

2022, June 26, Sunday
Introduction to Web Development
-Raju Mandal

Introduction to CSS

-assisted byDeshant Devkota

CSS is extremely powerful

CSS Cheat Sheet

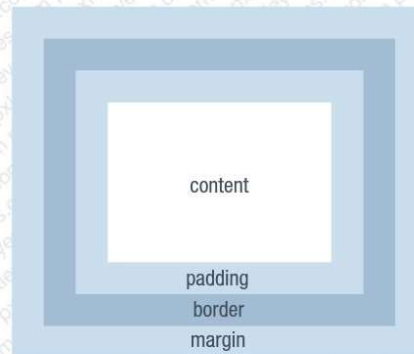
CSS Cheat Sheet

brought to you by pxleyes.com

Selectors

div	all DIV tags
div, span	all DIV tags and all SPAN tags
div span	all SPAN tags inside DIVs
#content	element with ID "content"
.box	all elements with CLASS "box"
ul#box	UL tag with ID "box"
span.box	all SPAN tags with CLASS "box"
*	all elements
#box *	all elements inside #box
a:link, a:active,	links in normal state, in clicked state,
a:visited	and in visited state
a:hover	link with mouse over it
div > span	all SPANs one-level deep in a DIV

Box Model



Positioning

position	places elements on screen, e.g. absolute, fixed, relative
float	stacks elements horizontally in a particular direction, e.g. left
top, left, right, bottom	specifies the offsets used in absolute, fixed, and relative positions, e.g. top:10px;left:10px
display	sets how the element is placed in the doc flow, e.g. block, inline, none
z-index	sets the stacking order of elements, e.g. z-index of 1 is below z-index of 2
overflow	sets what happens to content outside of container, e.g. auto, hidden

Text

font-family	font used, e.g. Helvetica, Arial
font-size	text size, e.g. 60px, 3em
color	text color, e.g. #000, #abcdef
font-weight	how bold the text is, e.g. bold
font-style	what style the text is, e.g. italic
text-decoration	sets a variety of effects on text, e.g. underline, overline, none
text-align	how text is aligned, e.g. center
line-height	spacing between lines, e.g. 2em
letter-spacing	spacing between letters, e.g. 5px
text-indent	indent of the first line, e.g. 2em
text-transform	applies formatting to text, e.g. uppercase, lowercase, capitalize
vertical-align	align relative to baseline, e.g. text-top

Borders and Lists

border	sets border style for all borders, in the format: border: (solid, dashed, dotted, double) (width) (color), e.g. border: solid 1px #000
border-top	sets border style for a specific border (same property syntax used for padding and margin, e.g. margin-left)
border-bottom	
border-left	
border-right	
list-style-type	sets style of bullets, e.g. square
list-style-position	sets how text wraps when bulleted, e.g. outside, inside
list-style-image	sets an image for a bullet, e.g. list-style-image:url(bullet.png)

Everything Else

background	sets background of an element, in the format: background: (color) (image) (repeat) (position), e.g. background: #000 url(bg.png) repeat-x top left
cursor	sets shape of cursor, e.g. pointer
outline	a border drawn around an element that doesn't affect the box model
border-collapse	sets how borders within tables behave, e.g. collapse
clear	sets on what side a new line starts in relation to nearby floated elements, e.g. left, right, both

Always write <!doctype html> in your files!

CSS Introduction:

CSS stands for Cascading Style Sheets.

HTML provides a skeleton for your webpage while CSS provides styling.

Why is it called Cascading?

Because the styles are cascaded. When you give multiple styles to different elements they all combine to create a virtual style sheet.

CSS History:

Håkon Wium Lie developed Cascading Style Sheets while working with Tim Berners-Lee and Robert Cailliau at CERN in 1994. There are versions of CSS. Version 1 was released in 1996 with CSS2 following in 1991 and so on. CSS3 was released in 1999.



CSS Syntax

```
selector{  
  property:value; /*declaration:*/  
  property:value;  
}
```

Example code:

```
p{  
  color:red;  
}
```

Don't ask why there is a semicolon though, it's a thing we programmers like to do, put semicolons at the end. It also makes reading through easier both for us and the compiler.

The selector is a DOM element. Or simply a thing in HTML. Example shows tag name for element <p> has been used as a selector.

The declaration block contains one or more declarations separated by semicolons. Each declaration has a property name and value separated by a colon.

Declaration blocks are encased in curly braces.

CSS Comments

Comments are those lines of code that aren't used by computers but are helpful to programmers who study the code. Different programming languages have different syntax for comments.

