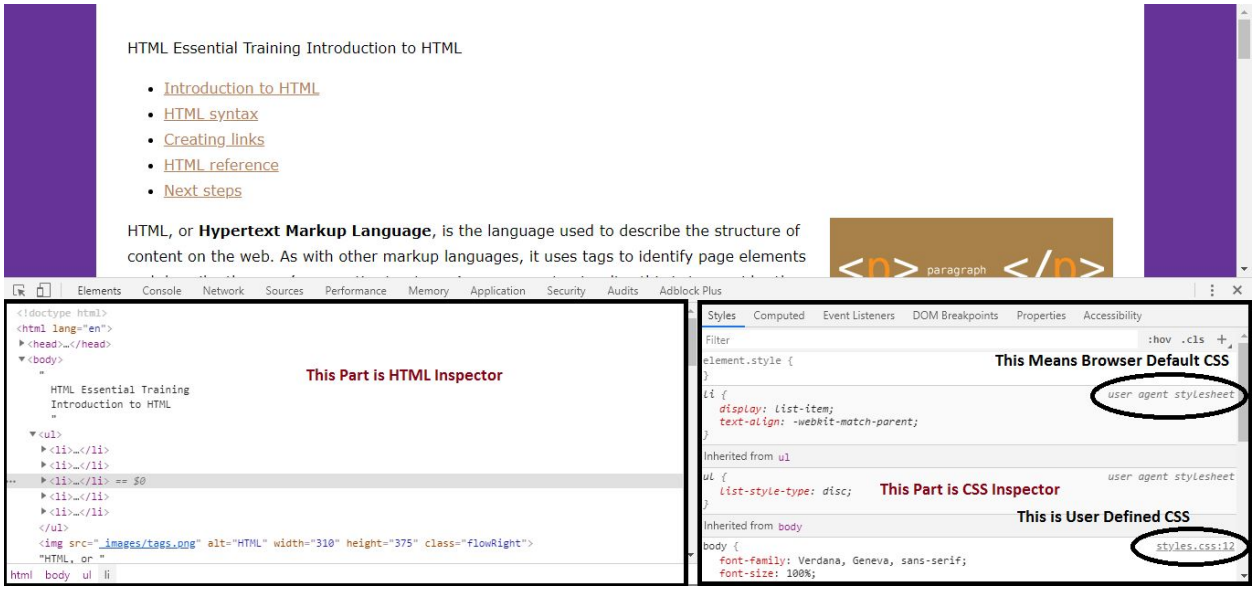
CASCADING STYLE SHEETS

- CSS is mainly used to style html pages and make responsive design

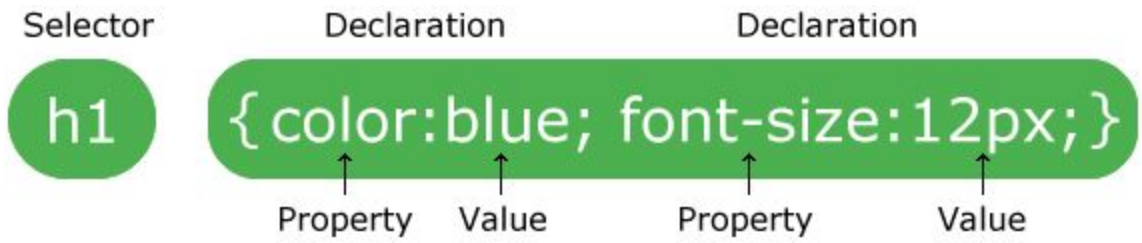
Using The Developer Tools

- Right Click on Browser Page; Find & Click on “Inspect”. Page inspection Tool will be opened.
-



CSS Syntax

-



- General Syntax for Css; This will render all <h1> tag in the page as blue in 12px font size
- We will learn more types of selectors below
- CSS has many properties and values to style html pages. (not all of them are necessary to learn at first; Just the basics will do)

