Quadrant II – Transcript and Related Materials

Programme: B.Sc.

Subject: Computer Science

Course Code: CSS 104/105/106

Course Title: Web Application Development Using FLASK/ASP.NET/DJANGO

Unit: III (CSS - Cascading Style Sheets)

Module Name: Pseudo-classes, Pseudo-elements, Cascade and Inheritance, Box model

Module No: 11

Name of the Presenter: Mrs. Liana da Costa

Notes

Pseudo-classes

- Pseudo-classes are another form of specification used in CSS.
- A pseudo-class is used to define a special state of an element.
- It can be combined with a CSS selector to add an effect to existing elements based on their states.
- They identify markup elements, and in some cases, specific user actions, to which a particular style is to be applied.

For example, it can be used to:

- Style an element when a user hovers over it
- Style visited and unvisited links differently
- Style an element when it gets focus

Pseudo-Classes (Syntax)

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

Note:

1. Pseudo-Class names are not case-sensitive.
2. Link Pseudo-Classes (In Order)

```
a:link {color: blue;}

a:visited {color: purple;}

a:hover {color: red;}

a:active {color: yellow;}
```

Anchor Pseudo-classes

Examples:

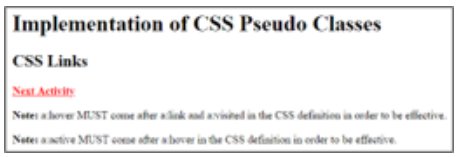
```
/* unvisited link */
a:link {
  color: #FF0000;
}

/* visited link */
a:visited {
  color: #00FF00;
}

/* mouse over link */
a:hover {
  color: #FF00FF;
}

/* selected link */
a:active {
  color: #0000FF;
}
```

Note:

The **:hover pseudo-class** – This applies a style only when the user 'points to' the visible element with a mouse pointer – It is appended to a selector as in **a:hover** or **#elementid:hover**

Anchor Pseudo-classes

```
<html>
<head>
<style>
/* unvisited link */
a:link {
  color: green;
}

/* visited link */
a:visited {
  color: red;
}

/* mouse over link */
a:hover {
  color: orange;
}

/* selected link */
a:active {
  color: blue;
}
</style>
</head>
<body>
<h1>Implementation of Pseudo-Classes</h1>
<h2>CSS Links</h2>
<p><b><a href="default.asp" target="_blank">This is a link</a></b></p>
<p><b>Note:</b> a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.</p>
<p><b>Note:</b> a:active MUST come after a:hover in the CSS definition in order to be effective.</p>

</body>
</html>
```



CSS - The :first-child Pseudo-class

The :first-child pseudo-class matches a specified element that is the first child of another element.

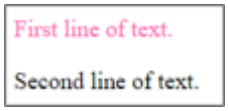
Example

To Match the first <p> element(The selector matches any <p> element that is the first child of any element)

Source File

```
<html>
<head>
<style>
p:first-child {
  color: hotpink;
}
</style>
</head>
<body>

<p>First line of text.</p>
<p>Second line of text.</p>

</body>
</html>
```

Output



CSS - The :lang Pseudo-class

The :lang pseudo-class allows you to define special rules for different languages.

Example: :lang defines the quotation marks for <q> elements with lang="quo"

Example

Source File

```
<html>
<head>
<style>
q:lang(quo) {
  quotes: "~" "~";
}
</style>
</head>
<body>

<p>Text1 Text2 Text3 <q lang="quo">Quotes will be inserted in the
paragraph</q> Text4 Text5 Text6</p>
<p><b>In this example, :lang defines the quotation marks for q elements
with lang="quo":</b></p>

</body>
</html>
```

Output

Text1 Text2 Text3~Quotes will be inserted in this paragraph~Text4 Text5 Text6.

In this example, :lang defines the quotation marks for q elements with lang="quo":

CSS - Pseudo-class

Example: Demonstration of how to add other styles to hyperlinks

Source File

```
<html>
<head>
<style>
a.one:link {color:blue;}
a.one:visited {color:red;}
a.one:hover {color:yellow;}

a.two:link {color:orange;}
a.two:visited {color:blue;}
a.two:hover {font-size:150%;}

a.three:link {color:pink;}
a.three:visited {color:green;}
a.three:hover {background:purple;}

a.four:link {color:yellow;}
a.four:visited {color:purple;}
a.four:hover {font-family:arial;}

a.five:link {color:blue;text-decoration:none;}
a.five:visited {color:cyan;text-decoration:none;}
a.five:hover {text-decoration:underline;}
</style>
</head>
<body>

<p><b>Move the Mouse over the links and watch them change the appearance</b></p>

<p><b><a class="one" href="default.asp" target="_blank">This link changes color</a></b></p>
<p><b><a class="two" href="default.asp" target="_blank">This link changes font-size</a></b></p>
<p><b><a class="three" href="default.asp" target="_blank">This link changes background-color</a></b>
</p>
<p><b><a class="four" href="default.asp" target="_blank">This link changes font-family</a></b></p>
<p><b><a class="five" href="default.asp" target="_blank">This link changes text-decoration</a></b>
</p>

</body>
</html>
```

