Web Programming

CSS3 Layout, Combinators

Instructor: Prof. SoYeop Yoo
School of Computing, Gachon University

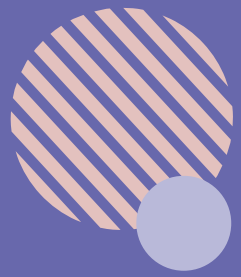
Class Selector vs. ID Selector (revisited)

Similarities

- Assign an identifier to define a special case for an element
- Apply a certain formatting style to specified areas

Differences

- ID (unique)
  - Each element can have only one ID
  - Each page can have only one element with that ID
- Class (not unique)
  - You can use the same class on multiple elements
  - You can use multiple classes on the same element



CONTENTS

List

CSS3 List Property

HTML5 List

HTML Tag	Description
<UL>	Show items in the list with bullets
<OL>	Displays items in the list by number or letter

CSS3 List Property

Property	Description
list-style-type	Properties that specify the type of bullets in the list
list-style-image	Properties that assign list bullets as images
list-style-position	Properties that specify the location of the list bullet inside or outside the area.

Different List Item Markers (1/2)

list-style-type Property

- Specify the type properties of bullets

```
ul {
        list-style-type: value;
}
ol {
        list-style-type: value;
}
```

- Values
  - disc, circle, square
  - decimal, decimal-leading-zero, lower-alpha, lower-greek, lower-latin, lower-roman, upper-alpha, upper-latin, upper-roman, armenian, cjk-ideographic, georgian, hebrew, hiragana, hiragana-iroha, katakana, katakana-iroha
  - none

Different List Item Markers (2/2)

Example

- Espresso
- Cappuccino
- Cafe Latte

- Espresso
- Cappuccino
- Cafe Latte

- Espresso
- Cappuccino
- Cafe Latte

1. Espresso
2. Cappuccino
3. Cafe Latte

01. Espresso
02. Cappuccino
03. Cafe Latte

a. Espresso
b. Cappuccino
c. Cafe Latte

A. Espresso
B. Cappuccino
C. Cafe Latte

```
ul.a { list-style-type: disc; }

ul.b { list-style-type: circle; }

ul.c { list-style-type: square; }

ol.d { list-style-type: decimal; }

ol.e { list-style-type: decimal-leading-zero; }

ol.f { list-style-type: lower-alpha; }

ol.g { list-style-type: upper-alpha; }
```

a. Espresso
b. Cappuccino
c. Cafe Latte

A. Espresso
B. Cappuccino
C. Cafe Latte

i. Espresso
ii. Cappuccino
iii. Cafe Latte

I. Espresso
II. Cappuccino
III. Cafe Latte

α. Espresso
β. Cappuccino
γ. Cafe Latte

Espresso
Cappuccino
Cafe Latte

```
ol.h { list-style-type: lower-latin; }

ol.i { list-style-type: upper-latin; }

ol.j { list-style-type: lower-roman; }

ol.k { list-style-type: upper-roman; }

ol.l { list-style-type: lower-greek; }

ol.m { list-style-type: none; }
```

Image as List Item Marker & Position (1/2)

Image as List Item Marker

- *list-style-image* property
- Properties that assign list bullets as images

```
ul {
        list-style-image: url("coffee.png");
}
```

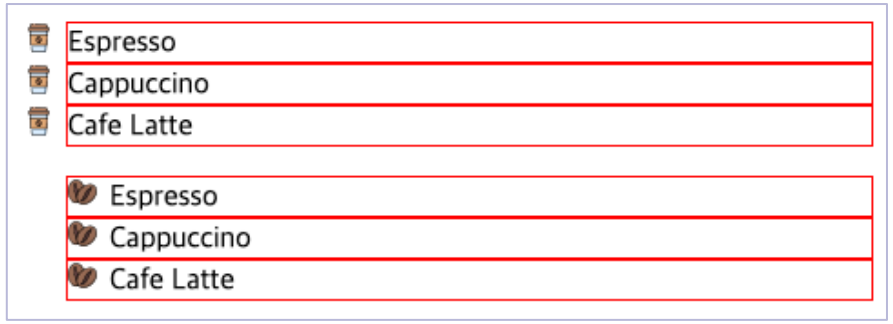
Position List Item Marker

- *list-style-position* property
- Properties that position bullets

```
ul {
        list-style-position: inside | outside;
}
```

