CSS

Agenda

- What is CSS?
- Tools
- CSS Structure
- Quiz
- Assignment

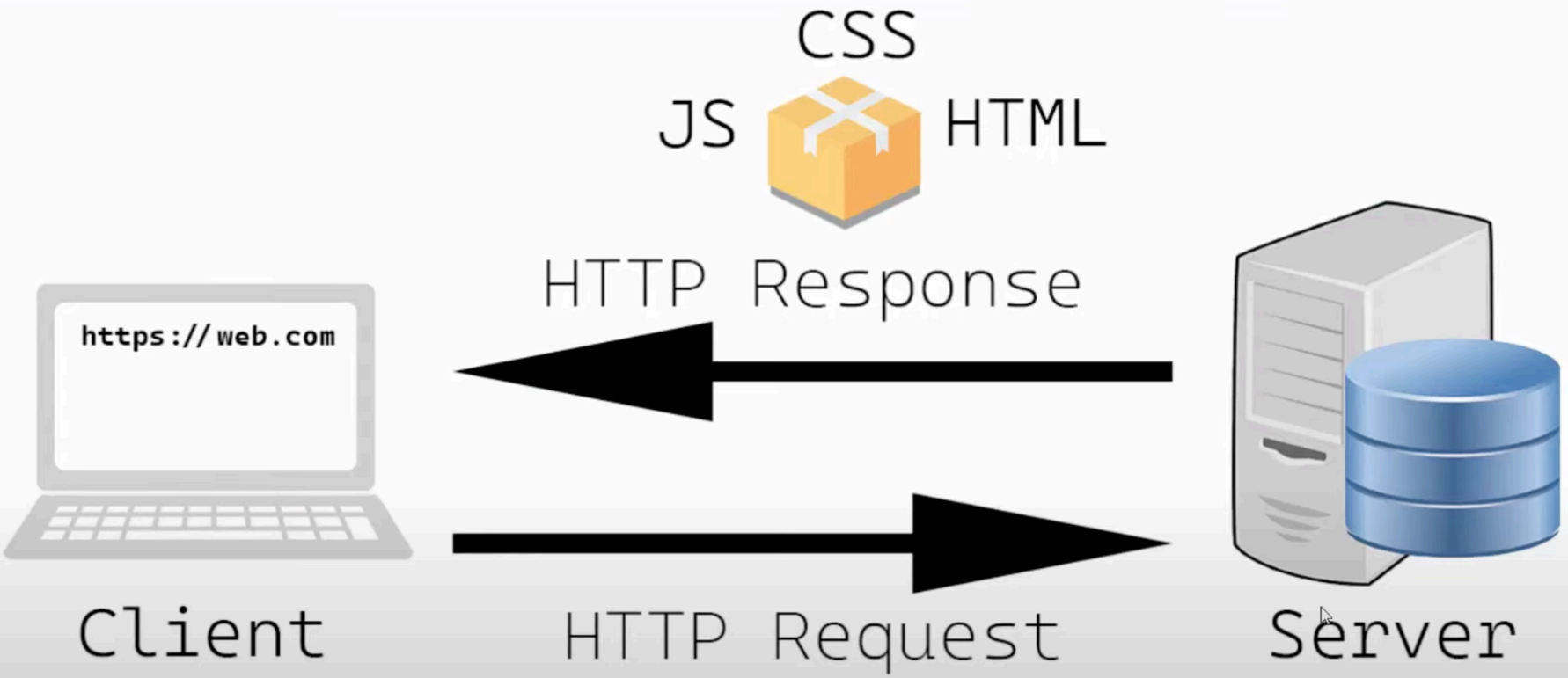
What Is CSS?

What Is CSS?

- CSS stands for Cascading Style Sheet
- CSS is **not** a programming language
- CSS is used for styling web pages
- CSS is versatile

What Is CSS? - Syntax

```
selector {
    property1: value;
    property2: value
}
```


CSS
JS
HTML
HTTP Response
https://web.com
Client
HTTP Request
Server

Tools needed

Tools

- Visual Studio Code – editor
  - Prettier – formatting tool
  - LiveServer
- Chrome / Firefox

