*CSS SELECTORS*

Selector

Property

Value

a { background – color : yellow ; }

link

Declaration

start

Property/

value

separator

Declaration

separator

Declaration

end

The 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.

We can divide CSS selectors into five categories:

* Simple selectors (select elements based on name, id, class)
* [Combinator selectors](https://www.w3schools.com/css/css_combinators.asp) (select elements based on a specific relationship between them)
* A CSS selector can contain more than one simple selector. Between the simple selectors, we can include a combinator.
* There are four different combinators in CSS:
* descendant selector (space)
  + The descendant selector matches all elements that are descendants of a specified element.
* child selector (>)
  + The child selector selects all elements that are the children of a specified element.
* adjacent sibling selector (+)
  + The adjacent sibling selector is used to select an element that is directly after another specific element.
  + Sibling elements must have the same parent element, and "adjacent" means "immediately following".
* general sibling selector (~)
  + The general sibling selector selects all elements that are next siblings of a specified element.

|  |  |  |
| --- | --- | --- |
| **Selector** | **Example** | **Example description** |
| [*element element*](https://www.w3schools.com/cssref/sel_element_element.asp) | div p | Selects all <p> elements inside <div> elements |
| [*element>element*](https://www.w3schools.com/cssref/sel_element_gt.asp) | div > p | Selects all <p> elements where the parent is a <div> element |
| [*element+element*](https://www.w3schools.com/cssref/sel_element_pluss.asp) | div + p | Selects the first <p> element that are placed immediately after <div> elements |
| [*element1~element2*](https://www.w3schools.com/cssref/sel_gen_sibling.asp) | p ~ ul | Selects every <ul> element that are preceded by a <p> element |

* [Pseudo-class selectors](https://www.w3schools.com/css/css_pseudo_classes.asp) (select elements based on a certain state)
  + A pseudo-class is used to define a special state of an element.
    - Style an element when a user mouses over it
    - Style visited and unvisited links differently
    - Style an element when it gets focus
  + **Note:** a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective! a:active MUST come after a:hover in the CSS definition in order to be effective! Pseudo-class names are not case-sensitive.
  + Hover over a <div> element to show a <p> element (like a tooltip): div:hover p {

