Date of Experiment: dd/mm/yyyy

Practical -7

Introduction to CSS

1. Create a webpage which displays "Hello World" with font size 20 pixels, bold format, in "Times New Roman" font and green in color using inline CSS, internal/embedded CSS and external CSS.
2. Create a webpage which implements CSS for the following appearance.
   - For h1 background-color: green
   - For p background-color: red
   - For div background-color: gray
   - For Body set Background image
3. Create a webpage which implements CSS to set different font type, color and size for H1 to H6.
4. Create a webpage which implements different selectors.
5. Create a webpage which implements different types of borders.
6. Create a webpage which implements different CSS text property.
7. Create a webpage which generate zebra stripe table. [Hint: Use the ***nth-child()*** selector and add *background-color* of your choice to all odd (or even) table rows/columns. Apply on tr or td elements]
8. Create a webpage which implements following **pseudo-classes** and **pseudo-elements**. [

   Syntax:

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

   ]

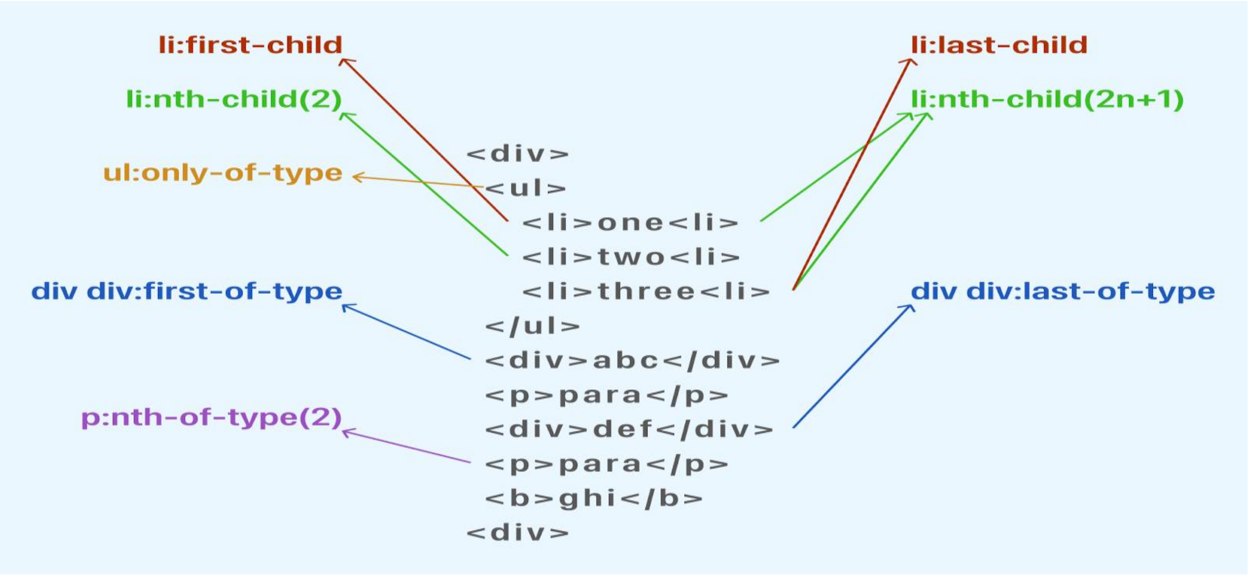
CSS Pseudo Classes

Selector	Example	Example description
:first-child	p:first-child	Selects every <p> elements that is the first child of its parent
:first-of-type	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
:last-child	p:last-child	Selects every <p> elements that is the last child of its parent
:last-of-type	p:last-of-type	Selects every <p> element that is the last <p> element of its parent
:nth-child(n)	p:nth-child(2)	Selects every <p> element that is the second child of its parent

:nth-last-child(n)	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
:nth-last-of-type(n)	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child
:nth-of-type(n)	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent
:only-of-type	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
:only-child	p:only-child	Selects every <p> element that is the only child of its parent

CSS Pseudo Elements

Selector	Example	Example description
::first-letter	p::first-letter	Selects the first letter of every <p> element
::first-line	p::first-line	Selects the first line of every <p> element



```
<div>
 <ul>
   <li>one<li>
   <li>two<li>
   <li>three<li>
 </ul>
 <div>abc</div>
 <p>para</p>
 <div>def</div>
 <p>para</p>
 <b>ghi</b>
<div>
```

How to add CSS?

