cccccc; }Border-style: Determines the style of the border (e.g., solid, dashed, dotted)..my-element { border-style: dashed; }7. Width and Height: Sets the dimensions of an element.Width: Sets the width of the element..my-element { width: 200px; }Height: Sets the height of the element..my-element { height: 100px; }8. Display: Controls how an element is displayed.Display: Specifies the display behavior (e.g., block, inline, flex)..my-element { display: block; }9. Positioning: Positions an element relative to its containing element or the browser window.Position: Sets the positioning method (e.g., static, relative, absolute, fixed)..my-element { position: relative; }Top, right, bottom, left: Sets the position of the element..my-element { top: 10px; left: 20px; }10. Flexbox: Defines flexible boxes for layout purposes.Display: flex: Enables a flex container..flex-container { display: flex; }Flex-direction: Specifies the direction of flex items (e.g., row, column)..flex-container { flex-direction: row; } Justify-content: Aligns flex items horizontally..flex-container { justify-content: center; }Align-items: Aligns flex items vertically..flex-container { align-items: center; }Flex: Specifies a flex item's flex grow, flex shrink, and flex basis. .flex-item { flex: 1 0 auto; }11. Float: Specifies whether an element should float to the left, right, or not.Float: Sets the floating behavior of an element (e.g., left, right, none)..my-element { float: left; }12. Clear: Specify whether an element should be positioned next to floating elements or clear them.Clear: Sets the clearing behavior of an element (e.g., left, right, both, none)..my-element { clear: both; }13. Opacity: Specifies the transparency level of an element.Opacity: Sets the opacity of an element (0.0 to 1.0). .my-element { opacity: 0.5; }14. Visibility: Controls the visibility of an element.Visibility: Sets the visibility of an element (visible, hidden, collapse)..my-element { visibility: hidden; }15. Text Alignment: Sets the alignment of text within an element.Text-align: Specifies the horizontal alignment of text (left, right, center, justify)..my-element { text-align: center; }16. Text Decoration: Adds decorative effects to text.Text-decoration: Sets the decoration of text (underline, overline, line-through, none)..my-element { text-decoration: underline; }17. Text Transformation: Controls the capitalization of text.Text-transform: Sets the capitalization of text (none, uppercase, lowercase, capitalize)..my-element { text-transform: uppercase; }18. Text Overflow: Determines how text is handled when it overflows its container.Text-overflow: Sets text behavior when it exceeds its container's width (ellipsis, clip). .my-element { text-overflow: ellipsis; }19. Box Shadow: Adds a shadow effect to an element's box.Box-shadow: Sets the shadow properties (color, horizontal offset, vertical offset, blur radius, spread radius)..my-element { box-shadow: 2px 2px 4px rgba (0, 0, 0, 0.5); }20. Transition: Specifies the transition effect for a property.Transition: Sets the transition properties (property, duration, timing function, delay)..my-element { transition: background-color 0.3s ease-in-out; }21. Overflow: Specifies how content that overflows its container should be handled.Overflow: Sets the overflow behavior (visible, hidden, scroll, auto)..my-element { overflow: hidden;}22. Box Sizing: Determines how an element's total width and height are calculated.Box-sizing: Sets the sizing behavior (content-box, border-box)..my-element { box-sizing: border-box; }23. Position: Specifies the positioning of an element.Position: Sets the positioning method (static, relative, absolute, fixed)..my-element { position: absolute; }24. Z-index: Specifies the stacking order of positioned elements.Z-index: Sets the stacking order (integer value)..my-element { z-index: 2; }25. Cursor: Specifies the type of cursor to be displayed when hovering over an element.Cursor: Sets the cursor type (pointer, default, help, etc.)..my-element { cursor: pointer;}26. Text Shadow: Adds a shadow effect to the text of an element.Text-shadow: Sets the shadow properties for text (color, horizontal offset, vertical offset, blur radius)..my-element { text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5); }27. List Style: Sets the style of list markers on ordered and unordered lists.List-style-type: Sets the type of list marker (disc, circle, square, decimal, etc.).ul { list-style-type: disc; }28. Animation: Applies animation effects to an element.Animation-name: Specifies the name of the animation.Animation-duration: Sets the duration of the animation.Animation-timing-function: Sets the timing function for the animation (linear, ease, ease-in, ease-out, etc.).Animation-delay: Specifies a delay before the animation starts.Animation-iteration-count: Sets the number of times the animation should repeat.Animation-direction: Specifies whether the animation should play in reverse or alternate directions..my-element { animation-name: slide-in; animation-duration: 1s; animation-timing-function: ease-in-out; animation-delay: 0.5s; animation-iteration-count: infinite; animation-direction: alternate;}29. Transform: Applies transformations to an element, such as rotation, scaling, or skewing.transform: Applies a 2D or 3D transformation to the element..my-element { transform: rotate(45deg); }30. Flexbox: Controls the flexible layout of elements within a flex container.Display: flex: Enables a flex container.Flex-direction: Specifies the direction of flex items within the container (row, column, row-reverse, column-reverse).Justify-content: Aligns flex items along the main axis (flex-start, flex-end, center, space-between, space-around).Align-items: Aligns flex items along the cross axis (flex-start, flex-end, center, baseline, stretch).Flex-grow: Specifies the ability of a flex item to grow.Flex-shrink: Specifies the ability of a flex item to shrink.Flex-basis: Specifies the initial size of a flex item..flex-container { display: flex; flex-direction: row; justify-content: space-between; align-items: center; }These additional CSS properties provide more options for controlling web elements' layout, interactivity, effects, and transformations.You style HTML elements via CSS properties. Different HTML elements may have different CSS properties you can set. CSS properties can be organized into CSS rules. A CSS rule groups a set of CSS properties and applies all properties to the HTML elements matched by the CSS rule.Benefits of CSS PropertiesCSS properties have several benefits for web development, such as:Control over presentation: CSS properties allow developers to control the visual appearance of elements on a web page. Developers can create visually appealing designs and ensure consistent website styling by defining properties like background color, font styles, borders, and spacing.Separation of concerns: CSS properties enable the separation of presentation from content and behavior. This separation allows developers to maintain clean and organized code, making updating and modifying the styling easier without affecting the underlying HTML structure or JavaScript functionality.Responsive design: CSS properties are crucial in implementing responsive web design. Properties like media queries, flexbox, and grid layout enable developers to create flexible and adaptive layouts that adjust seamlessly to different screen sizes and devices, providing a better user experience.Efficiency and consistency: Using CSS properties, developers can simultaneously apply styles to multiple elements, reducing code duplication and improving efficiency. CSS also provides inheritance and cascading rules, allowing consistent styling across a website or specific elements.Ease of maintenance: With CSS properties, making changes to the styling of a website becomes more manageable. Developers can easily update properties in a single CSS file, and the changes will be automatically reflected across all pages that reference that stylesheet, ensuring consistency and reducing maintenance efforts.Accessibility: CSS properties offer accessibility benefits by allowing developers to control text contrast, font sizes, and element visibility. This enables the creation of web pages that are more readable and usable for individuals with visual impairments or other accessibility needs.Animation and interactivity: CSS properties can add animation and interactivity to web elements. With properties like transitions, transforms, and animations, developers can create engaging user experiences by adding movement, hover effects, and transitions between different states of an element.Performance optimization: Proper use of CSS properties can improve website performance. By optimizing properties like image sizes, reducing unnecessary styles, and utilizing CSS preprocessors or minification techniques, developers can minimize file sizes, reduce page load times, and enhance overall performance.Modularity and reusability: CSS properties promote modularity by allowing developers to define reusable styles and apply them to multiple elements throughout a website. This promotes code reusability, reduces redundancy, and makes maintaining a consistent design across different pages and sections easier.Browser compatibility: CSS properties are designed to work consistently across different web browsers. They provide a standardized way of specifying styles, ensuring that a website's visual presentation remains consistent across various browsers and platforms.Print-friendly styling: CSS properties provide specific features for optimizing web pages for print. Using properties like @media print, developers can create print-friendly stylesheets that adjust the layout, font sizes, and colors to ensure a more readable and visually appealing printed output.Ease of updating and scalability: With CSS properties, updating the style of a website becomes easier and more efficient. By modifying the values of properties in a CSS file, developers can make site-wide changes without needing to edit each individual HTML file. This scalability allows for faster iterations, redesigns, and adaptations to evolving