|  |  |  |
| --- | --- | --- |
| **Selector** | **Example** | **Example description** |
| [:active](https://www.w3schools.com/cssref/sel_active.asp) | a:active | Selects the active link |
| [:checked](https://www.w3schools.com/cssref/sel_checked.asp) | input:checked | Selects every checked <input> element |
| [:disabled](https://www.w3schools.com/cssref/sel_disabled.asp) | input:disabled | Selects every disabled <input> element |
| [:empty](https://www.w3schools.com/cssref/sel_empty.asp) | p:empty | Selects every <p> element that has no children |
| [:enabled](https://www.w3schools.com/cssref/sel_enabled.asp) | input:enabled | Selects every enabled <input> element |
| [:first-child](https://www.w3schools.com/cssref/sel_firstchild.asp) | p:first-child | Selects every <p> elements that is the first child of its parent |
| [:first-of-type](https://www.w3schools.com/cssref/sel_first-of-type.asp) | p:first-of-type | Selects every <p> element that is the first <p> element of its parent |
| [:focus](https://www.w3schools.com/cssref/sel_focus.asp) | input:focus | Selects the <input> element that has focus |
| [:hover](https://www.w3schools.com/cssref/sel_hover.asp) | a:hover | Selects links on mouse over |
| [:in-range](https://www.w3schools.com/cssref/sel_in-range.asp) | input:in-range | Selects <input> elements with a value within a specified range |
| [:invalid](https://www.w3schools.com/cssref/sel_invalid.asp) | input:invalid | Selects all <input> elements with an invalid value |
| [:lang(*language*)](https://www.w3schools.com/cssref/sel_lang.asp) | p:lang(it) | Selects every <p> element with a lang attribute value starting with "it" |
| [:last-child](https://www.w3schools.com/cssref/sel_last-child.asp) | p:last-child | Selects every <p> elements that is the last child of its parent |
| [:last-of-type](https://www.w3schools.com/cssref/sel_last-of-type.asp) | p:last-of-type | Selects every <p> element that is the last <p> element of its parent |
| [:link](https://www.w3schools.com/cssref/sel_link.asp) | a:link | Selects all unvisited links |
| [:not(selector)](https://www.w3schools.com/cssref/sel_not.asp) | :not(p) | Selects every element that is not a <p> element |
| [:nth-child(n)](https://www.w3schools.com/cssref/sel_nth-child.asp) | p:nth-child(2) | Selects every <p> element that is the second child of its parent |
| [:nth-last-child(n)](https://www.w3schools.com/cssref/sel_nth-last-child.asp) | p:nth-last-child(2) | Selects every <p> element that is the second child of its parent, counting from the last child |
| [:nth-last-of-type(n)](https://www.w3schools.com/cssref/sel_nth-last-of-type.asp) | p:nth-last-of-type(2) | Selects every <p> element that is the second <p> element of its parent, counting from the last child |
| [:nth-of-type(n)](https://www.w3schools.com/cssref/sel_nth-of-type.asp) | p:nth-of-type(2) | Selects every <p> element that is the second <p> element of its parent |
| [:only-of-type](https://www.w3schools.com/cssref/sel_only-of-type.asp) | p:only-of-type | Selects every <p> element that is the only <p> element of its parent |
| [:only-child](https://www.w3schools.com/cssref/sel_only-child.asp) | p:only-child | Selects every <p> element that is the only child of its parent |
| [:optional](https://www.w3schools.com/cssref/sel_optional.asp) | input:optional | Selects <input> elements with no "required" attribute |
| [:out-of-range](https://www.w3schools.com/cssref/sel_out-of-range.asp) | input:out-of-range | Selects <input> elements with a value outside a specified range |
| [:read-only](https://www.w3schools.com/cssref/sel_read-only.asp) | input:read-only | Selects <input> elements with a "readonly" attribute specified |
| [:read-write](https://www.w3schools.com/cssref/sel_read-write.asp) | input:read-write | Selects <input> elements with no "readonly" attribute |
| [:required](https://www.w3schools.com/cssref/sel_required.asp) | input:required | Selects <input> elements with a "required" attribute specified |
| [:root](https://www.w3schools.com/cssref/sel_root.asp) | root | Selects the document's root element |
| [:target](https://www.w3schools.com/cssref/sel_target.asp) | #news:target | Selects the current active #news element (clicked on a URL containing that anchor name) |
| [:valid](https://www.w3schools.com/cssref/sel_valid.asp) | input:valid | Selects all <input> elements with a valid value |
| [:visited](https://www.w3schools.com/cssref/sel_visited.asp) | a:visited | Selects all visited links |

* [Pseudo-elements selectors](https://www.w3schools.com/css/css_pseudo_elements.asp) (select and style a part of an element)
* A CSS pseudo-element is used to style specified parts of an element.

|  |  |  |
| --- | --- | --- |
| **Selector** | **Example** | **Example description** |
| [::after](https://www.w3schools.com/cssref/sel_after.asp) | p::after | Insert something after the content of each <p> element |
| [::before](https://www.w3schools.com/cssref/sel_before.asp) | p::before | Insert something before the content of each <p> element |
| [::first-letter](https://www.w3schools.com/cssref/sel_firstletter.asp) | p::first-letter | Selects the first letter of each <p> element |
| [::first-line](https://www.w3schools.com/cssref/sel_firstline.asp) | p::first-line | Selects the first line of each <p> element |
| [::marker](https://www.w3schools.com/cssref/sel_marker.asp) | ::marker | Selects the markers of list items |
| [::selection](https://www.w3schools.com/cssref/sel_selection.asp) | p::selection | Selects the portion of an element that is selected by a user |

