Cascading Style Sheets (CSS)

- a simple mechanism for adding style (e.g., fonts, colors, spacing) to Web documents.
- used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.
- CSS saves a lot of work
- it can control the layout of multiple web pages all at once
- stored in the <**head**> section of the Web page or in a separate file with a .css extension
- external stylesheets are stored in CSS files

Cascading Style Sheet Rules

- a CSS rule consists of a selector and a declaration block.
- begins with the name of the tag or other element to which style applies

Style rules are formed as follows:

For single style rule:

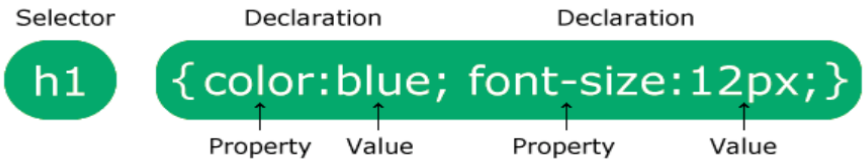
```
selector{
        property1: value

}
```

Multiple style declarations:

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

}
```



- the selector points to the HTML element you want to style.
- the declaration block contains one or more declarations separated by semicolons.
- each declaration includes a CSS property name and a value, separated by a colon.
- multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

Example of multiple style declaration: *Apply style rule h1*

```
<style>
    h1{
        color: red ;
        font-size: 14px;
        text-align: center
    }

</style>
```

Grouping

If the values of certain properties of selectors are the same, grouped declarations are allowed. For instance, if certain values for headings are similar, identical declarations can be given through a grouping:

Example#1: *Apply style rule h1, h2, h3, h4, h5, h6*

```
<style>
    h1,h2,h3,h4,h5,h6{
        color: red;
        font-family: serif;
        text-align: center
    }
</style>
```

Example#2: *Apply style rule **p** and **h***

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


    h1,h2,h3,h4{
        color: blue;
        font-size: 18px;
        text-align: left
    }
  </style>
```

Comments

Comments in cascading style sheets have the same format as that of comments in C programs. Everything within the slash-asterisk-asterisk-slash (/* */) symbols is ignored by the browser.

CSS Classes

In CSS, selectors are a way to select the element(s) you wish to style. The selection is made typically by using their element type, **class** or **id**.

The class selector selects the element(s) with their specific class name assigned while coding. **A class name can be any string value starting with a dot (.)**. It can have the upper or lower case letters, digits, characters like the hyphen and underscore are also acceptable. You cannot have space in between a class name.

Here's an example of a CSS class called **'mystyle'** that sets the color to blue and font size to 14px:

```
.mystyle {
    color: blue;
    font-size: 14px;
}
```

And here's the HTML with the CSS:

```
<p class= "mystyle">This paragraph is styled by the CSS .mystyle class defined by us.</p>
```

Various HTML elements can have the same styling by assigning them the desired class name.

```
<p class= "mystyle">This paragraph is styled by the CSS .mystyle class defined by us.</p>
<h1 class = "mystyle">Heading 1 </h1>
```

Uses of CSS Class Selectors

If you choose not to use classes in the example above, then you will have to define the style for every element individually. That means defining their specific color and font-sizes using the CSS style attribute. And repeat the same style attribute for the ul element and so forth. Let's assume you have styled 20 elements in red color and 14px font size without using any classes. If you now want to change the color to blue and increase the size to 18px, you will have to make changes 20 times. By defining all the style attributes at one place (in a class), just one change in the attribute values in that class will affect all those 20 elements!

Multiple Classes for an Element

You can define as **many classes** as you wish and assign them to an element one after the other **separated by space**. Let's say we want to center-align the text in our paragraph, so we define a new CSS class:

```
.mycenter {
    text-align: center;
}

.left_align{
    Property:value

}
```

And we apply it to our HTML:

```
<p class= "mystyle mycenter">This paragraph is styled by using two CSS classes .mystyle and .mycenter as defined above.</p>
```

The styles from the two assigned class names make our paragraph center-aligned, blue in color, and with a 14px font-size, precisely as expected.

