

# Module 3-1

Introduction to HTML and CSS

# Objectives

- Understand the basics of HTML
  - Document structure
  - Tags and attributes
- Understand the following tags:
  - Headings
  - Paragraphs
  - Links
  - Images
- Utilize elements to build a form
- Understand action attribute of form tag
- What is Semantic HTML
- Basics of CSS and how to create style declarations
- Basics of fonts and the browser
- Basics of colors and properties for working with them

# Building blocks of the web

- HTML
  - Provides content for page
- CSS
  - Provides presentation, formatting, and layout
- JS
  - Controls the behavior of different elements

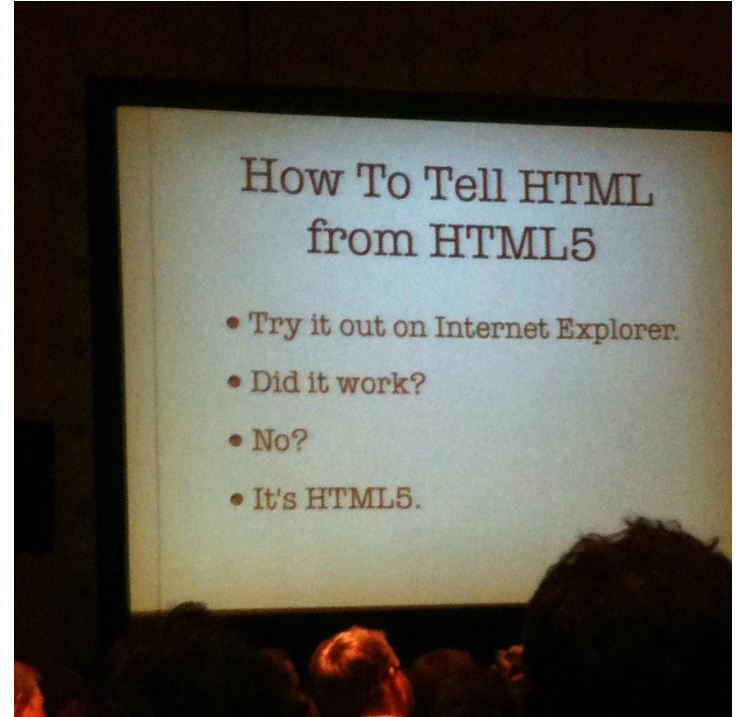


# HTML: Language Introduction

- HTML (Hyper Text Markup Language) is a declarative language interpreted by internet browsers.
- Unlike Java, there is no separate step needed to compile the code, the instructions written in HTML are simply interpreted by the browser.
- Standard markup language for creating Web pages
- Consists of series of elements
  - Tell the browser how to display the content

# HTML History

- 1989 Tim Berners-Lee invented www
- 1991 Tim Berners-Lee invented HTML
- 1993 Dave Raggett drafted HTML+
- 1995 Working group defined HTML 2.0
- 1997 W3C Recommendation HTML 3.2
- 2015 W3C Recommendation HTML5



# HTML: Language Introduction

---

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>

    <h1>My First Heading</h1>
    <p>My first paragraph.</p>

  </body>
</html>
```

# HTML: Elements

- HTML defined by a start tag, some content and (for non-void elements) an end tag. Here are a few common ones:
  - `<html>...</html>`
  - `<head>...</head>`
  - `<body>...</body>`
  - `<h1>...</h1>`
- The rules for tags are straightforward:
  - There is a begin tag and an end tag, the end tag is denoted by the slash.
  - Some tags can be defined in a self-closing manner. For example, the tag for an image can be constructed like so:
    - `` as an alternative to ` </img>`
    - `<br>` - single line break is a void element



# HTML: Attributes

- Provide additional information about HTML elements
- Always specified in the start tag
- Usually come in name/value pairs – src="photo.jpg"
  - src specifies the path to image to be displayed
    - ``
    - ``



# Common HTML tags

- Heading tags (h1, h2, h3, h4, h5, h6)
  - Range from most important (h1) to least important (h6)
  - Typically vary in size – largest (h1) to smallest (h6)
  - Best practice – only one h1 tag per page (main topic)
- Paragraph tags (p)
  - Defines paragraph
  - Starts a new line
- Emphasis (em)
  - Emphasizes text typically with italics
- Strong (strong)
  - Important text typically with bold



**This is heading 1**

**This is heading 2**

**This is heading 3**

**This is heading 4**

**This is heading 5**

**This is heading 6**

**Tip:** Use h1 to h6 elements only for headings. Do not use them just to make text bold or big. Use other tags for that.

# Common HTML tags

- Blockquote (blockquote)
  - Text is indented (unlike p tag)
  - Single cite attribute (contains a URL to the source of quote)
- Anchor (a)
  - Defines a hyperlink – link to another page or section
  - href attribute
- Image (img)
  - Links image to webpage
  - Placeholder signifying where image will appear on page
  - Two required attributes – src and alt

