

# Lab 03. CSS (1)

인터넷과웹기초



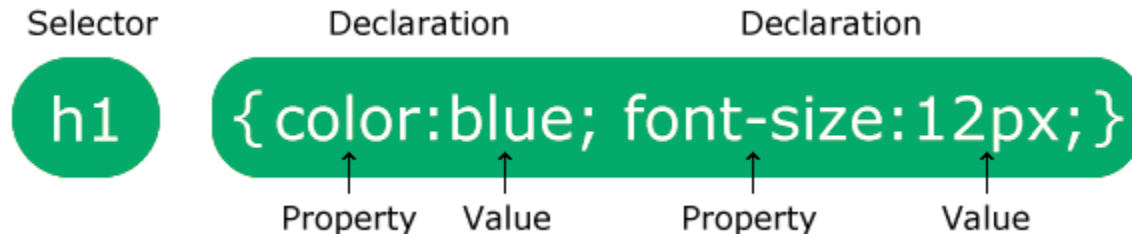
# Index

---

- CSS How To
- CSS Selectors
- CSS Combinators
- CSS Pseudo-classes
- CSS Pseudo-elements
- CSS Attribute Selectors
- CSS Specificity
- Colors

# Syntax

- A CSS rule consists of a selector and a declaration block



- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes CSS property name and a value, separated by a colon.

# How To

---

- When a browser reads a style sheet, it will format the HTML document according to the information in the style sheet.
- There are three ways of inserting a style sheet:
  - External CSS
  - Internal CSS
  - Inline CSS

# External CSS

- External CSS

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

```
body {
    background-color: lightblue;
}

h1 {
    color: navy;
    margin-left: 20px;
}
```

**This is a heading**

This is a paragraph.

- An external CSS style sheet must be saved with a .css extension.
- The external .css file should not contain any HTML tags.

# Internal CSS

- Internal CSS

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: linen;
}

h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

**This is a heading**

This is a paragraph.

- The internal style is defined inside the <style> element, inside the head section.

# Inline CSS

- Inline CSS

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: linen;
}

h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

**This is a heading**

This is a paragraph.

# Cascading Order

---

- CSS
  - Cascading Style Sheets
- Cascading
  - All the Styles in a page will cascade into a new virtual style sheet by the following rules, where number one has the highest priority:
    - Inline style (inside an HTML element)
    - External and internal style sheets (in the head section)
    - Browser default



# Multiple Style Sheets Example

- When they are same priority.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  color: orange;
}
</style>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>The style of this document is a combination of an external stylesheet,
and internal style</p>

</body>
</html>
```

```
h1 {
  color: navy;
}
```

**This is a heading**

The style of this document is a combination of an external stylesheet, and internal style



# Simple Selectors

# Selectors

---

- Selectors
  - CSS Selectors are used to select the HTML elements.
- Categories of selectors
  - Simple selectors
  - Combinator selectors
  - Pseudo-class selectors
  - Pseudo-elements selectors
  - Attribute selectors

# Simple selectors

- Select elements based on name, id, class ...

Selector	Example	Example description
<u><a href="#">#id</a></u>	#firstname	Selects the element with id="firstname"
<u><a href="#">.class</a></u>	.intro	Selects all elements with class="intro"
<u><a href="#">element.class</a></u>	p.intro	Selects only <p> elements with class="intro"
<u><a href="#">*</a></u>	*	Selects all elements
<u><a href="#">element</a></u>	p	Selects all <p> elements
<u><a href="#">element,element,..</a></u>	div, p	Selects all <div> elements and all <p> elements

# Simple selectors

- 1. id selector
  - #id { }

```
<!DOCTYPE html>
<html>
<head>
<style>
#para1 {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<p id="para1">Hello World!</p>
<p>This paragraph is not affected by the style.</p>

</body>
</html>
```

Hello World!

This paragraph is not affected by the style.

# Simple selectors

- 2. class selector
  - .class { }