Referencing CSS

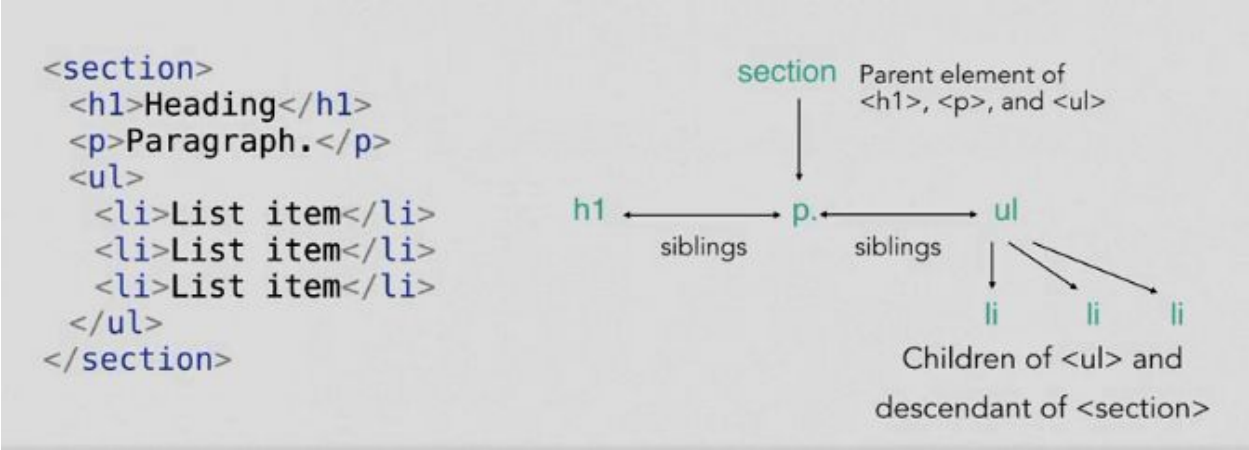
```
<!DOCTYPE html>
<html>

<head>
      <!-- External CSS -->
      <link rel="stylesheet" type="text/css" href="mystyle.css">

    <!-- mystyle.css contains:
          body {
              background-color: lightblue;
          }
          h1 {
              color: navy;
              margin-left: 20px;
          }
    -->

    <!-- Internal CSS -->
    <style>
      h1{
            color: red;
           /* Color: red !important; !important is used to ignore all other
color properties for h1*/
        }
    </style>
</head>

<body>
      <!-- Inline CSS -->
      <h1 style="color: green; font-size: 13px;">This is a heading</h1>
      <p>This is a paragraph.</p>
</body>

</html>
```

- Css can be referenced (linked to HTML) in 03 ways: External, Internal, Inline
- **External:** an external .css file is linked with <link> tag.
- **Internal:** Inserted in <style> tag in the page. You can put <style></style> anywhere in your html page. But, Standard is to put it within the <head> tag.

- **Inline:** Used inside HTML tags.
- You may ask which one will render in the browser. Well, Browsers render html, css pages from top to bottom. So, the bottom one will be prioritized. In our case, <h1> elements will first get navy color, then overridden with red, finally green.
- However, if you want to break this default behaviour, You can use “Specificity Rule” or “!important” (check the commented code) to override.
- **Note:** Internal, Inline css are not recommended as they have less portability (ex. If you want to use the same css in multiple pages, you must write it down separately, Where just a link tag will do the same in external css) and hard to maintain. !important is also discouraged to use unless it is really necessary.

CSS Selectors (selctors.html in css-resources-files folder)

- **HTML DOM (Document Object Model)**



- **Simple Selectors:**
  - Universal Selector (*)
  - Element Based (Type Selector)
  - ID Selector
  - Class Selector
- **Combinator Selector**
  - descendant selector (space)
  - child selector (>)
  - adjacent sibling selector (+)
  - general sibling selector (~)
- **Pseudo Classes**
- **Pseudo Elements**
- **Attribute Selectors**

Cascading Rule

Latest one will get precedence if specificity is same or larger

Specificity (or priority in browser rendering) Serial:
inherited style < universal < type < class < id
https://specificity.keegan.st/ (for testing in online)