CSS Structure

CSS Structure

CSS applies on HTML elements, on classes and on ids

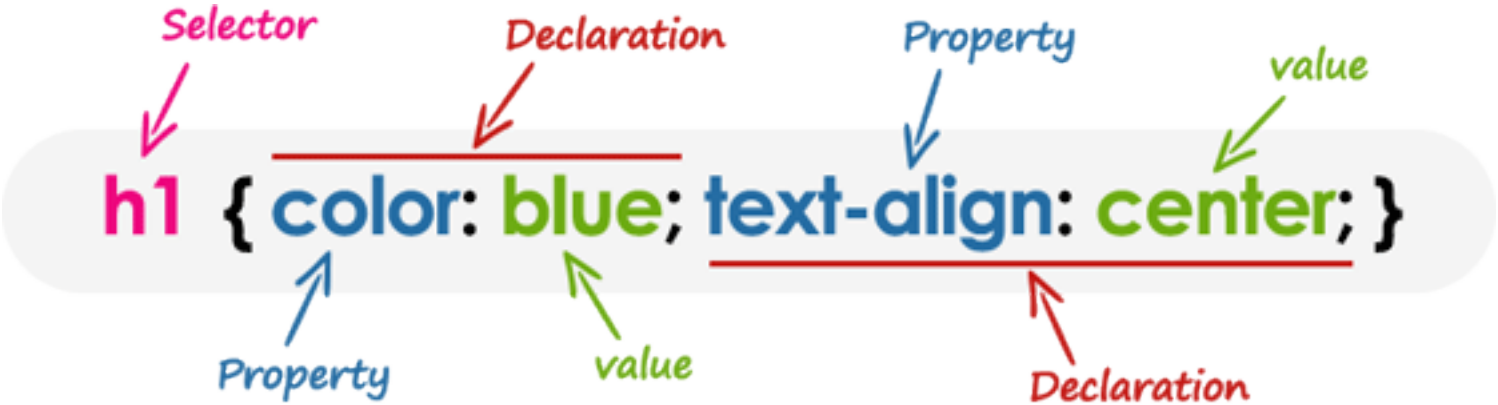
- HTML elements – well HTML elements
- Classes refer to a number of elements with same style
- IDs are unique in a page

CSS Structure - Possibilities

There are:

- Inline styles
- Styles in files

CSS Structure – Syntax



CSS Structure – HTML elements, Class and IDs Selectors

```
h1 {
    color: blue;
}
.title {
    color: teal;
}
#my-unique-title {
    color:  magenta;
}
```

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

CSS Structure – Universal Selector

The universal selector, denoted by an asterisk (*), matches every single element on the page.

```
* {
    color: blue
}
```

CSS Structure –Element Type Selectors

An element type selector matches all instance of the element in the document with the corresponding element type name.

```
h1 {
    color: blue;
}
```

```
p {
    color: blue;
}
```

```
div {
    font-size: 12px;
}
```

CSS Structure – Grouping selectors

An element type selector matches all instance of the element in the document with the corresponding element type name.

```
h1 {
   font-size: 36px;
   font-weight: normal;
}
h2 {
   font-size: 28px;
   font-weight: normal;
}
h3 {
   font-size: 22px;
   font-weight: normal;
}
```

```
h1, h2, h3 {
      font-weight: normal;
}
h1 {
   font-size: 36px;
}
h2 {
   font-size: 28px;
}
h3 {
   font-size: 22px;
}
```

CSS Structure –ID Selectors

The id selector is used to define style rules for a *single* or *unique* element.
The id selector is defined with a hash sign (#) immediately followed by the id value.

```
#error {
    color: red;
}
```

CSS Structure –Descendant Selectors & example

You can use these selectors when you need to select an element that is the descendant of another element, for example, if you want to target only those anchors that are contained within an unordered list, rather than targeting all anchor elements.

```
ul.menu li a {
   text-decoration: none;
}
h1 em {
   color: green;
}
```

CSS Structure – Child Selectors & example

A child selector is used to select only those elements that are the direct children of some element. A child selector is made up of two or more selectors separated by a greater than symbol (>).

```
ul > li {
   list-style: square;
}
ul > li ol {
   list-style: none;
}
```

CSS Structure – Adjacent Sibling Selectors

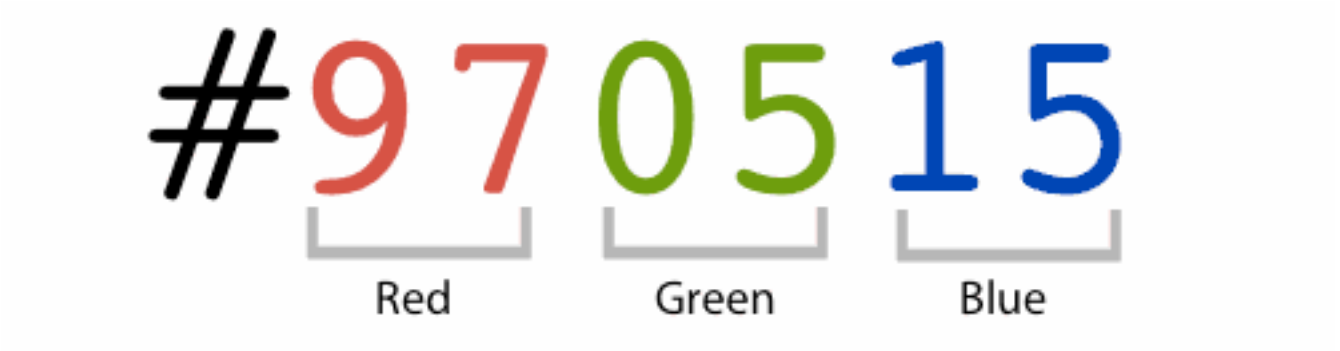
```
h1 + p {
     color: blue; font-size: 18px;
}
ul.task + p {
     color: #f0f;
     text-indent: 30px;
 }
```