Grouping Selectors

Let's say; you wish to give the same set of styling to a paragraph, a heading and a list We want them all to be in blue with 14px font size for the text.

```
h1 {
    color: blue;
    font-size: 14px;
}

p {
    color: blue;
    font-size: 14px;
}

li {
    color: blue;
    font-size: 14px;
}
```

Using a class:

```
.myclass {
            color: blue;
           font-size: 14px;
}
```

Hyperlink styles

- textual hyperlink is underlined blue text by default
- visited hyperlink is underlined purple

Pseudo-classes (for hyperlink styles)

Classes that use a variable to determine membership

a:link	regular hyperlinks
a:visited	Visited hyperlinks
focus	Selected using keyboard but not activated
a:hover	Mouse is positioned over the link
a:active	Link is clicked

Example:

```
<style>
        a: link{text-decoration:underline, font-size:1mm, color:blue}
        a: visited{color: green}

</style>
```

Font family

- set of fonts listed in order of preference

Example:

```
<style>
    p{
        font-family: “Arial”, “Helvetica”, sans-serif
    }
</style>
```

Generic font types (specified with no quotation marks):

- serif
- sans-serif
- cursive
- fantasy
- monospace

Font size

- specify an absolute size or specify a size in relation to the parent tag

Absolute size

- number followed by unit of measurement
- px for pixels (average font size is 10px)
    - **in** for inches
    - **mm** for millimeters
    - **pt** for point
    - **pc** of picas

Example for an entire tag in a style sheet:

```
<style>
        p{font-size: 12px}
</style>
```

Example (embedded in a single paragraph's tag):

```
<p style= "font-size:12">The text </p>
```

Font weight

- bold, bolder, lighter, or 100 (lightest) to 900 (darkest)

Font style

- normal, italic, or oblique

Example:

```
 <style>
          p{
              font-weight: bold;
              font-style: italic
          }
 </style>
```

(inline) Span

- shell for placing attributes (usually used to cover specific portions of the text and not an all-encompassing tag)
- class can be applied

Example (using class):

```
<style>
        .sample{
                    font-weight:bold;
                    font-style: italic
        }
</style>
```

Applying the sample class in a span:

```
<span class= "sample">Just this >/span>
```

Paragraph formatting

- formatting for the layout of the entire paragraph

First line indent

- **text-indent**

Example

```
<p style = "text-indent: 20px>Type your paragraph here…</p>
```

Padding

- adds as specified format of space between the border of an element and its contents (inside of the element)
- **padding**

Example

```
<p style= "padding:40px">Type your paragraph here…</p>
```

Margin

- adds a specified amount of space around an element (outside of the element)
- **margin**

Example

```
<p style= "margin:50px">Type your paragraph here… </p>
```

Borders

- **border-style**
- solid, dotted, dashed, double, groove, ride, inset outset

Example

```
<p style= "border-style:solid">Type your paragraph here…</p>
```

order padding

- **padding**
- same as paragraph padding

Border width and color

- **border-width**
- **border-color**

Example

```
<p style= "border-width:2px; border-color:blue">Type your paragraph here…</p>
```

Horizontal alignment

- **text-align**
- apply alignment only to block-level elements
- choices for text-align are **left, center, right, or justify**

Example

```
<p style= "text-align:center">Type your paragraph here…</p>
```

CSS Background Properties

Property	Description	Values
background	A shorthand property for setting all background properties in one declaration	*background-color* *background-image* *background-repeat* *background-attachment* *background-position*
background-attachment	Sets whether a background image is fixed or scrolls with the rest of the page	scroll fixed
background-color	Sets the background color of an element	*color-rgb* *color-hex* *color-name* transparent
background-image	Sets an image as the background	url(*URL*) none
background-position	Sets the starting position of a background image	top left top center top right center left center center center right bottom left bottom center bottom right *x% y%* *xpos ypos*
background-repeat	Sets if/how a background image will be repeated	repeat repeat-x repeat-y no-repeat