Example code: A comment in HTML:

```
<!-- This is an HTML comment -->
```

Example code: A comment in CSS:

```
<style>
```

```
/* This is a CSS comment */
```

```
p{  
  color: red;  
}
```

```
</style>
```

Do you know about inline, internal, and external CSS?

Inline CSS

Inline CSS is applied directly to the HTML tag using the style attribute to an HTML tag.

Example Code:

```
<h1 style="color:blue;text-align:center;">This is a heading</h1>
```

Internal CSS

Internal CSS is defined using the <style> element inside the head section of a page to which the CSS is applied. Use it to style a page uniquely.

Example Code:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: linen;
}

h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

External CSS

External CSS is applied by linking an external '.css' file to the HTML file. Use it to style a website.

Example Code: myweb.html:

```
<!DOCTYPE html>
<html>
```

```
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Example Code: mystyle.css:

```
body {
  background-color: lightblue;
}

h1 {
  color: navy;
  margin-left: 20px;
}
```

Here we have linked a CSS file where the style is specified using the <link> element at the top. 'myweb.html' uses 'mystyle.css' for styling.

CSS Selectors

Selector selects a thing in HTML or a HTML element.

We can exploit DOM relationships and pseudo relationships to select elements. Based on this CSS selectors can be divided into five categories:

- Simple selectors
- Combinator selectors
- Pseudo-class selectors
- Pseudo-element selectors
- Attribute selectors

'*' is called a CSS universal selector which selects all HTML elements.

Example Code:

```
*{
  margin: 0;
}
```

Similarly, you can group selectors using commas.

Example Code:

```
p, div {  
  background-color: red;  
}
```

Simple selectors

Based on tag/element name, id, and class.

Example Code:

```
p{  
  text-align: center;  
}
```

```
.center{/*selects all elements with class center*/}  
p.center{/*selects all <p> elements with class center*/}  
#myid{/*selects all elements with id myid*/}
```

Combinator selectors

Combinator selectors explain the relationship between selectors. Used to combine simple selectors.

There are four types of combinators in css:

- Descendent Selector:
div p {/ All <p> elements inside <div> element. i.e: All descendants*/}*
- Child Selector:
*div > p {/*All child <p> elements inside <div> element. i.e: Grandchilds won't be selected */}*
- Adjacent Sibling Selector:
div + p {/<p> elements immediately after <div>*/}*
- General Sibling Selector:
div ~ p {/ <p> elements after <div>*/}*

Example Code: Guess what this does:


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

Pseudo-class selectors

Pseudo-class selectors define a special state of an element like mouseover, visited, focus, etc.

Syntax:

```
selector:pseudo-class { }
```

Example Code:

```
a:visited {  
  color: red;  
}
```

```
div:hover { }  
a:active { }
```

Example Code: Try this:

```
img { display: none; }  
div:hover img { display: block; }/* place a <img> element inside <div>*/
```

Pseudo-element selectors

Pseudo-element is used to style specified parts of an element. eg: First letter, First line, before and after.

Syntax:

```
selector::pseudo-element { }
```

Example Code:

```
p::first-line { /*Separated by . */ }  
p::first-letter { }
```

Example Code: Try this to understand before and after:

```
h1::before{  
  content: url(smiley.gif);  
}
```

Try also:

```
::marker { /*Markers of list*/ }, ::selection
```

Attribute selectors

Attribute selectors select HTML elements with specific attributes.

Syntax:

`selector[attribute] { }`

`selector[attribute="value"] { }` and more.

CSS Colors:

Colors can be specified using 140 predefined color names or RGB, HEX, and other formats.

Try: Red, Blue, Yellow, Orange, Tomato, Dodgerblue, Violet

You can set color for text, backgrounds, and borders.

You can also set transparency.

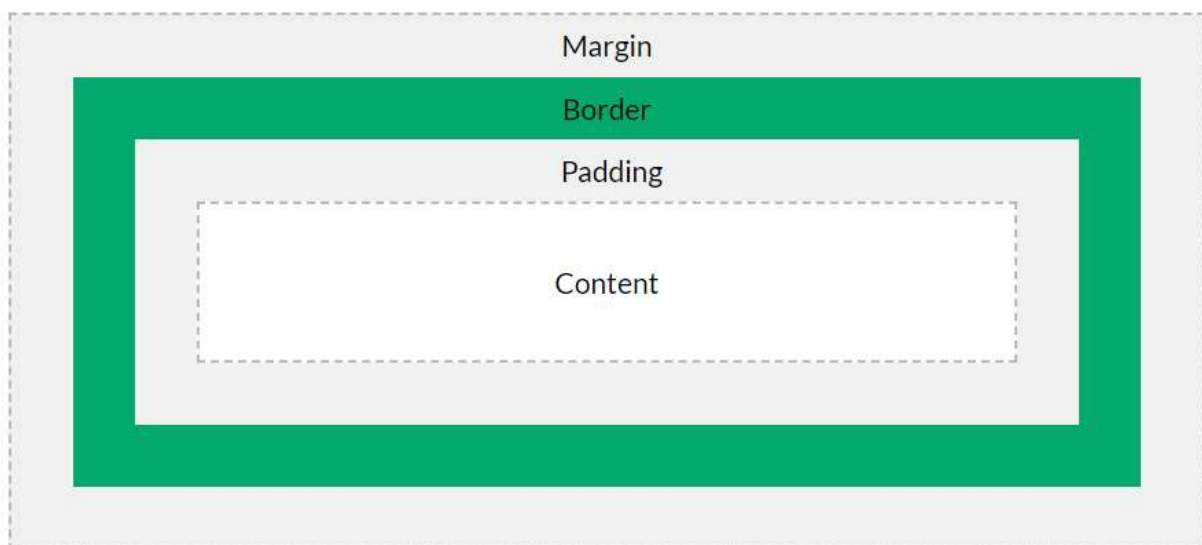
Example Code:

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
```

See this on how to set transparency:

```
<h1 style="background-color:rgba(255, 99, 71, 0.5);">...</h1>
```

CSS Box Model: Margin, Border, Padding, Content



Source: w3schools.com/css/css_boxmodel.asp

All HTML elements can be considered as a box. The CSS Box model is used to design the spacing and border inside, around, and outside this box. It consists of content, padding, border, and margin.

Content: The content of the box, where text and images appear.

Padding: Around the content or between the content and the border.

Border: Goes around the content and padding.

Margin: Outside the border.

Example Code: Calculate the area of this div element:

```
div {  
    width: 320px;  
    padding: 10px;  
    border: 5px solid gray; /* you can separately style for border width, style, color or use pointers  
to side*/  
    margin: 0;  
}
```

Also, try:

```
div {  
    width: 320px;  
    padding: 10px;  
    border-color: gray;  
    border-style: solid;  
    border-width: 2px;  
    margin: 0;  
}
```

You can specify directions for margins, border, and padding using top, bottom, left, and right. Try border-top, padding-left, and so on.

Similarly, the use of one, two, three, or four values can be used to specify these values in different directions.

Four Values:

```
selector {  
    margin: top right bottom left;  
}
```

```
p {  
    margin: 25px 50px 75px 100px;  
}
```

Three Values:

```
selector {  
    margin: top right-and-left bottom;  
}
```

Two Values:

```
selector {  
  margin: top-and-bottom right-and-left;  
}
```

One Value:

In all four directions.

Try also:

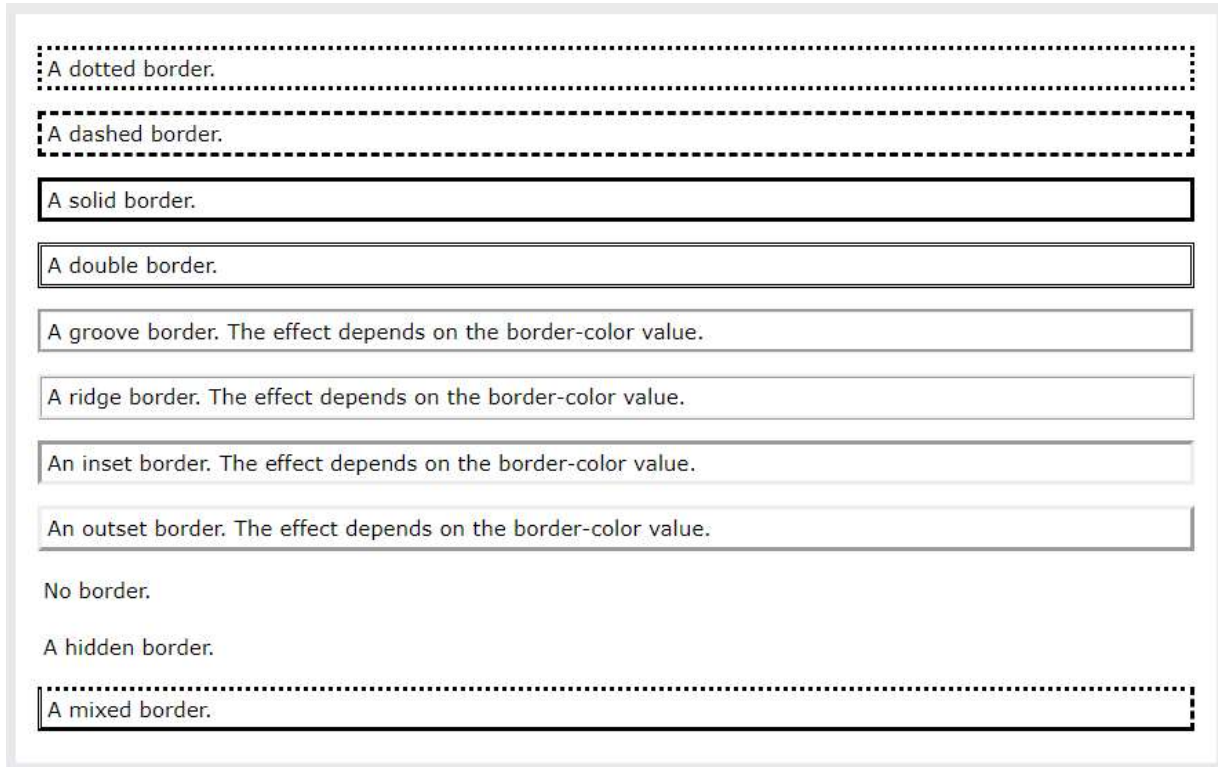
```
selector {  
  border-color: red green blue yellow;  
}
```

