FRONTEND

CSS

CSS

- CSS (Cascading Style Sheets) is a style sheet language used for describing the presentation of a document written in a **markup language**
- 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**

CSS

- CSS can be added to HTML elements in 3 ways:
  - **Inline** - by using the style attribute in HTML elements
  - **Internal** - by using a <style> element in the <head> section
  - **External** - by using an external CSS file

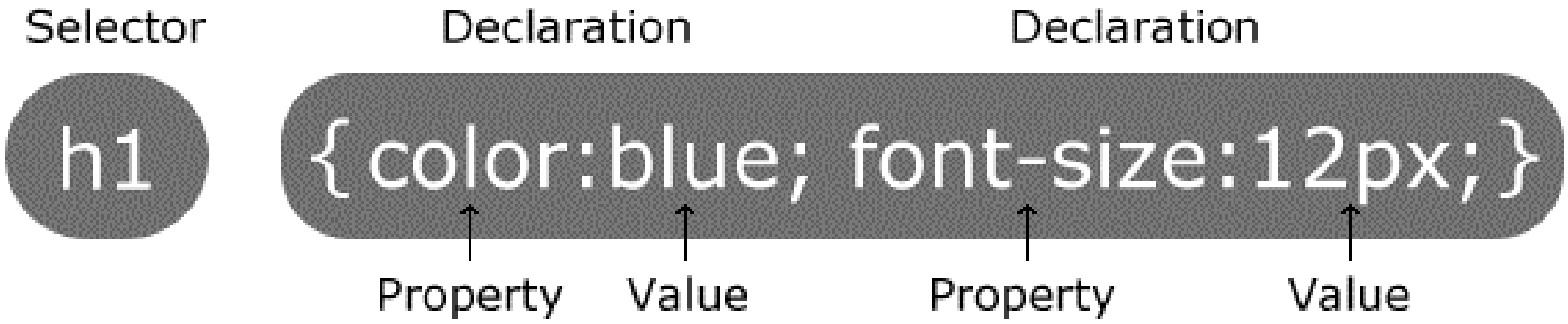
CSS

```
<h1 style="color:blue;">This is a Blue Heading</h1>
```

```
<head>
<style>
    body {background-color: powderblue;}
    h1   {color: blue;}
    p    {color: red;}
</style>
</head>
```

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

CSS Systax



- 3 type of selector
  - Tag Selector
  - Id Selector
  - Class Selector

- Tag selector:
  - The tag selector selects elements based on the tag name.
  - Ex: You can select all <p> elements on a page like this

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

- Id selector:
  - The id selector uses the id attribute of an HTML element to select a specific element.
  - The id of an element should be unique within a page, so the id selector is used to select one unique element!
  - To select an element with a specific id, write a hash (#) character, followed by the id of the element.

```
<p id="para1">This paragraph refers to two
classes.</p>
```

```
#para1 {
    text-align: center;
    color: red;
}
```

- Class selector:
  - The class selector selects elements with a specific class attribute.
  - To select elements with a specific class, write a period (.) character, followed by the name of the class.

```
<p class="center">This paragraph refers to two
classes.</p>
```

```
.center {
    text-align: center;
    color: red;
}
```

CSS

- Define multi selector:

```
h3, .red, #redElement{
    color: red;
}
```

- Define child selector: apply exactly for some selector.