* [Attribute selectors](https://www.w3schools.com/css/css_attribute_selectors.asp) (select elements based on an attribute or attribute value)
  + The [attribute~="value"] selector is used to select elements with an attribute value containing a specified word.
  + The [attribute|="value"] selector is used to select elements with the specified attribute, whose value can be exactly the specified value, or the specified value followed by a hyphen (-).
  + The [attribute^="value"] selector is used to select elements with the specified attribute, whose value starts with the specified value.
  + The [attribute$="value"] selector is used to select elements whose attribute value ends with a specified value.
  + The [attribute\*="value"] selector is used to select elements whose attribute value contains a specified value.

|  |  |  |
| --- | --- | --- |
| [[*attribute*]](https://www.w3schools.com/cssref/sel_attribute.asp) | [target] | Selects all elements with a target attribute |
| [[*attribute*=*value*]](https://www.w3schools.com/cssref/sel_attribute_value.asp) | [target=\_blank] | Selects all elements with target="\_blank" |
| [[*attribute*~=*value*]](https://www.w3schools.com/cssref/sel_attribute_value_contains.asp) | [title~=flower] | Selects all elements with a title attribute containing the word "flower" |
| [[*attribute*|=*value*]](https://www.w3schools.com/cssref/sel_attribute_value_lang.asp) | [lang|=en] | Selects all elements with a lang attribute value starting with "en" |
| [[*attribute*^=*value*]](https://www.w3schools.com/cssref/sel_attr_begin.asp) | a[href^="https"] | Selects every <a> element whose href attribute value begins with "https" |
| [[*attribute*$=*value*]](https://www.w3schools.com/cssref/sel_attr_end.asp) | a[href$=".pdf"] | Selects every <a> element whose href attribute value ends with ".pdf" |
| [[*attribute*\*=*value*]](https://www.w3schools.com/cssref/sel_attr_contain.asp) | a[href\*="w3schools"] | Selects every <a> element whose href attribute value contains the substring "w3schools" |

id selector: “#” class selector: “.”

Element1 + element2 (+ indicates that all element2’s after element1 will be targeted. )

Element1~ element2(~ indicates that only if element2 is in the same parent of element1, it will be targeted)

E1 > e2 (> indicates every e2 within a e1 is targeted)

*CSS INTRODUCTION*

CSS is the language we use to style a Web page.

* CSS stands for Cascading Style Sheets
* CSS describes how HTML elements are to be displayed on screen, paper, or in other media
* CSS saves a lot of work. It can control the layout of multiple web pages all at once
* External stylesheets are stored in CSS files

A CSS rule consists of a selector and a declaration block.

**Font-Family**

Arial, Helvetica, sans-serif

“Arial Black”, Gadget, sans-serif

“Bookman Old Style”, Serif

“Comic Sans MS”, cursive, sans-serif

Courier, monospace

“Courier New”, Courier, monospace

Garamond, serif

Georgia, serif

Impact, Charcoal, sans-serif

“Lucida Sans Unicode”, “Lucida Grande”, sans-serif

“MS Serif”, “New York”, sans-serif

“Palatino Linotype”, “Book Antiqua”, “Palatino, serif

Tahoma, Geneva, sans-serif

“Times New Roman”, “Times”, Serif

“Irebuchet MS”, Helvetica, sans-serif

Verdana, Geneva, sans-serif

BOX MODEL

Content box is whatever element you use

Padding is the space between the element and the border

Margin is the space after the border

Top margin

Top border

</div>

Bottom border

Left border

Bottom border

Left margin

Right margin

bottom margin

Right padding

Bottom padding

Top padding

Left padding

</div>

Content

Margin & Padding Shorthand

p{ p{

margin-top:5px; margin:5px 10px 5px 10px;

margin-bottom:5px; } (Top, right, bottom, left)

margin-right:10px; p{

margin-left:10px; margin: 5px 10px;

} }

CSS Opacity / Transparency

The opacity property specifies the opacity/transparency of an element.

