

# HTML & CSS Basics

**Mohammad Zarif**



# Menu

- CSS Syntax
  - Types of insert CSS
  - Color
  - Unit
- CSS Selectors
  - Simple
  - Combinator
  - Pseudo class
  - Pseudo element
  - Attribute
- HTML Forms (Section1)
  - <form>
  - <input> (and its types)



# CSS Syntax

selector

p

declaration

{ color:blue; }

↑  
property

↑  
value



# Types of insert CSS

- External CSS : with an external style sheet you can change your entire website just in a file!

- Syntax : (in <head> tag , add this line)

```
<link rel="stylesheet" href="path_of_style_sheet_file.css">
```

- Internal CSS : an internal style sheet may be used if one single HTML page has unique styles.

- Syntax : (in <head> tag, add this tag)

```
<style>
```

```
/* your styles */
```

```
</style>
```



# Types of inline CSS

- Inline CSS : an inline style may be used to apply a unique style for a single element.

- Syntax:

```
<p style="your styles (don't forget semicolon after each property!)">
```

```
    some text
```

```
</p>
```



# CSS Colors

Tomato

`rgb(255, 99, 71)`

`#ff6347`

`hsl(9, 100%, 64%)`

`rgba(255, 99, 71, 0.5)`

`hsla(9, 100%, 64%, 0.5)`

# HSL

In CSS, a color can be specified using hue, saturation, and lightness (HSL) in the form:

**`hsl(hue, saturation, lightness)`**

Hue is a degree on the color wheel from 0 to 360. 0 is red, 120 is green, and 240 is blue.

Saturation is a percentage value, 0% means a shade of gray, and 100% is the full color.

Lightness is also a percentage, 0% is black, 50% is neither light or dark, 100% is white



# CSS Units

There are two types of length units: **absolute** and **relative**

## ***Absolute Lengths***

The absolute length units are fixed and a length expressed in any of these will appear as exactly that size.

Absolute length units are not recommended for use on screen, because screen sizes vary so much. However, they can be used if the output medium is known, such as for print layout.

**Some of absolute lengths** : cm , mm , in, px, pt (1pt = 1/72 in), pc (picas) (1pc = 12 pt)





# CSS Units

## ***Relative Lengths***

Relative length units specify a length relative to another length property. Relative length units scales better between different rendering mediums.

Some of relative units :

em: Relative to the font-size of the element (2em means 2 times the size of the current font)

rem: Relative to font-size of the root element

vw : Relative to 1% of the width of the viewport

vh: Relative to 1% of the height of the viewport

%: Relative to the parent element



# CSS Selectors

A CSS selector selects the HTML element(s) you want to style.

CSS selectors are used to "find" (or select) the HTML elements you want to style.

We divide CSS selectors into five categories:

- Simple selectors (select elements based on name, id, class)
- Combinator selectors (select elements based on a specific relationship between them)
- Pseudo-class selectors (select elements based on a certain state)
- Pseudo-elements selectors (select and style a part of an element)
- Attribute selectors (select elements based on an attribute or attribute value)



# CSS Selectors

- Simple Selector
  - Element selector: select element based on tag name

- Example:

```
p {  
    color: red;  
}
```

- Id selector: select element based on **unique** identifier (not starts with number)

- Example:

```
#myId {  
    color: blue;  
}
```



# CSS Selectors

- Simple selector
  - Class Selector: select element based on className
    - Example:

```
.className {  
    color: green;  
}
```
  - Universal Selector: select all HTML elements on the page:
    - Example :

```
* {  
    text: orange;  
}
```



# CSS Selectors

- Combinator Selectors:
  - Descendant selector (space): matches all elements that are descendants of a specified element.
  - Child selector (>): select all elements that are the children of a specified element.
  - Adjacent sibling selector (+): selects an element that is directly after another specific element.
  - General sibling selector (~): selects all element that are the next siblings of a specific element.



# Pseudo class

A pseudo class is used to define a special state of an element.

Syntax:

```
selector:pseudo-class {  
    property: value;  
    ...  
}
```

Some famous pseudo classes:

- Hover (:hover) : activates when mouse hover on element
- Link (:link): activates when a link is in unvisited state
- Visited (:visited) : activates when a link is in visited state
- Active (:active) : activates when a link is in selected state



# Pseudo class

- First child (:first-child) : selects the first child of a element
- Language (:lang(language)) : allows to define some style for different language
- Focus (:focus) : actives when a form element is focused
- Nth child (:nth-child(n)) : selects the nth child of a element



# Pseudo element

A CSS pseudo-element is used to style specified parts of an element.

Syntax:

```
selector::pseudo-element {  
    property: value;  
    ...  
}
```

Some famous pseudo-elements:

- first line (`::first-line`): active specific styles for first line of a paragraph (just for block level elements)





# Pseudo element

- first letter (`::first-letter`): active specific styles for first letter of each word in paragraph (just for block level elements)
- Before (`::before`) : it can be insert some content before the content of an element
- After (`::after`) : it can be insert some content after the content of an element
- Marker (`::marker`) : selects the markers of list items
- Selection (`::selection`): actives when select a part of an element by user



# Attribute selector

It is possible to style HTML elements that have specific attributes or attribute values.

The [attribute] selector is used to select elements with a specified attribute.

Types of attribute selection :

- Simple

Syntax : selector[attribute] { ... }

- With value

Syntax: selector[attribute="value"] {...}



# Attribute selector

- Containing : is used to select elements with an attribute value containing a specified word.

Syntax: `selector[attribute~="value"] {...}`

- Exact value : selector is used to select elements with the specified attribute, whose value can be exactly the specified value, or the specified value followed by a hyphen (-).

Syntax: `selector[attribute|=“value”] {...}`

- Starts with ... : selector is used to select elements with the specified attribute, whose value starts with the specified value.

Syntax: `selector[attribute^="value"] {...}`



# Attribute selector

- Ends with ... : selector is used to select elements whose attribute value ends with a specified value.

Syntax: `selector[attribute$="value"] { ... }`

- Contains a value : selector is used to select elements whose attribute value contains a specified value.

Syntax: `selector[attribute*="value"] { ... }`



# HTML Forms

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

The HTML `<form>` element is used to create an HTML form for user input

The `<form>` element is a container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc.

Some form attributes :

- Action: The action attribute defines the action to be performed when the form is submitted.



# HTML Forms

- Target: The target attribute specifies where to display the response that is received after submitting the form.
- Method: The method attribute specifies the HTTP method to be used when submitting the form data.



# HTML Forms (Input Section1)

One of the most used form element is the `<input>` element.

The `<input>` element can be displayed in several ways, depending on the type attribute.



# Input

List of input types :

`<input type="button">`

`<input type="checkbox">`

`<input type="color">`

`<input type="date">`

`<input type="datetime-local">`

`<input type="email">`

`<input type="file">`

`<input type="hidden">`

`<input type="image">`

`<input type="month">`

`<input type="number">`

`<input type="password">`

`<input type="radio">`

`<input type="range">`

`<input type="reset">`

`<input type="search">`

`<input type="submit">`

`<input type="tel">`

`<input type="text">`

`<input type="time">`

`<input type="url">`

`<input type="week">`







**THE END**