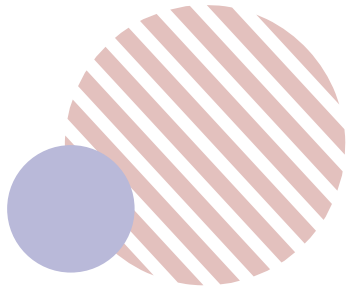
Image as List Item Marker & Position (2/2)

Example




```
<!DOCTYPE html>
<html>
      <head>
            <title>List Property</title>
            <style>
                  li {  border: 1px solid red;  }
                  ul.a {
                        list-style-image: url("coffee.png");
                        list-style-position: outside;
                  }
                  ul.b {
                        list-style-image: url("coffee-beans.png");
                        list-style-position: inside;
                  }
            </style>
      </head>
      <body>
            <ul class="a">
                  <li>Espresso</li>
                  <li>Cappuccino</li>
                  <li>Cafe Latte</li>
            </ul>
            <ul class="b">
                  <li>Espresso</li>
                  <li>Cappuccino</li>
                  <li>Cafe Latte</li>
            </ul>
      </body>
</html>
```

Overflow

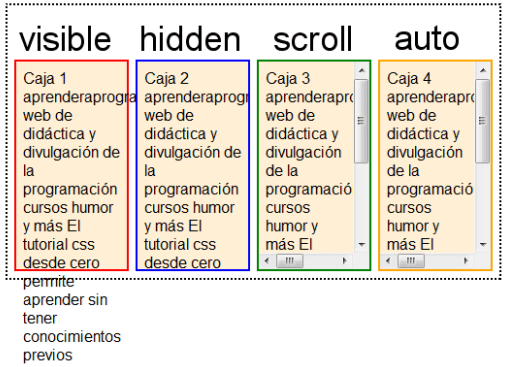


CSS3 Overflow Property (1/4)

overflow Property

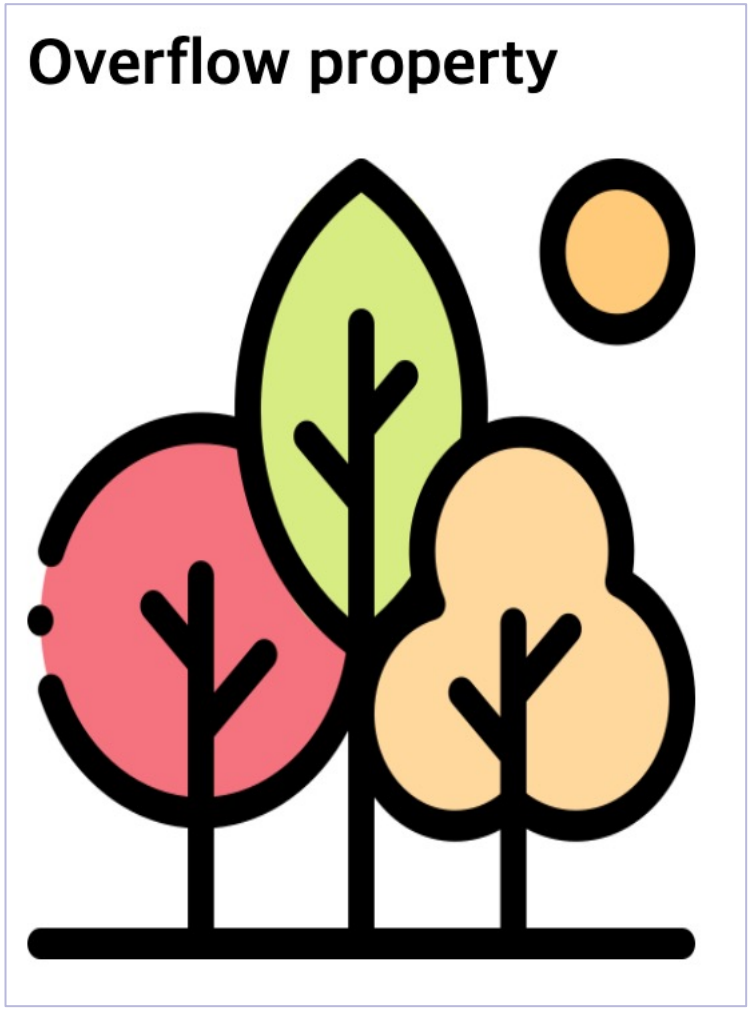
- Specify how the element will be shown when the content is not visible beyond the size of the element
- Values
  - visible, hidden, scroll, and auto

