

CSS selectors

Patterns used to select elements to style. Selectors can override each other going from least to most “specific,” element < class < id < inline.

element selectors

Styles elements of a certain type, such as paragraphs or headings.

Code example:

```
p {  
  background-color: yellow;  
}
```

.class selectors

Styles elements with a class attribute. Elements can be given multiple classes, and classes can be used throughout the page to style similar elements.

Code example:

```
.intro {  
  color: #00ff00;  
  background-color: #e6e6e6;  
}
```

#id selectors

Styles the element with the specified id. Ids should be unique and each only assigned to one element on the page.

Code example:

```
#firstname {  
  color: rgb(255, 255, 0);  
}
```

Inline style

Elements can also be styled with an inline style attribute

Code example:

My mother has blue eyes.

font-family

Specifies the font, can hold several font names as a “fallback.” The generic font names are sans-serif, serif, fantasy, monospace, and cursive.

Code example:

```
body {  
  font-family: "Times New Roman", serif;  
}
```

font-size

Sets the font size, in pixels, percentage, or relative sizing.

Code examples:

```
body {  
  font-size: 12px;  
}  
h1 {  
  font-size: 2em;  
}
```

font-weight

Sets characters to degrees of thickness. 400 is normal, and 700 is the same as bold.

Code examples:

```
h1 {  
  font-weight: bold;  
}  
h2 {  
  font-weight: 700;  
}
```

font-style

Can be used to sets the font style to italic or normal.

Code example:

```
p {  
  font-style: italic;  
}
```

text-decoration

Decoration added to text

Code example:

```
h3 {  
    text-decoration: underline;  
}  
a {  
    text-decoration: none;  
}
```

text-align

Sets the horizontal alignment of text in an element.

Code example:

```
.username {  
    text-align: right;  
}
```

line-height

Sets the height of a line in pixels, percentage, or relative sizing

Code example:

```
h1 {  
    line-height: 24px;  
}
```

CSS comments

```
/* this is a comment */  
.intro {  
    /* comments can contain text or "commented out" code  
    color: #f0f0f0; */  
    background-color: rgb(100, 0, 200);  
}
```

width and height

Set the width and/or height of an element. Values are a percent or a number followed by a length unit, like 100% or 200px.

Code examples:

```
img {  
  width: 100%;  
}
```

```
div {  
  height: 200px;  
}
```

overflow

overflow-x (horizontal)

overflow-y (vertical)

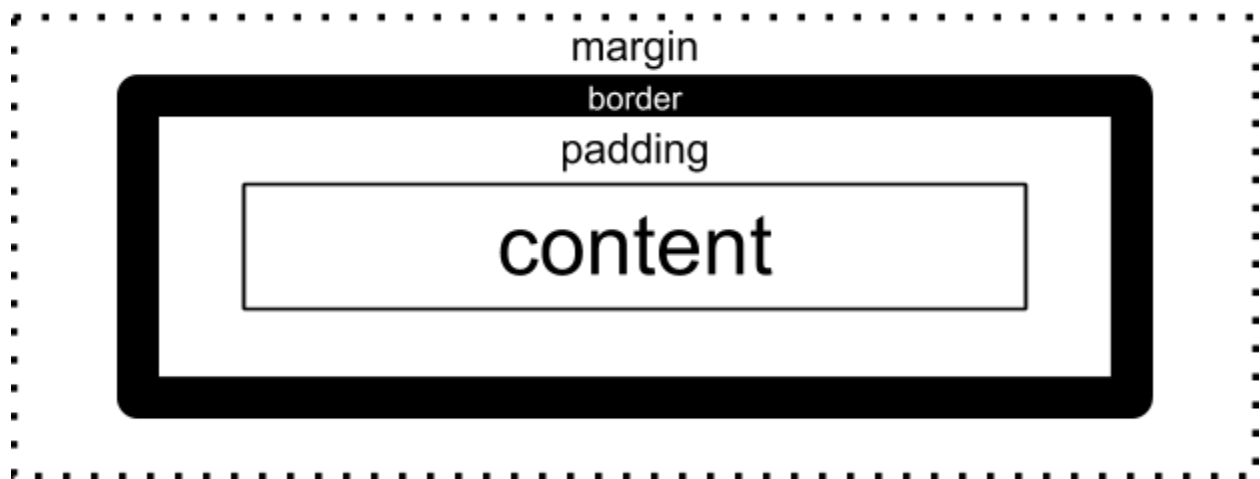
When content overflows its box, specify whether it should be hidden or to add scrollbars when an element's content is too big to fit. If setting overflow to anything other than its default value of `visible`, use `auto` to only show the scrollbars when the content does actually overflow, and only set overflow in the direction where you expect a user to scroll. Avoid scrollbars within scrollbars.

visible	default, content may flow outside its box
hidden	content is clipped, no scrolling
scroll	content is clipped, scrollbars always show
auto	clipped with scrollbars only if it overflows

Code examples:

```
p {  
  height: 200px;  
  overflow-y: auto;  
}
```

CSS box model



margin

`margin-top margin-right margin-bottom margin-left`

Margin is the transparent area around the box that separates the box from other elements. Can also be applied individually to the top, left, bottom, or right margins.

border (shorthand)

`border-width border-style border-color`

`border-top border-right border-bottom border-left`

Sets the border width, style, and color. Can also be applied individually to `border-style`, `border-width`, `border-color`, or to the top, left, bottom, or right borders. Border style options are none, dotted, dashed, solid, double, groove, ridge, inset, or outset.

padding

`padding-top padding-right padding-bottom padding-left`

Padding is the space between the border and the content. Can also be applied individually to the top, left, bottom, or right padding.

Code example:

```
div {  
  margin: 40px;  
  padding: 20px;  
  border: 5px solid red;  
}
```

background-image

Sets a background image for an element. It's a good idea to always set a `background-color` as well to be used in case the image is unavailable. By default, a background-image is placed at the top-left corner of an element, and repeated both vertically and horizontally. Set the `background-position`, `background-size`, and `background-repeat` to tile, stretch, or fit the background image to the element as desired.

background-position

Sets the starting position of a background image.

`xpos` can be `left`, `right`, `center`, or a CSS unit like `50%` or `20px`

`ypos` can be `top`, `bottom`, `center`, or a CSS unit like `50%` or `20px`

background-repeat

Sets if/how a background image will be repeated.

`background-repeat: repeat;` (default) image is repeated both ways

`background-repeat: repeat-x;` image is repeated horizontally

`background-repeat: repeat-y;` image is repeated vertically

`background-repeat:` image is not repeated

`no-repeat;`

background-size

Specifies the size of the background images using keywords `auto`, `cover`, or `contain` or with CSS width and height units like `50%` or `200px`

`background-size: auto;` (default) image is shown in its original size

`background-size: cover;` resized, stretched, clipped to cover the container

`background-size: contain;` resized to fit the container but not stretched to fill it

`background-size: 50% 50%;` resized to 50% of the width and height

`background-size: 50px;` width is 50px and height is auto

Code example:

```
body {  
    background-image: url('image.gif');  
    background-position: center;  
    background-repeat: no-repeat;  
    background-size: contain;  
}
```

}