Introduction to CSS

- What is CSS
- History of CSS
- CSS building blocks
- CSS Styling
- CSS Layout
- Responsive Web Design principles

What is CSS?

- CSS, a shorthand for Cascading Style Sheets, is one of the main building blocks of the Web
- the language that we use to style an HTML file, and tell the browser how should it render the elements on the page
- declarative-style computer programming language used to design website content

History of CSS

1996 - CSS 1 It was launched in 1996 with the capabilities of font properties. It is also used for adding color to the background and text side. In CSS 1, there were text alignment functionalities. It also has capabilities of padding, positioning, and generic classifications.

1998 - CSS 2 It has more features and functionalities than the previous version. And now users could use new features like relative, absolute, and also fixed positioning. There were media types, and bidirectional text features were also there. This version also saw many revisions in the same, and updates came as CSS2.1.

1999 - CSS 3 The latest version of CSS officially by W3C, and it was launched in 1999. It has a vast collection of font types, and you can use any font type from Google and Typecast.

CSS Building blocks

A CSS **rule set** has one part called **selector**, and the other part called **declaration**. The declaration contains various **rules**, each composed by a **property**, and a **value**.

```
<selector> {

property: value1;

property: value2;

}
```

CSS Building blocks

CSS building blocks, specifically ***ID*** and ***Class*** selectors, are fundamental tools for targeting and styling HTML elements.

ID Selector

- **Purpose:** The ID selector is used to target a single, unique HTML element within a document.
- **HTML Attribute:** An id attribute is assigned to an HTML element with a unique value.
- **CSS Syntax:** In CSS, the ID selector is represented by a hash symbol (#) followed by the ID value.

CSS Building blocks

– HTML –

```
<div id="main-header">
    <h1>Welcome</h1>
</div>
```

– CSS –

```
#main-header {
    background-color: blue;
    color: white;
}
```

CSS Building blocks

Class Selector:

- **Purpose:** The Class selector is used to target multiple HTML elements that share a common characteristic or styling.
- **HTML Attribute:** A `class` attribute is assigned to one or more HTML elements, and multiple classes can be assigned to a single element.
- **CSS Syntax:** In CSS, the Class selector is represented by a period (`.`) followed by the class name.

CSS Building blocks

```
– HTML –
<p class="highlight">
    This text is highlighted and important.
  </p>
  <span class="highlight">
    This is also highlighted.
  </span>
– CSS –
  .highlight {
    background-color: yellow;
  }
```

CSS Styling

To apply styles on the html tags/elements, we can do the following approaches:

- **Inline styling** By adding a ***style*** attribute in the html tags/elements
- **Internal styling** By declaring a ***<style>..</style>*** element inside the <head> .. </head> element
- **External styling** by declaring a ***<link rel=”stylesheet” href=”style.css”/>*** in the <head> .. </head>

CSS Styling (cont’d)

Fonts/Text

- **font-size:** 18px;
- **font-family:** ‘Arial’, ‘georgia’, ‘Helvetica’, ‘Tahoma’, ‘Times New Roman’, ‘Trebuchet MS’;
- **font-weight:** bold | normal | lighter | bolder;
- **text-align:** left | right | justify | center;

CSS Styling (cont’d)

Colors/Background

- **background-color:** red | #00adef | rgb(255,0,233);
- **color:** blue | #0023ed | rgb(54,233,200);
- **background-image:** url(<file path>);
- **opacity:** 0.5;

CSS Styling (cont’d)

Display

- **display:** block | inline-block | none | inline-block;
- **visibility:** hidden | visible;

CSS Styling (cont’d)

Size (height/width/max-width or height/min-width or height)

- **height:** 10px | 10% | auto;
- **width:** 10px | 10% | auto;
- **max/min-width:** 10px | 10% | auto;
- **max/min-height:** 10px | 10% | auto;

CSS Styling (cont’d)

Links (a:link | visited | hover | active)

- **color:** red | #00adef | rgb(255,0,233);
- **text-decoration:** none | underline;

CSS Styling (cont’d)

Lists (ul/ol)

- **list-style-type:** circle | square (for ul)/ upper-roman | lower-alpha**;**
- **list-style-image:** url(‘sample.png’);
- **list-style-position**: outside | inside;

CSS Styling (cont’d)

Table

- **border:** 1px solid; /*applies to table, th, and td elements */
- **width:** 100%; /* full-width table */
- **border-collapse:** collapse | separate;

CSS Styling (cont’d)