Property	Value	Description
overflow	visible	Contents are not clipped *(default)*
	hidden	Invisible areas outside the area
	scroll	You can see the contents of the element by scrolling out of the area.
	auto	Similar to scroll, but it adds scrollbars only when necessary



CSS3 Overflow Property (2/4)

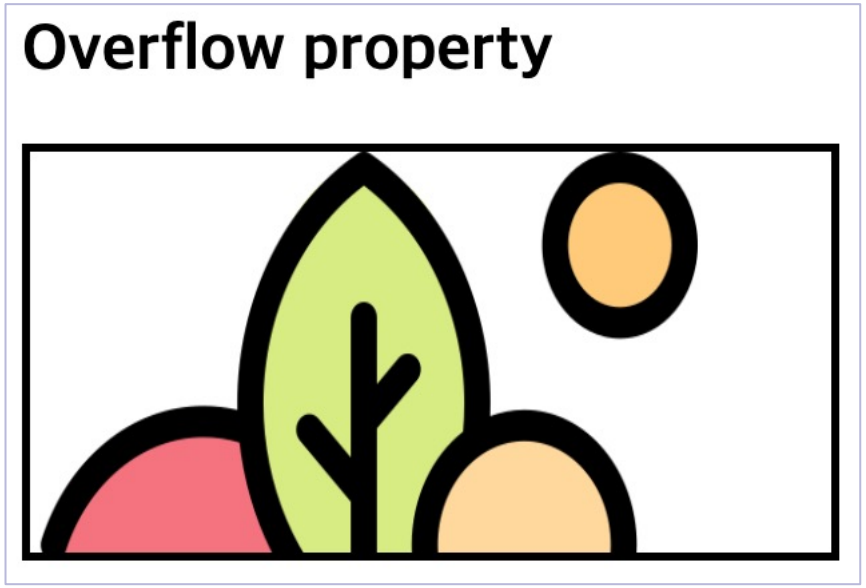
Example




```
<!DOCTYPE html>
<html>
     <head>
          <title>Overflow Property</title>
          <style>
               img {
                    width: 250px;
                    height: 300px;
               }
          </style>
     </head>
     <body>
          <h2>Overflow property</h2>
          <div>
                    <img src="trees.png" />
          </div>
     </body>
</html>
```

CSS3 Overflow Property (3/4)

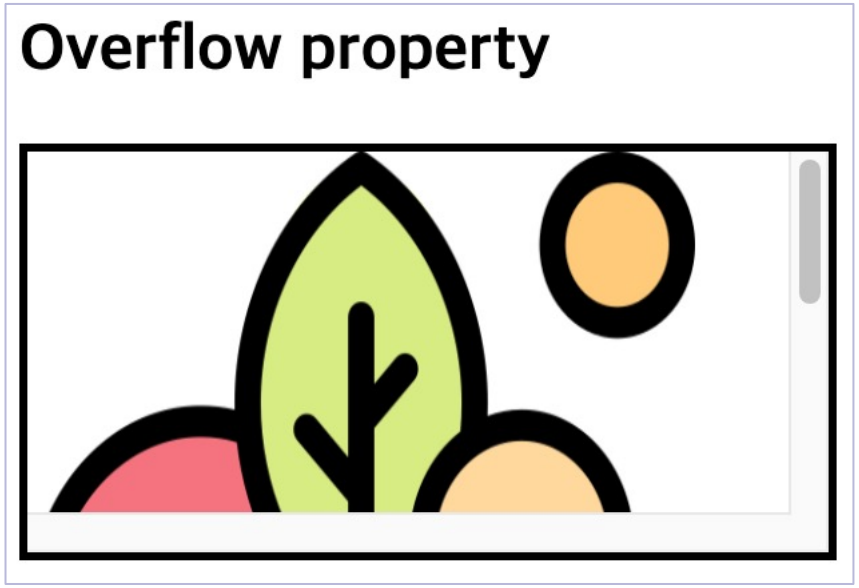
Example - hidden




```
<!DOCTYPE html>
<html>
	<head>
		<title>Overflow Property</title>
		<style>
			img {
				width: 250px;
				height: 300px;
			}
			div {
				width: 300px;
				height: 150px;
				border: 3px solid black;
				position: relative;
				overflow: hidden;
			}
		</style>
	</head>
	<body>
		<h2>Overflow property</h2>
		<div>
			<img src="trees.png" />
		</div>
	</body>
</html>
```