Output

Move the Mouse over the links and watch them change the appearance

This link changes color

This link changes font-size

This link changes background-color

This link changes font-family

This link changes text-decoration

CSS - 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
- Select the markers of list items

CSS - Pseudo-Elements (Syntax)

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

Examples:

Selector	Example	Example description
::after	p::after	Insert something after the content of each <p> element
::before	p::before	Insert something before the content of each <p> element
::first-letter	p::first-letter	Selects the first letter of each <p> element
::first-line	p::first-line	Selects the first line of each <p> element
::marker	::marker	Selects the markers of list items
::selection	p::selection	Selects the portion of an element that is selected by a user

The ::first-line Pseudo-element

The ::first-line pseudo-element is used to add a special style to the first line of a text.

Note:

The ::first-line pseudo-element can only be applied to block-level elements.

The following properties apply to the ::first-line pseudo-element

- font properties
- color properties
- background properties
- word-spacing
- letter-spacing
- text-decoration
- vertical-align
- text-transform
- line-height
- clear

Example: To format the first line of the text in all <p> elements

Source File

```
<html>
<head>
<style>
p::first-line {
  color: cyan;
  font-variant: small-caps;
  font-size: 40px;
  font-family: Helvetica, Arial, sans-serif;
  line-height: 1.6;
  letter-spacing: 1px;
}
P{
  color: blue;
  font-size: 20px;
  font-family: Helvetica, Arial, sans-serif;
}
</style>
</head>

<body>
<p>You can use the ::first-line pseudo-element to add a special effect to the first line of a text. Some more text and even more text, and more text, and more text, and more text, and more text, and more text, and more text, and more text, and more text, and so on.</p>
</body>
</html>
```

Output

The ::first-letter Pseudo-element

The ::first-letter pseudo-element is used to add a special style to the first letter of a text.

Note:

The ::first-letter pseudo-element can only be applied to block-level elements.

The following properties apply to the ::first-letter pseudo-element

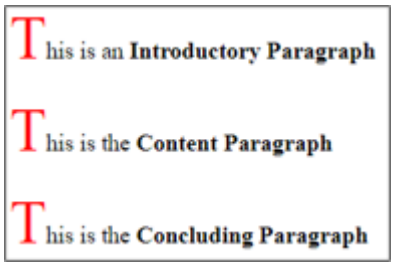
- font properties
- color properties
- background properties
- margin properties
- padding properties
- border properties
- text-decoration
- vertical-align (only if "float" is "none")
- text-transform
- line-height
- float
- clear

Example: To display the first letter of paragraphs in red and in a **larger size.**

Source File

```
<html>
<head>
<style>
p::first-letter {
  color: red;
  font-size: 250%;
}
</style>
</head>
<body>

<p>This is an <b> Introductory Paragraph </b></p>
<p>This is the <b> Content Paragraph </b></p>
<p>This is the <b> Concluding Paragraph </b></p>


</body>
</html>
```

Output



The ::before Pseudo-element

The ::before pseudo-element is used to insert some content before the content of an element.

Example: To insert **some text** before the content of each <h1> element.

Source File

```
<html>
<head>
<style>
h1::before {
  content: "Text inserted BEFORE ";
}
</style>
</head>
<body>

<h1><u>Heading One</u></h1>
<p>The ::before pseudo-element inserts content before the content of an element.</p>
<h1><u>Heading Two</u></h1>

</body>
</html>
```

Output

The ::after Pseudo-element

The ::after pseudo-element is used to insert some content after the content of an element.