CSS Structure – Colors - Hexadecimal & RGB(A)

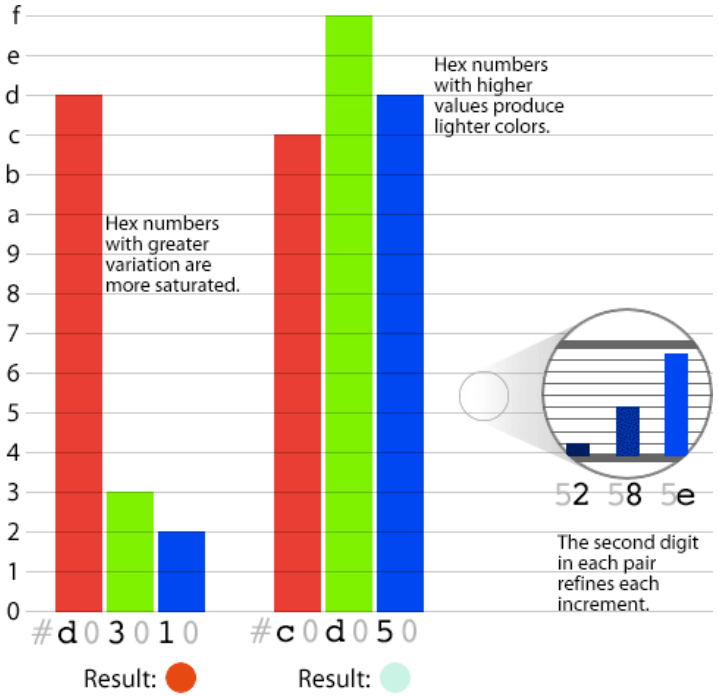
Colors can be represented as hexadecimal

Numbers – 0-9, Letters A-F

#ff00aa;



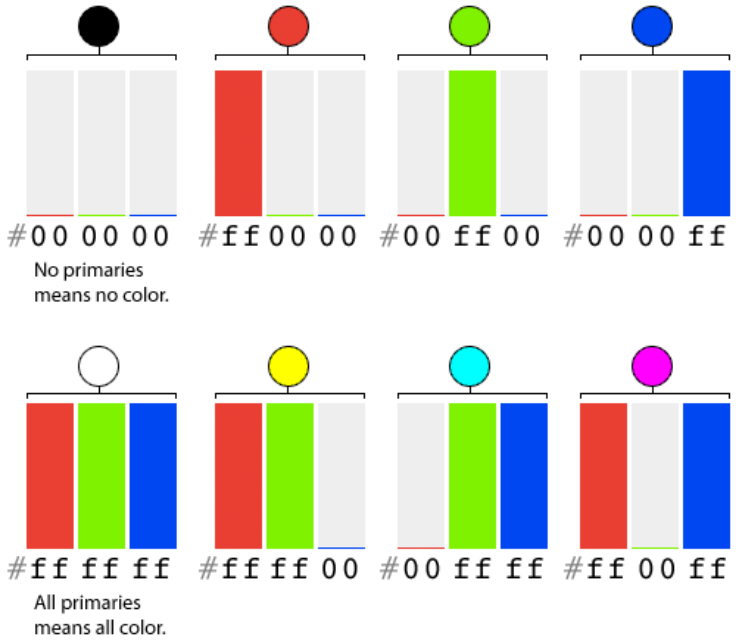
CSS Structure – Colors - Hexadecimal



CSS Structure – Colors – Hexadecimal

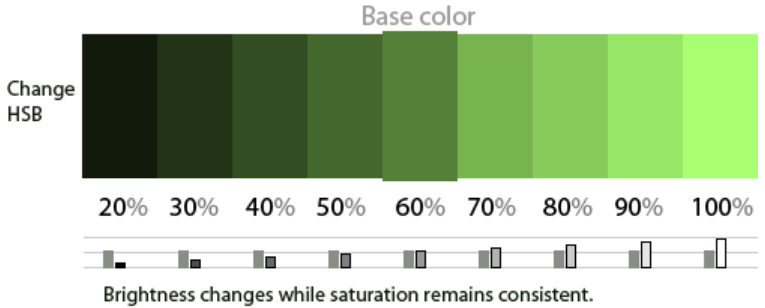
- #000000 is black, the starting point.
- #**ff**0000 stands for the brightest red.
- #00**ff**00 stands for the brightest green.
- #0000**ff** stands for the brightest blue.