CSS3 Overflow Property (4/4)

Example - scroll



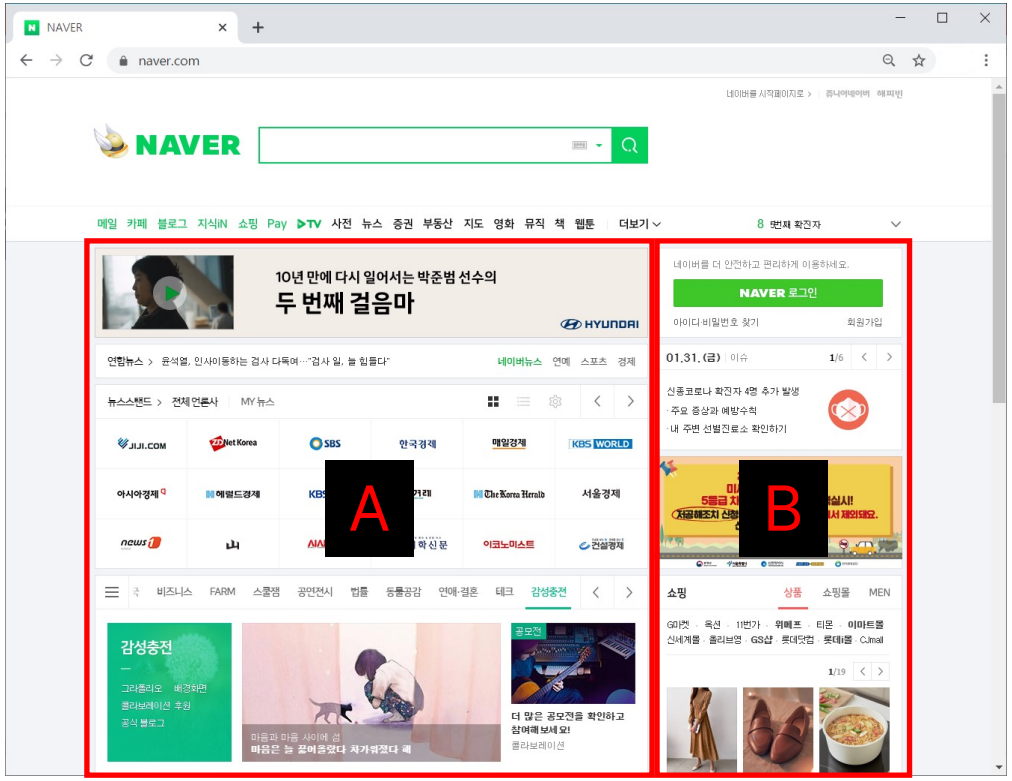

```
<!DOCTYPE html>
<html>
    <head>
        <title>Overflow Property</title>
        <style>
            img {
                width: 250px;
                height: 300px;
            }
            div {
                width: 300px;
                height: 150px;
                border: 3px solid black;
                position: relative;
                overflow: scroll;
            }
        </style>
    </head>
    <body>
        <h2>Overflow property</h2>
        <div>
            <img src="trees.png" />
        </div>
    </body>
</html>
```

Layout

Horizontal Layout (1/3)

Horizontal Layout

- Rules for horizontally aligning elements
  - The overflow property of the parent applies the hidden keyword
  - Assign float attributes to descendants



Horizontal Layout (2/3)

Example (1/2)


```
<!DOCTYPE html>
<html>
      <head>
            <title>Horizontal Layout</title>
      </head>
      <body>
            <h2>Welcome~!!</h2>
            <div>
                  <div><a href="#">Menu -1</a></div>
                  <div><a href="#">Menu -2</a></div>
                  <div><a href="#">Menu -3</a></div>
                  <div><a href="#">Menu -4</a></div>
            </div>
            <p>Click the menu you want!</p>
      </body>
</html>
```

Horizontal Layout (3/3)