```
<blockquote cite="http://www.worldwildlife.org/who/index.html">
```

For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.

```
</blockquote>
```

## The blockquote element

Here is a quote from WWF's website:

For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.

## The a element

[Visit Techelevator.com!](http://www.techelevator.com)



```

```

# Common HTML tags

- Ordered and unordered lists (ol and ul)
  - Both use list item tag (li) to itemize

```
<p>The ol element defines an ordered list:</p>
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

<p>The ul element defines an unordered list:</p>
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

## The ol and ul elements

The ol element defines an ordered list:

1. Coffee
2. Tea
3. Milk

The ul element defines an unordered list:

- Coffee
- Tea
- Milk



- Table (table)
  - Consists of table rows (tr) and table data (td)
  - Table head (th) can be used in place of table data

```
<table>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>February</td>
    <td>$80</td>
  </tr>
</table>
```

## The table element

Month	Savings
January	\$100
February	\$80

# Forms

- Allow you to get information from the user
- Two attributes (method and action)
  - Method – HTTP method to submit form with
    - POST and GET!
  - Action – URL to process the form submission

```
<form action="/action_page.php" method="get" target="_blank">  
  <label for="fname">First name:</label>  
  <input type="text" id="fname" name="fname"><br><br>  
  <label for="lname">Last name:</label>  
  <input type="text" id="lname" name="lname"><br><br>  
  <input type="submit" value="Submit">  
</form>
```

First name:

Last name:



# Form Elements

- Input
  - Most common used form element
  - Type attribute
    - text, radio, checkbox, submit, button
    - name – sent to the server to be recognized and get the value

```
<form action="/action_page.php">
  <input type="radio" id="dog" name="pet" value="dog">
  <label for="dog">Dog</label><br>
  <input type="radio" id="cat" name="pet" value="cat">
  <label for="cat">Cat</label><br>
  <input type="radio" id="other" name="pet" value="other">
  <label for="other">Other</label><br><br>
  <input type="submit" value="Submit">
</form>
```

The **input type="radio"** defines a radio button:

- ☐ Dog
- ☐ Cat
- ☐ Other

```
<!DOCTYPE html>
<html>

  <head>
    <title>Text Input Control</title>
  </head>

  <body>
    <form >
      First name: <input type = "text" name = "first_name" />
      <br>
      Last name: <input type = "text" name = "last_name" />
    </form>
  </body>

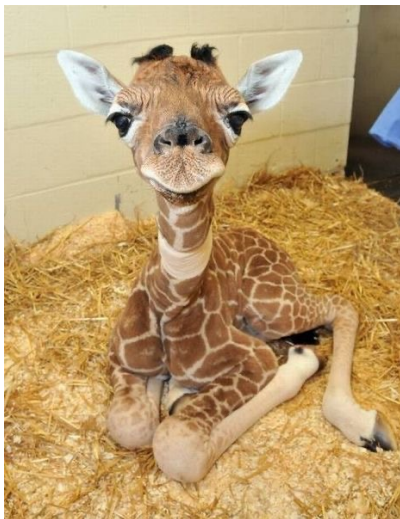
</html>
```

This will produce the following result –

First name:

Last name:

# Form Elements



Type	Description
button	A push-button with no default behavior displaying the value of the value attribute, empty by default.
text	The default value. A single-line text field. Line-breaks are automatically removed from the input value.
checkbox	A check box allowing single values to be selected/deselected.
radio	A radio button, allowing a single value to be selected out of multiple choices with the same name value.
date	A control for entering a date (year, month, and day, with no time). Opens a date picker or numeric wheels for year, month, day when active in supporting browsers.
email	A field for editing an email address. Looks like a text input, but has validation parameters and relevant keyboard in supporting browsers and devices with dynamic keyboards.
color	A control for specifying a color; opening a color picker when active in supporting browsers.
file	A control that lets the user select a file. Uses the accept attribute to define the types of files that the control can select.
number	A control for entering a number. Displays a spinner and adds default validation when supported. Displays a numeric keypad in some devices with dynamic keypads.
submit	A button that submits the form.

