Google | طويق

CSS 1

أكاديمية طويق
TUWAIQ ACADEMY

وزارة الاتصالات وتقنية المعلومات
MINISTRY OF COMMUNICATIONS AND INFORMATION TECHNOLOGY

الاتحاد السعودي للأمن السيبراني والبرمجة والدرونز
SAUDI FEDERATION FOR CYBERSECURITY, PROGRAMMING & DRONES

مهارات المستقبل

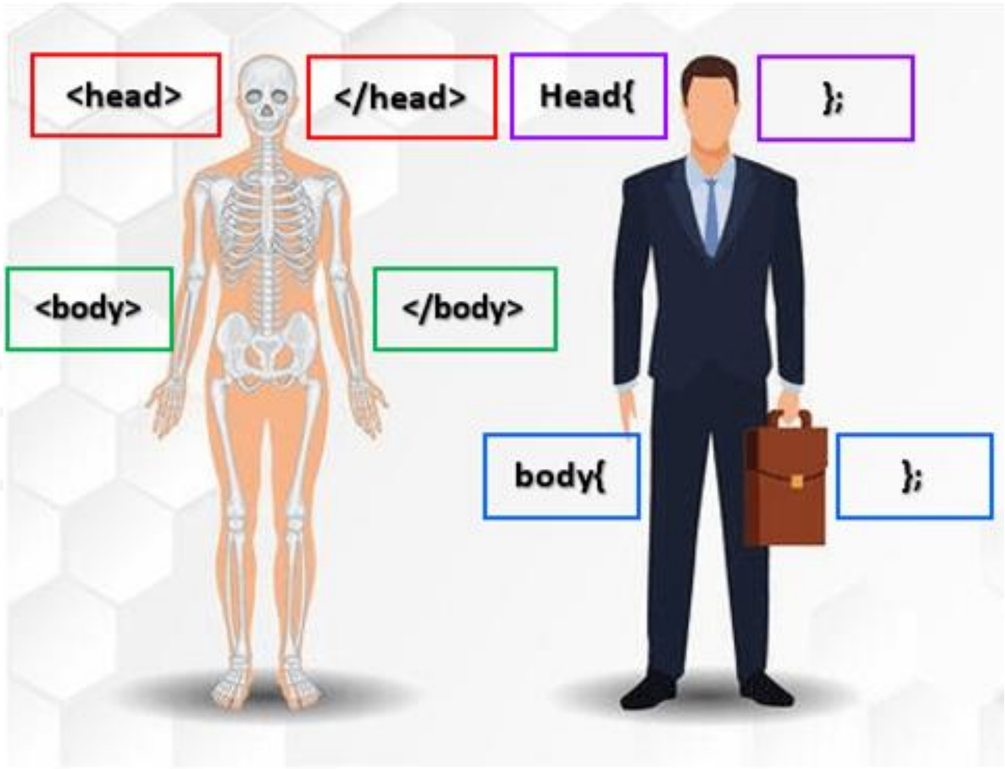
stc

Objective

- Inline CSS
- Internal CSS
- External CSS
- Box model and position

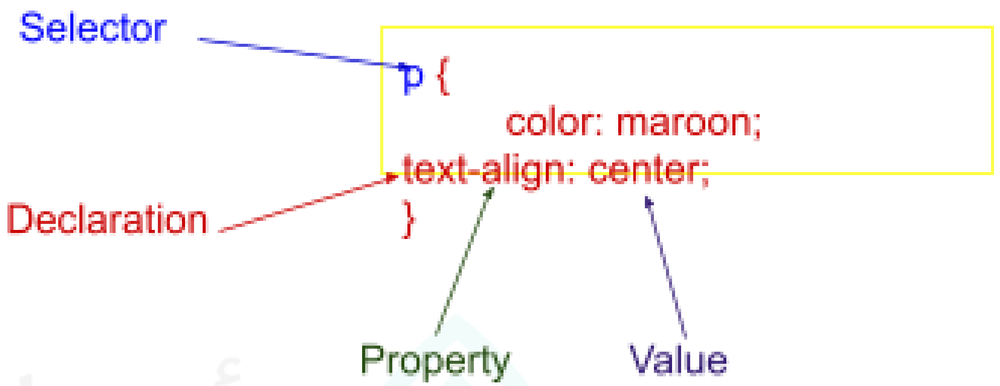
Introduction to CSS

- **Cascading Style Sheets (CSS):** a language used to change the presentation and look of a particular document written in a markup language