CSS Structure – Colors - Hexadecimal

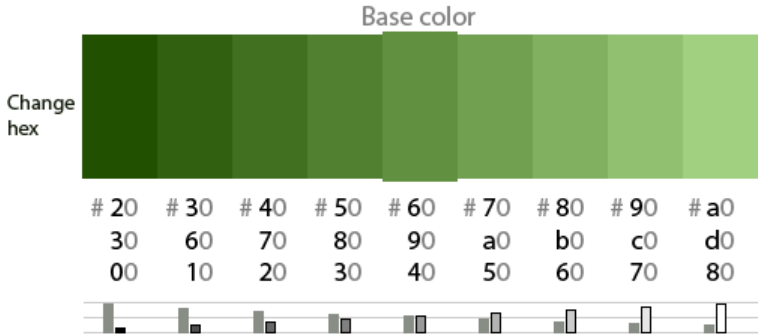


CSS Structure – Colors - Hexadecimal

To brighten or darken a color, one's inclination is often to adjust its brightness. **This makes a color run the gamut from murky to brilliant**, but loses its character on either end of the scale. For example, below a middle green becomes decidedly black when reduced to 20% brightness. Raised to 100%, the once-neutral green gains vibrancy.



Brightness changes while saturation remains consistent.



Brightness increases as saturation drops.

CSS Structure – Colors – RGB(A)

- Red Green Blue (Alpha)
- rgb(255, 255, 255) white
- rgb(0, 0, 0) black
- rgb(255, 0, 0) = ?
- rgb(0, 255, 0) = ?
- rgb(0, 0, 255) = ?
- rgba(255, 100, 100, 0.5)

-CSS Structure – Units & Positions

Units: Px, rem, em, %
Positions: Absolute, Relative

```
.container {
    width: 100px;
    height: 100px;
    background-color: #abcc11;
    padding: 20px;
    margin: 20px;
}
```

```
.contained {
    width: 50%;
     height: 50%;
     border: 1px solid blue;
}
```

CSS Structure – Units - PX

Units: Px, rem, em, %

1 px = 1px

Setting the font size in pixel values (e.g. 14px, 16px, etc.) is a good choice when you need the pixel accuracy. Pixel is an absolute unit of measurement which specifies a fixed length.

CSS Structure – Units - EM

Units: Px, rem, em, %

1 em = xPX relative to parent element

The **em** unit refers to the font size of the parent element. When defining the font-size property, 1em is equal to the size of the font that applies to the *parent of the element*.

So, if you set a font-size of 20px on the body element, then 1em = 20px and 2em = 40px.

However, if you haven't set the font size anywhere on the page, then it is the browser default, which is normally 16px. Therefore, by default 1em = 16px, and 2em = 32px.

CSS Structure – Units - REM

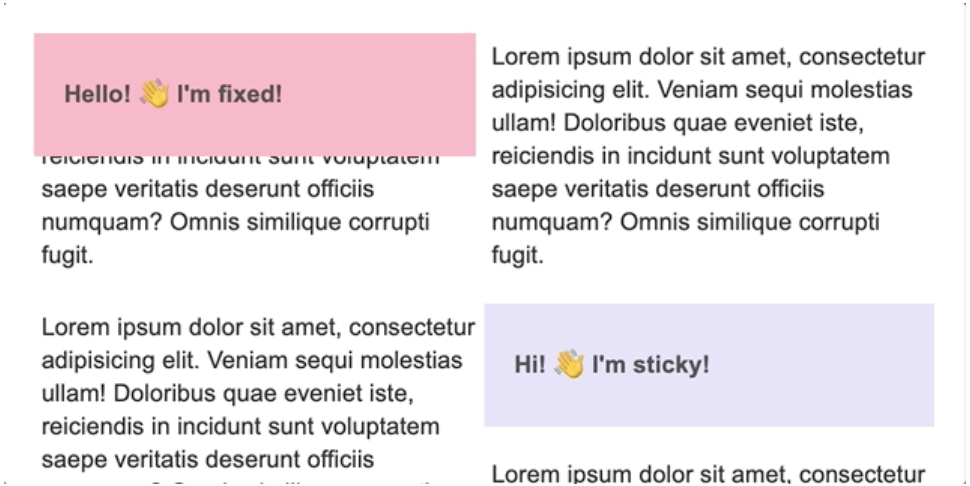
Units: Px, rem, em, %

To make things even more simpler CSS3 has introduced rem unit (short for "root em") which is always relative to the font-size of the root element (html), regardless of where the element lies in the document (unlike em which is relative to parent element's font size).

This means that 1rem is equivalent to the font size of the html element, which is 16px by default in most browsers.