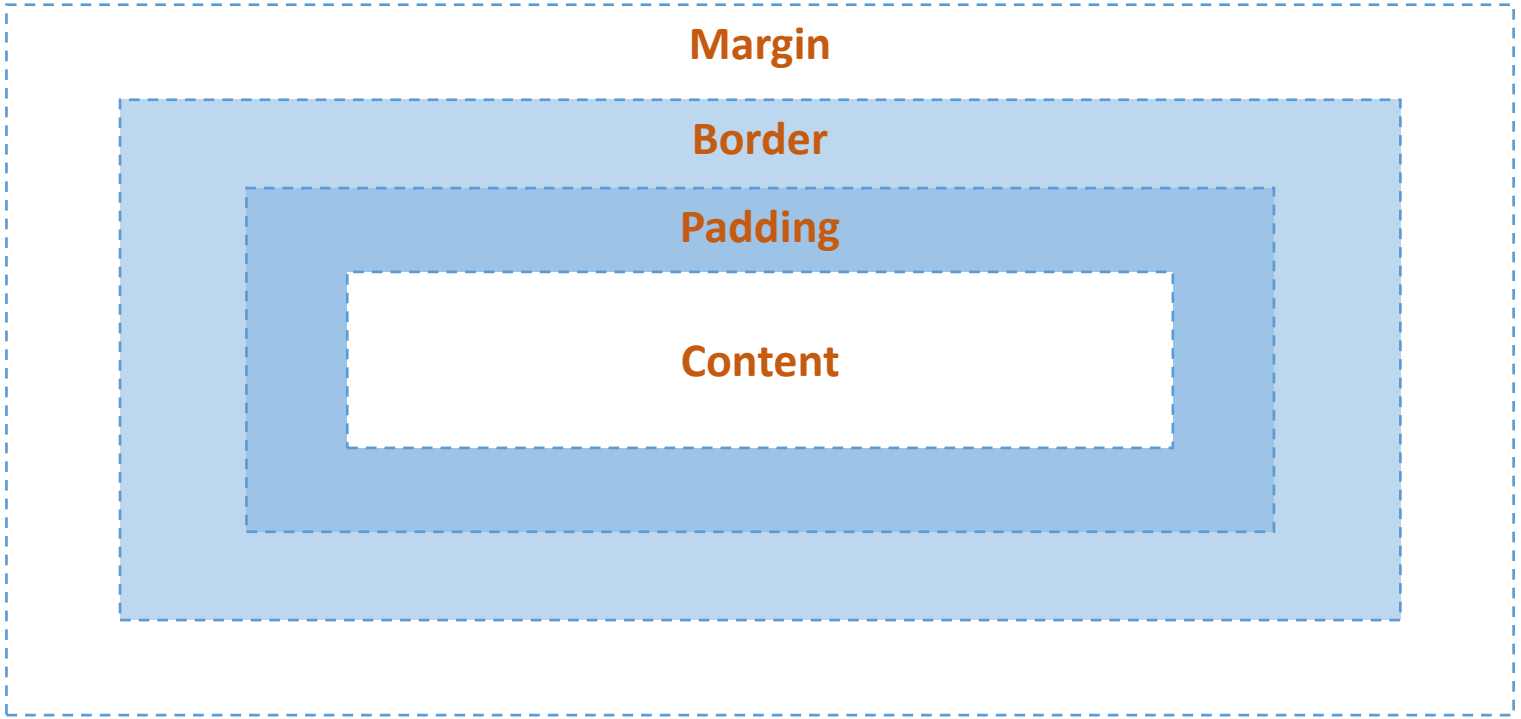
```
#header #menu #item p{
    color: red;
}
```

- Nested selector

```
#header #menu #item > p{
    color: red;
}
```

HTML Box Model

- Your browser renders every HTML element as a box, you can format all component of box.

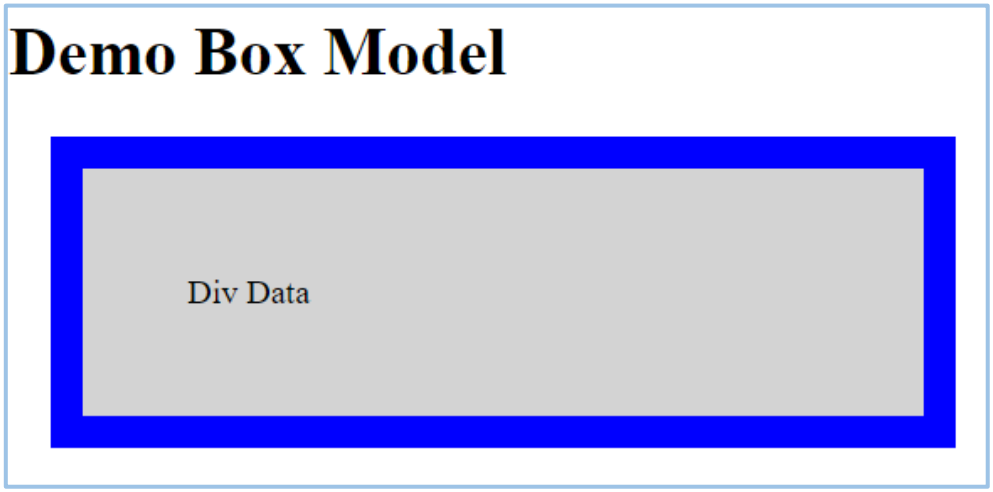


HTML Box Model

- The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content.
  - **Margin** - Clears an area outside the border. The margin is transparent
  - **Border** - A border that goes around the padding and content
  - **Padding** - Clears an area around the content. The padding is transparent
  - **Content** - The content of the box, where text and images appear

HTML Box Model

```
<style>
div {
  background-color: lightgrey;
  width: 300px;
  border: 15px solid blue;
  padding: 50px;
  margin: 20px;
}
</style>
<h1>Demo Box Model</h1>
<div>Div Data</div>
```



HTML Box Model

- height and width:
  - Use the height and width CSS properties to change the height and width of an element's content area.

