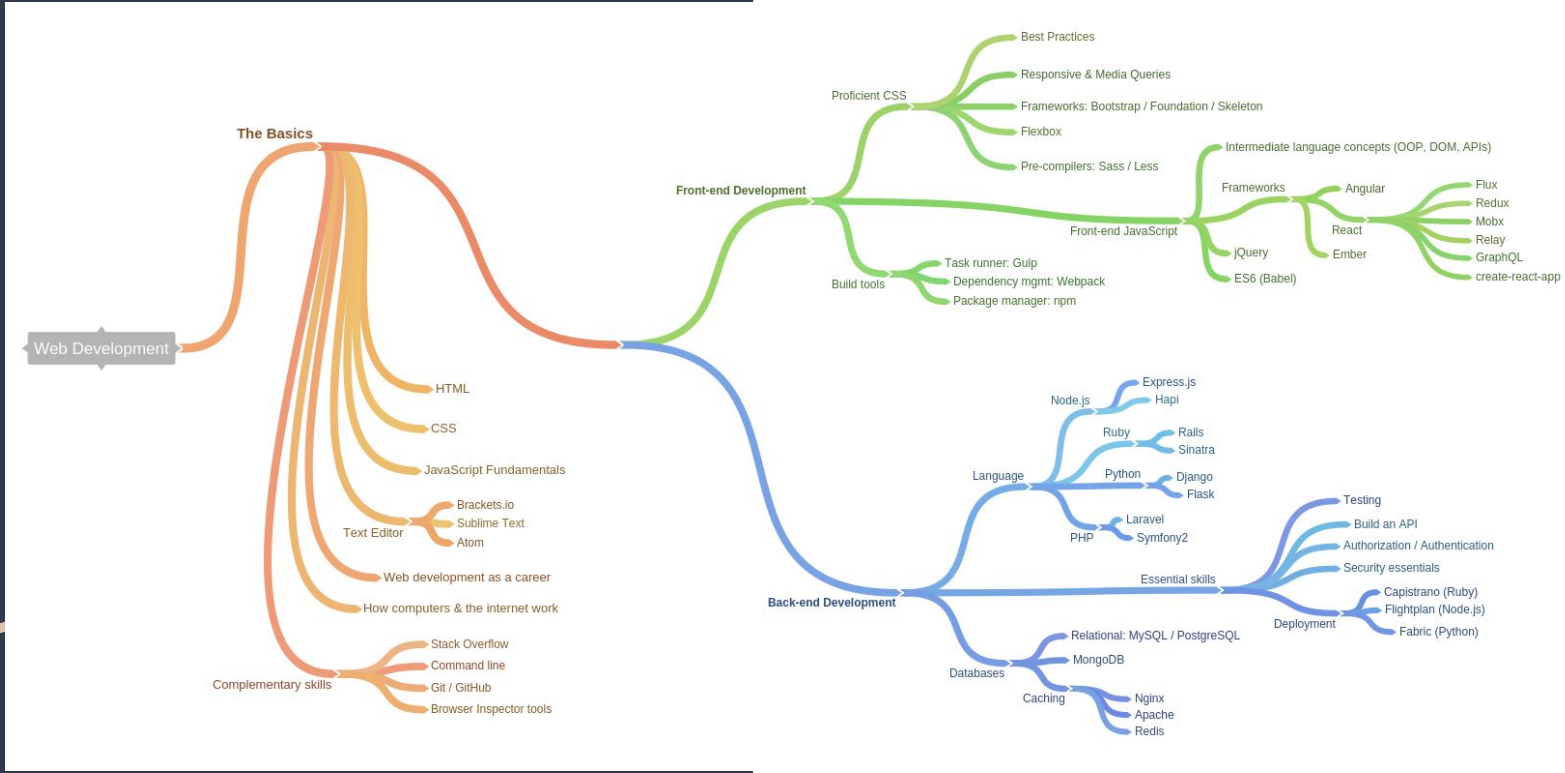
Web Development

Cascading Style Sheets - CSS

Web Development

The Basics
HTML, CSS, JavaScript,…



Cascading Style Sheets
CSS

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language like HTML.