CSS Structure – Position blocks

Positions: Absolute, Relative



CSS Structure – Margins & Paddings

- **Margins** are applied on exterior
- **Paddings** are applied on interior

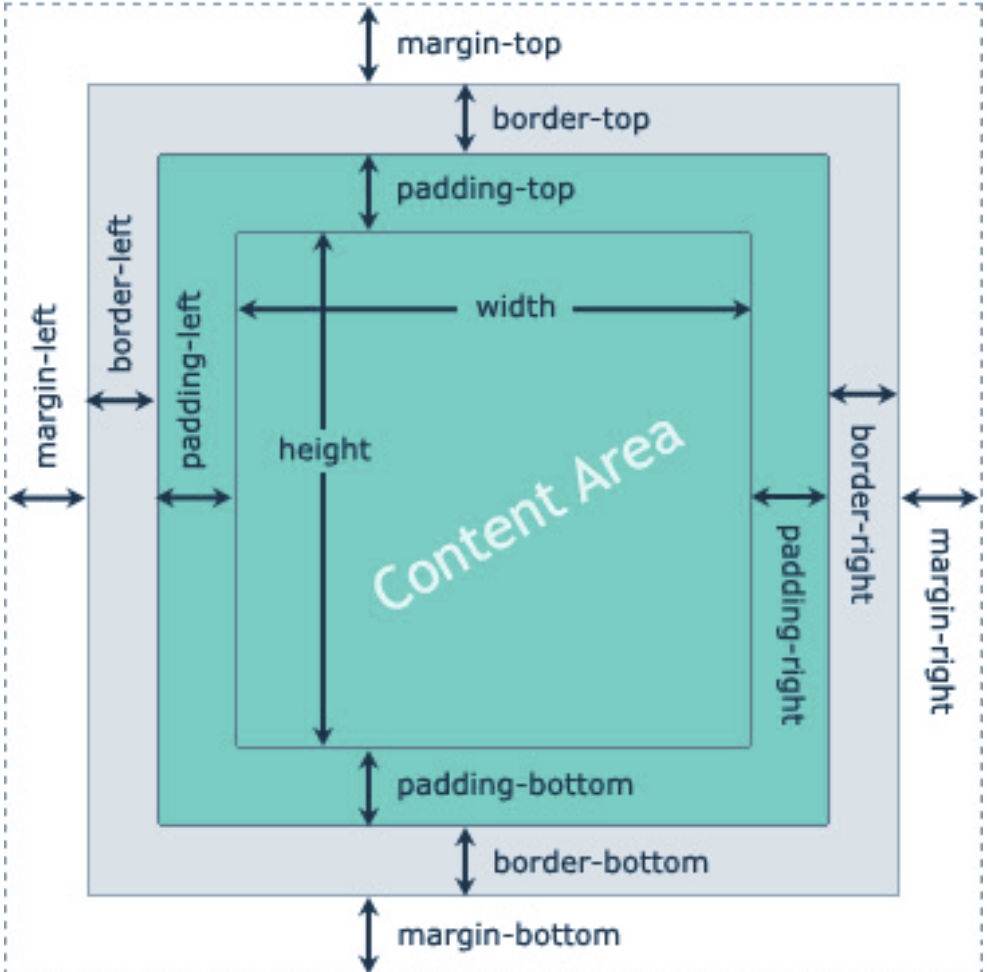
```
.container {
    width: 100px;
    height: 100px;
    background-color: #abcc11;
    padding: 20px;
    margin: 20px;
}
```

CSS Structure – Position blocks – Box Model

Every element that can be displayed on a web page is comprised of one or more rectangular boxes. CSS box model typically describes how these rectangular boxes are laid out on a web page.