Example (2/2)


```html
<!DOCTYPE html>
<html>
	<head>
		<title>Horizontal Layout</title>
		<style>
			div.container {
				overflow: hidden;
			}
			div.item {
				float: left;
				margin: 0 3px;
				padding: 10px;
				border: 1px solid black;
			}
		</style>
	</head>
	<body>
		<h2>Welcome~!!</h2>
		<div class="container">
			<div class="item"><a href="#">Menu -1</a></div>
			<div class="item"><a href="#">Menu -2</a></div>
			<div class="item"><a href="#">Menu -3</a></div>
			<div class="item"><a href="#">Menu -4</a></div>
		</div>
		<p>Click the menu you want!</p>
	</body>
</html>
```

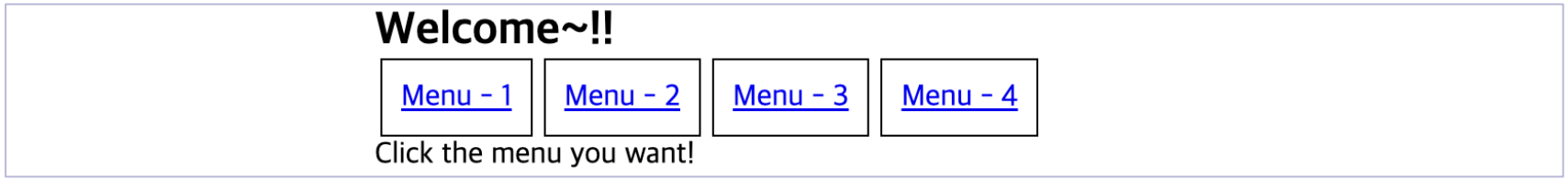
Central Sort Layout (1/2)

Central Layout

- The most used layout; positions the content in the center of a specified width
- Rules of central sorting
  - Specify the width property
  - Specify the margin property as ‘0 auto’

Central Sort Layout (2/2)