- **Cornerstone technology** of the World Wide Web, alongside HTML and JavaScript.
- Describes **how HTML elements are to be displayed** on screen, paper, or in other media
- Layout of multiple web pages **all at once**
- Used to define styles for your web pages, including the design, layout and variations in **display for different devices and screen sizes**

CSS

- HTML was never intended to **contain tags for formatting a web page**
- HTML was created to describe the content of a web page, like:
  <h1>This is a heading</h1>
  This is a paragraph.
- When tags like **<font>, and color attributes were added to the HTML 3.2** specification
- Development of large websites, ***where fonts and color information were added to every single page***, became **a long and expensive process**.

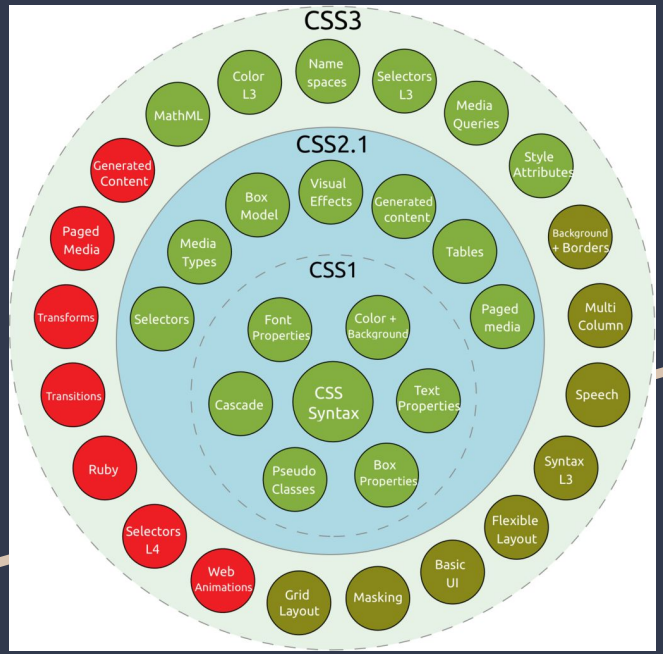
CSS Release

- To solve this problem, the World Wide Web Consortium (W3C) created **CSS**, was developed in 1996
- CSS removed the style formatting from the HTML page
- In mid-1998, the second standard, **CSS2**, was released
- W3C developed a new standard, **CSS2.1**, which reflected the level of acceptance of CSS2.
- Internet Explorer 8 and later (IE8+), Chrome 5 and later (C5+), and Firefox 3 and later (FX3+) fully support CSS2.1, which was at the *working draft* stage as of spring 2011.

CSS Release

CSS1

- Font properties such as typeface and emphasis
- Color of text, backgrounds, and other elements
- Text attributes such as spacing between words, letters, and lines of text
- Alignment of text, images, tables and other elements
- Margin, border, padding, and positioning for most elements
- Unique identification and generic classification of groups of attributes



- A superset of CSS 1, **CSS 2** includes
  - a number of new capabilities like absolute, relative,
  - fixed positioning of elements,
  - he concept of media types,..
- **CSS level 2 revision 1,** often referred to as "CSS 2.1"
  - fixes errors in CSS 2,
  - removes poorly supported or not fully interoperable features
  - adds already implemented browser extensions to the specification
- Unlike CSS 2, which is a large single specification defining various features, **CSS 3** is divided into several separate documents called **modules**.
- Each module adds new capabilities or extends features defined in CSS 2, preserving backward compatibility.

CSS Syntax

- CSS has a simple syntax and uses a number of English keywords to specify the names of various style properties.
- A style sheet consists of a list of *rules*.

- A CSS rule-set consists of
    - Selector (one or more)
    - Declaration block

CSS Syntax

Selector	Declaration
	{property:value; property:value;}
p	{color:red; text-align:left;}

- Values may be
  - keywords, such as "center" or "inherit",
  - numerical values, such as 200px , 80%

CSS Selectors

- CSS selectors are used to "find" (or select) HTML elements based on **element name, id, class, attribute,** and more.
    - The element Selector : Affects all p elements

      p { text-align: left; color: red; }
    - The id Selector : select a specific element

      #p1 { text-align: center; color: red; }

CSS Selectors

- The class Selector : selects elements with a specific class attribute

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

```
<h1 class="center">Red heading</h1>
```

```
<p class="center">Red paragraph</p>
```

- only specific HTML elements should be affected by a class

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

```
<p class="center"> Red paragraph</p>
```

```
<h1 class="center">Not affected</h1>
```