CSS Selectors

- We have thus far only worked with one type of selector. Here the selector is always an element.
- You can also use class and ID selectors
  - Class selector: .class
  - ID selector: #ID
  - Universal selector: *
  - Attributes
  - Pesudo classes
  - Combinators




```
p {
    color: red;
    font-family: Arial, Helvetica;
    background-color: blue;
    }

body{
    text-align: center;
}
```

Properties

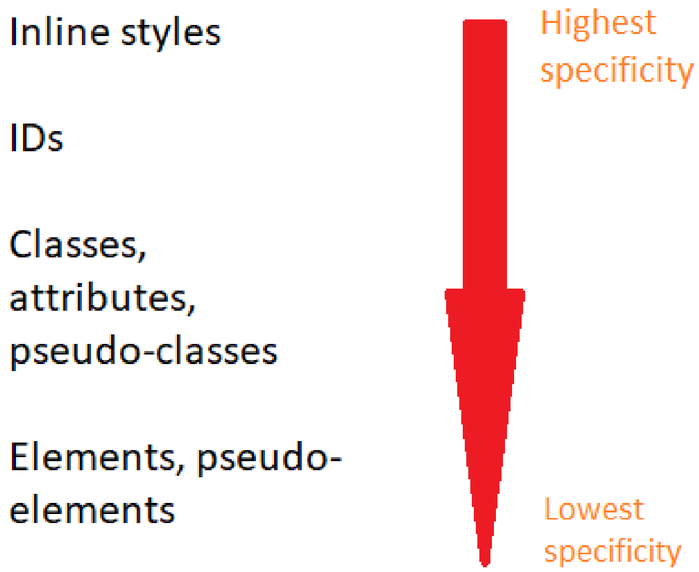
- properties are the styles you want to apply to the targeted element.
  - list properties
  - font properties
  - border properties
  - Other, here

Values

- Values are written immediately after the colon that separates them from CSS properties.
  - text
  - integers
  - unit, measurement
  - URL
  - colours
  - auto, none, inherit

Cascade: Specificity

- “Cascade” has to do with the order in which rules are applied
- NB! The more specific a rule is, the higher its precedence



Inline Style

- HTML elements are described using attributes and properties

```
<!-- Inline CSS Styles -->

<h1 style="font-weight: bold; font-size: 42px;">
  I will be bold and 42 pixels!
</h1>

<p style="color: blue;">I will be blue!</p>
```

- To style an element individually by changing that element’s properties

- Like all other attributes, the style attribute goes inside the element's beginning tag, right after the tag name. After specifying that you are changing the style attribute, you type =, and then, within double quotes, list the properties you want to change and after a colon specify the value for that property.

Internal CSS

- The example shows how you can define a CSS rule **in the head part** of your HTML template. This is called internal CSS. The example below shows a CSS rule that will cause all h1 to be in the color blue.

- CSS follows the following syntax: **A style sheet consists of a selector and a declaration:**

  - **The selector** indicates which HTML element you want to style

  - **The declaration block** contains one or more declarations separated by ;. A declaration always ends with a ; and is surrounded by {}. Each declaration includes a property and a value, separated by a :

```
<!DOCTYPE html>
<html>
    <head>
        <style>
            h1 (1)
                {
                    color:blue; (2)
                }
        </style>
    </head>
    <body>
    (3) <h1>Welcome To Wikitechy</h1>

        <p>Just a normal paragraph</p
        >
    </body>
</html>
```

