

Professional Skills
Agile Fundamentals
Jira
Git
DevOps
Cloud Fundamentals
Databases Introduction
Java Beginner
Maven
Testing (Foundation)
Java Intermediate
HTML
CSS <ul style="list-style-type: none">Introduction to CSS3FontsTransformSelectorsSelectors - Pseudo-Class / Pseudo-ElementsSpatial AttributesAlignmentOverflowDisplay and PositioningBackground and BordersBootstrap IntroductionBootstrap NavBootstrap CollapseBootstrap GridBootstrap Forms & InputsBootstrap ModalBootstrap Cards
Javascript
Spring Boot
Selenium

Selectors

Contents

- Overview
 - CSS rules
 - Hierarchical Inheritance
 - Descendent Selectors
 - Child Selectors
 - Adjacent Selectors
 - Sibling Selectors
- Tutorial
- Exercises

Overview

CSS allows us to apply different styling rules to different HTML elements. When applying rules we need to consider if we want to target one unique element or multiple HTML elements. This is done by using the `id` attribute for a unique element and `class` for targeting multiple HTML elements.

A CSS rule is made up of a selector and a semicolon-separated list of declarations inside brackets. Each declaration has a property and a value separated by a colon. If an element in an associated HTML document matches a selector in the style sheet, then the declarations will be applied to that element:

CSS rules

```
/* An example of a CSS rule */
p {color: blue;}
```

- The `p` is the selector
- `{color:blue;}` is the declaration
- `color` is considered a property
- `blue` is the value.
- Followed by a `;` semi-colon to show the end of that particular rule.

The below code snippet will select **all** `<p>` elements and apply the following CSS rules to it:

- Text to be center aligned.
- Text colour to be red.

```
p {
  text-align: center;
  color: red;
}
```

We can also create a generic rule that selects multiple html elements to be affected by a certain rule. The following rule will be applied to `h1`, `h2` and `p` HTML elements:

```
h1, h2, p {
  text-align: center;
  color: red;
}
```

Sonarqube
Advanced Testing (Theory)
Cucumber
MongoDB
Express
NodeJS
React
Express-Testing
Pre-Project Assessment
Markdown
IDE Cheatsheet

CSS rules can also be applied by using **id**. The identity of an element is unique within a page, so the **id** selector is used to select **one** unique element! We use a **#** followed by the **id** of the element.

```
<p id="para1">This is ready for CSS. </p>
```

```
#para1 {  
  color:red;  
  text-align: centre;  
}
```

(note: an **id** name cannot start with a number.)

The class selector selects HTML elements with a specific class attribute. To select elements with a specific class, write a period (.) character, followed by the class name:

```
<h1 id="myHeading">This is a heading</h1>  
<p class="center">This is ready for CSS.</p>
```

```
.center {  
  text-align: center;  
  color: red;  
}  
#myHeading {  
  font-size: 80px;  
}
```

We can also specify the only HTML elements that should be affected by a class:

```
<h1>This is a heading</h1>  
<p class="center">This is ready for CSS.</p>
```

```
p.center {  
  text-align: center;  
  color: red;  
}
```

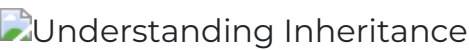
CSS universal selector, selects all HTML elements on the page we do this by using an asterisk wildcard *****:

```
* {  
  text-align: center;  
  color: blue;  
}
```

Hierarchical Inheritance

When a tag is surrounded by another tag, the tags are nested as shown below:

- Tags in the same parent are called siblings.
- Tags immediately next to each other are adjacent siblings.
- Parent elements contain other elements (children).
- Child elements will often inherit styles from a parent element.
- Descendent elements are any elements within another element.



Descendent Selectors

Descendent selectors are used by putting a space between the parent and child – all descendants will be styled.

The below would make all text in any **li** that is a descendant of a **ul** purple:

```
ul li { color: purple }
```

Child Selectors

Child selectors are used by putting a **>** between the parent and the child – any direct child will be styled

The below would make all text in any **p** that is a direct descendant of a section brown:

```
section > p { color: brown }
```

Adjacent Selectors

Adjacent selectors are used by putting a **+** between the siblings – the last sibling listed will be styled.

The below would make all text in any **p** that **immediately follows** an **h2** element black:

```
h2 + p { color: black }
```

Sibling Selectors

Sibling selectors are used by putting a **~** between two elements, which matches all elements that are siblings of the selected element, not just the first one that appears immediately afterwards.

The below would make all text in any **p** that immediately follows an **h2** element **at any point** black:

```
h2 ~ p { color: black }
```

Tutorial

There is no tutorial for this module.

Exercises

1. Style the following html document with CSS for each of the elements
 - Add a blue colour to the **** element
 - Identify a descendant and apply a descendant combinator to set the colour to red.
 - Identify a direct child relationship and apply a lime green colour.
 - Identify an adjacent-sibling relationship and apply a purple colour
 - Identify all siblings and apply a background-colour of grey
 - BONUS: Research pseudo-class and apply a hover effect on **<p class="hover">**

```
<!Doctype html>
<html>
<head>
  <title>L3_CSS_Combinators _Demo</title>
</head>
<body>
  <h1 id="heading" class="font"> A <em>heading</em></h1>
  <p class="font">A paragraph <em>with emphasis</em></p>
  <header>
    <h1>Header</h1>
  </header>

  <article>
    <h1>Article</h1>
    <p>paragraph 1</p>
    <p>paragraph 2</p>
    <footer>Article footer</footer>
  </article>

  <footer>
    Body footer
  </footer>

  <p class="hover">hover me</p>

  <article>
    <h1>Article</h1>
    <p>Paragraph 1</p>
    <p>Paragraph 2</p>
    <blockquote>Quote</blockquote>
    <p>Paragraph 3</p>
    <footer>Article footer</footer>
  </article>
  <footer>Body footer</footer>
</body>
</html>
```

▼ Solution

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>L3_CSS_Combinators _Demo1</title>
  <style>
    em {
      color:#0094ff;
    }
    /* Descendent combinator: more specific */
    p em{
      color:#ff0000;
    }
    /* Direct child */
    body > footer {
      color:#00ff21;
    }
    /* Adjacent-sibling combinator */
    h1 + p{
      color:#4800ff;
    }
    /* all siblings */
    h1 ~ p {
      background-color:#808080;
    }
    /* p:hover -- pseduo class */
    p:hover:hover{
      background-color:#ffd800;
    }
  </style>
</head>
<body>
  <h1 id="heading" class="font"> A <em>heading</em></h1>
  <p class="font">A paragraph <em>with emphasis</em></p>
  <header>
    <h1>Header</h1>
  </header>
  <article>
    <h1>Article</h1>
    <p>paragraph 1</p>
    <p>paragraph 2</p>
    <footer>Article footer</footer>
  </article>
  <footer>
    Body footer
  </footer>
  <p class="hover">hover me</p>
  <article>
    <h1>Article</h1>
    <p>Paragraph 1</p>
    <p>Paragraph 2</p>
    <blockquote>Quote</blockquote>
    <p>Paragraph 3</p>
    <footer>Article footer</footer>
  </article>
  <footer>Body footer</footer>
</body>
</html>
```