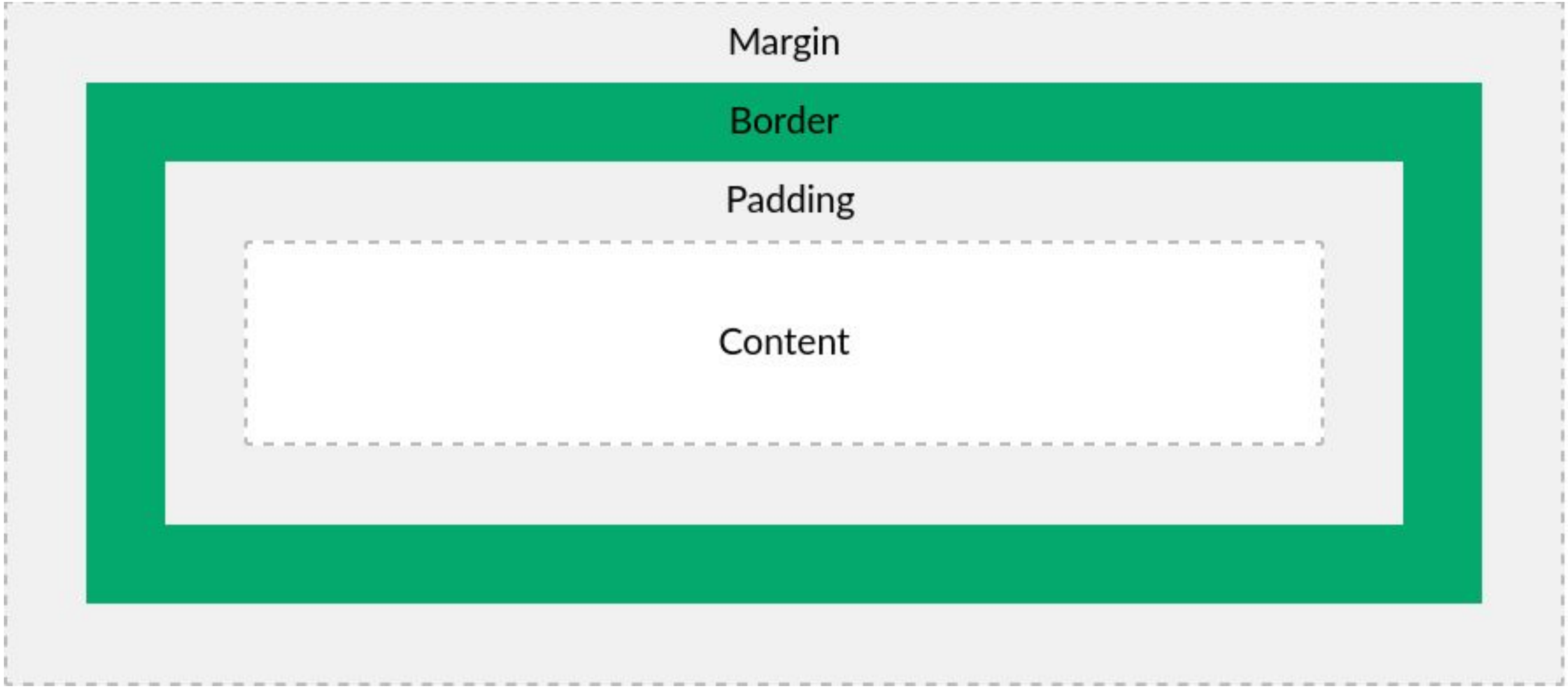
Font references:

[https://www.tutorialbrain.com/css_tutorial/css_font_family_list/]

Color references:

[https://www.w3.org/wiki/CSS/Properties/color/keywords]

CSS Layout (Box model)



CSS Layout (cont'd)

Border used to set visible border width to most HTML elements within the body

- **border-style:** dotted | dashed | solid | none | hidden;
- **border-width:** 5px | medium | thick;
  - **Can have specific width size**
    - **20px 10px; /* 20px on top and bottom, 10px on sides**
    - **10px 5px 4px 2px; /* top, right, bottom, left**
- **border-color:** red | #001234 | rgb(123,222,111);
- **Border-radius:** 5px;

CSS Layout (cont’d)

Margin properties are used to create space around elements, outside of any defined borders.

- **margin-top|right|bottom|left:** 10px | auto | 90% | inherit;
- **margin:** 20px; /* all sides 20px */

CSS Layout (cont'd)

Padding used to create space around an element's content, inside of any defined borders.

- **padding-top|right|bottom|left:** 10px | auto | 50% | inherit;
- **padding:** 20px /* all sides are 20px of size */

Responsive Web Design principles

There are many devices that can access the web, and they come in all shapes and sizes.

Responsive web design is the practice of designing flexible websites that can respond to different screen sizes, orientations, and resolutions.

Responsive Web Design principles (cont’d)

Create a Media query

Media Queries are a new technique introduced in CSS3 that change the presentation of content based on different viewport sizes. The viewport is a user's visible area of a web page, and is different depending on the device used to access the site.

```
@media (max-width: 100px) { /* CSS Rules */ }

@media (min-height: 350px) { /* CSS Rules */ }
```

Responsive Web Design principles (cont’d)

Make an Image Responsive

Making images responsive with CSS is actually very simple. You just need to add these properties to an image:

```
img {
  max-width: 100%;
  height: auto;
}
```

The **max-width** of 100% will make sure the image is never wider than the container it is in, and the **height** of **auto** will make the image keep its original aspect ratio.

Responsive Web Design principles (cont'd)

Make typography responsive

Instead of using **em** or **px** to size text, you can use viewport units for responsive typography. Viewport units, like percentages, are relative units, but they are based off different items. Viewport units are relative to the viewport dimensions (width or height) of a device, and percentages are relative to the size of the parent container element.

```
body{
width: 50vw;
}
```

Responsive Web Design principles (cont’d)

The four different viewports are:

- **vw (viewport width)**: 10vw would be 10% of the viewport's width.
- **vh (viewport height)**: 3vh would be 3% of the viewport's height.
- **vmin (viewport minimum)**: 70vmin would be 70% of the viewport's smaller dimension (height or width).
- **vmax (viewport maximum)**: 100vmax would be 100% of the viewport's bigger dimension (height or width).

Responsive Web Design principles (cont'd)

Use a Retina Image for Higher Resolution Displays

With the increase of internet connected devices, their sizes and specifications vary, and the displays they use could be different externally and internally. Pixel density is an aspect that could be different on one device from others and this density is known as Pixel Per Inch(PPI) or Dots Per Inch(DPI)

Responsive Web Design principles (cont’d)

The simplest way to make your images properly appear on High-Resolution Displays, such as the MacBook Pros "retina display" is to define their **width** and **height** values as only half of what the original file is.

```
img{
    height: 250px; /* for example, image height is 500px */
    width: 250px; /* for example. Image width is 500px */
}
```