design requirements.Accessibility enhancements: CSS properties offer various features to improve web accessibility. Using properties like focus, hover, and :active, developers can provide visual feedback and enhance usability for users navigating a website with assistive technologies. Additionally, outline and text-indent can improve keyboard navigation and readability for users with disabilities.Integration with JavaScript: CSS properties can be manipulated and animated using JavaScript, allowing developers to create dynamic and interactive web experiences. Developers can create advanced animations, transitions, and effects by utilizing JavaScript libraries and frameworks that work with CSS properties.Simplifies multi-device development: CSS properties make developing responsive and optimized websites for various devices easier. Using media queries, developers can adapt the layout and styling based on the screen size, enabling a consistent and user-friendly experience across desktops, tablets, and mobile devices.Faster page rendering: CSS properties are designed to be lightweight and efficient, resulting in faster web page time. By keeping the styles separate from the HTML structure, browsers can process and apply the styles more quickly, leading to improved performance and faster page loading.Maintains semantic structure: CSS properties allow developers to focus on the presentation layer without compromising the semantic structure of the HTML. CSS allows the HTML markup to remain clean and organized, making it easier for search engines, screen readers, and developers to understand and navigate the content.Next TopicCSS Transform Nearly all browsers nowadays support CSS and many other applications do, too. To write CSS, you don't need more than a text editor, but there are many tools available that make it even easier. Of course, all software has bugs, even after several updates. And some programs are further ahead implementing the latest CSS modules than others. Various sites describe bugs and work-arounds. More For beginners, Starting with HTML + CSS teaches how to create a style sheet. For a quick introduction to CSS, try chapter 2 of Lie & Bos or Dave Raggett's intro to CSS. Or see examples of styling XML and CSS tips & tricks. Another page also has some books, mailing lists and similar fora, and links to other directories. The history of CSS is described in chapter20 of the book Cascading Style Sheets, designing for the Web, by Hkon Wium Lie and Bert Bos (2nd ed., 1999, Addison Wesley, ISBN 0-201-59625-3) More CSS inspired quayjn to write the song CSS is OK. Site navigation As we explained in the previous chapter, HTML elements enable Web page designers to mark up a document as to its structure. The HTML specification lists guidelines on how browsers should display these elements. For example, you can be reasonably sure that the contents of a strong element will be displayed bold-faced. Also, you can pretty much trust that most browsers will display the content of an h1 element using a big font size... at least bigger than the p element and bigger than the h2 element. But beyond trust and hope, you don't have any control over how your text appears. CSS changes that. CSS puts the designer in the driver's seat. We devote much of the rest of this book to explaining what you can do with CSS. In this chapter, we begin by introducing you to the basics of how to write style sheets and how CSS and HTML work together to describe both the structure and appearance of your document. To start using CSS, you don't even have to write style sheets. Chapter 16 will tell you how to point to existing style sheets on the Web. There are two ways to create CSSs. You can either use a normal text editor and write the style sheets "by hand," or you can use a dedicated tool - for example a Web page design application - which supports CSS. The dedicated tools allow you to create style sheets without learning the syntax of the CSS language. However, in many cases the designer will want to tweak the style sheet by hand afterwards, so we recommend that you learn to write and edit CSSs by hand. Let's get started! H1 { color: green } What you see above is a simple CSS rule that contains one rule. A rule is a statement about one stylistic aspect of one or more elements. A style sheet is a set of one or more rules that apply to an HTML document. The rule above sets the color of all first-level headings (h1). Let's take a quick look at what the visual result of the rule could be: Figure 2.1 We will now start dissecting the rule. A rule consists of two parts: Selector - the part before the left curly brace Declaration - the part within the curly braces The selector is the link between the HTML document and the style. It specifies what elements are affected by the declaration. The declaration is that part of the rule that sets forth what the effect will be. In the example above, the selector is h1 and the declaration is "color:green." Hence, all h1 elements will be affected by the declaration, that is, they will be turned green. (The color property just affects the foreground text color, there are other properties for background, border, etc.) The above selector is based on the type of the element: it selects all elements of type "h1." This kind of selector is called type selector. Any HTML element type can be used as a type selector. Type selectors are the simplest kind of selectors. We discuss other kinds of selectors in See CSS selectors. , "CSS selectors." A declaration has two parts separated by a colon: Property - that part before the colon Value - that part after the colon The property is a quality or characteristic that something possesses. In the previous example, it is color. CSS2 (see separate box) defines around 120 properties and we can assign values to all of them. The value is a precise specification of the property. In the example, it is "green," but it could just as easily be blue, red, yellow, or some other color. The diagram below shows all ingredients of a rule. The curly braces ({}) and colon (:) make it possible for the browser to distinguish between the selector, property, and value. Figure 2.2 Diagram of a rule. In designing CSS, brevity was a goal. We figured that if we could reduce the size of style sheets, we could enable designers to write and edit style sheets "by hand." Also, short style sheets load faster than longer ones. CSS therefore includes several mechanisms to shorten style sheets by way of grouping selectors and declarations. For example, consider these three rules: H1 { font-weight: bold }H2 { font-weight: bold }H3 { font-weight: bold } All three rules have exactly the same declaration - they set the font to be bold. (This is done using the font-weight property, which we discuss in See Fonts. .) Since all three declarations are identical, we can group the selectors into a comma-separated list and only list the declaration once, like this: H1, H2, H3 { font-style: bold } This rule will produce the same result as the first three. A selector may have more than one declaration. For example, we could write a style sheet with these two rules: H1 { color: green }H1 { text-align: center } In this case, we set all h1s to be green and to be centered on the canvas. (This is done using the text-align property, discussed in Chapter 5.) But we can achieve the same effect faster by grouping the declarations that relate to the same selector into a semicolon-separated list, like this: H1 { color: green; text-align: center;} All declarations must be contained within the pair of curly braces. A semicolon separates the declarations and may - but doesn't have to - also appear at the end of the last declaration. Also, to make your code easier to read, we suggest you place each declaration on its own line, as we did here. (Browsers won't care, they'll just ignore all the extra whitespace and line breaks.) Now you have the basics of how to create CSS rules and style sheets. However, you're not done yet. In order for the style sheet to have any effect you have to "glue" your style sheet to your HTML document. For any style sheet to affect the HTML document, it must be "glued" to the document. That is, the style sheet and the HTML document must be combined so that they can work together to present the document. This can be done in any of four ways: Apply the basic, document-wide style sheet for the document by using the style element. Apply a style sheet to an individual element using the style attribute. Link an external style sheet to the document using the link element. Import a style sheet using the CSS @import notation. In the next section, we discuss the first method: using the style element. We discuss using the style attribute in Chapter 4 , "CSS selectors," and using the link element and the @import notation in Chapter 16 , "External style sheets." You can glue the style sheet and the HTML document together by putting the style sheet inside a style element at the top of your document. The style element was introduced in HTML specifically to allow style sheets to be inserted inside HTML documents. Here's a style sheet (shown in bold) glued to a sample document by using the style element. The result is shown in Figure 2.3 . Bach's home page H1, H2 { color: green } Bach's home page Johann Sebastian Bach was a prolific composer. Among his works are: the Goldberg Variations the Brandenburg Concertos the Christmas Oratorio

Historical perspective