```
<!DOCTYPE html>
<html>
<head>
<style>
{
  .center {
    text-align: center;
    color: red;
  }
</style>
</head>
<body>

<h1>Red and center-aligned heading</h1>
<p class="center">Red and center-aligned paragraph.</p>

</body>
</html>
```

**Red and center-aligned heading**

Red and center-aligned paragraph.

# Simple selectors

- 3. element selector
  - tag { }

```
<!DOCTYPE html>
<html>
<head>
<style>
{
  p {
    text-align: center;
    color: red;
  }
</style>
</head>
<body>

<p>Every paragraph will be affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>

</body>
</html>
```

Every paragraph will be affected by the style.

Me too!

And me!

# Simple selectors

- 4. universal selector
  - \* {}

```
<!DOCTYPE html>
<html>
<head>
<style>
{
* {
  text-align: center;
  color: blue;
}
}</style>
</head>
<body>

<h1>Hello world!</h1>

<p>Every element on the page will be affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>

</body>
</html>
```

**Hello world!**

Every element on the page will be affected by the style.

Me too!

And me!



# Simple selectors

- 5. grouping selector
  - tag, tag, ... { }

```
<!DOCTYPE html>
<html>
<head>
<style>
h1, h2, p {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<h1>Hello World!</h1>
<h2>Smaller heading!</h2>
<p>This is a paragraph.</p>

</body>
</html>
```

**Hello World!**

**Smaller heading!**

This is a paragraph.



# Combinators

# Combinator

---

- Combinator
  - It explains the relationship between the selectors.
- There are four different combinators in CSS
  - Descendant selector (space)
  - Child selector (>)
  - Adjacent sibling selector (+)
  - General sibling selector (~)

# Combinator

- 1. Descendant selector
  - It matches all elements that are descendants of a specified element.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
{div p {  
  background-color: yellow;  
}}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h2>Descendant Selector</h2>
```

```
<p>The descendant selector matches all elements that are descendants of a  
specified element.</p>
```

```
<div>
```

```
  <p>Paragraph 1 in the div.</p>
```

```
  <p>Paragraph 2 in the div.</p>
```

```
  <section><p>Paragraph 3 in the div.</p></section>  
</div>
```

```
<p>Paragraph 4. Not in a div.</p>
```

```
<p>Paragraph 5. Not in a div.</p>
```

```
</body>
```

```
</html>
```

## Descendant Selector

The descendant selector matches all elements that are descendants of a specified element.

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div.

Paragraph 4. Not in a div.

Paragraph 5. Not in a div.

# Combinator

- 2. Child selector

- It selects all elements that are the children of a specified element.

```
<!DOCTYPE html>
<html>
<head>
<style>


## Child Selector



The child selector (>) selects all elements that are the children of a specified element.



Paragraph 1 in the div.



Paragraph 2 in the div.



Paragraph 3 in the div (inside a section element).



Paragraph 4 in the div.



Paragraph 5. Not in a div.



Paragraph 6. Not in a div.


```

# Combinator

- 3. Adjacent sibling selector

```
<!DOCTYPE html>
<html>
<head>
<style>
div + p {
  background-color: yellow;
}
</style>
</head>
<body>
```

```
<h2>Adjacent Sibling Selector</h2>
```

```
<p>The + selector is used to select an element that is directly after
another specific element.</p>
```

```
<p>The following example selects the first p element that are placed
immediately after div elements:</p>
```

```
<div>
  <p>Paragraph 1 in the div.</p>
</div>
```

```
<p>Paragraph 2. After a div.</p>
```

```
<div>
  <p>Paragraph 3 in the div.</p>
</div>
```

```
<p>Paragraph 4. After a div.</p>
<p>Paragraph 5. After a div.</p>
```

```
</body>
</html>
```

## Adjacent Sibling Selector

The + selector is used to select an element that is directly after another specific element.