CSS Selectors

- Grouping Selectors : elements with the same style definitions, separate each selector with a comma

- h2, h3, p {
  text-align: left;
  color: blue;
  }

Sources

- CSS information can be provided from **various sources**.
- These sources can be the **web browser, the user and the author.**
- The information from the author can be further classified into **inline, media type, importance, selector specificity, rule order, inheritance and property definition.**
- CSS style information can be in a **separate document** or it can be embedded into an HTML document.
- Multiple style sheets can be imported.
- **Different styles** can be applied depending on the output device being used; screen version can be quite different from the printed version

Source Type

CSS priority scheme (highest to lowest)

Priority	CSS source type	Description
1	Importance	The '!**important**' annotation overwrites the previous priority types
2	Inline	A style applied to an HTML element via HTML 'style' attribute
3	Media Type	A property definition applies to all media types, unless a media specific CSS is defined
4	User defined	Most browsers have the accessibility feature: a user defined CSS
5	Selector specificity	A specific contextual selector (**#heading p**) overwrites generic definition
6	Rule order	Last rule declaration has a higher priority
7	Parent inheritance	If a property is not specified, it is inherited from a parent element
8	CSS property definition in HTML document	CSS rule or CSS inline style overwrites a default browser value
9	Browser default	The lowest priority: browser default value is determined by W3C initial value specifications

CSS Inserting Ways

There are three ways of inserting a style sheet

- External style sheet
    - reference to the external style sheet file inside the <link> element
    - not contain any html tags
    - saved with x .css extension
- Internal style sheet
- Inline style

External style sheet

style1.css

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

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

Html file

```
<head>
<link rel="stylesheet" type="text/css" href="style1.css">
</head>
```

Internal style sheet

- Internal styles are defined within the <style> element
- inside the <head> section of an HTML page

```
<head>
<style>
body {
  background-color: black;
}
h1 {
  color: white;
  margin-left: 30px;
}
</style>
</head>
<body>
<h1>This is a heading</h1>
</body>
</html>
```

Inline Styles

- An inline style may be used to apply a unique style for a single element.
- To use inline styles, **the style attribut**e added to the relevant element.

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

<h1 style="color:blue;margin-left:30px;">This is a heading</h1>
<p>First paragraph</p>

</body>
</html>
```

Highest Priority

All the styles in a page will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:

1. Inline style (inside an HTML element)
2. External and internal style sheets (in the head section)
3. Browser default

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
      <style> #xyz { color: blue; } </style>
  </head>
  <body>
    <p id="xyz" style="color: green;"> To demonstrate specificity </p>
</body>
</html>
```

CSS

- Colors
- Backgrounds
- Borders
- Margins
- Padding
- Height/width
- Box Model
- Text
- Fonts
- Icons
- Tables
- Align
- List
- Outline
- Navigation Bar
- Dropdowns
- Image Gallery
- Links
- Forms
- …..

Colors

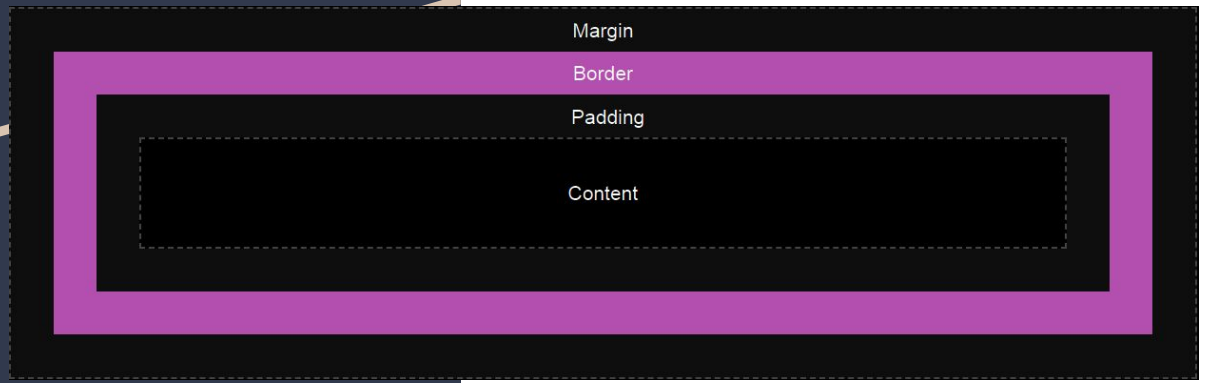
- <h1 style="color:Tomato;">Hello World</h1>
- <h1 style="background-color:rgb(255, 99, 71);">...</h1>
- <h1 style="background-color:#ff6347;">...</h1>
- <h1 style="background-color:hsl(9, 100%, 64%);">...</h1>