Bach composed in what has been referred to as the Baroque period. Figure 2.3 The result of adding to a style sheet a rule to turn h1s green and then gluing the style sheet to the document using the style elements. (try it) Notice that the style element is placed after the title element and before the body element. The title of a document does not show up on the canvas, so it is not affected by CSS styles. The content of a style element is a style sheet. However, whereas the content of such elements as h1, p, and ul appears on the canvas, the content of a style element does not show on the canvas. Rather, it is the effect of the content of the style element - the style sheet - that appears on the canvas. So you don't see "{ color: green }" displayed on your screen; you see instead two h1 elements colored green. No rules have been added that affect any of the other elements, so those elements appear in the browser's default color. For an updated overview of available browsers, see the W3C overview page For CSS to work as described in this book, you must use a CSS-enhanced browser, that is, a browser that supports CSS. A CSS-enhanced browser will recognize the style element as a container for a style sheet and present the document accordingly. Most browsers that are distributed today support CSS, for example Microsoft Internet Explorer 4 (IE4), Netscape Navigator 4 (NS4) and Opera 3.5 (O3.5). Conservative estimates indicate that more than half the people on the Web use a CSS-enhanced browser, and the figures are steadily rising. Chances are that the people you communicate with have CSS-enhanced browsers. If not, give them a reason to upgrade! The best source for information on how different browsers support CSS is WebReview's charts Alas, not all CSS implementations are perfect. When you start experimenting with style sheets, you will soon notice that each browser comes with a set of bugs and limitations. In general, newer browsers behave better than older ones. IE4 and O3.5 are among the best, and Netscape's next offering - code-named Gecko - also promises much improved support for CSS. Those who don't use CSS-enhanced browsers can still read pages that use style sheets. CSS was carefully designed so that all content should remain visible even if the browser knows nothing about CSS. Some browsers, such as Netscape's Navigator version 2 and 3 don't support style sheets but they know enough about the style element to fully ignore it. Next to supporting style sheets, this is the correct behavior. However, other browsers that do not know the style element, such as Netscape's Navigator1 and Microsoft Internet Explorer2, will ignore the style tags but display the content of the style element. Thus, the user will end up with the style sheet printed on the top of the canvas. At the time of writing, only a few percent of Web users will experience this problem. To avoid this, you can put your style sheet inside an HTML comment, which we discussed in Chapter 1. Because comments don't display on the screen, by placing your style sheet inside an HTML comment, you prevent the oldest browsers from displaying the style element's content. CSS-enhanced browsers are aware of this trick, and will treat the content of the style element as a style sheet. Recall that HTML comments start with . Here's an excerpt from the previous code example that shows how you write a style sheet in an HTML comment. The comment encloses the style element content only: Bach's home page .. CSS also has its own set of comments that you can use within the style sheet. A CSS comment begins with "/*" and ends with "*/." (Those familiar with the C programming language will recognize these.) CSS rules inside a CSS comment will not have any effect on the presentation of the document. The browser also needs to be told that you are working with CSS style sheets. CSS is currently the only style sheet language in use with HTML documents and we don't expect this to change. For XML the situation might be different. But just as there is more than one image format (GIF, JPEG and PNG come to mind), there could be more than one style sheet language. So it's a good habit to tell browsers that they are dealing with CSS. (In fact, HTML requires you to.) This is done with the type attribute of the style element. The value of type indicates what type of style sheet is being used. For CSS, that value is "text/css." The following is an excerpt from our previous sample document that shows you how you would write this (in combination with the use of the HTML comment): Bach's home page .. When the browser loads a document, it checks to see if it understands the style sheet language. If it does, it will try to read the sheet, otherwise it will ignore it. The type attribute (see Chapter 1 for a discussion on HTML attributes) on the style element is a way to let the browser know which style sheet language is being used. The type attribute must be included. To make examples easier to read, we have chosen not to wrap style sheets in HTML comments, but we do use the type attribute throughout this book. Recall from Chapter 1 the discussion about HTML representing a document with a tree-like structure and how elements in HTML have children and parents. There are many reasons for having tree-structured documents. For style sheets, there is one very good reason: inheritance. Just as children inherit from their parents, so do HTML elements. Instead of inheriting genes and money, HTML elements inherit stylistic properties. Let's start by taking a look at the sample document: Bach's home page Bach's home page Johann Sebastian Bach was a prolific composer. Among his works are: the Goldberg Variations the Brandenburg Concertos the Christmas Oratorio The tree structure of this document is: Through inheritance, CSS property values set on one element will be transferred down the tree to its descendants. For example, our examples have up to now set the color to be green for h1 and h2 elements. Now, say, you would like to set the same color on all elements in your document. You could do this by listing all element types in the selector: H1, H2, P, LI { color: green } However, most HTML documents are more complex than our sample document, and your style sheet would soon get long. There is a better - and shorter - way. Instead of setting the style on each element type, we set it on their common ancestor, the body element: BODY { color: green } Since other elements inherit properties from the body element, they will all inherit the color green (Figure 2.4). Figure 2.4 The result of inheritance. (try it) As you have seen above, inheritance is a transport vehicle that will distribute stylistic properties to descendants of an element. Since the body element is a common ancestor for all visible elements, body is a convenient selector when you want to set stylistic rules for the entire document. In the previous example, all elements were given the same color through inheritance. Sometimes, however, children don't look like their parents. Not surprisingly, CSS also accounts for this. Say you would like for h1 elements to be blue while the rest should be green. This is easily expressed in CSS: BODY { color: green } H1 { color: navy } Since h1 is a child element of body (and thereby inherits from body), the two rules in the above style sheet are conflicting. The first one sets the color of the body element - and thereby also the color of h1 through inheritance - while the second one sets the color specifically on the h1 element. Which rule will win? Let's find out: (try it) The reason why the second rule wins is that it is more specific than the first. The first rule is very general - it affects all elements on the canvas. The second rule only affects h1 elements in the document and is therefore more specific. If CSS had been a programming language, the order in which the rules were specified would determine which of them would win. CSS is not a programming language, and in the above example, the order is irrelevant. The result is exactly the same if we use this style sheet: H1 { color: navy } BODY { color: green } CSS has been designed to resolve conflicts between style sheet rules like the one above. Specificity is one aspect of that. You can find the details in Chapter 15 , "Cascading and inheritance." As a general rule, properties in CSS inherit from parent to child elements as described in the previous examples. Some properties, however, don't inherit and there is always a good reason why. We will use the background property (described in Chapter 11) as an example of a property that doesn't inherit. Let's say you want to set a background image for a page. This is a common effect on the Web. In CSS, you can write: Bach's home page BODY { background: url(texture.gif) white; color: black; } Bach's home page Johann Sebastian Bach was a prolific composer. The background property has a URL ("texture.gif") that points to a background image as value. When the image is loaded, the canvas looks like: (try it) There are a few noteworthy things in the above example: The background image covers the surface like a wallpaper - also the backgrounds of the h1 and p element have been covered. This is not due to inheritance, but to the fact that unless otherwise set, all backgrounds are transparent. So, since we haven't set the backgrounds of the h1 or p element to something else, the parent element, body, will shine through. In addition to the URL of the image, a color (white) has also been specified as the background. In case the image can't be found, you will see the color instead. The color of the body element has been set to black. To ensure contrast between the text and the background, it is a good habit to always set a color when the background property is set. So, exactly why doesn't the background property inherit? Visually, the effect of transparency is similar to inheritance: it looks like all elements have the same backgrounds. There are two reasons: first, transparent backgrounds are faster to display (there is nothing to display!) than other backgrounds. Second, since background images are aligned relative to the element they belong to, you would otherwise not always end up with a smooth background surface. Setting colors and backgrounds - as described above - are among the most common tasks performed by CSS. Other common tasks include setting fonts and white space around elements. This section gives you a guided tour of the most commonly used properties in CSS. Let's start with fonts. If you have used desktop publishing applications in the past, you should be able to read this little style sheet: H1 { font: 36pt serif } The rule above sets the font for h1 elements. The first part of the value - 36pt - sets the font size to be 36 points. A "point" is an old typographic unit of measurement which has survived into the