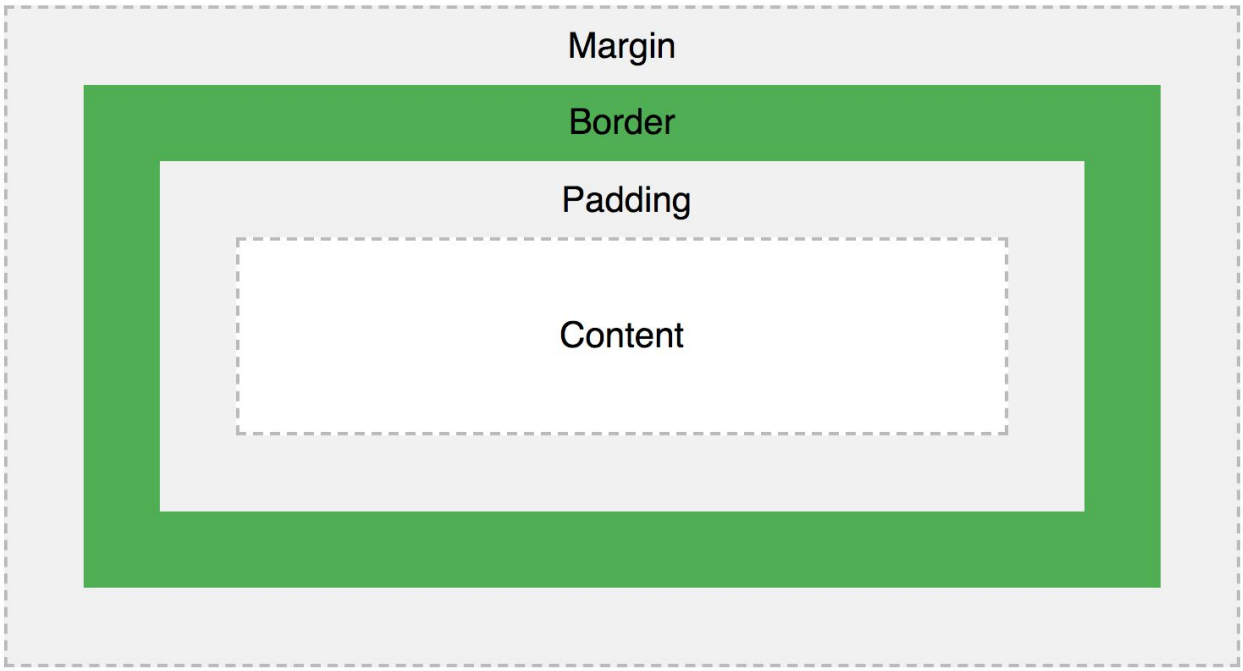
!important Rule
This will override both of the above

Basic CSS Properties & Concepts (basic-css-concepts.html in same folder)

- Color

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

- Background
- Border
- Margin
- Padding
- CSS Box Model



-
- CSS Box Sizing
    - By default, the width and height of an element is calculated like this:
      width + padding + border = actual width of an element

height + padding + border = actual height of an element

This means: When you set the width/height of an element, the element often appears bigger than you have set (because the element's border and padding are added to the element's specified width/height).

The box-sizing property allows us to include the padding and border in an element's total width and height.

If you set box-sizing: border-box; on an element, padding and border are included in the width and height.

- Display
    - Widely used values: block, inline, inline-block, none, visibility: hidden
- Position
    - Widely used values: static, relative, fixed, absolute, sticky
- Overflow
    - Widely used values: visible, hidden, scroll, auto
- Text
    - Check the reference code file
- Font

    -
    - Check the reference code file
- Lists
    - Check the reference code file
- Tables
    - Check the reference code file
- Opacity
    - Opacity: 0.5/ 50% (1 means 100%, 0 means 0% opacity)
- Float
    - Float Property Example: https://www.w3schools.com/css/tryit.asp?filename=trycss_layout_float
    - Clear Property Example https://www.w3schools.com/css/tryit.asp?filename=trycss_layout_clear
- Align
    - https://www.w3schools.com/css/css_align.asp (check the examples)

Advanced CSS Properties & Concepts

- Styling Image
    - https://www.w3schools.com/css/css3_images.asp
- Web Font
    - @font-face, web fonts
- Media Queries
    - https://www.w3schools.com/css/css_rwd_mediaqueries.asp
- CSS Variables
    - https://www.w3schools.com/css/css3_variables.asp
- CSS Flexbox
    - https://www.w3schools.com/css/css3_flexbox.asp

CSS Frameworks

- Less, SASS, Bootstrap etc.

Bootstrap Basics

- Responsive Grid System & How it works in bootstrap

Resources

- https://www.w3schools.com/css/default.asp (css)
- https://www.w3schools.com/bootstrap4/default.asp (bootstrap)
- https://www.lynda.com/CSS-tutorials/CSS-Essential-Training/5038219-2.html (video tutorial)
- https://medium.com/@madhum86/css-font-sizing-pixels-vs-em-vs-rem-vs-percent-vs-viewport-units-b1485716afe7 (to read more on px, em, rem concepts)