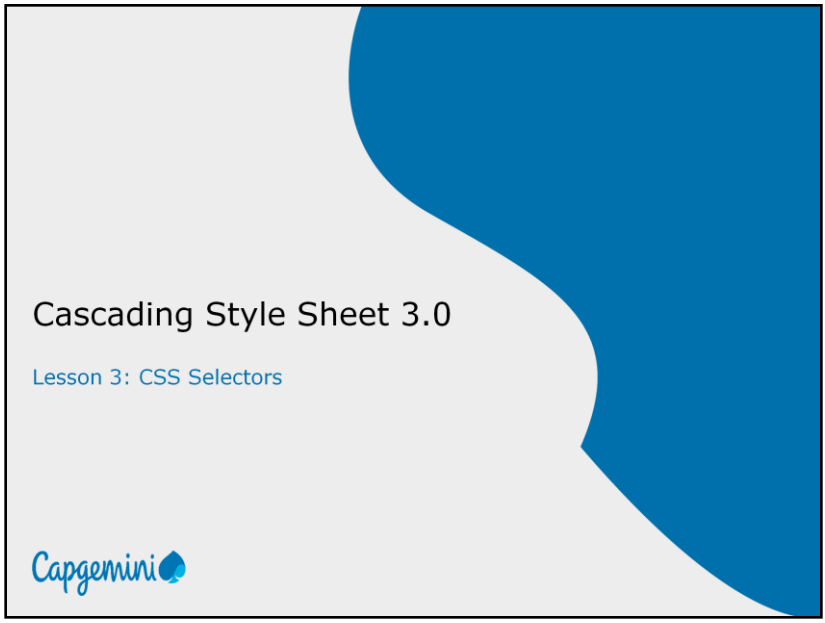

Cascading Style Sheet 3.0
Lesson 3: CSS Selectors
Capgemini

Lesson Objectives

In this lesson, you will be learning about:

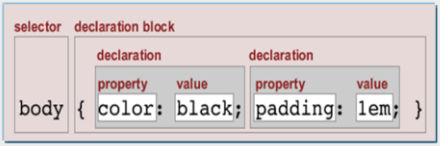
- Introduction to Selectors
- Universal Selector
- Type Selector
- Class Selector
- ID Selector
- Attribute Selector
- Pseudo Classes

3.1 Introduction to Selectors

Selectors

Introduction:

- Selectors are one of the most important aspects of CSS as they are used to "select" elements on an HTML page so that they can be styled.
- The selector "selects" the elements on an HTML page that are affected by the rule set.
- A rule or "rule set" is a statement that tells browsers how to render particular elements on an HTML page
- A rule set consists of a selector followed by a declaration block.
- Rule structure



Text Color :The color property is used to set the color of the text.

Text Alignment:The text-align property is used to set the horizontal alignment of a text.

Text can be centered, or aligned to the left or right, or justified.

Text Decoration:The text-decoration property is used to set or remove decorations from text.

The text-decoration property is mostly used to remove underlines from links for design purposes:

Text Transformation:The text-transform property is used to specify uppercase and lowercase letters in a text.

It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word.

Text Indentation:The text-indentation property is used to specify the indentation of the first line of a text.

Text Shadow: In CSS3, the text-shadow property applies shadow to text

Word Wrapping:In CSS3, the word-wrap property allows you to force the text to wrap - even if it means splitting it in the middle of a word

3.1 Introduction to Selectors

Selectors

Example

- h1 { color: blue; margin-top: 1em; }
- p { padding: 5px; }
- td { background-color: #ddd; }

3.2 Universal Selector

Universal Selector

The universal selector matches any element type.
Example:

This rule set will be applied to every element in a document:

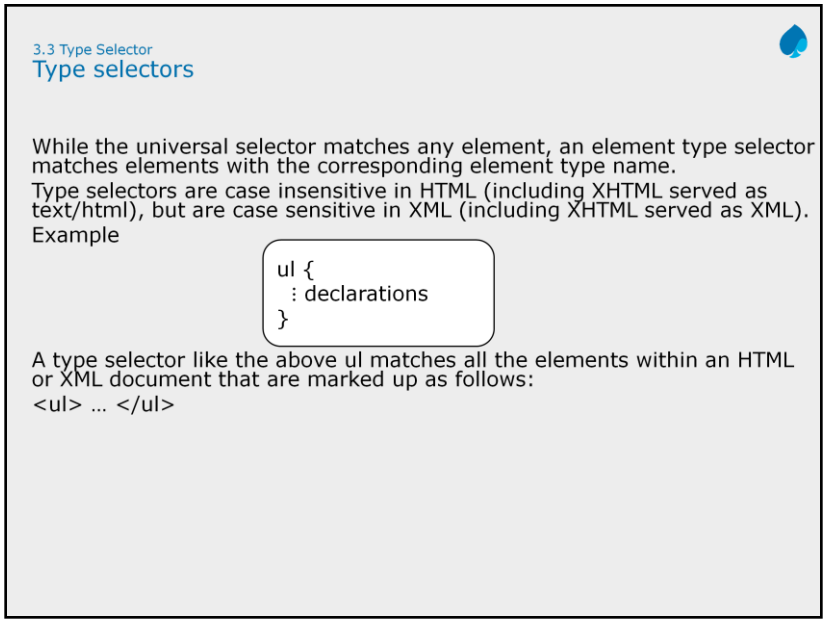
```
* {
    margin  : 0;
    padding: 0;
  }
```