Example: To insert **an image** after the content of each <h1> element

Source File

```
<html>
<head>
<style>
h1::after {
  content: url(smiley.gif);
}
</style>
</head>
<body>

<h1>This is Heading One</h1>
<p>The ::after pseudo-element inserts content
after the content of an element.</p>

<h1>This is Heading Two</h1>

</body>
</html>
```

Output

The ::marker Pseudo-element

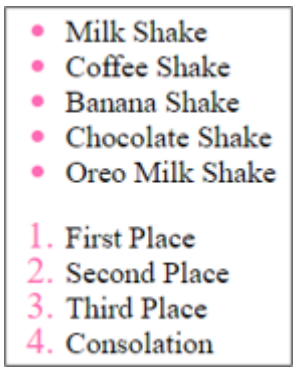
The ::marker pseudo-element selects the markers of list items.

Example: To style the markers of list items

Source File

```
<html>
<head>
<style>
::marker {
  color: hotpink;
  font-size: 20px;
}
</style>
</head>
<body>
<ul>
  <li>Milk Shake</li>
  <li>Coffee Shake</li>
  <li>Banana Shake</li>
  <li>Chocolate Shake</li>
  <li>Oreo Milk Shake</li>
</ul>

<ol>
  <li>First Place</li>
  <li>Second Place</li>
  <li>Third Place</li>
  <li>Consolation</li>
</ol>
</body>
</html>
```

Output



The ::selection Pseudo-element

The ::selection pseudo-element matches the portion of an element that is selected by a user.

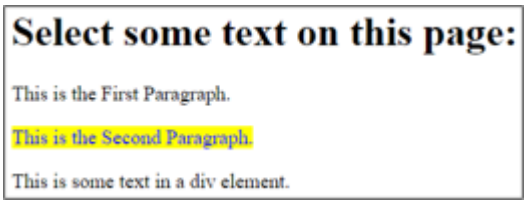
The following CSS properties can be applied to ::selection: color, background, cursor, and outline.

Example: To make the selected text blue on a yellow background

Source File

```
<html>
<head>
<style>
::selection {
  color: blue;
  background: yellow;
}
</style>
</head>
<body>
<h1>Select some text on this page:</h1>
<p>This is the First Paragraph.</p>
<p>This is the Second Paragraph.</p>
<div>This is some text in a div element.
</div>
</body>
</html>
```

Output



Cascade

CSS stands for **Cascading Style Sheets**,

- *Cascading* - the way that the cascade behaves is key to understanding CSS.
- Style sheets **cascade**
    - the order of CSS rules matter
    - when two rules apply that have equal specificity the one that comes last in the CSS is the one that will be used.

Example: We have two rules that could apply to the h1.

Source File

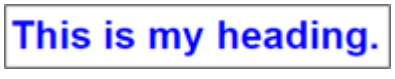
CSS Code

```
h1 {
   color: red;
}
h1 {
   color: blue;
}
```

HTML Code

```
<h1>This is my heading.</h1>
```

Output



Understanding the Cascade

1) Source order

- If you have more than one rule, which have the same weight, then the one that comes last in the CSS will win.

2) Specificity

- a class selector has more weight than an element selector, so the properties defined on the class will override those applied directly to the element.
- define generic styles for the basic elements, and then create classes for those which are different.
- IDs have an even higher specificity than classes (you can only have one element with each unique ID on a page, but many elements with the same class — ID selectors are very specific in what they target)

Source File

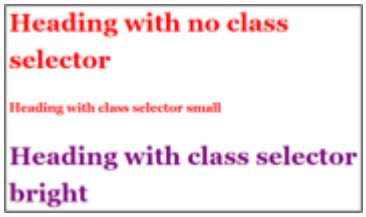
CSS Code

```css
h2 {
    font-size: 2em;
    color: red;
    font-family: Georgia, 'Times New Roman', Times, serif;
}

.small {
    font-size: 1em;
    color green;
}

.bright {
    color: purple;
}
```

HTML Code

```html
<h2>Heading with no class selector</h2>
<h2 class="small">Heading with class selector small</h2>
<h2 class="bright">Heading with class selector bright</h2>
```

Output



3) Importance: !important is used to make a particular property and value the most specific thing, thus overriding the normal rules of the cascade.

Source File

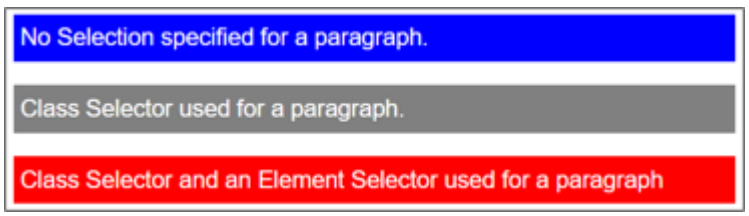
CSS Code

```
#winning {
    background-color: red;
    border: 1px solid black;
}

.better {
    background-color: gray;
    border: none !important;
}

p {
    background-color: blue;
    color: white;
    padding: 5px;
}
```

HTML Code

```
<p>No Selection specified for a paragraph.</p>
<p class="better ">Class Selector used for a paragraph.</p>
<p class="better" id="winning">Class Selector and an Element Selector used for a paragraph</p>
```

Output



Note: The only way to override this **`!important`** declaration would be to include another `!important` declaration on a declaration with the same specificity later in the source order, or one with higher specificity.

