

## LAB 03 – BASIC CSS (cont.)

### 1/. CSS Z-index

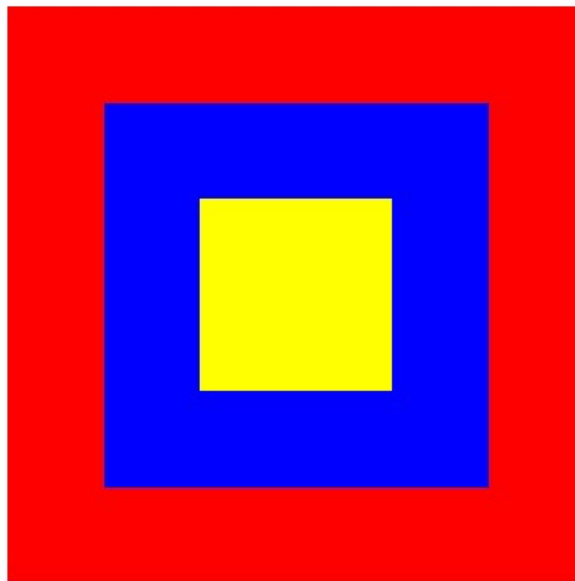
The z-index property specifies the stack order of an element. An element with greater stack order is always in front of an element with a lower stack order.

z-index only works on positioned elements (position: absolute, position: relative, position: fixed, or position: sticky) and flex items.

z-index: auto   [ <i>number</i> ]   initial   inherit;	
auto	Sets the stack order equal to its parents. This is default
number	Sets the stack order of the element. Negative numbers are allowed
initial	Sets this property to its default value.
inherit	Inherits this property from its parent element.

Example:

I want to design like this. How can you do that?



You can try:

```
<style>
  #div01 {
    height: 200px;
    width: 200px;
    background-color:  blue;
    position: absolute;
    z-index: 2;
    margin: 50px;
  }

  #div02 {
    height: 300px;
    width: 300px;
    background-color:  red;
    position: absolute;
    z-index: 1;
  }

  #div03 {
    height: 100px;
    width: 100px;
    background-color:  yellow;
    position: absolute;
    z-index: 3;
    margin: 100px;
  }
</style>
```

## 2/. CSS Pseudo-class, CSS Pseudo-element a/.

### CSS Pseudo-class:

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

The syntax of pseudo-classes:

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

Some common pseudo-classes:

Selector	Description
:first-child	Selects every elements that is the first child of its parent
:first-of-type	Selects every element that is the first element of its parent
:focus	Selects the element that has focus
:hover	Selects elements on mouse over
:link	Selects all unvisited elements
:active	Selects the active elements
:visited	Selects all visited elements
:last-child	Selects every elements that is the last child of its parent
:last-of-type	Selects every element that is the last element of its parent
:nth-child(n)	Selects every element that is the n child of its parent
.....	.....

You can see more at: [https://www.w3schools.com/css/css\\_pseudo\\_classes.asp](https://www.w3schools.com/css/css_pseudo_classes.asp) b/.

### CSS Pseudo-element:

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

The syntax of pseudo-elements:

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

All CSS pseudo-elements:

Selector	Description
::after	Insert something after the content of each element

::before	Insert something before the content of each element
::first-letter	Selects the first letter of each element
::first-line	Selects the first line of each element
::marker	Selects the markers of list items
::selection	Selects the portion of an element that is selected by a user

You can see more at: [https://www.w3schools.com/css/css\\_pseudo\\_elements.asp](https://www.w3schools.com/css/css_pseudo_elements.asp)

Example:

I want to design like this. How can you do that?

This is the first paragraph!

**T**his is the second paragraph!

THIS IS THE THIRD PARAGRAPH!

And after design, I want to do another thing like this:

When I mouse over the first paragraph, it can have borders around with solid style, color orange, width is 5px, background color is cyan, rounded border, padding, width or height and align center.