CSS can be added to HTML documents in 3 ways:

Inline - by using the style attribute inside HTML elements

```
            <h2 style="color:red;">CAUTION: Icy Road Conditions</h2>
            <h2>Please Slow Down!</h2>
```

Internal - by using a <style> element in the <head> section

```
<head>
        <style type="text/css">
                  h2 {color:red;}
         </style>
</head>
```

External - by using a <link> element to link to an external CSS file

style.css (separate file):

```
h2 {color:red;}
```

example.html file:

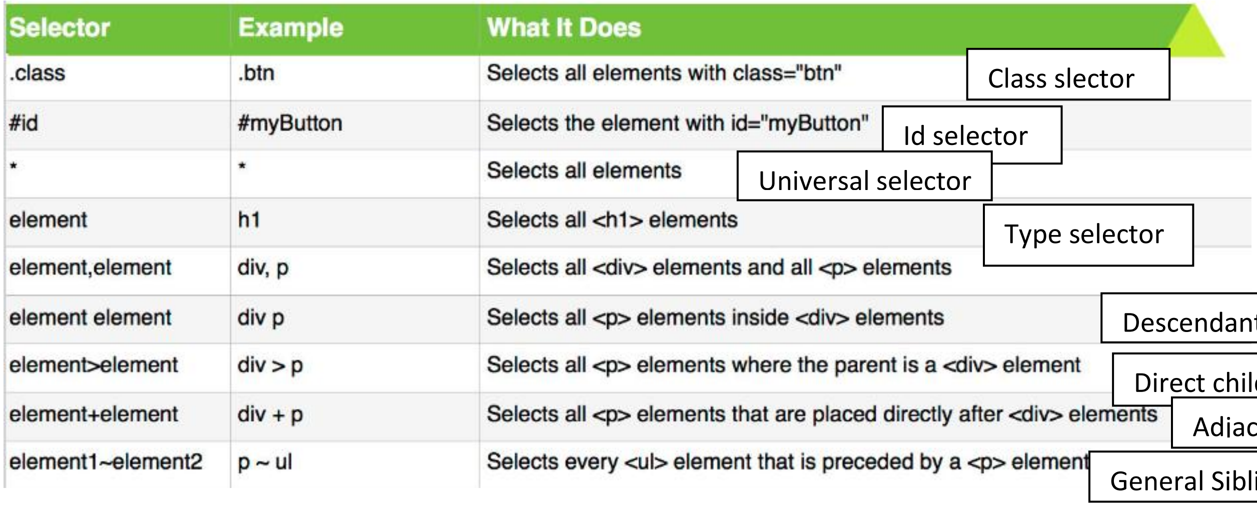
```
<head>
        <link rel="stylesheet" type="text/css" href="style.css" />
        ..
</head>
```

The <link> element instructs the browser to load the external file specified by the href attribute and to apply the CSS style declarations contained there.

CSS selectors can be grouped into the following categories based on the type of elements they can select.

Basic selectors

- The universal selector
- The type selector
- The class selector
- The id selector
- The child selector
- The descendant selector
- The adjacent sibling selector



Selector	Example	What It Does	
.class	.btn	Selects all elements with class="btn"	Class slector
#id	#myButton	Selects the element with id="myButton"	Id selector
*	*	Selects all elements	Universal selector
element	h1	Selects all <h1> elements	Type selector
element,element	div, p	Selects all <div> elements and all <p> elements	
element element	div p	Selects all <p> elements inside <div> elements	Descendant selector
element>element	div > p	Selects all <p> elements where the parent is a <div> element	Direct child
element+element	div + p	Selects all <p> elements that are placed directly after <div> elements	Adjacent Sibling
element1~element2	p ~ ul	Selects every <ul> element that is preceded by a <p> element	General Sibling

CSS Font

- CSS font properties define the font family, boldness, size, and the style of a text.

	Property Name
1. Font Color	(color)
2. Font Family	(font-family)
3. Font Size	(font-size)
4. Font Style	(font-style)
5. Font Weight	(font-weight)
6. Font Variant	(font-variant)

Background Property

	Property Name
▪ Background Color	(background-color)
▪ Background Image	(background-image)
▪ Background Image Repeat	(background-repeat)
▪ Fixed Background Image	(background-attachment)
▪ Background Image Positioning	(background-position)