CCS Text Properties

Property	Description	Values
color	Sets the color of a text	*color*
direction	Sets the text direction	ltr rtl
line-height	Sets the distance between lines	normal *number* *length* %
letter-spacing	Increase or decrease the space between characters	normal *length*
text-align	Aligns the text in an element	left right center justify
text-decoration	Adds decoration to text	none underline overline line-through blink
text-indent	Indents the first line of text in an element	*length* %
text-shadow		none *color* *length*
text-transform	Controls the letters in an element	none capitalize uppercase lowercase
unicode-bidi		normal embed bidi-override
white-space	Sets how white space inside an element is handled	normal pre nowrap
word-spacing	Increase or decrease the space between words	normal *length*

Attaching CSS to Your Document

The CSS we create will act as a style sheet for our web pages. This is what will control the type, color, layout and even interactive pieces. In order for our HTML pages to make use of the CSS rules, we need to make sure that our HTML page references, or attaches, them in some way.

There are three common ways to attach your stylesheets:

A. Inline
B. Embedded /Internal
C. External

A. Inline Styles

To do this, you just have to use the style attribute and add the CSS declaration as the attribute value.

```
html>
<head>

</head>

<p style="color: blue">This paragraph will be "deep pink".</p>

</html>
```

While this is very easy, it's also not very extensible.
For example, to change all the paragraphs to red, we'd have to add the style attribute to each paragraph on in our HTML document.

B. Embedded / Internal Styles

You can also add CSS styles to the top of your HTML page, inside the head element.

```
<html>
<head>
  <title>Embedded / Internal Stylesheets Example</title>

  <style>
    p {
        color: blue;
        font-weight:bold
    }
  </style>

</head>

<p>This paragraph will appear in bold with font color blue.</>

</html>
```

This is a great way to quickly test a new style on your page. **Reserve internal styles for testing and experimenting with new CSS rules.**

C. External Styles

The best method for attaching your CSS style sheets is to use **external styles**. With this method, you will write all your CSS in a separate file with a .css extension. You can then link to the CSS file from each of your HTML pages.

In the example below, we're linking to a CSS document called styles.css.

```
<html>
    <head>
            <title>External Style Sheet Example</title>
            <link rel="stylesheet" href="styles.css" media="screen">
   </head>
<html>
```

External stylesheets use the <link> tag inside the head element.

rel

The ***rel*** attribute explains the **relation** the link has to our document. The value in this case will always be stylesheet, since that is what we're creating a link to.

href

The ***href*** attribute is the link to our stylesheet. This works exactly the same as the href used in a tags.

media

The ***media*** attribute describes what type of media our stylesheets should apply to. There are a number of possible values, including both screen and print. You will most often use screen.

What is a Div Tag and How to Style it with CSS

The HTML division tag, called "div" for short, is a special element that lets you group similar sets of content together on a web page. You can use it as a generic container for associating similar content.

The div tag is one of the most used tags and doesn't seem to be going anywhere despite the introduction of semantic elements (these elements let you use several tags as a container).

When to Use the div Tag

The **div** tag is multi-purpose – you can use it to do several things on a web page. You'll mostly use it in web layouts and CSS art, but it's super flexible.

Creating a Basic Layout

You'll primarily use the div tag to group similar content together so you can style it easily. A great example of this is using div to group different sections of a webpage together. You can put together the header, nav, sections, and footer of a page in an individual div tag so they can be styled together.

Class

Another way of selecting HTML elements is by using the **class** attribute. In HTML, we can assign different classes to our elements. Each element can have multiple classes, and each class can also be applied to multiple elements as well.

In the code below, we have assigned the class of container to the div element. In the stylesheet, we select our class using **.className** format and giving it a 10px margin.

CSS:

```
.container {
        margin: 10px;
}
```

html:

```
<div class= "container">
   <h1> This is heading </h1>
</div>
```

ID

Like classes, we can also use **ID**s to select HTML elements and apply styling to them. The only difference between class and ID is that one ID can be assigned to only one HTML element.