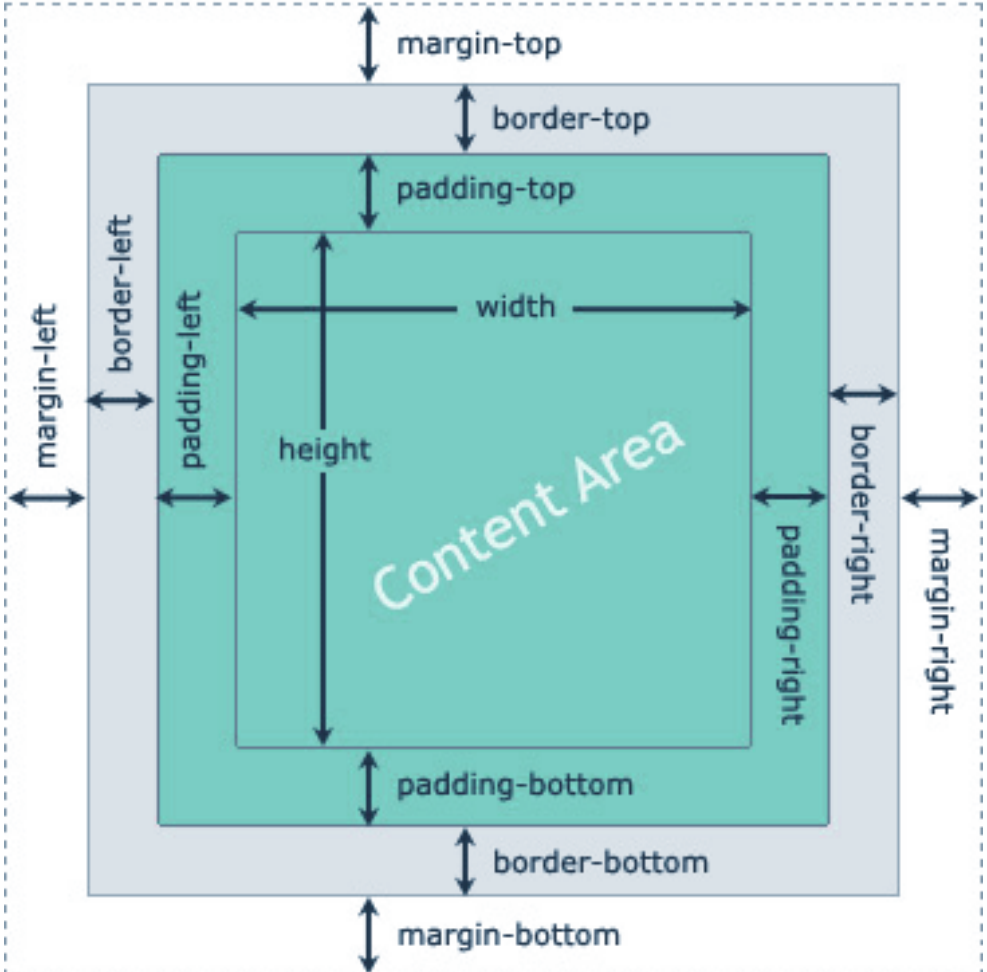
These boxes can have different properties and can interact with each other in different ways, but every box has a ***content area*** and optional surrounding ***padding***, ***border***, and ***margin areas***.