# Form Elements

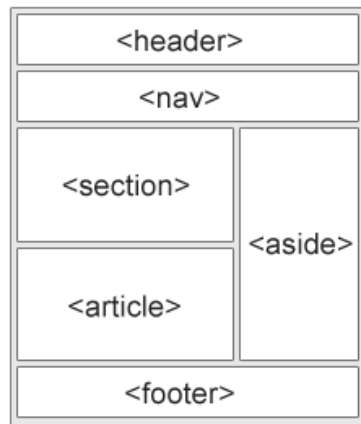
Attribute	Description
name	Name of the input form control. Submitted with the form as part of a name/value pair.
type	Type of input form control.
value	Current value of the form control. Submitted with the form as part of a name/value pair.
disabled	Whether the form control is disabled.
minlength	Minimum length (number of characters) of value.
maxlength	Maximum length (number of characters) of value placeholder.
required	Boolean. A value is required or must be checked for the form to be submittable.



# Semantic HTML

- Elements that have meaning
- Describes meaning to both browser and developer
  - Non-semantic – div and span
  - Semantic – form, table, article

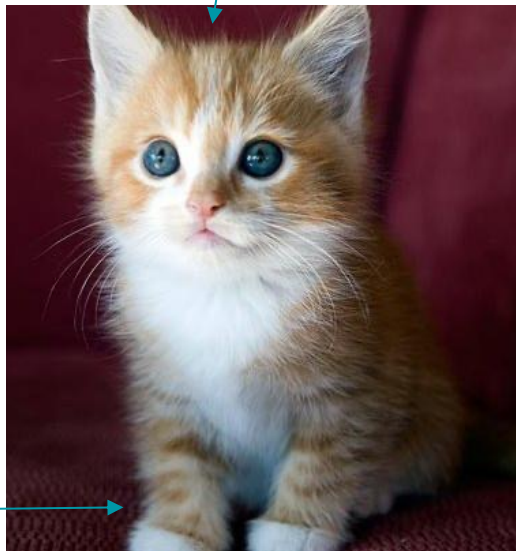
- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>
- <header>
- <main>
- <mark>
- <nav>
- <section>
- <summary>
- <time>



# Common Semantic elements

- Header (not head)
  - Headings, navigation links, company name, etc.
- Footer
  - Copyright info, company name, links to related pages

Header



Footer



# Let's Code!

# What is CSS?

- CSS (Cascading Style Sheets) are used to change the default style of HTML elements.
  - Brings page to life with layout and style.
- CSS code can be inlined (included on an element), internal (included within the <head>...</head> section of a HTML document), or external (it can be included in its own separate file).
  - The latter is much preferred.
  - File names should end with a \*.css extension.



# Inline CSS

- Placed in tag with style attribute
- Use as sparingly as possible!

```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>
```

---

**This is a heading**

This is a paragraph.

# Internal CSS

- Typically used for unique page in web site
- Style element placed in head section

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: linen;
}

h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

**This is a heading**

This is a paragraph.

# External CSS

- One file can be used for all pages in web site
- Link element is placed in head section, referencing external style sheet

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

"mystyle.css"

```
body {
  background-color: lightblue;
}

h1 {
  color: navy;
  margin-left: 20px;
}
```

**This is a heading**

This is a paragraph.