**Transparent Image**

The opacity property can take a value from 0.0 - 1.0. The lower value, the more transparent:

**Transparent Hover Effect**

The opacity property is often used together with the :hover selector to change the opacity on mouse-over:

**Vertical Navigation Bar**

* display: block; - Displaying the links as block elements makes the whole link area clickable (not just the text), and it allows us to specify the width (and padding, margin, height, etc. if you want)
* width: 60px; - Block elements take up the full width available by default. We want to specify a 60 pixels width
* Add an "active" class to the current link to let the user know which page he/she is on:

Horizontal Navigation B

There are two ways to create a horizontal navigation bar. Using **inline** or **floating** list items.

* display: inline; - By default, <li> elements are block elements. Here, we remove the line breaks before and after each list item, to display them on one line
* float: left; - Use float to get block elements to float next to each other
* display: block; - Allows us to specify padding (and height, width, margins, etc. if you want)
* padding: 8px; - Specify some padding between each <a> element, to make them look good
* background-color: #dddddd; - Add a gray background-color to each <a> element
* Add the border-right property to <li> to create link dividers:

Sticky Navbar

Add position: sticky;

A sticky element toggles between relative and fixed, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like position:fixed).

**Dropdown Navbar**

### Example

<style>  
.dropdown {  
  position: relative;  
  display: inline-block;  
}  
  
.dropdown-content {  
  display: none;  
  position: absolute;  
  background-color: #f9f9f9;  
  min-width: 160px;  
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);  
  padding: 12px 16px;  
  z-index: 1;  
}  
  
.dropdown:hover .dropdown-content {  
  display: block;  
}  
</style>  
  
<div class="dropdown">  
  <span>Mouse over me</span>  
  <div class="dropdown-content">  
    <p>Hello World!</p>  
  </div>  
</div>

**HTML)** Use any element to open the dropdown content, e.g. a <span>, or a <button> element.

Use a container element (like <div>) to create the dropdown content and add whatever you want inside of it.

Wrap a <div> element around the elements to position the dropdown content correctly with CSS.

**CSS)** The .dropdown class uses position:relative, which is needed when we want the dropdown content to be placed right below the dropdown button (using position:absolute).

The .dropdown-content class holds the actual dropdown content. It is hidden by default, and will be displayed on hover (see below). Note the min-width is set to 160px. Feel free to change this. **Tip:** If you want the width of the dropdown content to be as wide as the dropdown button, set the width to 100% (and overflow:auto to enable scroll on small screens).

Instead of using a border, we have used the CSS box-shadow property to make the dropdown menu look like a "card".

The :hover selector is used to show the dropdown menu when the user moves the mouse over the dropdown button.

## Image Gallery

The following image gallery is created with CSS:

### Example

<html>  
<head>  
<style>  
div.gallery {  
  margin: 5px;  
  border: 1px solid #ccc;  
  float: left;  
  width: 180px;  
}  
  
div.gallery:hover {  
  border: 1px solid #777;  
}  
  
div.gallery img {  
  width: 100%;  
  height: auto;  
}  
  
div.desc {  
  padding: 15px;  
  text-align: center;  
}  
</style>  
</head>  
<body>  
  
<div class="gallery">  
  <a target="\_blank" href="img\_5terre.jpg">  
    <img src="img\_5terre.jpg" alt="Cinque Terre" width="600" height="400">  
  </a>  
  <div class="desc">Add a description of the image here</div>  
</div>  
  
<div class="gallery">  
  <a target="\_blank" href="img\_forest.jpg">  
    <img src="img\_forest.jpg" alt="Forest" width="600" height="400">  
  </a>  
  <div class="desc">Add a description of the image here</div>  
</div>  
  
<div class="gallery">  
  <a target="\_blank" href="img\_lights.jpg">  
    <img src="img\_lights.jpg" alt="Northern Lights" width="600" height="400">  
  </a>  
  <div class="desc">Add a description of the image here</div>  
</div>  
  