You can try:

```
<div>
  <p>
    This is the first paragraph!
  </p>
  <p>
    This is the second paragraph!
  </p>
  <p>
    This is the third paragraph!
  </p>
</div>
```

```

<style>
  p {
    font-size: larger;
  }
  p:nth-child(2)::first-letter {
    color: red;
    font-size: xx-large;
    font-weight: bold;
  }
  p:nth-child(2)::first-line {
    color: green;
    font-weight: bold;
  }
  p:first-child {
    color: blue;
    font-weight: bold;
    text-decoration: underline;
  }
  p:last-of-type {
    color: blueviolet;
    background-color: antiquewhite;
    font-variant: small-caps;
  }
  p:first-of-type:hover {
    border: 5px solid orange;
    background-color: aqua;
    border-radius: 5px;
    width: 400px;
    padding: 10px;
    text-align: center;
  }
</style>

```

### 3/. CSS Link

Links can be styled with any CSS property (e.g. color, font-family, background, etc.).

In addition, links can be styled differently depending on what state they are in.

The four links states are:

- `a:link` - a normal, unvisited link
- `a:visited` - a link the user has visited
- `a:hover` - a link when the user mouses over it
- `a:active` - a link the moment it is clicked

The `text-decoration` property is mostly used to remove underlines from links.

Example:

I want to design like this. How can you do that?



You can try:

```
<body>
  <div>
    <a id="link01" href="https://www.google.com">Browse to Google</a>
    <a id="link02" href="https://www.amazon.com">Browse to Amazon</a>
  </div>
</body>
```

```
<style>
  div {
    width: 200px;
    border: 5px double blueviolet;
    padding: 10px;
  }
  #link01 {
    background-color: palegoldenrod;
    color: black;
    border: 5px solid blue;
    padding: 10px 20px;
    text-align: center;
    text-decoration: none;
    display: block;
    border-radius: 10px;
    margin: 10px;
  }
  #link02 {
    background-color: palegoldenrod;
    color: green;
    border-top: 5px dashed red;
    border-bottom: 5px dashed red;
    padding: 10px 20px;
    text-align: center;
    text-decoration: none;
    display: block;
    margin: 10px;
  }
</style>
```



#### 4/. CSS List

The CSS list properties allow you to:

- Set different list item markers for ordered lists
- Set different list item markers for unordered lists
- Set an image as the list item marker
- Add background colors to lists and list items Example:

I want to design like this. How can you do that?



You can try:

```
<body>
  <div>
    <ol>
      <li>Item 01</li>
      <li>Item 02</li>
      <li>Item 03</li>
    </ol>
    <ul>
      <li>Item random 01</li>
      <li>Item random 02</li>
      <li>Item random 03</li>
    </ul>
  </div>
</body>
```



```

<style>
  div {
    width: 350px;
    border: 5px dashed red;
    padding: 10px;
    background-color: aquamarine;
  }
  ol {
    background-color: palegoldenrod;
    color: black;
    display: inline-block;
    margin: 10px;
    padding: 20px;
    border: 3px solid orange;
  }
  ul {
    list-style-type: circle;
    background-color: palegoldenrod;
    display: inline-block;
    margin: 10px;
    padding: 20px;
    border: 5px double hotpink;
  }
  li {
    padding: 5px;
  }
  ul li {
    color: red;
    font-weight: bold;
    font-style: italic;
  }
  ol li {
    color: blue;
    font-weight: bold;
    font-size: large;
  }
</style>

```

## 5/. CSS Table

- Table Borders: To specify table borders in CSS, use the border property.
- Table Size: The width and height of a table are defined by the width and height properties.
- Table Alignment:
  - + The text-align property sets the horizontal alignment (like left, right, or center) of the content in `<th>` or `<td>`
  - + The vertical-align property sets the vertical alignment (like top, bottom, or middle) of the content in `<th>` or `<td>`
- Table Styles:
  - + To control the space between the border and the content in a table, use the padding property on `<td>` and `<th>` elements.
  - + Add the border-bottom property to `<th>` and `<td>` for horizontal dividers.
  - + Use the `:hover` selector on `<tr>` to highlight table rows on mouse over.
  - + Use the `nth-child()` selector and add a style to all even (or odd) table rows.
- Table Responsive: A responsive table will display a horizontal scroll bar if the screen is too small to display the full content.

Add a container element (like `<div>`) with `overflow-x:auto` around the `<table>` element to make it responsive.

Example:

I want to design like this. How can you do that?