Example

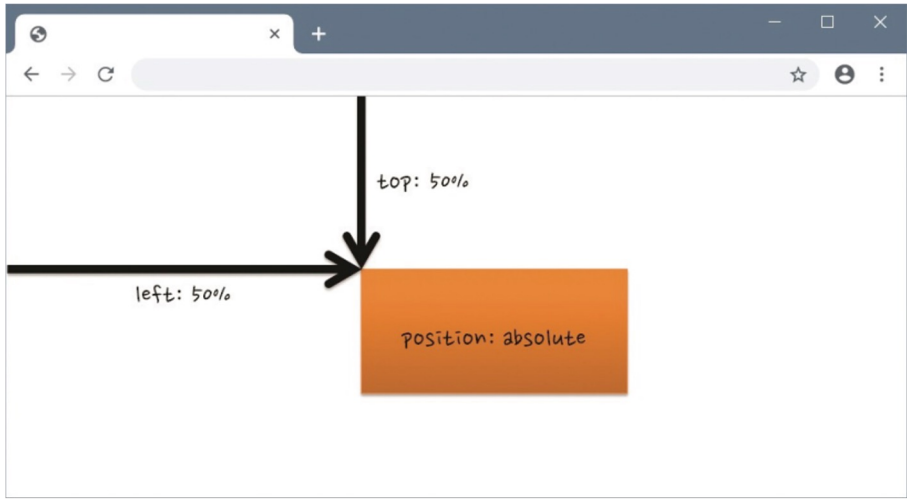


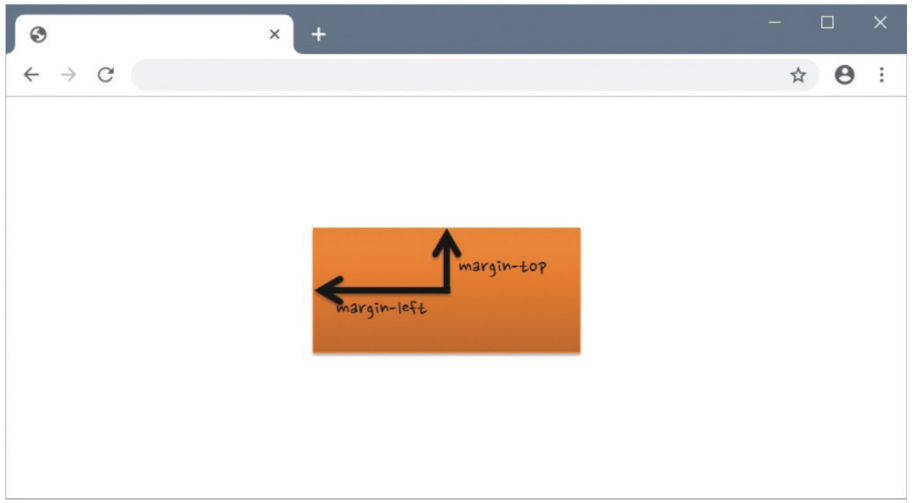

```
<head>
    <title>Central Sort Layout</title>
    <style>
        * {
            margin: 0;
            padding: 0;
        }
        body {
            margin: 0 auto;
            width: 700px;
        }
        div.container { overflow: hidden; }
        div.item {
            float: left;
            margin: 0 3px;
            padding: 10px;
            border: 1px solid black;
        }
    </style>
</head>
```

Central Placement of Elements (1/2)

Central Placement

- Rules to use when placing elements centrally
  - Set the *position* property of the div tag to absolute
  - Both the left and top properties are specified at *50%*

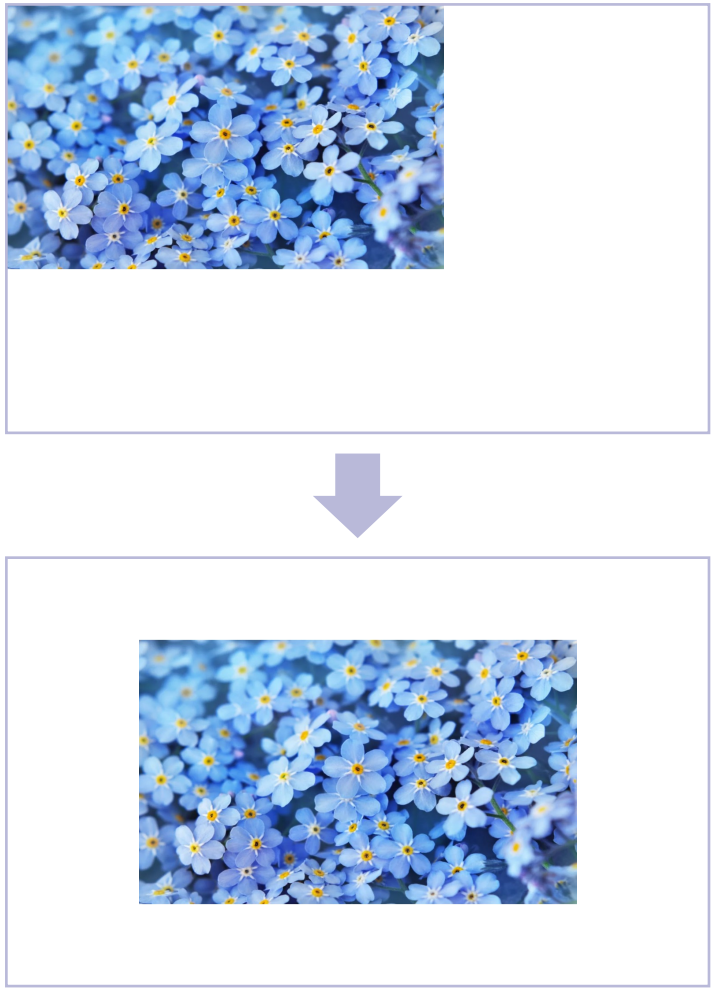




  - Enter a negative value in the *margin-left* and *margin-top* properties of the div tag
  - The value you enter must be exactly half the width and height of the div tag

Central Placement of Elements (2/2)