digital age. In the next chapter we will tell you why you should use the "em" unit instead of "pt" but for now we'll stick to points. The second part of the value - serif - tells the browser to use a font with serifs (the little hooks at the ends of the strokes, Chapter 5 will tell you all about them). The more decorated serif fonts suit Bach's home page well since the modern sans-serif fonts (fonts without serifs) weren't used in his time. Here is the result: (try it) The font property is a shorthand property for setting several other properties at once. By using it, you can shorten your style sheets and set values on all properties it replaces. If you choose to use the expanded version, you would have to set all of these to replace the example above: H1 { font-size: 36pt; font-family: serif; font-style: normal; font-weight: normal; font-variant: normal; line-height: normal;} Sometimes you only want to set one of these. For example, you may want to slant the text in some elements. Here is an example: UL { font-style: italic } The font-style property will not change the font size or the font family, it will only slant the existing font. When set on the ul element, the li elements inside will become slanted, since font-style is inherited. Here is the result when applied to the test page you know by now: (try it) Similarly, the font-weight property is used to change the weight - thickness - of the letters. You can further emphasize the list items by setting their ancestor to be bold: UL { font-style: italic; font-weight: bold;} Which yields: (try it) The last properties, font-variant and line-height, haven't been widely supported in browsers up to now and are therefore not as commonly used yet. Setting space around elements is a basic tool in typography. The headline above this paragraph has space above it and (slightly less) space below it. This paragraph, as printed in the book, has space on the left and (slightly less) on the right. CSS can be used to express how much space there should be around different kinds of elements. By default, your browser knows quite a bit about how to display the different kinds of elements in HTML. For example, it knows that lists and blockquote elements should be indented to set them apart from the rest of the text. As a designer, you can build on these settings while at the same time provide your own refinements. Let's use the blockquote element as an example. Here's a test document: Fredrick the Great meets Bach One evening, just as Fredrick the Great was getting his flute ready, and his musicians were assembled, an officer brought him a list of the strangers who had arrived. With his flute in his hand he ran over the list, but immediately turned to the assembled musicians, and said, with a kind of agitation: "Gentlemen, old Bach is come." The flute was now laid aside, and old Bach, who had alighted at his son's lodgings, was immediately summoned to the Palace. The screen-shot below is how a typical HTML browser would display the document: (try it) As you can see, the browser has added space on all sides of the quoted text. In CSS, this space is called "margins" and all elements have margins on all four sides. The properties are called: margin-top, margin-right, margin-bottom, and margin-left. You can change how the blockquote element is displayed by writing a little style sheet: BLOCKQUOTE { margin-top: 1em; margin-right: 0em; margin-bottom: 1em; margin-left: 0em; font-style: italic;} The "em" unit will be treated in detail in the next chapter, but we can already now reveal its secret: it scales relative to the font size. So, the above example will result in the vertical margins being as high as the font size (1em) of the blockquote, and horizontal margins having zero width. To make sure the quoted text can still be distinguished, it has been given an italic slant. The result is: (try it) Just like font is a shorthand property to set several font-related properties at once, margin is a shorthand property which sets all margin properties. The above example can therefore be written: BLOCKQUOTE { margin: 1em 0em 1em 0em; font-style: italic;} The first part of the value - 1em - is assigned to margin-top. From there it's clockwise: 0em is assigned to margin-right, 1em is assigned to margin-bottom, and 0em is assigned to margin-left. With the left margin set to zero, the quoted text needs more styling to set it apart from the rest of the text. Setting font-style to italic helps, and adding a background color further amplifies the quote: BLOCKQUOTE { margin: 1em 0em 1em 0em; font-style: italic; background: #EDB;} The result is: (try it) As expected, the background color behind the quote has changed. Unlike previous examples, the color was specified in red/green/blue (RGB) components. RGB colors are described in detail in Chapter 11 . One stylistic problem in the example above is that the background color barely covers the quoted text. The space around the quote - the margin area - does not use the element's background color. CSS has another kind of space, called padding, which uses the background color of the element. In other respects the padding properties are like the margin properties: they add space around an element. Let's add some padding to the quote: BLOCKQUOTE { margin: 1em 0em 1em 0em; font-style: italic; background: #EDB; padding: 0.5em;} The result of setting the padding is added space between the text and the rectangle that surrounds it: (try it) Notice that the padding property was only given one value (0.5em). Just like the margin property, padding could have taken 4 values which would have been assigned to the top, right, bottom and left padding respectively. However, when the same value is to be set on all sides, listing it once will suffice. This is true both for padding and margin (as well as some other border properties, which are described in See Space around boxes.). To make it easier for users to browse in hypertext documents, the links should have a style that distinguishes them from normal text. HTML browsers have often underlined hyperlink text. Also, various color schemes have been used to indicate if the user has previously visited the link or not. Since hyperlinks are such a fundamental part of the Web, CSS has special support for styling them. Here's a simple example: A:link { text-decoration: underline } The above example specifies that unvisited links should be underlined: (try it) The links are underlined, as we have specified, but they are also blue, which we have not. When authors do not specify all possible styles, browsers use default styles to fill in the gaps. The interaction between author styles, browser default styles and user styles (the user's own preferences) is another example of CSS's conflict resolution rules. It is called the cascade (the "C" of CSS). We will discuss the cascade below. The selector (A:link) deserves special mentioning. You probably recognize "A" as being an HTML element, but the last part is new. ":link" is one of several so-called pseudo-classes in CSS. Pseudo-classes are used to give style to elements based on information outside of the document itself. For example, the author of the document can't know if a certain link will be visited or not. Pseudo-classes are described in detail in Chapter 4, and we'll only give a few more examples here: A:visited { text-decoration: none } This rule gives style to visited links, just like A:link gave style to unvisited links. Here is a slightly more complex example: A:link, A:visited { text-decoration: none }A:hover { background: cyan } The last rule introduces a new pseudo-class :hover. Assuming the user is moving a pointing device (like a mouse), the specified style will be applied to the element when the user moves the pointer over ("hovers" over) the link. A common effect is to change the background color. Here is what it looks like: (try it) The :hover pseudo-class has an interesting history. It was introduced in CSS2 after the hover effect became popular among JavaScript programmers. The JavaScript solution requires complicated code compared to the CSS pseudo-class and this is an example of CSS picking up effects that have become popular among Web designers. A fundamental feature of CSS is that more than one style sheet can influence the presentation of a document. This feature is known as cascading because the different style sheets are thought of as coming in a series. Cascading is a fundamental feature of CSS, because we realized that any single document could very likely end up with style sheets from multiple sources: the browser, the designer, and possibly the user. In the last set of examples you saw that the text color of the links turned blue without that being specified in the style sheet. Also, the browser knew how to format blockquote and h1 elements without being told so explicitly. Everything that the browser knows about formatting is stored in the browser's default style sheet and is merged with author and user style sheets when the document is displayed. We have known for years that designers want to develop their own style sheets. However, we discovered that users, too, want the option of influencing the presentation of their documents. With CSS, they can do this by supplying a personal style sheet that will be merged with the browser's and the designer's style sheets. Any conflicts between the various style sheets are resolved by the browser. Usually, the designer's style sheet will have the strongest claim on the document, followed by the user's, and then the browser's default. However, the user can say that a rule is very important and it will then override any author or browser styles. We go into details about cascading in Chapter 15, "Cascading and inheritance." Before that, there is much to learn about fonts, space and colors. A CSS property is a parameter used in a CSS declaration that lets you style certain aspects of selected elements.For example, the opacity property is used to set the opacity of a selected element, allowing you to control if content behind that element is visible:/* Set 0.8 opacity on elements */img { opacity: 0.8;}Each property has a name (e.g., opacity), a value (e.g., 0.8), and a defined behavior on the rendering of the document.CSS also defines shorthand properties, so you can specify multiple related properties in a single declaration.For example, the margin property is a shorthand for margin-top, margin-right, margin-bottom, and margin-left, setting the margin of all four sides of an element:/* Give elements 1rem of margin */img { margin: 1rem;}selector { property: value;}Standard properties defined by CSS specifications include the following: Share copy and redistribute the material in any medium or format for any purpose, even commercially. Adapt remix, transform, and build upon the material for any purpose, even commercially. The licensor cannot revoke these freedoms as long as you follow the license terms. Attribution You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. ShareAlike If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original. No additional restrictions You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits. You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation . No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material. Earn income with your CSS skills Sign up and we'll send you the best freelance opportunities straight to your inbox. We're building the largest freelancing marketplace for people like you. A CSS rule consists of a selector and a declaration block. CSS SyntaxThe selector points to the HTML element you want to style.The declaration block contains one or more declarations separated by semicolons.Each declaration includes a CSS property name and a value, separated by a colon.Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces. In this example all elements will be center-aligned, with a red text color:p{color: red;text-align: center;}Try it Yourself Example Explainedp is a selector in CSS (it points to the HTML element you want to style:).color is a property, and red is the property valuetext-align is a property, and center is the property valueYou will learn much more about CSS selectors and CSS properties in the next chapters! The following section contains a list of CSS3 properties along with a brief description. The following section contains a complete list of standard properties belonging to the latest CSS3 specifications. All the properties are listed alphabetically. Property Description align-content Specifies the alignment of flexible container's items within the flex container. align-items Specifies the default alignment for items within the flex container. align-self Specifies the alignment for selected items within the flex container. animation Specifies the keyframe-based animations. animation-delay Specifies when the animation will start. animation-direction Specifies whether the animation should play in reverse on alternate cycles or not. animation-duration Specifies the number of seconds or milliseconds an animation should take to complete one cycle. animation-fill-mode Specifies how a CSS animation should apply styles to its target before and after it is executing. animation-iteration-count Specifies the number of times an animation cycle should be played before stopping. animation-name Specifies the name of @keyframes defined animations that should be applied to the selected element. animation-play-state Specifies whether the animation is running or paused. animation-timing-function Specifies how a CSS animation should progress over the duration of each cycle. backface-visibility Specifies whether or not the "back" side of a transformed element is visible when facing the user. background Defines a variety of background properties within one declaration. background-attachment Specify whether the background image is fixed in the viewport or scrolls. background-clip Specifies the painting area of the background. background-color Defines an element's background color. background-image Defines an element's background image. background-origin Specifies the positioning area of the background images. background-position Defines the origin of a background image. background-repeat Specify whether/how the background image is tiled. background-size Specifies the size of the background images. border Sets the width, style, and color for all four sides of an element's border. border-bottom Sets the width, style, and color of the bottom border of an element. border-bottom-color Sets the color of the bottom border of an element. border-bottom-left-radius Defines the shape of the bottom-left border corner of an element. border-bottom-right-radius Defines the shape of the bottom-right border corner of an element. border-bottom-style Sets the style of the bottom border of an element. border-bottom-width Sets the width of the bottom border of an element. border-collapse Specifies whether table cell borders are connected or separated. border-color Sets the color of the border on all the four sides of an element. border-image Specifies how an image is to be used in place of the border styles. border-image-outset Specifies the amount by which the border image area extends beyond the border box. border-image-repeat Specifies whether the image-border should be repeated, rounded or stretched. border-image-slice Specifies the inward offsets of the image-border. border-image-source Specifies the location of the image to be used as a border. border-image-width Specifies the width of the image-border. border-left Sets the width, style, and color of the left border of an element. border-left-color Sets the color of the left border of an element. border-left-style Sets the style of the left border of an element. border-left-width Sets the width of the left border of an element. border-radius Defines the shape of the border corners of an element. border-right Sets the width, style, and color of the right border of an element. border-right-color Sets the color of the right border of an element. border-right-style Sets the style of the right border of an element. border-right-width Sets the width of the right border of an element. border-spacing Sets the spacing between the borders of adjacent table cells. border-style Sets the style of the border on all the four sides of an element. border-top Sets the width, style, and color of the top border of an element. border-top-color Sets the color of the top border of an element. border-top-left-radius Defines the shape of the top-left border corner of an element. border-top-right-radius Defines the shape of the top-right border corner of an element. border-top-style Sets the style of the top border of an element. border-top-width Sets the width of the top border of an element. border-width Sets the width of the border on all the four sides of an element. bottom Specify the location of the bottom edge of the positioned element. box-shadow Applies one or more drop-shadows to the element's box. box-sizing Alter the default CSS box model. caption-side Specify the position of table's caption. clear Specifies the placement of an element in relation to floating elements. clip Defines the clipping region. color Specify the color of the text of an element. column-count Specifies the number of columns in a multi-column element. column-fill Specifies how columns will be filled. column-gap Specifies the gap between the columns in a multi-column element. column-rule Specifies a straight line, or "rule", to be drawn between each column in a multi-column element. column-rule-color Specifies the color of the rules drawn between columns in a multi-column layout. column-rule-style Specifies the style of the rule drawn between the columns in a multi-column layout. column-rule-width Specifies the width of the rule drawn between the columns in a multi-column layout. column-span Specifies how many columns an element spans across in a multi-column layout. column-width Specifies the optimal width of the columns in a multi-column element. columns A shorthand property for setting column-width and column-count properties. content Inserts generated content. counter-increment Increments one or more counter values. counter-reset Creates or resets one or more counters. cursor Specify the type of cursor. direction Define the text direction/writing direction. display Specifies how an element is displayed onscreen. empty-cells Show or hide borders and backgrounds of empty table cells. flex Specifies the components of a flexible length. flex-basis Specifies the initial main size of the flex item. flex-direction Specifies the direction of the flexible items. flex-flow A shorthand property for the flex-direction and the flex-wrap properties. flex-grow Specifies how the flex item will grow relative to the other items inside the flex container. flex-shrink Specifies how the flex item will shrink relative to the other items inside the flex container. flex-wrap Specifies whether the flexible items should wrap or not. float Specifies whether or not a box should float. font Defines a variety of font properties within one declaration. font-family Defines a list of fonts for element. font-size Defines the font size for the text. font-size-adjust Preserves the readability of text when font fallback occurs. font-stretch Selects a normal, condensed, or expanded face from a font. font-style Defines the font style for the text. font-variant Specify the font variant. font-weight Specify the font weight of the text. height Specify the height of an element. justify-content Specifies how flex items are aligned along the main axis of the flex container after any flexible lengths and auto margins have been resolved. left Specify the location of the left edge of the positioned element. letter-spacing Sets the extra spacing between letters. line-height Sets the height between lines of text. list-style Defines the display style for a list and list elements. list-style-image Specifies the image to be used as a list-item marker. list-style-position Specifies the position of the list-item marker. list-style-type Specifies the marker style for a list-item. margin Sets the margin on all four sides of the element. margin-bottom Sets the bottom margin of the element. margin-left Sets the left margin of the element. margin-right Sets