External CSS

- If your website consists of many HTML files, you are likely to want to be able to apply the same style rules to all the web pages
- Create a separate file with the **extension .css**
- Within this file write all the style rules that you would like to specify
- **Link this external CSS file** to all the HTML files in where you would like the style rules applied

```html
<!-- HTML document linking to external CSS file -->

<!DOCTYPE html>
<html>
  <head>
    <!-- Links to external stylesheet named style.css -->
    <link rel="stylesheet" href="style.css">
  </head>

  <body>
    <h1 id="heading-1">This heading will be yellow.</h1>
    <p class="paragraph-text">This paragraph will have
    white (#fff) text and be underlined.</p>

    <h1 id="heading-2">This heading will purple.</h1>
    <p class="paragraph-text">This paragraph will be white
    and underlined.</p>
  </body>

</html>
```

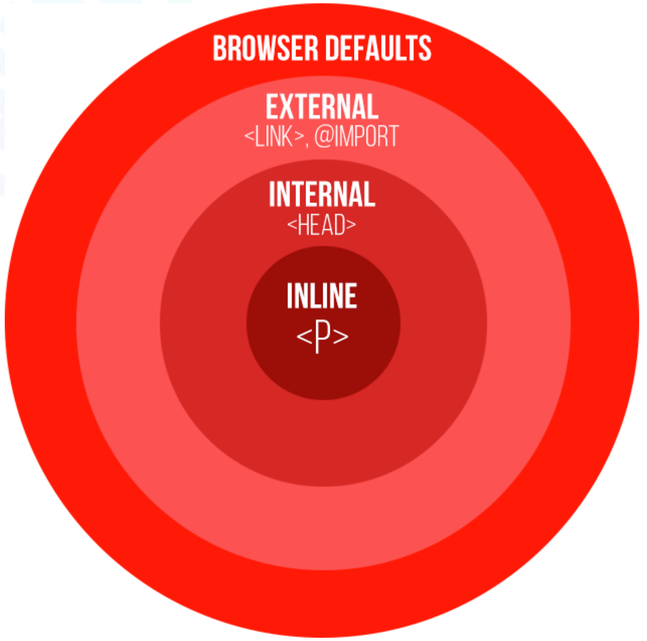
```css
/* External CSS Styles (in style.css file) */

h1#heading-1 {
  color: yellow;
}

h1#heading-2 {
  color: purple;
}

p.paragraph-text {
  color: #fff;
  text-decoration: underline;
}
```

Cascade

You could use a combination of internal CSS (declared in the head of your HTML document) and inline style. How would the style rules apply?

Essentially, **the closer to the element the style is the higher the precedence.**

For example, if you had the internal CSS in your page but you wanted one paragraph to be styled differently from the rest, you would simply use inline style for that one paragraph and that would overwrite the rule specified by the internal CSS.



🤔Internal, External or Inline: Which is Best?

Generally, it is better to use **external CSS** wherever possible because:

- Readability
- Maintainability, If you wanted to update the look and feel of a website, this could easily be done by simply replacing the external CSS file if only external CSS is used for the website.
- Easier to debug errors since all the CSS is in one place.

CSS Validator

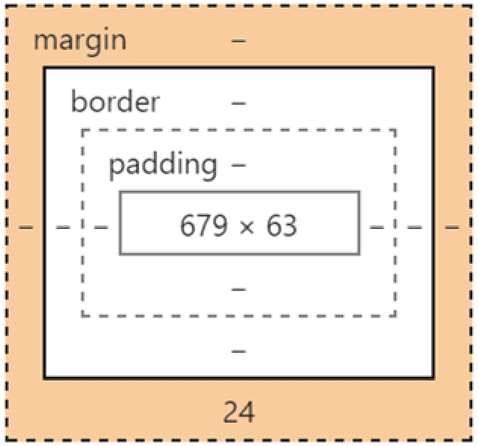
You need to follow the rules for formatting your CSS rules exactly or unexpected errors will occur when you try to view your web page in the browser

Being able to identify and correct these errors becomes easier with time and is an extremely important skill to develop.

To help you identify errors in your CSS, use this helpful tool.