How to center a div?

```
div {  
  width: 300px;  
  margin: auto;  
  border: 1px solid red;  
}
```

Different border-style

You can have different styles for the border as shown in the figure below:



Source: https://www.w3schools.com/css/css_border.asp

Example Code:

```
div.dotted{  
  border-style: dotted;  
}
```

```
div.mixed{  
  border-style: dotted dashed;/* use one to four values, same as in colors */  
}
```

CSS icons

To add icons you can use icons libraries like font awesome, bootstrap, or google.

1. First link the library using either link or script.
2. Then use classes provided by the library to insert icons.
3. Use CSS to target those classes and change styles like the size and color of icons.

Example Code:

```
<head>
<script src="https://kit.fontawesome.com/bd14ae6de41948.js"
crossorigin="anonymous"></script>
</head>
```

Or

```
<head>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
</head>
```

Or

```
<head>
<link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
</head>
```

Font Awesome icons

Register at fontawesome.com using email to get a code and embed it in the document head to be able to use font awesome icons.

See: <https://fontawesome.com/search>

Example Code:

```
<!DOCTYPE html>
<html>

<head>
  <title>Font Awesome Icons</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <script src="https://kit.fontawesome.com/bd14ae6de41948.js"
crossorigin="anonymous"></script>
  <!-- Use your code from font awesome -->
  <!-- About meta tags, inserted here because it's common to see this tags everywhere,
vscode has a shortcut template with meta tags used in this document examples -->

  <style>
    .fas.fa-cloud.style1{
      font-size: xx-large;
    }
  </style>
</head>

<body>

  <p>Inserting Font Awesome icons:</p>
  <i class="fas fa-cloud"></i>
  <i class="fas fa-heart"></i>

  <p>Styling Font Awesome icons with inline css (size and color):</p>
  <i class="fas fa-heart" style="font-size:24px;"></i>
  <i class="fas fa-heart" style="font-size:36px;"></i>
  <i class="fas fa-heart" style="font-size:48px;color:red;"></i>
  <i class="fas fa-heart" style="font-size:60px;color:lightblue;"></i>

  <p>Styling Font Awesome icons with internal css (size and color):</p>
  <i class="fas fa-cloud style1"></i>

</body>
```


</html>

Output:

Inserting Font Awesome icons:



Styling Font Awesome icons with inline css (size and color):



Styling Font Awesome icons with internal css (size and color):



CSS Position

You can position HTML elements using CSS and this position can be specified in various ways.

To use the positioning you must first specify the type of positioning and then use left, right, top, and left to specify the actual position parameters.

Types of position:

- static: normal positioning, not affected by top, bottom, left, and right
- relative: in relation to the normal position, leaves a gap and will obstruct other elements. (will see about the z-index later)
- fixed: relative to the viewport, thus always stays in the same place and doesn't scroll. No gap where it should have been located.
- absolute: relative to the nearest positioned ancestor.
- sticky: based on scroll positions. Use top and bottom. Scrolls until the given offset position is met then sticks to place.