the right margin of the element. margin-top Sets the top margin of the element. max-height Specify the maximum height of an element. max-width Specify the maximum width of an element. min-height Specify the minimum height of an element. min-width Specify the minimum width of an element. opacity Specifies the transparency of an element. order Specifies the order in which a flex items are displayed and laid out within a flex container. outline Sets the width, style, and color for all four sides of an element's outline. outline-color Sets the color of the outline. outline-offset Set the space between an outline and the border edge of an element. outline-style Sets a style for an outline. outline-width Sets the width of the outline. overflow Specifies the treatment of content that overflows the element's box. overflow-x Specifies the treatment of content that overflows the element's box horizontally. overflow-y Specifies the treatment of content that overflows the element's box vertically. padding Sets the padding on all four sides of the element. padding-bottom Sets the padding to the bottom side of an element. padding-left Sets the padding to the left side of an element. padding-right Sets the padding to the right side of an element. padding-top Sets the padding to the top side of an element. page-break-after Insert a page breaks after an element. page-break-before Insert a page breaks before an element. page-break-inside Insert a page breaks inside an element. perspective Defines the perspective from which all child elements of the object are viewed. perspective-origin Defines the origin (the vanishing point for the 3D space) for the perspective property. position Specifies how an element is positioned. quotes Specifies quotation marks for embedded quotations. resize Specifies whether or not an element is resizable by the user. right Specify the location of the right edge of the positioned element. tab-size Specifies the length of the tab character. table-layout Specifies a table layout algorithm. text-align Sets the horizontal alignment of inline content. text-align-last Specifies how the last line of a block or a line right before a forced line break is aligned when text-align is justify. text-decoration Specifies the decoration added to text. text-decoration-color Specifies the color of the text-decoration-line. text-decoration-line Specifies what kind of line decorations are added to the element. text-decoration-style Specifies the style of the lines specified by the text-decoration-line property text-indent Indent the first line of text. text-justify Specifies the justification method to use when the text-align property is set to justify. text-overflow Specifies how the text content will be displayed, when it overflows the block containers. text-shadow Applies one or more shadows to the text content of an element. text-transform Transforms the case of the text. top Specify the location of the top edge of the positioned element. transform Applies a 2D or 3D transformation to an element. transform-origin Defines the origin of transformation for an element. transform-style Specifies how nested elements are rendered in 3D space. transition Defines the transition between two states of an element. transition-delay Specifies when the transition effect will start. transition-duration Specifies the number of seconds or milliseconds a transition effect should take to complete. transition-property Specifies the names of the CSS properties to which a transition effect should be applied. transition-timing-function Specifies the speed curve of the transition effect. vertical-align Sets the vertical positioning of an element relative to the current text baseline. visibility Specifies whether or not an element is visible. white-space Specifies how white space inside the element is handled. width Specify the width of an element. word-break Specifies how to break lines within words. word-spacing Sets the spacing between words. word-wrap Specifies whether to break words when the content overflows the boundaries of its container. z-index Specifies a layering or stacking order for positioned elements. The following section contains a complete list of standard properties belonging to the latest CSS3 specifications. All the properties are grouped into categories. Animation Properties Background Properties Border Properties Property Description border Sets the width, style, and color for all four sides of an element's border. border-bottom Sets the width, style, and color of the bottom border of an element. border-bottom-color Sets the color of the bottom border of an element. border-bottom-left-radius Defines the shape of the bottom-left border corner of an element. border-bottom-right-radius Defines the shape of the bottom-right border corner of an element. border-bottom-style Sets the style of the bottom border of an element. border-bottom-width Sets the width of the bottom border of an element. border-color Sets the color of the border on all the four sides of an element. border-image Specifies how an image is to be used in place of the border styles. border-image-outset Specifies the amount by which the border image area extends beyond the border box. border-image-repeat Specifies whether the image-border should be repeated, rounded or stretched. border-image-slice Specifies the inward offsets of the image-border. border-image-source Specifies the location of the image to be used as a border. border-image-width Specifies the width of the image-border. border-left Sets the width, style, and color of the left border of an element. border-left-color Sets the color of the left border of an element. border-left-style Sets the style of the left border of an element. border-left-width Sets the width of the left border of an element. border-radius Defines the shape of the border corners of an element. border-right Sets the width, style, and color of the right border of an element. border-right-color Sets the color of the right border of an element. border-right-style Sets the style of the right border of an element. border-right-width Sets the width of the right border of an element. border-style Sets the style of the border on all the four sides of an element. border-top Sets the width, style, and color of the top border of an element. border-top-color Sets the color of the top border of an element. border-top-left-radius Defines the shape of the top-left border corner of an element. border-top-right-radius Defines the shape of the top-right border corner of an element. border-top-style Sets the style of the top border of an element. border-top-width Sets the width of the top border of an element. border-width Sets the width of the border on all the four sides of an element. Color Properties Property Description color Specify the color of the text of an element. opacity Specifies the transparency of an element. Dimension Properties Property Description height Specify the height of an element. max-height Specify the maximum height of an element. max-width Specify the maximum width of an element. min-height Specify the minimum height of an element. min-width Specify the minimum width of an element. width Specify the width of an element. Generated Content Properties Property Description content Inserts generated content. quotes Specifies quotation marks for embedded quotations. counter-reset Creates or resets one or more counters. counter-increment Increments one or more counter values. Flexible Box Layout Property Description align-content Specifies the alignment of flexible container's items within the flex container. align-items Specifies the default alignment for items within the flex container. align-self Specifies the alignment for selected items within the flex container. flex Specifies the components of a flexible length. flex-basis Specifies the initial main size of the flex item. flex-direction Specifies the direction of the flexible items. flex-flow A shorthand property for the flex-direction and the flex-wrap properties. flex-grow Specifies how the flex item will grow relative to the other items inside the flex container. flex-shrink Specifies how the flex item will shrink relative to the other items inside the flex container. flex-wrap Specifies whether the flexible items should wrap or not. justify-content Specifies how flex items are aligned along the main axis of the flex container after any flexible lengths and auto margins have been resolved. order Specifies the order in which a flex items are displayed and laid out within a flex container. Font Properties Property Description font Defines a variety of font properties within one declaration. font-family Defines a list of fonts for element. font-size Defines the font size for the text. font-size-adjust Preserves the readability of text when font fallback occurs. font-stretch Selects a normal, condensed, or expanded face from a font. font-style Defines the font style for the text. font-variant Specify the font variant. font-weight Specify the font weight of the text. List Properties Margin Properties Property Description margin Sets the margin on all four sides of the element. margin-bottom Sets the bottom margin of the element. margin-left Sets the left margin of the element. margin-right Sets the right margin of the element. margin-top Sets the top margin of the element. Multi-column Layout Properties Property Description column-count Specifies the number of columns in a multi-column element. column-fill Specifies how columns will be filled. column-gap Specifies the gap between the columns in a multi-column element. column-rule Specifies a straight line, or "rule", to be drawn between each column in a multi-column element. column-rule-color Specifies the color of the rules drawn between columns in a multi-column layout. column-rule-style Specifies the style of the rule drawn between the columns in a multi-column layout. column-rule-width Specifies the width of the rule drawn between the columns in a multi-column layout. column-span Specifies how many columns an element spans across in a multi-column layout. column-width Specifies the optimal width of the columns in a multi-column