The example below displays how we assign an ID to the paragraph element and later use the ID selector in the stylesheet to select the paragraph and apply the style to it.

html:

```
<div>
   <p id= "para1"> This is a paragraph </p>
</div>
```

CSS:

```
#para1 {
   color: green;
   font-size: 16px;
}
```

Fonts & Colors

CSS provides us with literally hundreds of options for playing around with fonts and colors and making our HTML elements look pretty. We can choose from two types of font family names:

1. Generic Family: a group of font families with a similar look (like 'Serif' or 'Monospace')
2. Font Family: a specific font family (like "Times New Roman" or "Arial")

For colors, we can use predefined color names, or RGB, HEX, HSL, RGBA, HSLA values.

html:

```
<div class= “container”>
    <h1 class= “heading1”>  CSS is Coooooool!!!!  </h1>
</div>
```

CSS:

```
.container {
   width: 500px;
   height: 100px;
   background-color: lightcyan;
   text-align: center;
}


.heading1 {
   font-family: “Courier New”;
   color: tomato;
}
```

As you can see in the above example, we have a div element with the class of container. Inside this div, there is an h1 tag with some text inside it.

In the stylesheet, we select the container class and set its width, height, background-color, and text-align.

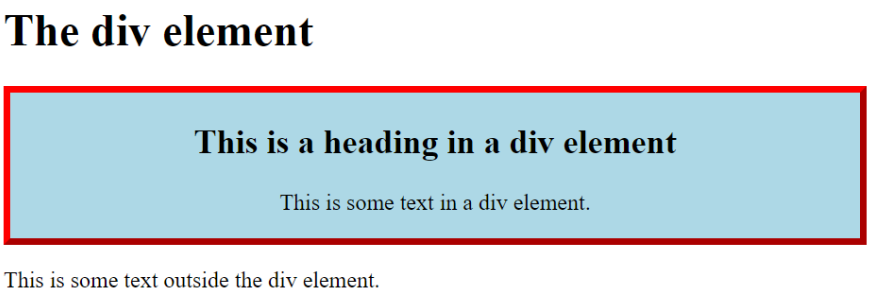
Finally, we select the .heading1 class—which is applied to the h1 tag—and give it the attributes of font-family and color.

Sample #1:

```
<html>
  <head>
      <style>
          .myDiv {
              border: 5px outset red;
              background-color: lightblue;
              text-align: center;
          }
      </style>
  </head>

  <body>
      <h1>The div element</h1>
      <div class="myDiv">
          <h2>This is a heading in a div element</h2>
          <p>This is some text in a div element.</p>
      </div>
      <p>This is some text outside the div element.</p>
  </body>
</html>
```

Output:



Sample #2:

Creating Sidebars

html:

```
<html>
<head>
  <title>External Style Sheet Example</title>

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

<body>
        <div id="banner"></div>

</body>
</html>
```

CSS:

```
 #banner{
        padding: 5px;
        background-color: blue;
        height:140px;
        }

#left_sidebar1{
        font-family:Arial;
        font-size: 15px;
        background:#003366;
        width:173px;
        height:620px;
        margin-top:1%;
        float: left;
        color:white;
        padding:5px 5px;
        position:absolute;
        }

#right_sidebar1{
        font-family:Arial;
        font-size: 15px;
        background:#003366;
        width:182px;
        height:620px;
        float: right;
        color:white;
        padding:5px 5px;
        margin-top:1%;

        }

        #center1{
        border-width: thin;
        border-style: solid;
        border-color: black;
        background:#003366;
        font-family:Arial;
        font-size: 15px;
        color:white;
        padding:5px;
        width:580px;
        height:620px;
        margin-left:19.5%;
        }
```

References:

https://www.w3schools.com/css
https://blog.hubspot.com/website/what-is-css-class
http://web.simmons.edu/~grovesd/comm244/notes/week3/css-linking
https://www.freecodecamp.org/news/html-div-what-is-a-div-tag-and-how-to-style-it-with-css/
https://www.youtube.com/watch?v=Y9rHsdCxU8Q
https://www.w3schools.com/howto/howto_css_sidebar_responsive.asp
https://blog.hubspot.com/website/css-sidebar