Module 5.5-[Bonus] Box model,pseudo class, position

5_5-1 Understand inline, block, inline-block elements

Shortcut:

```
Div*3=>   <div></div>
          <div></div>
          <div></div>
```

```
div*3>h2
    <div>
        <h2></h2>
    </div>
    <div>
        <h2></h2>
    </div>
    <div>
        <h2></h2>
    </div>
```

```
    div*3>h2+p
    <div>
        <h2></h2>
        <p></p>
    </div>
    <div>
        <h2></h2>
        <p></p>
    </div>
    <div>
        <h2></h2>
        <p></p>
    </div>
```

```
     div*3>h2{Room-$}+p{Hamid}
    <div>
        <h2>Room-1</h2>
        <p>Hamid</p>
    </div>
    <div>
        <h2>Room-2</h2>
        <p>Hamid</p>
```

```
    </div>
    <div>
        <h2>Room-3</h2>
        <p>Hamid</p>
    </div>
```

```
<!--Inline element-->
    <span>Hamid</span>
    <span>Hosen</span>
    <span>Azad</span>

    <a href="www.google.com">go to google</a>
    <a href="www.google.com">go to google</a>
    <a href="www.google.com">go to google</a>
    <a href="www.google.com">go to google</a>

    <!--Block element-->
    <div>
        <h2>Room-1</h2>
        <p>Hamid</p>
    </div>
    <div>
        <h2>Room-2</h2>
        <p>Hamid</p>
    </div>
    <div>
        <h2>Room-3</h2>
        <p>Hamid</p>
    </div>
```

HTML Block and Inline Element:

Block Level element:

A block-level element always starts on a new line.

A block-level element always takes up the full width available (stretches out to the left and right as far as it can).

A block level element has a top and a bottom margin, whereas an inline element does not.

<address>

<article>
<aside>
<blockquote>
<canvas>
<dd>

<dl>
<dt>
<fieldset>
<figcaption>
<figure>
<footer>
<form>
<h1>-<h6>
<header>
<hr>

<main>
<nav>
<noscript>

<p>
<pre>
<section>
<table>
<tfoot>

<video>

Inline element:

An inline element does not start on a new line.

An inline element only takes up as much width as necessary.

<a>
<abbr>
<acronym>
<b>
<bdo>
<big>

<button>
<cite>
<code>
<dfn>

```
<em>
<i>
<img>
<input>
<kbd>
<label>
<map>
<object>
<output>
<q>
<samp>
<script>
<select>
<small>
<span>
<strong>
<sub>
<sup>
<textarea>
<time>
<tt>
<var>
```

Block line to Inline convert: CSS- > `display: inline;`

Inline to Block line convert : CSS-> `display: block;`

Block Line: height and width work

Inline : height and width not work but (CSS-display:inline-block use) than height and weight work.

5_5-2 CSS box model, div vs span, border image slice

Div vs span:

Span and div are both generic HTML elements that group together related parts of a web page. ... A **div element is used for block-level organization and styling** of page elements, whereas a span element is used for inline organization and styling.

5_5-3 Pseudo class hover, class hover, visited, focus

Hover:

```
h1:hover
{
    color: royalblue;
    background-color: skyblue;
    margin: 50px;
    padding: 50px;
    font-size: 50px;
    font-weight: bold;
}
```

Focus:

```
.middle:focus
{
    color: springgreen;
    font-size: 20px;
    border-radius: 50px;
    border-color: red;
}
```

Visited:

```
a:visited
{
    color: royalblue;
}
```

What are Pseudo-Elements?

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

For example, it can be used to:

- Style the first letter, or line, of an element
- Insert content before, or after, the content of an element

A CSS pseudo-element is **a keyword added to a selector that lets you style a specific part of the selected element(s)**. For example, ::first-line can be used to change the font of the first line of a paragraph.

5_5-4 First child, nth child, pseudo element before after

First child nth child:

```
        li:first-child
        {
            color: red;
        }
        li:last-child
        {
            background-color: salmon;
        }
        li:nth-child(2n)
        {
            color: springgreen;
        }
        li:nth-child(2n+1)
        {
            background-color: green;
        }

    <ul>
        <li>Hamid</li>
        <li>Hosen</li>
        <li>Azad</li>
        <li>Fahim</li>
        <li>Riad</li>
        <li>Moinul</li>
        <li>Moinul</li>
        <li>Moinul</li>
        <li>Moinul</li>
        <li>Moinul</li>
    </ul>
```

Pseudo element before after:

```
        h3::before
```

```
        {
            content: "Hamid Hosen";
        }
        h3::after
        {
            content: 'Hosen';
            color: green;
        }
        .spacial::after
        {
            content: "Azad";
            color: greenyellow;
        }

    <h3>middle -1</h3>
    <h3 class="spacial">middle -2</h3>
    <h3>middle -3</h3>
    <h3 class="spacial">middle -4</h3>
    <h3>middle -5</h3>
    <h3>middle -6</h3>
```

5_5-5 Position static relative absolute fixed sticky z-index

Link= https://www.w3schools.com/css/css_positioning.asp

The position Property

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

There are five different position values:

- `static`
- `relative`
- `fixed`
- `absolute`
- `sticky`

```
    <style>
      .swimming {
```

```
      position: relative;
    }
    .swimming div {
      border: 1px dashed red;
      width: 200px;
    }
    #boy
    {
        position: relative;
        top: 75px;
        z-index: 100;
    }
    #ring {
      position: relative;
      z-index: 1;
    }
    /*
    #ring {
      position: sticky;
      left: 0px;
      top: 100px;
    }
    /*
      #ring {
      position: fixed;
      left: 0px;
      top: 100px;
    }
      #ring {
      position: absolute;
      left: 0px;
      top: 100px;
    }
    */
      #water {
      position:relative;
      top: -75px;
    }

    .dummy {
      height: 100px;
      width: 100px;
      background-color: yellow;
    }
  </style>
```