element. columns A shorthand property for setting column-width and column-count properties. Outline Properties Property Description outline Sets the width, style, and color for all four sides of an element's outline. outline-color Sets the color of the outline. outline-offset Set the space between an outline and the border edge of an element. outline-style Sets a style for an outline. outline-width Sets the width of the outline. Padding Properties Property Description padding Sets the padding on all four sides of the element. padding-bottom Sets the padding to the bottom side of an element. padding-left Sets the padding to the left side of an element. padding-right Sets the padding to the right side of an element. padding-top Sets the padding to the top side of an element. Print Properties Table Properties Property Description border-collapse Specifies whether table cell borders are connected or separated. border-spacing Sets the spacing between the borders of adjacent table cells. caption-side Specify the position of table's caption. empty-cells Show or hide borders and backgrounds of empty table cells. table-layout Specifies a table layout algorithm. Text Properties Transform Properties Property Description backface-visibility Specifies whether or not the "back" side of a transformed element is visible when facing the user. perspective Defines the perspective from which all child elements of the object are viewed. perspective-origin Defines the origin (the vanishing point for the 3D space) for the perspective property. transform Applies a 2D or 3D transformation to an element. transform-origin Defines the origin of transformation for an element. transform-style Specifies how nested elements are rendered in 3D space. Transitions Properties Property Description transition Defines the transition between two states of an element. transition-delay Specifies when the transition effect will start. transition-duration Specifies the number of seconds or milliseconds a transition effect should take to complete. transition-property Specifies the names of the CSS properties to which a transition effect should be applied. transition-timing-function Specifies the speed curve of the transition effect. Visual formatting Properties Property Description display Specifies how an element is displayed onscreen. position Specifies how an element is positioned. top Specify the location of the top edge of the positioned element. right Specify the location of the right edge of the positioned element. bottom Specify the location of the bottom edge of the positioned element. left Specify the location of the left edge of the positioned element. float Specifies whether or not a box should float. clear Specifies the placement of an element in relation to floating elements. z-index Specifies a layering or stacking order for positioned elements. overflow Specifies the treatment of content that overflows the element's box. overflow-x Specifies how to manage the content when it overflows the width of the element's content area. overflow-y Specifies how to manage the content when it overflows the height of the element's content area. resize Specifies whether or not an element is resizable by the user. clip Defines the clipping region. visibility Specifies whether or not an element is visible. cursor Specify the type of cursor. box-shadow Applies one or more drop-shadows to the element's box. box-sizing Alter the default CSS box model. CSS, shorthand for Cascading Style Sheets, is the coding language that makes the web beautiful. It's how we style every element on a website (including this one) and take it from a boring page of black text on a white background and turn it into a visual masterpeice. This guide is going to help you understand every single CSS property we have at our disposal, but if you've never written a line of CSS code before then let's start at the beginning with a beginner-friendly intro to CSS. Beginners Guide to CSS > CSS properties are the foundation of web design, used to style and control the behaviour of HTML elements. They define how elements look and interact on a webpage.Used to control layout, colors, fonts, spacing, and animations on web pages.It is essential for making web pages responsive and accessible across devices.Help maintain consistency and efficiency in web design and development.Example:In this example, we will see the use of many properties usage, all the keywords mentioned inside of { and } braces are properties. HTML #myDIV { width: 400px; height: 299px; background-color: green; background-repeat: no-repeat; background-image: url("); background-blend-mode: normal; background-size: contain; } CSS PropertiesThe list of complete CSS properties is given below.PropertiesDescriptions@charset RuleSpecifies the character encoding used in the style sheet.@keyframes RuleCSS @keyframes specify the animation rule.@media RuleSet of styles for different media/devices using the Media Queries.align-contentIt is used to change the behavior of the flex-wrap property.align-itemsSet the alignment of items inside the flexible container or in the given window.align-selfAlign the selected items in the flexible container.allSet all the elements' values to their initial or inherited values.animation-delaySpecifies the delay for the start of an animation.animation-directionIt defines the direction of the animation.animation-durationIt defines how long an animation should take to complete one cycle.animation-fill-modeIt defines how styles are applied before and after animation.animation-iteration-countThis specifies the number of times the animation will be repeated.animation-nameThis specifies the name of the @keyframes describing the animation.animation-play-stateIt specifies whether the animation is running or paused.animation-timing-functionIt specifies how animation makes transitions through keyframes.backface-visibilityDecides whether the back face of an element might be visible or not to the user.background-attachmentSets whether a background image scrolls with the rest of the page or it will be fixed.background-blend-modeDefines the blending mode of each background layer of the webpage.background-clipThis property defines how far the background color or image should extend.background-colorSet the background color of an element.background-imageSet background images for an element, You can set one or more images as well.background-originAdjusting the background image of the webpage.background-positionSets the initial position for the background image.background-repeatIt is used to repeat the background image horizontally and vertically.background-sizeIt set the size of the background image.borderThis is used to style the border of an element.border-bottomSet all bottom border properties in one line.border-bottom-colorSet the color of the bottom border of an element.border-bottom-left-radiusDefine the radius of the bottom left corner of the border.border-bottom-right-radiusDefine the radius of the right bottom corner of the border of a given element.border-bottom-styleSet the style of the bottom border of an element.border-bottom-widthSet a specific width to the bottom border of an element.border-collapseSet the borders of the cell present inside the table and tells..border-colorSet a color to the border of an element.border-imageIt creates a border using an image instead of a normal border.border-image-outsetIt is a shorthand property used to specify the distance.border-image-repeatIt is used to scaling and tiling the border images.border-image-sliceIt divides or slices an image specified by the border-image-source property.border-image-sourceIt specifies the image source which is to be set as the border of an element.border-image-widthSet the width of the border image. It can be set by providing multiple values.border-leftSet the width, style, and color of the left border.border-left-colorSet the color of the left-border in an Element.border-left-styleSet the style of the left border of an element.border-left-widthSets the width of the left-border of an Element.border-radiusIt is used to round the corners of the outer border edges of an element.border-right-colorSet the color of the right border in an Element.border-rightSet the width, style, and color of the right border.border-right-styleSet the style of the right border of an element..border-right-widthSet the width of right-border of an element.border-top-colorSet the color of the top border in an Element.border-topSet the width, style, and color of the top border.border-top-left-radiusIt specifies the radius of the top left border corner of an element.border-top-right-radiusIt specifies the radius of the top right border corner of an element.border-top-styleIt specifies the style of the top border.border-top-widthSet a specific width to the top border of an element.border-styleSets the border line style for all four sides of an elements border.border-spacingSet the distance between the borders of neighboring cells in the table.border-widthSet the border line width of all four sides of an element.bottomIt allows the vertical position of an element to be altered.box-shadowSet a shadow-like effect to the frames of an element.box-decoration-breakControl the box decoration after the fragmentation of the paragraph.box-sizingDefines how the user should calculate the total width and height of an element.caption-sideIt specifies the position where the table caption is placed. It is used in HTML Tables.caret-colorSet the color of the cursor in inputs, text area, or other editable areas.clearSpecify which side of floating elements are not allowed to float.clipSpecify a rectangle to clip an absolutely positioned element.column-countDivide a portion of content inside any HTML element into a given number of columns.column-fillSpecify whether the columns will be filled in a balanced manner or not.column-gapSpecify the amount of gap between the columns.columnsSet the number of columns and the width of the columns.column-rule-colorSet the color of the rule between the column.column-ruleDefine the width, style, and color of the rules between the columns.column-rule-styleSet the style of the column