RGB Value, Hex value, HSL Value (hue, saturation, lightness), RGBA, HSLA

Box Model

- All HTML elements can be considered as boxes. In CSS, the term **box model** is used about **design and layout.**
- The CSS box model is essentially a box that wraps around every HTML element.

- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent



Auto Keyword

- The value auto can be used with the property margin to horizontally center an element within its container.
- The margin property will take the width of the element and will split the rest of the space equally between the left and right margins.

Web Site Layout

- A website is often divided into headers, menus, content and a footer:
- one of the most common

```
.header {
  background-color: #F1F1F1;
  text-align: center;
  padding: 20px;
}
```

Navigation Bar

```
/* 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;
}
```

Layout Content

- **1-column** (often used for mobile browsers)
- **2-column** (often used for tablets and laptops)
- **3-column layout** (only used for desktops)

Responsive Web Design

- Responsive web page that works well on any device - phone, tablet, desktop or anything in between
- Makes your web page look good on all devices.
- Uses only HTML and CSS.
- Not a program or a JavaScript
- It is called responsive web design when you use CSS and HTML to
    - resize, hide, shrink, enlarge, or move the content to make it look good **on any screen**.

Responsive vs Adaptive Web Design

- Responsive Web Design provides the optimal viewing experience of a website, no matter what type of device the user is seeing it on.
-
- Adaptive web design is different from responsive design in that there isn’t one layout that always changes. Instead, there are several distinct layouts for multiple screen sizes.

RWD-ViewPort

- The viewport is the **user's visible area** of a web page.

- The viewport varies with the device, and will be smaller on a mobile phone
- Before tablets and mobile phones, web pages were designed only for computer screens, web pages to have a **static design and a fixed size**.

- using tablets and mobile phones, fixed size web pages were too large to fit the viewport.
- To fix this, browsers on those devices scaled down the entire web page to fit the screen.
- HTML5 introduced a method to let web designers take control over the viewport, through the **<meta> tag**.

Set the ViewPort

- Use the meta viewport tag to control the width and scaling of the browser's viewport.
- Include **width=device-width** to match the screen's width in device-independent pixels.
- Include **initial-scale=1** to establish a 1:*1 relationship between CSS pixels and device-independent pixels*.
- Ensure your page is accessible by not disabling user scaling.
- Also set the following attributes on the viewport:
  - minimum-scale
  - maximum-scale
  - user-scalable

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Media Queries

- Media query is a CSS technique introduced in CSS3.
- It uses the @media rule to include a block of CSS properties only if a certain condition is true.

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width,
initial-scale=1.0">
<style>
body {
  background-color: lightgreen;
}

@media only screen and (max-width: 600px) {
  body {
    background-color: lightblue;
  }
}
</style>
</head>
<body>

<p>Resize the browser window. When the width of this document is
600 pixels or less, the background-color is "lightblue", otherwise
it is "lightgreen".</p>

</body>
</html>
```

RWD Images

If the width property is set to a percentage and the height is set to "auto",
the image will be responsive and scale up and down

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width,
initial-scale=1.0">
<style>
img {
  width: 100%;
  height: auto;
}
</style>
</head>
<body>

<img src="img_chania.jpg" width="460" height="345">
<p>Resize the browser window to see how the image will
scale.</p>

</body>
</html>
```

Element picture

HTML5 <picture> Element
HTML5 introduced the <picture> element, which lets you define more than one image.

```
<picture>
  <source srcset="smallflower.jpg"
media="(max-width: 400px)">
  <source srcset="flowers.jpg">
  <img src="flowers.jpg" alt="Flowers">
</picture>
```

CSS Framework

- Many existing CSS Frameworks that offer Responsive Design
- To make responsive web pages, Bootstrap is popular framework, uses HTML, CSS and jQuery

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Bootstrap Example</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/b
ootstrap.min.css">
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jqu
ery.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/boo
tstrap.min.js"></script>
</head>
<body>
..
```