Example

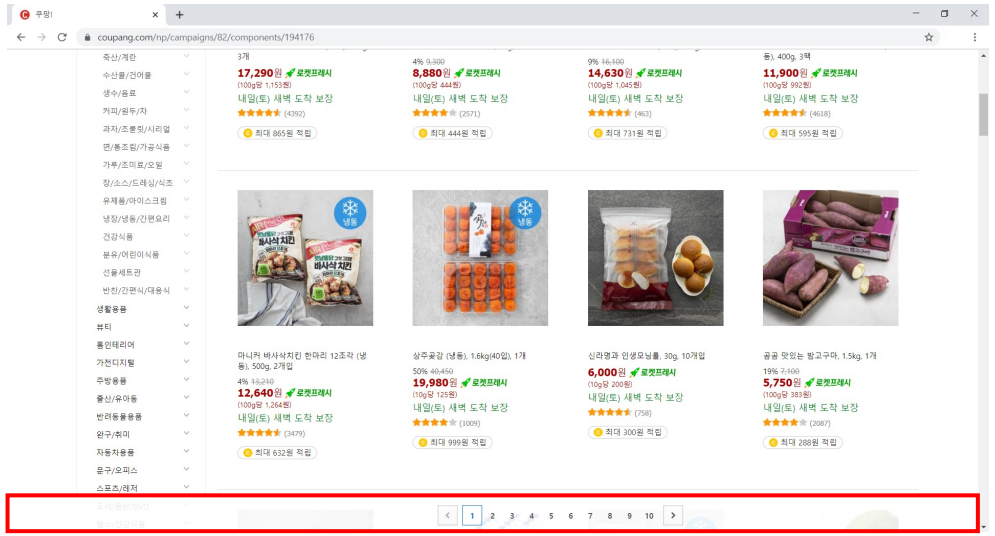


```
<!DOCTYPE html>
<html>
	<head>
		<title>Central Placement of Elements</title>
		<style>
			* {
				margin: 0;
				padding: 0;
			}
			img {
				width: 500px;
				height: 300px;
			}
			#container {
				position: absolute;
				left: 50%;
				top: 50%;
				margin-left: -250px;
				margin-top: -150px;
			}
		</style>
	</head>
	<body>
		<div id="container">
			<img src="flowers.jpg" />
		</div>
	</body>
</html>
```

Place Elements in a Fixed Position (1/3)