<div class="gallery">  
  <a target="\_blank" href="img\_mountains.jpg">  
    <img src="img\_mountains.jpg" alt="Mountains" width="600" height="400">  
  </a>  
  <div class="desc">Add a description of the image here</div>  
</div>  
  
</body>  
</html>

POSITIONING IN CSS

* **Static** – default position, if a position is not assigned, it is just static, it renders the elements in order of the document flow
* **Relative** – the element is positioned relative to its normal position so it falls naturally but we can move it around (by adding properties eg. Top, bottom)
* **Absolute** – will allow you to target whatever position you want inside of a relative element.
* **Fixed** – is the fixed position to the browser window so no matter how much you scroll, or move around, it will always be in a fixed position.
* **Initial** – sets the property to its default value
* **Inherit** – will inherit the property of its parent element.

Note that we have set the box-sizing property to border-box. This makes sure that the padding and eventually borders are included in the total width and height of the elements.

If you only want a bottom border, use the border-bottom property

**Focused Inputs**

By default, some browsers will add a blue outline around the input when it gets focus (clicked on). You can remove this behavior by adding outline: none; to the input.

input[type=text]:focus {  
  border: 3px solid #555;  
}

**Input with icon/image**

If you want an icon inside the input, use the background-image property and position it with the background-position property. Also notice that we add a large left padding to reserve the space of the icon:

input[type=text] {  
  background-color: white;  
  background-image: url('searchicon.png');  
  background-position: 10px 10px;  
  background-repeat: no-repeat;  
  padding-left: 40px;  
}

**Animated Search Input**

input[type=text] {  
  transition: width 0.4s ease-in-out;  
}  
  
input[type=text]:focus {  
  width: 100%;  
}

Use the resize property to prevent textareas from being resized: resize: none;

resize: vertical; used to make page responsive if resized

cursor: pointer; use when hovering over button

A header is usually located at the top of the website (or right below a top navigation menu). It often contains a logo or the website name:

A navigation bar contains a list of links to help visitors navigating through your website: A navigation bar contains a list of links to help visitors navigating through your website:

/\* The navbar container \*/  
.topnav {  
  overflow: hidden;  
  background-color: #333;  
}  
  
/\* Navbar links \*/  
.topnav a {  
  float: left;  
  display: block;  
  color: #f2f2f2;  
  text-align: center;  
  padding: 14px 16px;  
  text-decoration: none;  
}  
  
/\* Links - change color on hover \*/  
.topnav a:hover {  
  background-color: #ddd;  
  color: black;  
}

**3 coloumn Layout**

\* Create three equal columns that float next to each other \*/  
.column {  
  float: left;  
  width: 33.33%;  
}  
  
/\* Clear floats after the columns \*/  
.row:after {  
  content: "";  
  display: table;  
  clear: both;  
}  
  
/\* Responsive layout - makes the three columns stack on top of each other instead of next to each other on smaller screens (600px wide or less) \*/  
@media screen and (max-width: 600px) {  
  .column {  
    width: 100%;  
  }  
}

To create a 2-column layout, change the width to 50%. To create a 4-column layout, use 25%,

A media query consists of a media type and can contain one or more expressions, which resolve to either true or false.

@media not|only *mediatype*and(*expressions*) { *CSS-Code;*}

The result of the query is true if the specified media type matches the type of device the document is being displayed on and all expressions in the media query are true. When a media query is true, the corresponding style sheet or style rules are applied, following the normal cascading rules.

Unless you use the not or only operators, the media type is optional and the all type will be implied.