Inheritance

Inheritance: means that some CSS properties by default inherit values set on the current element's parent element, and some don't.

Example: if you set a color and font-family on an element, every element inside it will also be styled with that color and font, unless you've applied different color and font values directly to them.

Source File

CSS Code

```
body {
   color: blue;
}
span {
   color: black;
}
```

HTML Code

```
<p>As the body has been set to have a color of blue this is
inherited through the descendants.</p>
<p>We can change the color by targetting the element with a
selector, such as this <span>span</span>.</p>
```

Output

As the body has been set to have a color of blue this is inherited through the descendants.

We can change the color by targetting the element with a selector, such as this span.

Source File

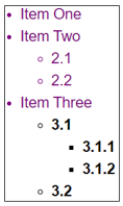
CSS Code

```
.main {
   color: purple;
   border: 2px solid #ccc;
   padding: 1em;
}
.special {
   color: black;
   font-weight: bold;
}
```

HTML Code

```
<ul class="main">
   <li>Item One</li>
   <li>Item Two
      <ul>
         <li>2.1</li>
         <li>2.2</li>
      </ul>
   </li>
   <li>Item Three
      <ul class="special">
         <li>3.1
            <ul>
               <li>3.1.1</li>
               <li>3.1.2</li>
            </ul>
         </li>
         <li>3.2</li>
      </ul>
   </li>
</ul>
```

Output



Controlling Inheritance

CSS provides four special universal property values for controlling inheritance. Every CSS property accepts these values.

inherit

Sets the property value applied to a selected element to be the same as that of its parent element. Effectively, this **"turns on inheritance"**.

initial

Sets the property value applied to a selected element to the initial value of that property.

unset

Resets the property to its natural value, which means that if the property is naturally inherited it acts like inherit, otherwise it acts like initial.

Example

Source File

CSS Code

```
.body {
   color: green;
}

.my-class-1 a {
   color: inherit;
}

.my-class-2 a {
   color: initial;
}

.my-class-3 a {
   color: unset;
}
```

HTML Code

```
<ul>
    <li>Default <a href="#">Default link</a> color</li>
    <li class="my-class-1">Inherit the <a href="#">Inherit link</a>
color</li>
    <li class="my-class-2">Reset the <a href="#">Reset link</a> color</li>
    <li class="my-class-3">Unset the <a href="#">Unset link</a> color</li>
</ul>
```

Output

- Default Default link color
- Inherit the Inherit link color
- Reset the Reset link color
- Unset the Unset link color

Resetting all property values

- CSS shorthand property **all** can be used to apply one of these inheritance values to (almost) all properties at once.
- Its value can be any one of the inheritance values (inherit, initial, unset or revert).
- convenient way to undo changes made to styles so that you can get back to a known starting point before beginning new changes.

Source File

CSS Code

```
blockquote {
    background-color: yellow;
    border: 4px solid blue;
    color: red;
}

.fix-this {
    all: unset;
}
```

HTML Code

```
<blockquote>
    <p>This blockquote is styled</p>
</blockquote>

<blockquote class="fix-this">
    <p>This blockquote is not styled</p>
</blockquote>
```

Output

CSS Box Model

- The CSS box model is essentially a box that wraps around every HTML element.
- Used when talking about design and layout.
- It consists of margins, borders, padding, and the actual content.
- The box model allows us to add a border around elements, and to define space between elements.

- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

Demonstration of the Box Model

Source File

```
<html>
<head>
<style>
div {
  background-color: lightgrey;
  width: 300px;
  border: 25px solid blue;
  padding: 80px;
  margin: 40px;
}
</style>
</head>
<body>
<h2>Demonstrating the Box Model</h2>
<p>The CSS box model is essentially a box that wraps around every HTML element. It consists of:
borders, padding, margins, and the actual content.</p>

<div>This text is the content of the box. We have added a 80px padding, 40px margin and a 25px blue
border.</div>

</body>
</html>
```

Output