CSS Structure – Box Model



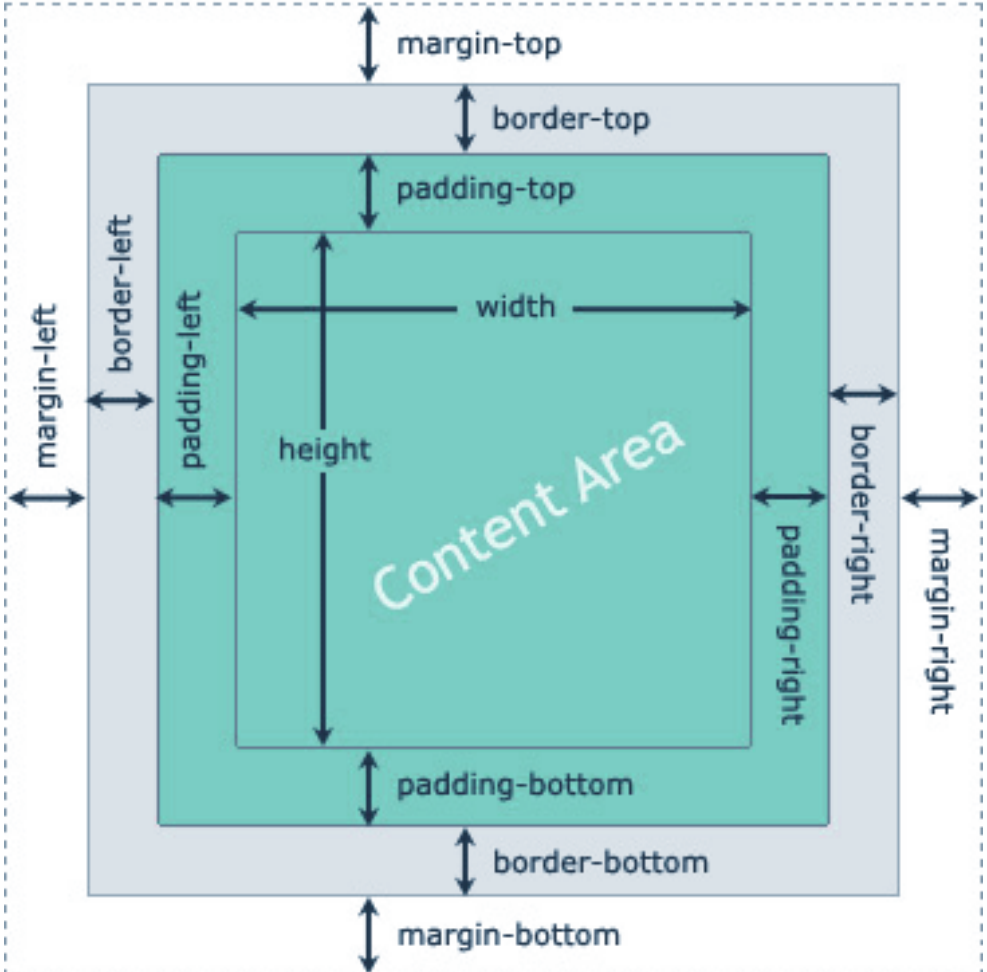
CSS Structure – Box Model

Dimensions



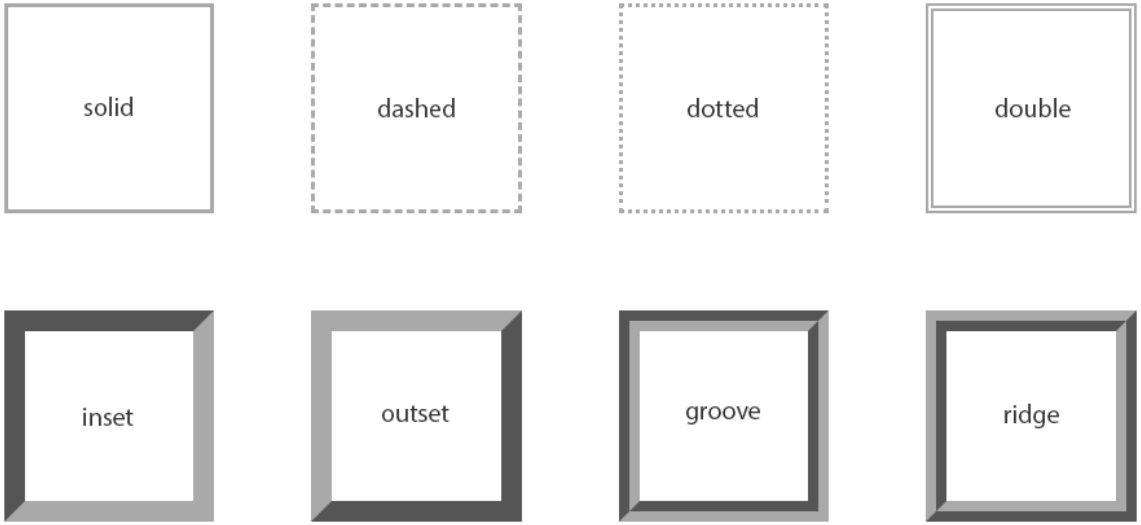
CSS Structure – Box Model

Padding



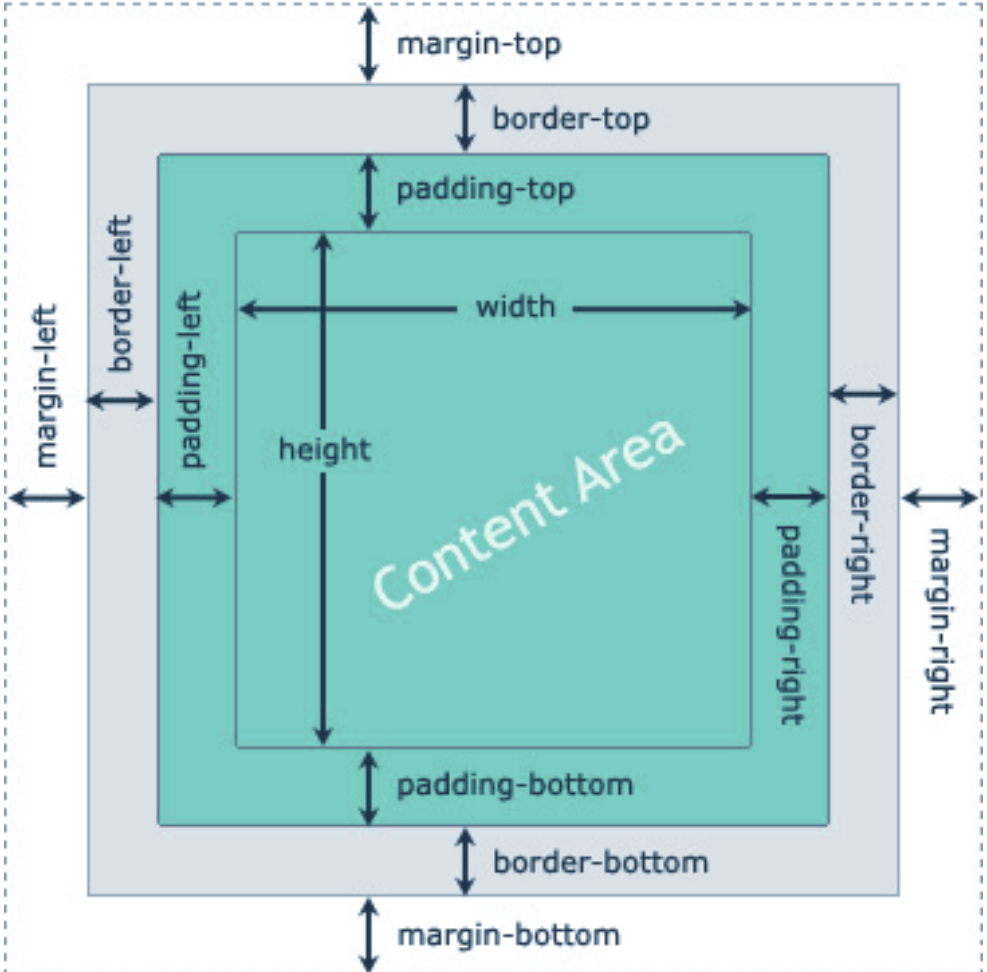
CSS Structure – Box Model

Border



CSS Structure – Box Model

Margin



CSS Structure – Pseudo classes

- nth-child(x);
- :hover
- :link
- :visited
- :active
- :first-child

-CSS Structure – Pseudo style on elements

- ::placeholder
- ::before
- ::after

-CSS Structure – ALL properties

- https://www.tutorialrepublic.com/css-reference/css3-properties.php

-CSS – Further resources

- https://developer.mozilla.org/en-US/docs/Web/CSS
- https://www.w3schools.com/css/
- https://css-tricks.com/
- https://www.tutorialrepublic.com/css-reference/css-color-values.php

Quiz - TBD

Assignment