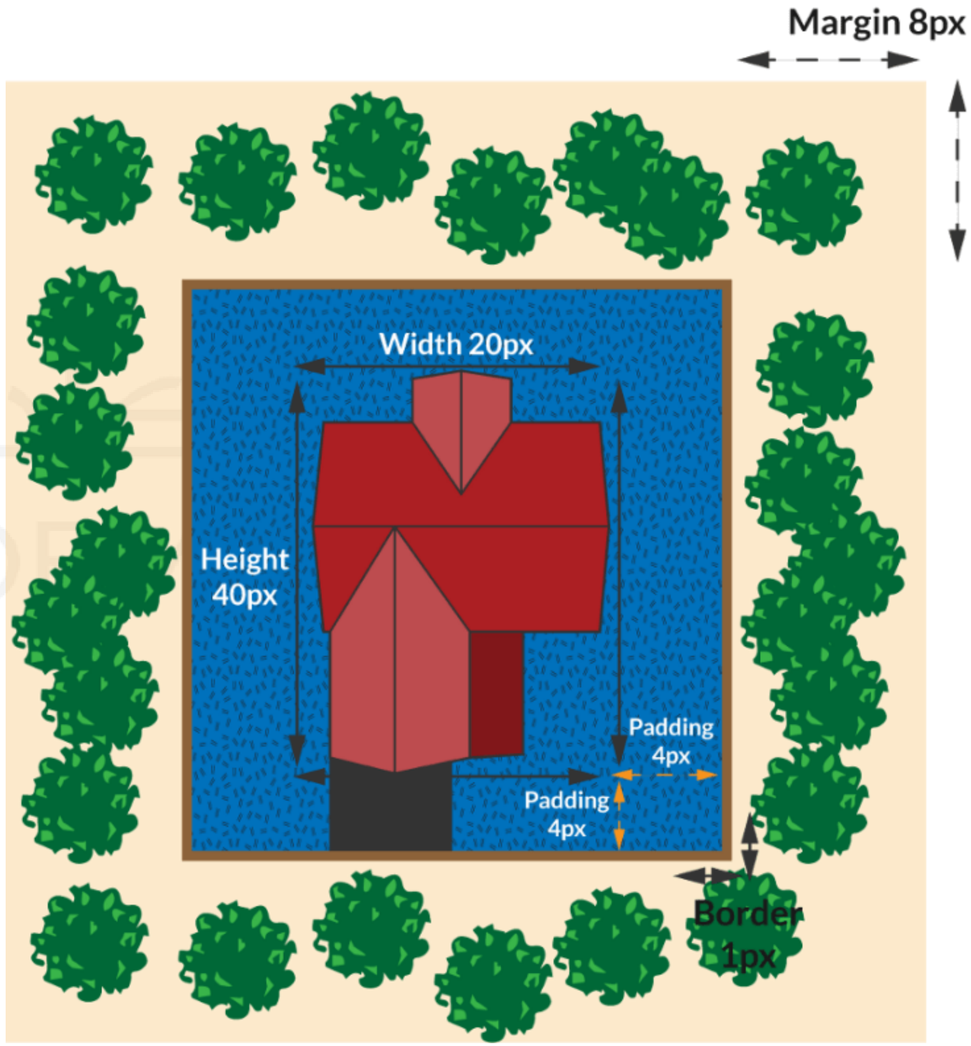
Box model

- In CSS, the term **"box model"** is used when talking about design and layout.
- **There are 5 important properties that allow you to size and distribute your HTML elements:**
  - Width.
  - Height.
  - 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.