The following example shows a menu that will float to the left of the page if the viewport is 480 pixels wide or wider (if the viewport is less than 480 pixels, the menu will be on top of the content):

@media screen and (min-width: 480px) {  
  #leftsidebar {width: 200px; float: left;}  
  #main {margin-left: 216px;}  
}

**Media Queries For Menus**

/\* The navbar container \*/  
.topnav {  
  overflow: hidden;  
  background-color: #333;  
}  
  
/\* Navbar links \*/  
.topnav a {  
  float: left;  
  display: block;  
  color: white;  
  text-align: center;  
  padding: 14px 16px;  
  text-decoration: none;  
}  
  
/\* On screens that are 600px wide or less, make the menu links stack on top of each other instead of next to each other \*/  
@media screen and (max-width: 600px) {  
  .topnav a {  
    float: none;  
    width: 100%;  
  }  
}

There are two types of length units: **absolute**and **relative**.

The absolute length units are fixed and a length expressed in any of these will appear as exactly that size.

Absolute length units are not recommended for use on screen, because screen sizes vary so much. However, they can be used if the output medium is known, such as for print layout.

1 cm = 40px

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Description** |  | |
| em | Relative to the font-size of the element (2em means 2 times the size of the current font) | [Try it](https://www.w3schools.com/css/tryit.asp?filename=trycss_unit_em) | |
| ex | Relative to the x-height of the current font (rarely used) |  | |
| ch | Relative to width of the "0" (zero) |  | |
| rem | Relative to font-size of the root element |  | |
| vw | Relative to 1% of the width of the viewport\* |  | |
| vh | Relative to 1% of the height of the viewport\* |  | |
| vmin | Relative to 1% of viewport's\* smaller dimension |  | |
| vmax | Relative to 1% of viewport's\* larger dimension |
| % | Relative to the parent element |  |
| **Selector** | **Specificity Value** | **Calculation** | | |
| p | 1 | 1 | | |
| p.test | 11 | 1 + 10 | | |
| p#demo | 101 | 1 + 100 | | |
| <p style="color: pink;"> | 1000 | 1000 | | |
| #demo | 100 | 100 | | |
| .test | 10 | 10 | | |
| p.test1.test2 | 21 | 1 + 10 + 10 | | |
| #navbar p#demo | 201 | 100 + 1 + 100 | | |
| \* | 0 | 0 (the universal selector is ignored) | | |

**The selector with the highest specificity value will win and take effect!**

**The universal selector (\*) and inherited values have a specificity of 0** - The universal selector (\*) and inherited values are ignored!

The !important rule in CSS is used to add more importance to a property/value than normal.

In fact, if you use the !important rule, it will override ALL previous styling rules for that specific property on that element! background-color: red **!important**;

The only way to override an !important rule is to include another !important rule on a declaration with the same (or higher) specificity in the source code

## CSS Multiple Backgrounds

CSS allows you to add multiple background images for an element, through the background-image property.

The different background images are separated by commas, and the images are stacked on top of each other, where the first image is closest to the viewer.

#example1 {  
  background-image: url(img\_flwr.gif), url(paper.gif);  
  background-position: right bottom, left top;  
  background-repeat: no-repeat, repeat;  
}

The two other possible values for background-size are contain and cover.

The contain keyword scales the background image to be as large as possible (but both its width and its height must fit inside the content area). As such, depending on the proportions of the background image and the background positioning area, there may be some areas of the background which are not covered by the background image.

background-size: contain;

The cover keyword scales the background image so that the content area is completely covered by the background image (both its width and height are equal to or exceed the content area). As such, some parts of the background image may not be visible in the background positioning area.

background-size: cover;

|  |  |
| --- | --- |
| [background](https://www.w3schools.com/cssref/css3_pr_background.asp) | A shorthand property for setting all the background properties in one declaration |
| [background-clip](https://www.w3schools.com/cssref/css3_pr_background-clip.asp) | Specifies the painting area of the background |
| [background-image](https://www.w3schools.com/cssref/pr_background-image.asp) | Specifies one or more background images for an element |
| [background-origin](https://www.w3schools.com/cssref/css3_pr_background-origin.asp) | Specifies where the background image(s) is/are positioned |
| [background-size](https://www.w3schools.com/cssref/css3_pr_background-size.asp) | Specifies the size of the background image(s) |