Assignment - 1

- Using selectors and pseudoselectors and **only** <table> <td> <tr>

Create a chessboard. The board should be alternating colors and an eight-by-eight grid.

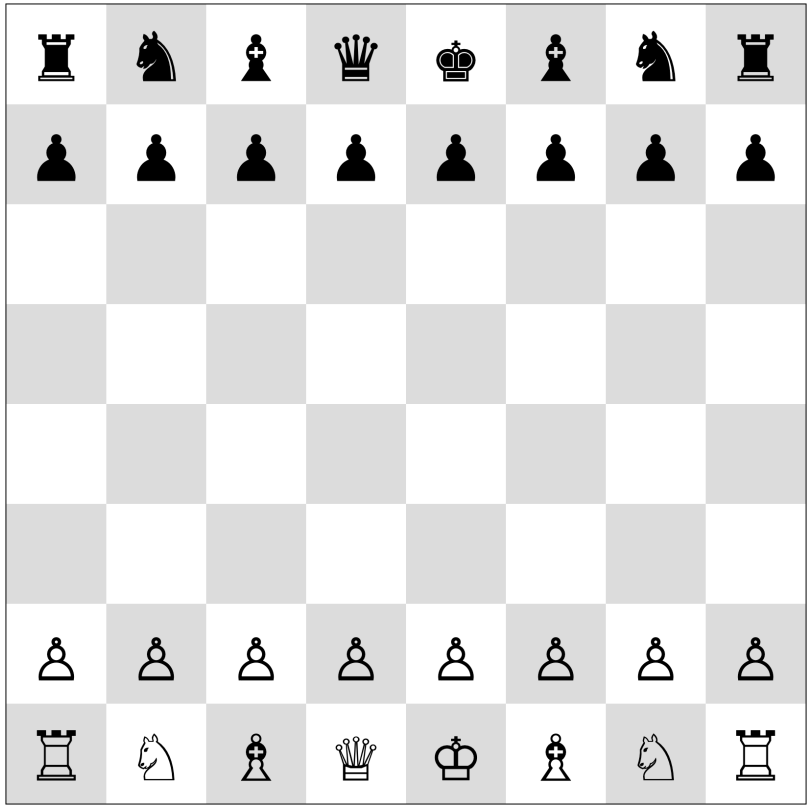
Assignment - 1

Here are the UNICODE characters you'll need for the pieces.

- White King: ♔
- White Queen: ♕
- White Rook: ♖
- White Bishop: ♗
- White Knight: ♘
- White Pawn: ♙

- Black King: ♚
- Black Queen: ♛
- Black Rook: ♜
- Black Bishop: ♝
- Back Knight: ♞
- Black Pawn: ♟

Assignment - 1



Assignment - 2

- Using all you've learned so far build something of your liking using HTML & CSS (e.g. – your CV)
- It should contain a HTML (with as many tags as possible) & CSS
- You can use all the documentation provided here - https://www.tutorialrepublic.com/css-tutorial/ (CSS Basic, CSS Box Model, CSS Advanced)