<i>Item 01</i>	<i>Item 02</i>	<i>Item 03</i>
<b>Item 04</b>	<b>Item 05</b>	<b>Item 06</b>
<i>Item 07</i>	<i>Item 08</i>	<i>Item 09</i>
<b>Item 10</b>		<b>Item 11</b>

You can try:

```
<div>
  <table>
    <tr>
      <td>Item 01</td>
      <td>Item 02</td>
      <td>Item 03</td>
    </tr>
    <tr>
      <td>Item 04</td>
      <td>Item 05</td>
      <td>Item 06</td>
    </tr>
    <tr>
      <td>Item 07</td>
      <td>Item 08</td>
      <td>Item 09</td>
    </tr>
    <tr>
      <td colspan="2">Item 10</td>
      <td>Item 11</td>
    </tr>
  </table>
</div>
```

```

<style>
  div {
    width: 500px;
    height: 300px;
    border: 3px solid brown;
    background-color: aquamarine;
  }
  table {
    width: 400px;
    height: 200px;
    text-align: center;
    vertical-align: center;
    margin: 50px;
  }
  td {
    border-bottom: 5px solid blue;
    border-left: 5px solid green;
    border-right: 5px solid green;
  }
  tr:nth-child(even) {
    background-color: orange;
    font-weight: bold;
    font-size: large;
  }
  tr:nth-child(odd) {
    background-color: hotpink;
    font-style: italic;
    font-size: large;
  }
</style>

```

## 6/. CSS Form

- Styling Input Fields: Use the width property to determine the width of the input field.
- Padded Inputs: Use the padding property to add space inside the text field.

- Bordered Inputs: Use the border property to change the border size and color, and use the border-radius property to add rounded corners.
- Colored Inputs: Use the background-color property to add a background color to the input, and the color property to change the text color.
- Focused Inputs: By default, some browsers will add a blue outline around the input when it gets focus (clicked on). You can remove this behavior by adding outline: none; to the input. Use the :focus selector to do something with the input field when it gets focus.
- Input with icon/image: If you want an icon inside the input, use the backgroundimage property and position it with the background-position property.
- .....etc.....

Example:

I want to design like this. How can you do that?

**User Name:**

**Password:**

**You are:** ☐ **Guest** ☐ **Member**

You can try:

```
<div>
  <form>
    <table>
      <tr>
        <td align="right"><label>User Name: </label></td>
        <td>
          <input type="text" placeholder="Please input username" size="50">
        </td>
      </tr>
      <tr>
        <td align="right"><label>Password: </label></td>
        <td>
          <input type="password" placeholder="Please input password" size="50">
        </td>
      </tr>
      <tr>
        <td align="right"><label>You are: </label></td>
        <td>
          <input type="radio" name="role"><label class="role">Guest</label>
          <input type="radio" name="role"><label class="role">Member</label>
        </td>
      </tr>
      <tr>
        <td colspan="2" align="center">
          <input type="button" value="Log in">
          <input type="reset" value="Reset">
        </td>
      </tr>
    </table>
  </form>
</div>
```



```

<style>
  div {
    width: 500px;
    height: 250px;
    border: 5px dashed orange;
    background-color: lemonchiffon;
    border-radius: 10px;
  }
  input[type="text"], input[type="password"]{
    padding: 10px;
    margin: 10px;
    color: blueviolet;
    background-color: blanchedalmond;
    border: 3px solid green;
    border-radius: 10px;
  }
  input[type="radio"]{
    padding: 10px;
    margin: 10px;
  }

```

```

  input[type="button"], input[type="reset"]{
    padding: 10px;
    margin: 10px;
    width: 80px;
    background-color: aquamarine;
    border: 3px solid salmon;
    border-radius: 10px;
    color: brown;
    font-weight: bold;
    font-size: large;
  }
  label {
    color: red;
    font-weight: bold;
  }
  .role {
    color: blue;
    font-weight: bold;
  }
</style>

```



## 7/. Priority in CSS

Every CSS selector has its place in the specificity hierarchy.

There are four categories which define the specificity level of a selector:

- Inline styles - Example: `<h1 style="color: pink;">`
- IDs - Example: `#navbar`
- Classes, pseudo-classes, attribute selectors - Example: `.test`, `:hover`, `[href]`
- Elements and pseudo-elements - Example: `h1`, `:before`

Start at 0, add 100 for each ID value, add 10 for each class value (or pseudo-class or attribute selector), add 1 for each element selector or pseudo-element.

### **Note:**

Inline style gets a specificity value of 1000, and is always given the highest priority.

There is one exception to this rule: if you use the `!important` rule, it will even override inline styles.