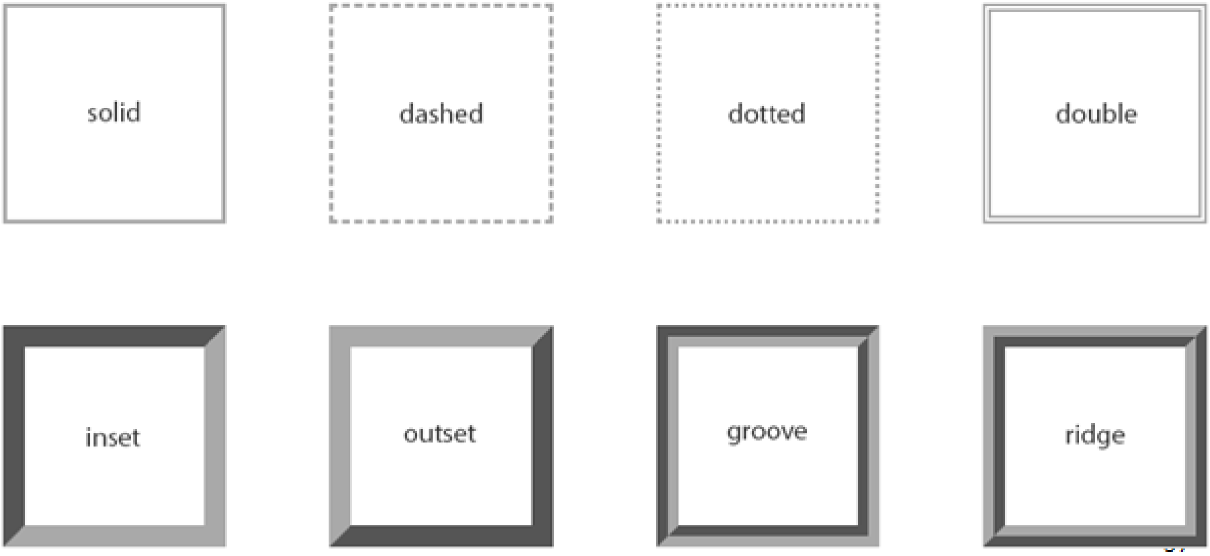
CSS Text Property

- While CSS Font covers most of the traditional ways to format your text, CSS Text allows you to control the spacing, decoration, and alignment of your text.

	Property Name
1. Text Decoration	(text-decoration)
2. Text Indent	(text-indent)
3. Text Align	(text-align)
4. Text Transform	(text-transform)
5. White Space	(white-space)
6. Word Spacing	(word-spacing)
7. Letter Spacing	(letter-spacing)
8. Line Height	(line-height)

Border

- **Border style:** border-top-style, border-right-style, border-bottom-style, border-left-style
- border-style : The border style properties specify the line style of a box's border (solid, double, dashed, etc.).



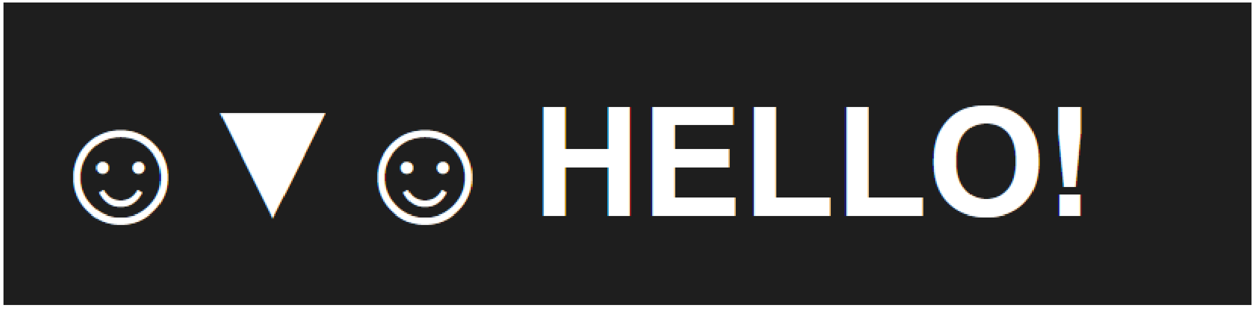
Example

```
<html>
          <head>
               <title>Pseudo Elements in CSS</title>
          <style>
          body {
             background-color: #1e1e1e;
          }

          p {
             font-family: Arial;
             font-weight: bold;
             font-size: 100px;
             text-align: center;
             color: white;
             position: relative;
             top: 40%;
          }
```

```
      p::before {
        content: "☺ ▼ ☺";
        margin-right: 20px;
      }
  </style>
      </head>
        <body>
          <p>HELLO!</p>
        </body>
</html>
```

Output



```
☺ ▼ ☺ HELLO!
```