It's important not to confuse the universal selector with a wildcard character—the universal selector doesn't match "zero or more elements." Consider the following HTML fragment:

```
<body>
  <div>
    <h1>The <em>Universal</em> Selector</h1>
    <p>We must <em>emphasize</em> the following:</p>
    <ul>
      <li>It's <em>not</em> a wildcard.</li>
      <li>It matches elements regardless of <em>type</em>.</li>
    </ul>
    This is an <em>immediate</em> child of the division.
  </div>
</body>
```

The selector div * em will match the following em elements:

"Universal" in the h1 element (* matches the <h1>)
"emphasize" in the p element (* matches the <p>)
"not" in the first li element (* matches the <ul> or the <li>)
"type" in the second li element (* matches the <ul> or the <li>)
However, it won't match the immediate element, since that's an immediate child of the div element—there's nothing between and for the * to match.

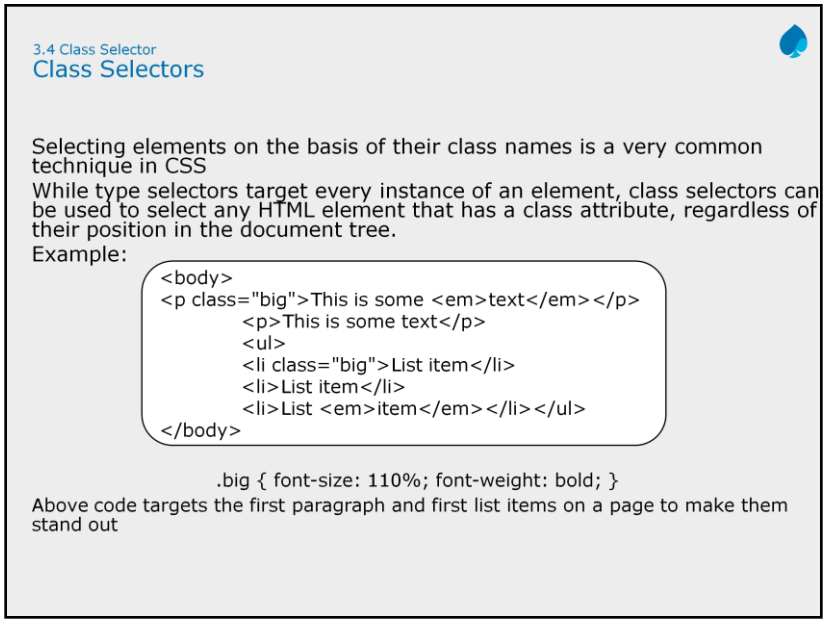


The most common and easy to understand selectors are type selectors. Type selectors will select any HTML element on a page that matches the selector, regardless of their position in the document tree. For example:

em {color: blue; }

This rule will select any element on the page and color it blue. As you can see from the document tree diagram below, all elements will be colored blue, regardless of their position in the document tree

There are a huge range of elements that you can select using type selectors, which means you can change the appearance of any or every element on your page using only type selectors.



Combining class and type selectors:

If you want to be more specific, you can use class and type selectors together. Any type selectors can be used.

```
div.big { color: blue; }
td.big { color: yellow; }
label.big { color: green; }
form.big { color: red; }
```

3.5 ID Selector

ID Selector

An ID selector matches an element that has a specific id attribute value. Since id attributes must have unique values, an ID selector can never match more than one element in a document.
In its simplest form, an ID selector looks like this:

```
#navigation
{
   : declarations
}
```

This selector matches any element whose id attribute value is equal to "navigation"

```
#firstname
{
background-color:yellow;
}
```

Code:

```
<!DOCTYPE html>
<html>
<head>
<style>
#firstname
{
                    background-color:yellow;
}
</style>
</head>
<body>

<h1>Welcome to My Homepage</h1>

<div class="intro">
<p id="firstname">My name is iGATE.</p>
<p id="hometown">I live in Bangalore.</p>
</div>

<p>My best friend was Patni.</p>

</body>
</html>
```

3.6 Attribute Selector

Attribute Selector

All HTML elements can have associated properties, called attributes. These attributes generally have values. Any number of attribute/value pairs can be used in an element's tag - as long as they are separated by spaces. They may appear in any order.