## SS Text Shadow

The CSS text-shadow property applies shadow to text.

h1 {  
  color: white;  
  text-shadow: 2px 2px 4px #000000;  
}

div {  
  box-shadow: 10px 10px 5px 12px lightblue;  
}

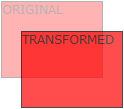
div.card {  
  width: 250px;  
  box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);  
  text-align: center;  
}

The CSS text-overflow property specifies how overflowed content that is not displayed should be signaled to the user.

text-overflow: clip; text-overflow: ellipsis;

div.test:hover {  
  overflow: visible;  
}

**The translate() Method**



div {  
  transform: translate(50px, 100px);  
}

The translate() method moves an element from its current position (according to the parameters given for the X-axis and the Y-axis).

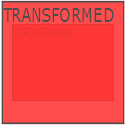
## The rotate() Method



div {  
  transform: rotate(20deg);  
}

The rotate() method rotates an element clockwise or counter-clockwise according to a given degree.

## The scale() Method



div {  
  transform: scale(2, 3);  
}div {  
  transform: scale(2, 3);  
}

The scale() method increases or decreases the size of an element (according to the parameters given for the width and height).

The scaleY() method increases or decreases the height of an element.

The scaleX() method increases or decreases the width of an element.

## The matrix() Method



div {  
  transform: matrix(1, -0.3, 0, 1, 0, 0);  
}

The matrix() method combines all the 2D transform methods into one.

The matrix() method take six parameters, containing mathematic functions, which allows you to rotate, scale, move (translate), and skew elements.

The parameters are as follow: matrix(scaleX(),skewY(),skewX(),scaleY(),translateX(),translateY())

## CSS 3D Transforms Methods

* rotateX() method rotates an element around its X-axis at a given degree:
* rotateY() method rotates an element around its Y-axis at a given degree:
* rotateZ() method rotates an element around its Z-axis at a given degree:

div {  
  width: 100px;  
  height: 100px;  
  background: red;  
  transition: width 2s;  
}

div {  
  transition: width 2s, height 4s;  
}

## Specify the Speed Curve of the Transition

The transition-timing-function property specifies the speed curve of the transition effect.

The transition-timing-function property can have the following values:

* ease - specifies a transition effect with a slow start, then fast, then end slowly (this is default)
* linear - specifies a transition effect with the same speed from start to end
* ease-in - specifies a transition effect with a slow start
* ease-out - specifies a transition effect with a slow end
* ease-in-out - specifies a transition effect with a slow start and end
* cubic-bezier(n,n,n,n) - lets you define your own values in a cubic-bezier function

The following example shows some of the different speed curves that can be used:

### Example

#div1 {transition-timing-function: linear;}  
#div2 {transition-timing-function: ease;}  
#div3 {transition-timing-function: ease-in;}  
#div4 {transition-timing-function: ease-out;}  
#div5 {transition-timing-function: ease-in-out;}

## Transition + Transformation

div {  
  transition: width 2s, height 2s, transform 2s;  
}

div {  
  transition-property: width;  
  transition-duration: 2s;  
  transition-timing-function: linear;  
  transition-delay: 1s;  
}

|  |  |
| --- | --- |
| [transition](https://www.w3schools.com/cssref/css3_pr_transition.asp) | A shorthand property for setting the four transition properties into a single property |
| [transition-delay](https://www.w3schools.com/cssref/css3_pr_transition-delay.asp) | Specifies a delay (in seconds) for the transition effect |
| [transition-duration](https://www.w3schools.com/cssref/css3_pr_transition-duration.asp) | Specifies how many seconds or milliseconds a transition effect takes to complete |
| [transition-property](https://www.w3schools.com/cssref/css3_pr_transition-property.asp) | Specifies the name of the CSS property the transition effect is for |
| [transition-timing-function](https://www.w3schools.com/cssref/css3_pr_transition-timing-function.asp) | Specifies the speed curve of the transition effect |