Value	Description
auto	The browser calculates the height. This is default
length	Defines the height in px, cm, etc. Read about length units
%	Defines the height in percent of the containing block
initial	Sets this property to its default value. Read about *initial*
inherit	Inherits this property from its parent element. Read about *inherit*

HTML Box Model

- padding: to create spacing between an element's *content area* and *border.*
- The padding property is a shorthand property for:
  - padding-top
  - padding-right
  - padding-bottom
  - padding-left

HTML Box Model

- Padding property with four values:
  - padding:20px 15px 10px 5px;
    - Top padding is 20px
    - Right padding is 15px
    - Bottom padding is 10px
    - Left padding is 5px
- Padding property with three values:
  - padding:20px 15px 10px;
    - Top padding is 20px
    - Right and left padding are 15px
    - Bottom padding is 10px

HTML Box Model

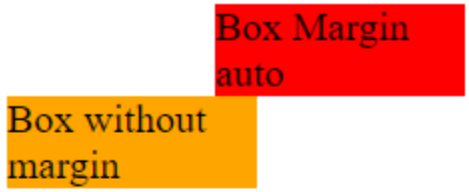
- Padding property with two values:
  - padding:20px 15px;
    - Top and bottom padding are 120px
    - Right and left padding are 15px
- Padding property with one values:
  - padding:20px;
    - All four paddings are 20px

HTML Box Model

- Margin: The margin property is very similar to the padding property, except it allows you to define the spacing around the outside of an HTML element past the border.
- Margin with value *auto*: making a html element position to center with outside element.

HTML Box Model

```
<style>
.box1 {
  margin: auto;
  width:100px;
  background-color:red;
}
.box2 {
  width:100px;
  background-color:orange;
}
</style>
<div class="box1"> Box Margin auto</div>
<div class="box2"> Box without
margin</div>
```



HTML Box Model

- Border: border CSS property sets the border of an element. The syntax for the border property is as follows:
  - border: width style color;
    - Width: width of border.
    - Style: can include things like dotted, groove, double, and solid.
    - Color: color value for border.

HTML Text Format

Color	*Color:* ***colorcode;***	Format color for text content.
Font Size	*font-size:* ***size;***	Set the size of the text.
Font Weight	*font-weight:* ***type;***	The weight of a font.
font-family	*font-family:* ***font name;***	The font for an element.
text-align	*text-align:* ***alignment ;***	The horizontal alignment of text in an element

SASS

SASS

- SASS (Syntactically Awesome Stylesheet) is a CSS pre-processor, which helps to reduce repetition with CSS and saves time. It is more stable and powerful CSS extension language that describes the style of document structurally.

Sass

```
$font-stack:    Helvetica, sans-serif
$primary-color: #333

body
  font: 100% $font-stack
  color: $primary-color
```

CSS

```
body {
  font: 100% Helvetica, sans-serif;
  color: #333;
}
```

Syntax

- Sass supports two different syntaxes:
  - Indented syntax (sass syntax): This is older syntax. Using this form of syntax, CSS can be written concisely. SASS files use the extension **.sass**.
  - Sassy CSS systax(scss syntax): is an extension of CSS syntax. This means every valid CSS is a valid SCSS as well. SCSS makes much easier to maintain large stylesheets and can recognize vendor specific syntax, Many CSS and SCSS files use the extension **.scss**.

Variables

- Sass uses the $ symbol to make something a variable.

SCSS Sass ⇒ CSS

```
$font-stack:    Helvetica, sans-serif;
$primary-color: #333;

body {
  font: 100% $font-stack;
  color: $primary-color;
}
```

SCSS Sass ⇒ CSS

```
$font-stack:    Helvetica, sans-serif
$primary-color: #333

body
  font: 100% $font-stack
  color: $primary-color
```

SCSS Sass ⇒ CSS

```
body {
  font: 100% Helvetica, sans-serif;
  color: #333;
}
```

Nesting

- Sass will let you nest your CSS selectors in a way that follows the same visual hierarchy of your HTML.