The following example selects the first p element that are placed immediately after div elements:

Paragraph 1 in the div.

Paragraph 2. After a div.

Paragraph 3 in the div.

Paragraph 4. After a div.

Paragraph 5. After a div.

# Combinator

- 4. General Sibling Selector

```
<!DOCTYPE html>
<html>
<head>
<style>


```
<h2>General Sibling Selector</h2>
```



```
<p>The general sibling selector (~) selects all elements that are next
siblings of a specified element.</p>
```



```
<p>Paragraph 1.</p>
```



```
<div>
  <p>Paragraph 2.</p>
</div>
```



```
<p>Paragraph 3.</p>
<code>Some code.</code>
<p>Paragraph 4.</p>
```



```
</body>
</html>
```



## General Sibling Selector



The general sibling selector (~) selects all elements that are next siblings of a specified element.



Paragraph 1.



Paragraph 2.



Paragraph 3.




Some code.



Paragraph 4.


```



# Pseudo-classes & Pseudo-elements



# Pseudo-classes

---

- Pseudo-classes
  - It is used to define a special state of an element.

- Syntax

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

- Examples

- [Anchor Pseudo-class](#)
- [Hover on <div>](#)
- [Simple Tooltip Hover](#)
- [First child pseudo-class](#)
- [Use of :focus](#)

# Pseudo-elements

---

- Pseudo-elements
  - It is used to style specified parts of an element.

- Syntax

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

- Example

- [::first-line](#)
- [::first-letter](#)
- [::before](#)
- [::after](#)
- [::marker](#)
- [::selection](#)



# Attribute selectors

# Attribute selectors

---

- Attribute selectors
  - It is possible to style HTML elements that have specific attributes or attribute values.

- [attribute] selector

```
a[target] {  
    background-color: yellow;  
}
```

- [attribute="value"] selector

```
a[target="_blank"] {  
    background-color: yellow;  
}
```



CSS specificity

# Specificity hierarchy

- Specificity
  - CSS selector has its place in the specificity hierarchy
- Specificity level
  - Inline styles : 1000
  - IDs : 100
  - Classes/Pseudo-classes/attributes : 10
  - Elements/Pseudo-elements : 1
  - Universal selector : 0

```
<!DOCTYPE html>
<html>
<head>
<style>
.test {color: green;}
p {color: red;}
</style>
</head>
<body>
```

```
<p class="test">Hello World!</p>
```

```
</body>
</html>
```

Hello World!

```
<!DOCTYPE html>
<html>
<head>
<style>
p {color: red;}
</style>
</head>
<body>
```

```
<p>Hello World!</p>
```

```
</body>
</html>
```

Hello World!



# CSS Colors

# Colors

- Colors

- It is specified using predefined color names, or RGB, HEX, HSL, RGBA, HSLA values.
- CSS/HTML support 140 standard color names.
  - Tip. Using html/css extension in vscode can help you

- Example

```
<!DOCTYPE html>
<html>
<body>

<h1 style="background-color:DodgerBlue;">Hello World</h1>

<p style="background-color:Tomato;">
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam
nonummy nibh euismod tincidunt
volutpat.
Ut wisi enim ad minim veniam,
suscipit lobortis nisl ut aliquam
</p>

</body>
</html>
```

**Hello World**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.



# Colors

- RGB colors

- Using RGB value
- `rgb(red, green, blue)`

`rgb(255, 0, 0)`

`rgb(60, 179, 113)`

`rgb(255, 165, 0)`

- HEX colors

- Using hexadecimal value
- `#rrggbb`

`#ff0000`

`#3cb371`

`#ffa500`

- HSL colors

- Using hue, saturation, lightness
- `hsl(hue, saturation, lightness)`

`hsl(0, 100%, 50%)`

`hsl(147, 50%, 47%)`

`hsl(39, 100%, 50%)`

---

# QnA



**부산대학교**  
PUSAN NATIONAL UNIVERSITY