In the example below, the code segments highlighted in blue are attributes and the segments highlighted in red are attribute values

```
<h1 id="section1"/>
<img src="small.gif" width="100" height="10
0"/>
<img title="mainimage" alt="main image"/>
<a href="foo.htm"/>
<p class="maintext"/>
<form style="padding: 10px"/>
```

3.6 Attribute Selector

Attribute Selector

Attribute selectors are used to select elements based on their attributes or attribute value. For example, you may want to select any image on an HTML page that is called "small.gif". This could be done with the rule below, that will only target images with the chosen name:

There are four types of attribute selectors.

- Example for Select based on attribute

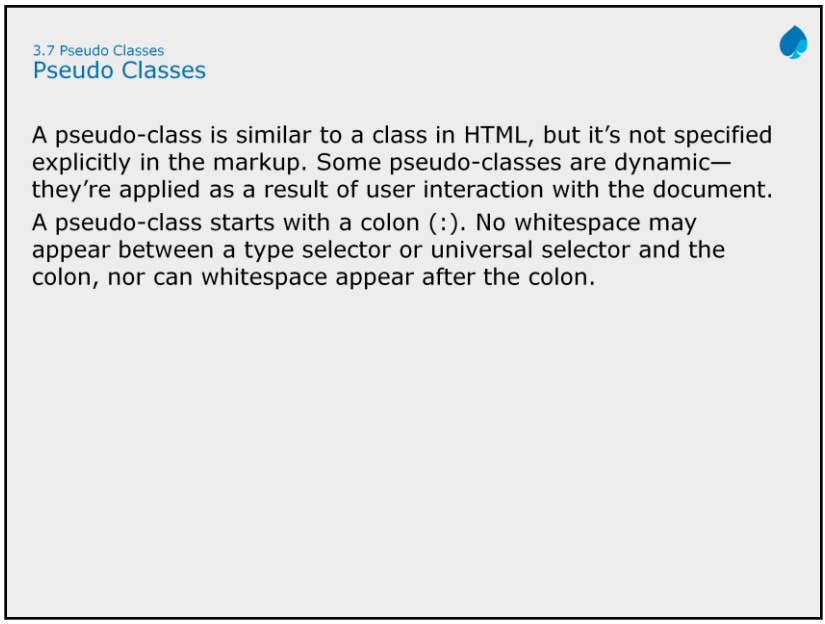
```
img[title]     { border: 1px solid
#000; }
img[width]  { border: 1px solid
#000; }
```

  - The example above will select an element (in this case "img") with the relevant attribute
- Example for Select based on value

```
img[src="small.gif"] { border: 1px solid

                       #000; }
```

  - The above example selects any image whose attribute (in this case "src") has a value of "small.gif"



CSS1 introduced the :link, :visited, and :active pseudo-classes, but only for the HTML a element. These pseudo-classes represented the state of links—unvisited, visited, or currently being selected—in a web page document. In CSS1, all three pseudo-classes were mutually exclusive.

CSS2 expanded the range of pseudo-classes and ensured that they could be applied to any element. :link and :visited now apply to any element defined as a link in the document language. While they remain mutually exclusive, the :active pseudo-class now joins :hover and :focus in the group of dynamic pseudo-classes. The :hover pseudo-class matches elements that are being designated by a pointing device (for example, elements that the user's hovering the cursor over); :active matches any element that's being activated by the user; and :focus matches any element that is currently in focus (that is, accepting input).

CSS2 also introduced the :lang pseudo-class to allow an element to be matched on the basis of its language, and the :first-childpseudo-class to match an element that's the first child element of its parent.

CSS3 promises an even greater range of powerful pseudo-classes.

Remember that pseudo-classes, like ID selectors and attribute selectors, act like modifiers on type selectors and the universal selector: they specify additional constraints for the selector pattern, but they don't specify other elements. For instance, the selector li:first-child matches a list item that's the first child of its parent; it doesn't match the first child of a list item.

Demo : Selector

demoType.html
demoId.html
demoClass.html
demoAttributeSelector.html
demoPseudoClasses.html

Lesson Summary

In this lesson, you have learnt about:

- Universal Selector
- Type Selector
- Class Selector
- ID Selector
- Attribute Selector
- PseudoClasses