Example Code:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS Positioning</title>

  <style>
    div.sticky {
      position: sticky; /*Change this property value to different options to see changes */
      bottom: 20px;
      top: 10px;
      padding: 5px;
      background-color: #cae8ca;
      border: 2px solid #4CAF50;
    }
  </style>
</head>

<body>

  <div style="padding-bottom:2000px;"></div>
```

```
<div class="sticky">I am sticky!</div>

<div style="padding-bottom:2000px;"></div>

</body>

</html>
```

CSS Z-index

Z-index is all about “ko agi, ko paxi”.

Think of elements occurring as a stack.

You have come to an issue where the positioning of the elements overlaps each other.

Think of this case: you place an image and some texts in a div and want the image to occur as a background in the stack.

Such properties are specified using the z-index. Z-index is the order of the stack which can be negative or positive. An element with a higher z-index will appear at the front or above.

Example Code: Source w3schools:

```
<!DOCTYPE html>
<html>
<head>
<style>
.container {
  position: relative;
}

.black-box {
  position: relative;
  z-index: 1;
  border: 2px solid black;
  height: 100px;
  margin: 30px;
}

.gray-box {
  position: absolute;
```

```
z-index: 3; /* gray box will be above both green and black box */
background: lightgray;
height: 60px;
width: 70%;
left: 50px;
top: 50px;
}
```

```
.green-box {
  position: absolute;
  z-index: 2; /* green box will be above black box */
  background: lightgreen;
  width: 35%;
  left: 270px;
  top: -15px;
  height: 100px;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Z-index Example</h1>
```

<p>An element with greater stack order is always above an element with a lower stack order.</p>

```
<div class="container">
  <div class="black-box">Black box (z-index: 1)</div>
  <div class="gray-box">Gray box (z-index: 3)</div>
  <div class="green-box">Green box (z-index: 2)</div>
</div>
```

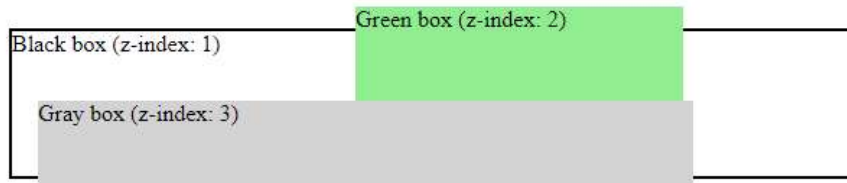
```
</body>
```

```
</html>
```

Output:

Z-index Example

An element with greater stack order is always above an element with a lower stack order.



How to set an image as a background

You can either use the `background-image` property to set an image as a background or make use of the `z-index`.

Using background-image property

Example Code:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Background Image</title>

<style>
  * {
    margin: 0;

  }

  body {
    background-image: url('bg.jpg');
    display: flex; /* Will see later */
    height: 100vh;
    align-items: center;
    justify-content: center;
  }
}
```

```

    div {
      padding: 100px;
      background-color: rgba(188, 209, 228, 0.5);
      color: white;
      font-size: xx-large;
    }
  </style>
</head>

<body>
  <div>
    <p>Lorem, ipsum dolor sit amet consectetur adipiscing elit. Consequatur, commodi!</p>
  </div>
</body>

</html>

```

Using z-index

Example Code:

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Background Image</title>

  <style>
    *{
      margin: 0;
    }

    div {
      padding: 100px;
      background-color: rgba(188, 209, 228, 0.5);
      color: white;
      font-size: xx-large;
    }

    img{
      width: 100%;

```

```

        position: absolute;
        z-index: -1;
        top: 0;
        left: 0;
    }
</style>
</head>

<body>
    <div>
        
        <p>Lorem, ipsum dolor sit amet consectetur adipiscing elit. Consequatur, commodi!</p>
    </div>
</body>

</html>

```

There are many things and properties being used in this example, break them down and it feels simple.

CSS Scroll Bars

CSS can target many things. One of which is the scroll bar.

Use the pseudo-element '::-webkit-scrollbar' selector to select the scrollbar and its elements as given in the example below:

Example Code:

```

<!DOCTYPE html>
<html>

<head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <style>
        /* width */
        ::-webkit-scrollbar {
            width: 10px;
        }

        /* Track */
        ::-webkit-scrollbar-track {

```

```
        background: #f1f1f1;
    }

    /* Handle */
    ::-webkit-scrollbar-thumb {
        background: rgb(25, 47, 211);
        border-radius: 100px;
    }

    /* Handle on hover */
    ::-webkit-scrollbar-thumb:hover {
        background: #555;
    }
</style>
</head>

<body>

    <div style="padding: 1000px;"> HELLO </div>

</body>

</html>
```