# External CSS

- Comments follow block comments in java => /\* \*/
- Type selector selects all elements with node name
- Curly braces denote selector blocks
- Contains one or more style declarations
- name : value pairs

"mystyle.css"

```
body {  
    background-color: lightblue;  
}  
  
h1 {  
    color: navy;  
    margin-left: 20px;  
}
```



# CSS Font Families

- Serif
- San-serif
- Monospace
- Cursive
- Fantasy

```
<!DOCTYPE html>
<html>
<head>
<style>
.p1 {
  font-family: "Times New Roman", Times, serif;
}
.p2 {
  font-family: Arial, Helvetica, sans-serif;
}
.p3 {
  font-family: "Lucida Console", "Courier New", monospace;
}
</style>
</head>
<body>

<h1>CSS font-family</h1>
<p class="p1">This is a paragraph, shown in the Times New Roman font.</p>
<p class="p2">This is a paragraph, shown in the Arial font.</p>
<p class="p3">This is a paragraph, shown in the Lucida Console font.</p>

</body>
</html>
```



Generic Font Family	Examples of Font Names
Serif	Times New Roman Georgia Garamond
Sans-serif	Arial Verdana Helvetica
Monospace	Courier New Lucida Console Monaco
Cursive	<i>Brush Script MT</i> <i>Lucida Handwriting</i>
Fantasy	<b>Copperplate</b> Papyrus

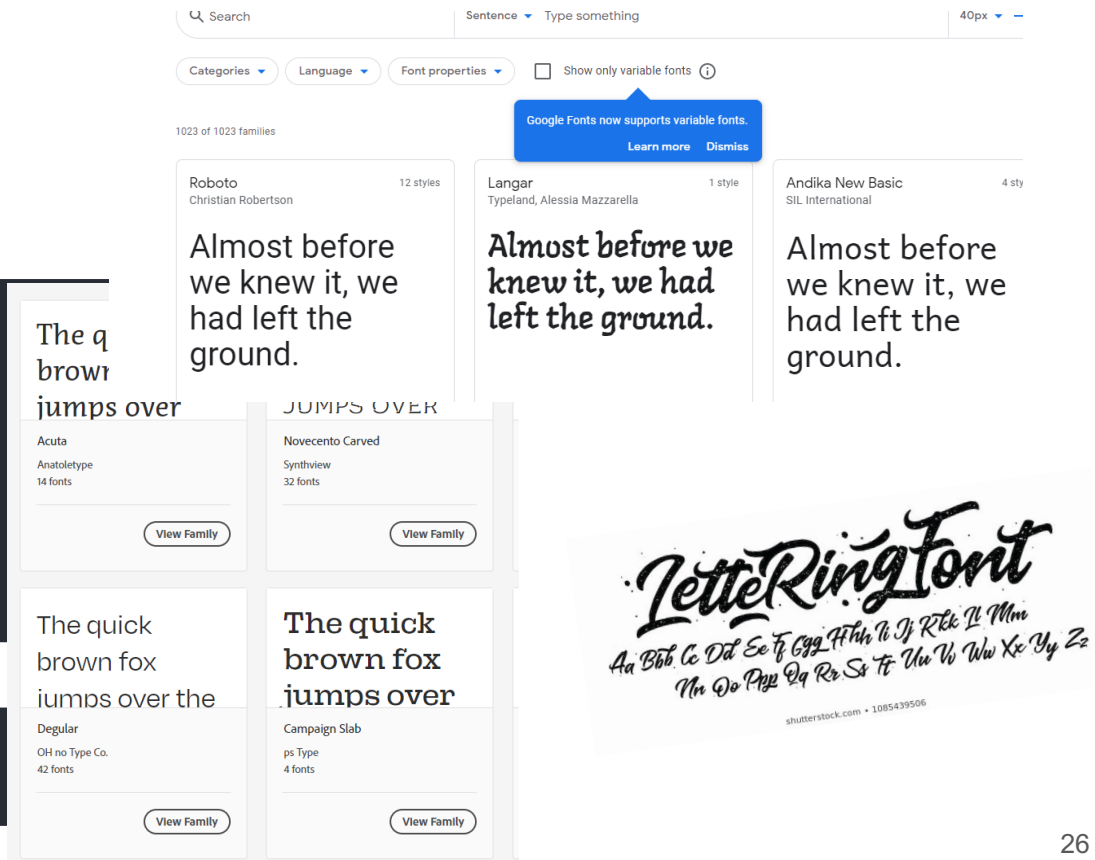
# Web Font Providers

- [Google Fonts](#)
- [Adobe Fonts](#)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>CSS Basics</title>
    <link rel="stylesheet" href="css/styles.css" />
    <link
      href="https://fonts.googleapis.com/css2?family=Roboto&display=swap"
      rel="stylesheet"
    />
  </head>
</html>
```

Next, update your selector to use the new font family:

```
body {
  font-family: 'Roboto', sans-serif;
}
```



# Font Properties

- Font size
- Font weight
- Font style
- Line height
- Text align
- Text decoration
- Text transform

```
h1 {  
  font-size: 40px;  
}
```

```
h1 {  
  font-size: 2.5em; /* 40px/16=2.5em */  
}
```

```
body {  
  font-size: 100%;  
}
```

```
h1 {  
  text-align: center;  
}
```

```
h2 {  
  text-align: left;  
}
```

```
h3 {  
  text-align: right;  
}
```

```
h1 {  
  text-decoration: overline;  
}
```

```
h2 {  
  text-decoration: line-through;  
}
```

```
h3 {  
  text-decoration: underline;  
}
```

```
p.small {  
  line-height: 0.8;  
}
```

```
p.big {  
  line-height: 1.8;  
}
```

```
p.uppercase {  
  text-transform: uppercase;  
}
```

```
p.lowercase {  
  text-transform: lowercase;  
}
```

```
p.capitalize {  
  text-transform: capitalize;  
}
```



# CSS Colors

- Predefined color names
- RGB
- HEX
- HSL
- RGBA
- HSLA

AliceBlue #F0F8FF	AntiqueWhite #FAEBD7	Aqua #00FFFF
Aquamarine #7FFFD4	Azure #F0FFFF	Beige #F5F5DC
Bisque #FFE4C4	Black #000000	BlanchedAlmond #FFEBCD
Blue #0000FF	BlueViolet #8A2BE2	Brown #A52A2A
	CadetBlue #5CB8A0	Chartreuse #7FFF00

rgb(0, 0, 255)
rgb(238, 130, 238)
rgb(106, 90, 205)

#0000FF
#ee82ee
#6a5acd

rgba(255, 99, 71, 0)	rgba(255, 99, 71, 0.2)	hsl(0, 100%, 50%)	hsl(240, 100%, 50%)
rgba(255, 99, 71, 0.4)	rgba(255, 99, 71, 0.6)	hsl(147, 50%, 47%)	hsl(300, 76%, 72%)
rgba(255, 99, 71, 0.8)	rgba(255, 99, 71, 1)	hsl(39, 100%, 50%)	hsl(248, 53%, 58%)

Let's Code! But first ... review!

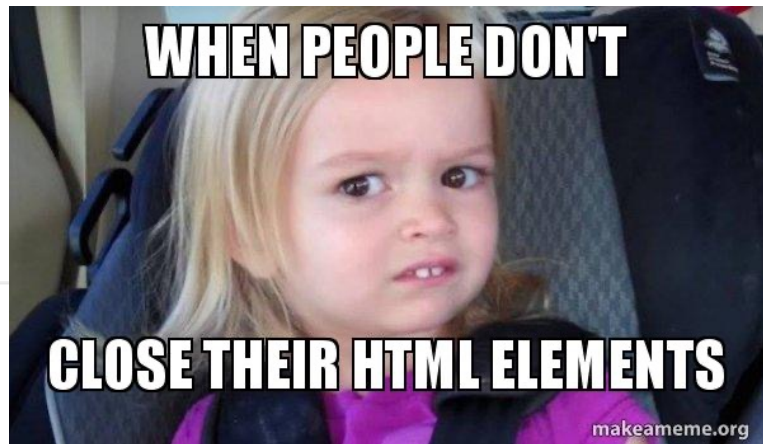
# Objectives

- Understand the basics of HTML
  - Document structure
  - Tags and attributes

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>

    <h1>My First Heading</h1>
    <p>My first paragraph.</p>

  </body>
</html>
```



# Objectives

- Understand the basics of HTML
  - Document structure
  - Tags and attributes
- Understand the following tags:
  - Headings
  - Paragraphs
  - Links
  - Images



```

```

# Objectives

- Understand the basics of HTML
  - Document structure
  - Tags and attributes
- Understand the following tags:
  - Headings
  - Paragraphs
  - Links
  - Images
- Utilize elements to build a form

## Add User

---

Username:

First Name:

Last Name:



# Objectives

- Understand the basics of HTML
  - Document structure
  - Tags and attributes
- Understand the following tags:
  - Headings
  - Paragraphs
  - Links
  - Images
- Utilize elements to build a form
- Understand action attribute of form tag

## Shipping Address

Name:

Address:

City:

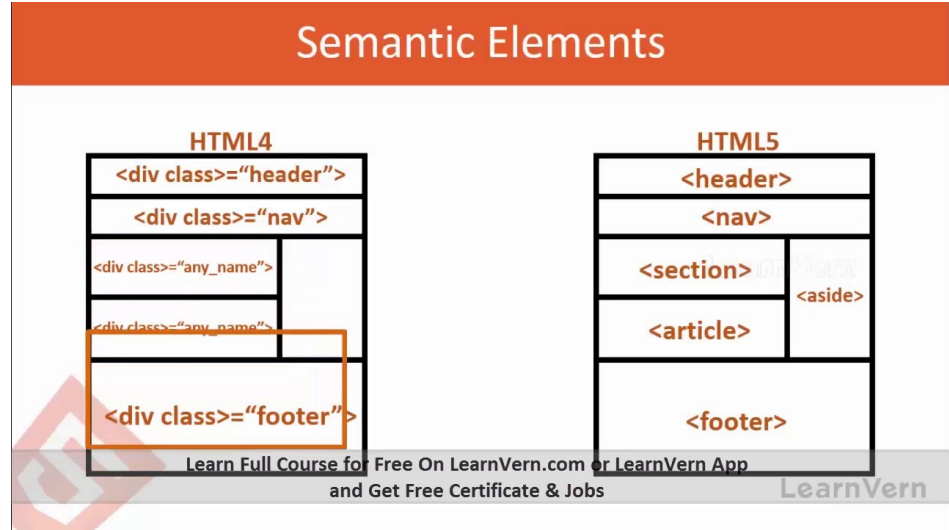
State:

Zip:

```
<form action="formprocessor.html" method="get">
  <fieldset>
    <legend>Shipping Address</legend>
    <p>Name: <input type="text" name="Name" /></p>
    <p>Address: <input type="text" name="Address" /></p>
    <p>City: <input type="text" name="City" /></p>
    <p>State: <input type="text" name="State" /></p>
    <p>Zip: <input type="text" name="Zip" /></p>
  </fieldset>
</form>
```

# Objectives

- Understand the basics of HTML
  - Document structure
  - Tags and attributes
- Understand the following tags:
  - Headings
  - Paragraphs
  - Links
  - Images
- Utilize elements to build a form
- Understand action attribute of form tag
- What is Semantic HTML



# Objectives

- Understand the basics of HTML
  - Document structure
  - Tags and attributes
- Understand the following tags:
  - Headings
  - Paragraphs
  - Links
  - Images
- Utilize elements to build a form
- Understand action attribute of form tag
- What is Semantic HTML
- Basics of CSS and how to create style declarations



# Objectives

- Understand the basics of HTML
  - Document structure
  - Tags and attributes
- Understand the following tags:
  - Headings
  - Paragraphs
  - Links
  - Images
- Utilize elements to build a form
- Understand action attribute of form tag
- What is Semantic HTML
- Basics of CSS and how to create style declarations
- Basics of fonts and the browser



# Objectives

- Understand the basics of HTML
  - Document structure
  - Tags and attributes
- Understand the following tags:
  - Headings
  - Paragraphs
  - Links
  - Images
- Utilize elements to build a form
- Understand action attribute of form tag
- What is Semantic HTML
- Basics of CSS and how to create style declarations
- Basics of fonts and the browser
- Basics of colors and properties for working with them



FFFFFF	000000	333333	666666	999999	cccccc	CCCC99	9999CC	666699
660000	663300	996633	003300	003333	003399	000066	330066	660066
990000	993300	CC9900	006600	336666	0033FF	000099	660099	990066
CC0000	CC3300	FFCC00	009900	006666	0066FF	0000CC	663399	CC0099
FF0000	FF3300	FFFF00	00CC00	009999	0099FF	0000FF	9900CC	FF0099
CC3333	FF6600	FFFF33	00FF00	00CCCC	00CCFF	3366FF	9933FF	FF00FF
FF6666	FF6633	FFFF66	66FF66	66CCCC	00FFFF	3399FF	9966FF	FF66FF
FF9999	FF9966	FFFF99	99FF99	66FFCC	99FFFF	66CCFF	9999FF	FF99FF
FFCCCC	FFCC99	FFFFCC	CCFFCC	99FFCC	CCFFFF	99CCFF	CCCCFF	FFCCFF