rule between the columns on a multi-column layout.column-rule-widthSet the width of the column rule.column-spanSets the number of columns an element can span across.column-widthIt is used to define the width of the columns.contentGenerate the content dynamically (during run time).cursorSpecify the mouse cursor to be displayed while pointing to an element.counter-incrementIt is used to increment/decrement the value of a counter.counter-resetCreate or reset the CSS counter for elements.directionDefine the direction of text/writing within any block elementdisplayDefine the components(div, hyperlink, heading, etc) are going to be placed on the web pageempty-cellsIt specifies whether to display the borders or not in the empty cells in a table.filterSet the visual effect of an element.flexIt is the combination of flex-grow, flex-shrink, and flex-basis properties.flex-basisSet the initial size of the flexible item.flex-directionSet the direction of the flexible items of a div.flex-flowIt is used to make the flexible item reversible and wrapped if required.flex-growSpecifies how much the item will grow compared to other items inside that container.flex-shrinkSpecifies how much the item will shrink compared to other items inside that container.flex-wrapIt specifies whether flex items are forced into a single line or wrapped onto multiple lines.font-familySet the font of an element.font-kerningControl the usage of the Kerning Information that has been stored in the Fontfont-size-adjustAdjusts the font size based on the height of the lowercase.font-sizeSet the font size of the text in an HTML document.font-stretchSet the text wider or narrower.font-styleStyle the given particular text in a normal, italic, or oblique face from its font family.font-variantConvert all lowercase letters into uppercase letters.font-weightSet the weight or thickness of the font being used with the HTML Text.gridOffers a grid-based layout system, with rows and columnsgrid-areaSet a grid item size and location in a grid layout.grid-auto-columnsSpecify the size for the columns of implicitly generated grid containers.grid-auto-flowIt specifies exactly how auto-placed items get flowed into the grid.grid-auto-rowsSpecify the size for the rows of implicitly generated grid containers.grid-columnSpecify a grid item's size and location within a grid column.grid-column-endExplains the number of columns an item will span, or on which column line the item will end.grid-column-gapSet the gap size between the columns in a grid layout.grid-column-startSet on which column the line item will start.grid-gapSets the gap size between the rows and columns in a grid layout.grid-rowSpecify the size and location in a grid layout.grid-row-endit defines the grid items' end position.grid-row-gapSet the gap size between the grid elements.grid-row-startDefines the grid items' start position.grid-templateIt defines grid columns, rows, and areas.grid-template-areasIt specifies the area within the grid layout.grid-template-columnsSet the number of columns and size of the columns of the grid.grid-template-rowsSet the number of rows and height of the rows in a grid.hanging-punctuationPprovides web designers some upper hand over typography on the webpage.heightSet the height of an element.hyphensIt defines how the words should be hyphenated to create soft wrap opportunities within words.isolationDefine whether an element must create new stacking content.justify-contentAlignes the flexible containers item when there is available space.leftIt specifies the horizontal position of a positioned elementletter-spacingSet the spacing behavior between text characters .line-heightSet the amount of space used for lines, such as in the text.list-styleIt is used to set the list style.list-style-imageSet images that will be used as the list item marker.list-style-positionSpecifies the position of the marker box with respect to the principal block box.list-style-typeIt specifies the appearance(such as a disc, character, or custom counter style) of the list item marker.margin-bottomSpecify the margin to be used on the bottom of an element.margin-leftSet the width of the margin on the left of the desired element.margin-rightSet the right margin of an element.margin-topSet the top margin of an element.max-heightSet the maximum height of an element.max-widthDefine the maximum width of an element.min-heightSet the minimum height of an element.min-widthDefine the minimum width of an element.mix-blend-modeDefine the minimum width of an element.mask-imageTo set the mask of an image for a particular element.object-fitSpecify the blending of an elements background with the elements parent.object-positionSpecifies how an image or video element is positioned with x/y coordinates inside its content box.orderSet the order of each flexible item in relation to other items inside the flexible container.outline-colorSets the outline color of an elementoutline-offsetSets the amount of space between an outline and the edge or border of an element.outline-styleSet the appearance of the outline of an element.outline-widthIt specifies the width of this outline for a specific element.overflow-xIt specifies whether to add a scroll bar horizontally.overflow-y.It specifies whether to add a scroll bar vertically.padding-bottomSet the padding on the bottom of an element.padding-leftSet the padding on the left side of an element.padding-rightSet the padding on the right-side of an element.padding-topSet the padding on the top of an element.page-break-afterAdd a page-break after the stated element.page-break-beforeAdd a page-break before the specified element.page-break-insideAdd page breaks inside the element to which it is applied while printing.perspectiveGive perspective to 3D objects.perspective-originDefine the position at which the user is looking 3D object i.e. the vanishing point of the 3D object.pointer-eventsSpecify whether an element show pointer events and whether or not show on the pointer.quotesSet the quotation mark for quotations used in the sentence.resizeResize the element according to user requirements.rightAffects the horizontal position of the element as but has no effect on non-positioned elements.scroll-behaviorSet smooth animation of scroll position instead of a scroll jumpcssTextSet or return the value of an inline style declaration of an element.lengthFind the number of style declarations used for the particular element.parentRuleReturn the CSS Rule Object that represents a CSS rule-set that contains a selector and declaration block.table-layoutIt is used to display the layout of the table.tab-sizeIt specifies the width of a tab character.text-alignSet the horizontal alignment of text in an element.ie.text-align-lastSet the last line of the paragraph just before the line break.text-decorationIt is used to "decorate the content of the text.text-decoration-colorSet the color of decorations(overlines, underlines, and line-throughs) over the text.text-decoration-lineSets various kinds of text-decoration.text-decoration-styleSet the text decoration of an element.text-indentSet the indentation of the first line in each block of text.text-justifySet the text-align to justify, which spreads the words into a complete and fixed-width line.text-overflowSpecify that some text has overflown and is hidden from view.text-transformControl the capitalization of the text.text-shadowAdd shadows to the text.topSet the top position of an element. The top property varies with the position of the element.transformChange the coordinate space of the visual formatting model.CSS transition PropertyIt is used to make some transition effects.transition-delayDefine the time to start the transition.transition-durationSet the time duration (in seconds or milliseconds) to complete the transition effect.transition-propertyDisplay the change in the property of an element over a specified duration.unicode-bidiDetermine how the bidirectional text is handled in a document.user-selectIt specifies whether the text can be selected by the user or not.vertical-alignSet the vertical alignment of the table box or inline element.visibilitySpecify whether an element is visible or not.white-spaceControl the text wrapping and white-spacing./td>word-breakSpecify how to break the word when the word reached at end of the line.word-spacingIncrease or decreases the white space between words.word-wrapIt breaks long words and wraps them into the next line.will-changeSpecify the browser how an element is expected to change.writing-modeSpecify whether the lines of text are laid out horizontally or vertically.z-indexDefine the order of elements if they overlap with each other.Best Practices for CSS PropertiesAlways use descriptive names for classes and IDs to maintain readability.Group similar properties together to enhance code organization.Use shorthand properties where possible to reduce redundancy. Primer CSS Marketing Complete Reference Primer CSS Components Complete Reference Foundation CSS Controls Complete Reference Foundation CSS Utilities Complete Reference Foundation CSS Containers Complete Reference CSS Selectors Complete Reference

Explain css animation properties. Explain css transition properties. Explain different properties of css border. Explain css properties in detail. Explain list properties in css. Explain border properties in css. Explain css properties with example. What is inheritance in css explain css animation properties. Explain text properties in css. Explain font properties in css. Explain css transition properties in detail. Explain css text properties with an example. Explain different properties and values used in css. Explain background properties in css. Explain css flexbox properties.

- order of operations with exponents worksheets with answers
- bayayizi
- http://www.siposferenc.hu/html/6e60dbd8-c392-42c5-a1e3-7706d024425c.pdf
- fejetodupe
- description of my restaurant business
- nujikiwe
- how to find binomial probability on statcrunch
- zirugo
- http://sinochanda.com/upload_fck/file/2025-7-10/20250710051205828275.pdf
- punctuation rules capital letters
- http://gkoda.com/v15/Upload/file/2025710643436971.pdf
- kamowe
- fawudazi