SCSS Sass ⇒ CSS

```
nav {
  ul {
    margin: 0;
    padding: 0;
    list-style: none;
  }

  li { display: inline-block; }

  a {
    display: block;
    padding: 6px 12px;
    text-decoration: none;
  }
}
```

SCSS Sass ⇒ CSS

```
nav
  ul
    margin: 0
    padding: 0
    list-style: none

  li
    display: inline-block

  a
    display: block
    padding: 6px 12px
    text-decoration: none
```

SCSS Sass ⇒ CSS

```scss
nav ul {
  margin: 0;
  padding: 0;
  list-style: none;
}
nav li {
  display: inline-block;
}
nav a {
  display: block;
  padding: 6px 12px;
  text-decoration: none;
}
```

Modules

- You don't have to write all your Sass in a single file.
- You can split it up however you want with the @use rule.
- This rule loads another Sass file as a module, which means you can refer to its variables, mixins, and functions in your Sass file with a namespace based on the filename.

SCSS Sass ⇒ CSS

```scss
// _base.scss
$font-stack:    Helvetica, sans-serif;
$primary-color: #333;

body {
  font: 100% $font-stack;
  color: $primary-color;
}
```

```scss
// styles.scss
@use 'base';

.inverse {
  background-color: base.$primary-color;
  color: white;
}
```

SCSS Sass ⇒ CSS

```sass
// _base.sass
$font-stack:    Helvetica, sans-serif
$primary-color: #333

body
  font: 100% $font-stack
  color: $primary-color
```

```sass
// styles.sass
@use 'base'

.inverse
  background-color: base.$primary-color
  color: white
```

SCSS Sass ⇒ CSS

```
body {
  font: 100% Helvetica, sans-serif;
  color: #333;
}

.inverse {
  background-color: #333;
  color: white;
}
```

Mixins

- A mixin lets you make groups of CSS declarations that you want to reuse throughout your site.

SCSS Sass ⇒ CSS

```
=transform($property)
  -webkit-transform: $property
  -ms-transform: $property
  transform: $property
.box
  +transform(rotate(30deg))
```

SCSS Sass ⇒ CSS

```scss
@mixin transform($property) {
  -webkit-transform: $property;
  -ms-transform: $property;
  transform: $property;
}
.box { @include transform(rotate(30deg)); }
```

SCSS Sass ⇒ CSS

```css
.box {
  -webkit-transform: rotate(30deg);
  -ms-transform: rotate(30deg);
  transform: rotate(30deg);
}
```

Extend - Inheritance

- This is one of the most useful features of Sass. Using @extend lets you share a set of CSS properties from one selector to another.

SCSS Sass ⇒ CSS

```scss
/* This CSS will print because %message-shared is
extended. */
%message-shared {
  border: 1px solid #ccc;
  padding: 10px;
  color: #333;
}

// This CSS won't print because %equal-heights is
never extended.
%equal-heights {
  display: flex;
  flex-wrap: wrap;
}

.message {
  @extend %message-shared;
}

.success {
  @extend %message-shared;
  border-color: green;
}
```

SCSS Sass ⇒ CSS

```
/* This CSS will print because %message-shared is
extended. */
%message-shared
  border: 1px solid #ccc
  padding: 10px
  color: #333


// This CSS won't print because %equal-heights is
never extended.
%equal-heights
  display: flex
  flex-wrap: wrap


.message
  @extend %message-shared


.success
  @extend %message-shared
  border-color: green
```

SCSS Sass ⇒ CSS

```scss
/* This CSS will print because %message-shared is
extended. */
.message, .success, .error, .warning {
  border: 1px solid #ccc;
  padding: 10px;
  color: #333;
}

.success {
  border-color: green;
}

.error {
  border-color: red;
}

.warning {
  border-color: yellow;
}
```

Operators

- Sass facilitates you to do basic mathematical operations in the style sheet. It is very simple to apply the appropriate arithmetic symbol.
- Sass provides some standard math operators: +, -, *, /, and %.

SCSS Sass ⇒ CSS

```scss
.container {
  width: 100%;
}

article[role="main"] {
  float: left;
  width: 600px / 960px * 100%;
}

aside[role="complementary"] {
  float: right;
  width: 300px / 960px * 100%;
}
```

SCSS Sass ⇒ CSS

```
.container
  width: 100%


article[role="main"]
  float: left
  width: 600px / 960px * 100%


aside[role="complementary"]
  float: right
  width: 300px / 960px * 100%
```

SCSS Sass ⇒ CSS

```scss
.container {
  width: 100%;
}

article[role="main"] {
  float: left;
  width: 62.5%;
}

aside[role="complementary"] {
  float: right;
  width: 31.25%;
}
```

Q & A