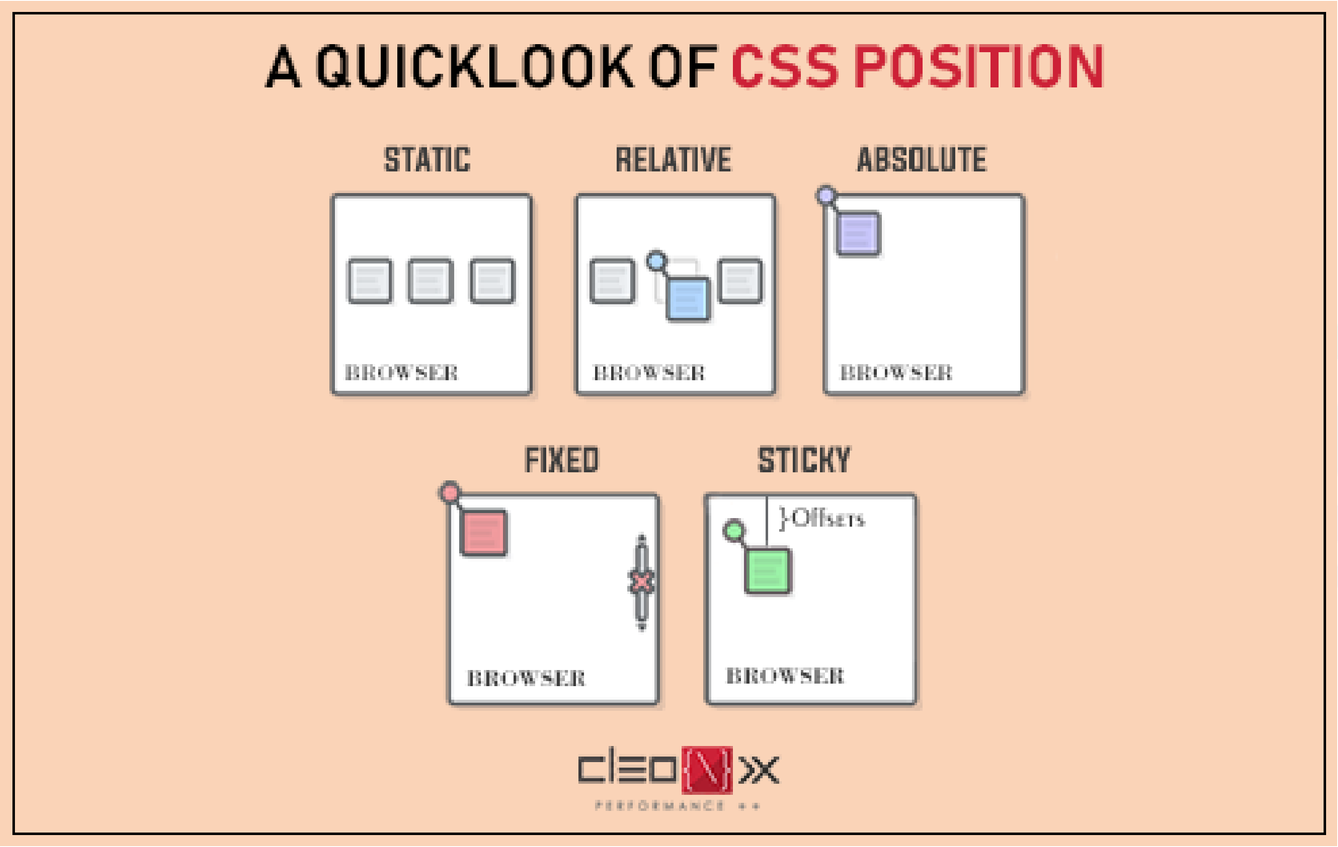


Box model

```
.myProperty{

    width: 20px;

    height: 40px;

    padding: 4px;

    border: 1px solid brown;

    margin: 8px;

    background-color:blue;

}
```



Position


A QUICKLOOK OF CSS POSITION
STATIC
RELATIVE
ABSOLUTE
BROWSER
BROWSER
BROWSER
FIXED
STICKY
}Offsets
BROWSER
BROWSER
cleoNX
PERFORMANCE ++

Position

- The position property specifies the type of positioning method used for an element.

- **There are five different position values:**

1. Static (default), this will not effect by top, bottom, right, left.

2. Relative, setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position.

3. Absolute, change position based on it’s parent.

* If we want to change child position just in parent, we should give relative position for the parent.

4. Fixed, it will be in the same position always, **EX:** got to the top button.

5. Sticky, when you scroll and reach at specific point, the element will be sticky,

   EX: when read an article and there is a links on the top, when scroll then links at the top will go down with you

Resources

- CSS-Tricks
- HTML Color Codes
- Free Frontend
- W3Schools Online Web Tutorials
- CodeAnalogies | CSS,JavaScript Tutorials
- Positioning
- Interneting Is Hard | Web Development Tutorials For Complete Beginners
- Learn CSS Layout

Summary

- Inline CSS
- Internal CSS
- External CSS
- Box model and position