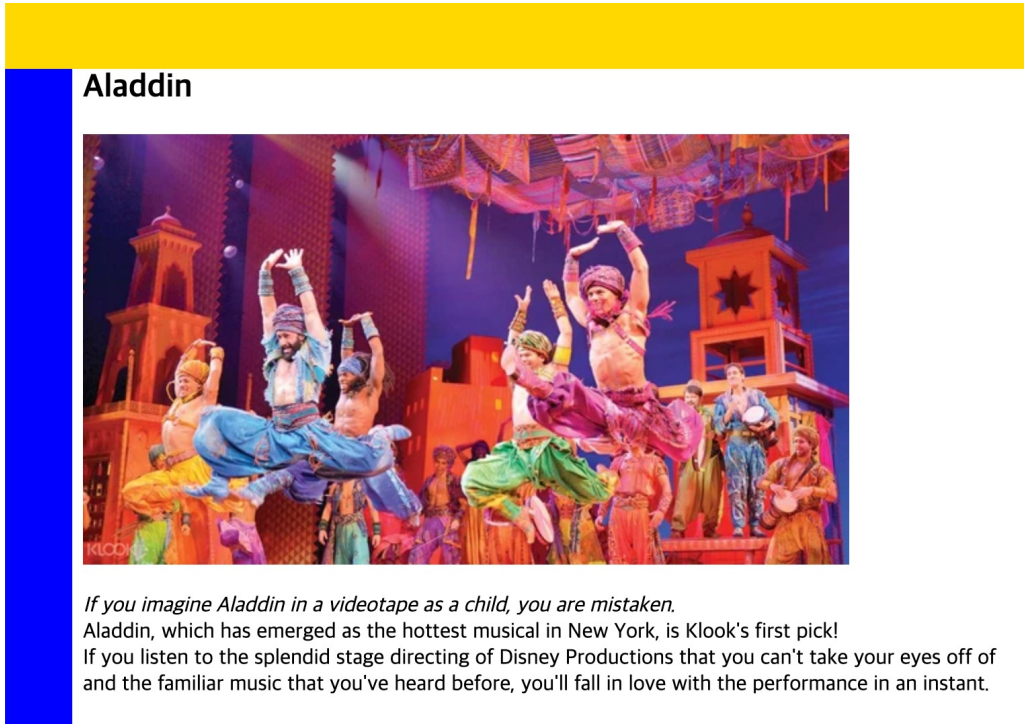
Fixed Position

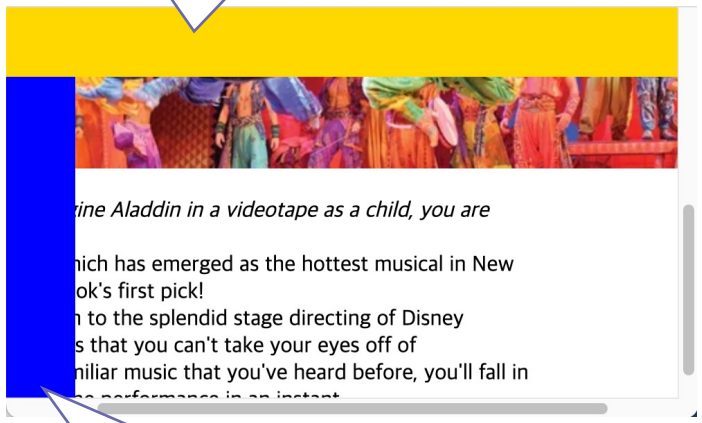
- Fix it to a specific location on a web page
- Used a lot in shopping malls or web applications
- Rules to use when an element is placed in a fixed position
  - Apply *fixed* to position property
  - Enter a value in the left, the top, the right, and the bottom property to set the location
  - Set the size to the width property and height property



Place Elements in a Fixed Position (2/3)

Example





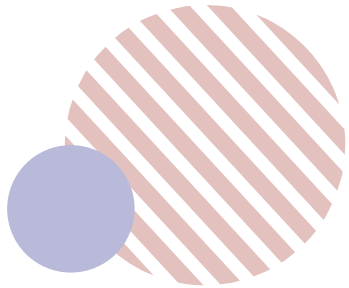
Place Elements in a Fixed Position (2/3)

Example

```
<!DOCTYPE html>
<html>
      <head>
            <title>Fixed Bar Placement</title>
            <style>
                  .container {
                        margin-top: 50px;
                        margin-left: 50px;
                  }
                  .top-bar {
                        background-color: gold;
                        position: fixed;
                        left: 0;
                        top: 0;
                        right: 0;
                        height: 50px;
                  }
                  .left-bar {
                        background-color: blue;
                        position: fixed;
                        left: 0;
                        top: 50px;
                        bottom: 0;
                        width: 50px;
                  }
            </style>
      </head>
```

```
      <body>
            <div class="top-bar"></div>
            <div class="left-bar"></div>
            <div class="container">
                  <h2>Aladdin</h2>
                  <div id="div1">
                        <img src="poster.jpg" />
                        <p>
                              <em>
                                    If you imagine Aladdin in a videotape as
                                    a child, you are mistaken.
                              </em>
                              <br />
                              Aladdin, which has emerged as the hottest music
                              al in New York, is
                              Klook's first pick! <br />
                              If you listen to the splendid stage directing o
                              f Disney Productions
                              that you can't take your eyes off of
                              <br />and the familiar music that you've heard
                              before, you'll fall in
                              love with the performance in an instant.<br />
                        </p>
                  </div>
            </div>
      </body>
</html>
```

Combinators



CSS Combinators (1/5)

Combinators

- A combinator is something that explains the relationship between the selectors.
- A CSS selector can contain more than one simple selector.
- Between the simple selectors, we can include a combinator.
- There are four different combinators in CSS:
  - descendant selector (space)
    ```
    div p { background-color: yellow; }
    ```
    - The descendant selector matches all elements that are descendants of a specified element
  - child selector (>)
    ```
    div > p { background-color: yellow; }
    ```
    - The child selector selects all elements that are the children of a specified element
  - general sibling selector (~)
    ```
    div ~ p { background-color: yellow; }
    ```
    - The general sibling selector selects all elements that are siblings of a specified element
    - Sibling elements must have the same parent element
  - adjacent sibling selector (+)
    ```
    div + p { background-color: yellow; }
    ```
    - The adjacent sibling selector selects all elements that are the adjacent siblings of a specified element
    - "adjacent" means "immediately following"

CSS Combinators (2/5)

Example – descendent selector

This is a paragraph inside a div element.

This is another paragraph inside a div element.

This is a paragraph inside a span element, inside a div element.

This is a paragraph, not inside a div element.

This is another paragraph, not inside a div element.

This is a paragraph inside a div element.

This is another paragraph inside a div element.

This is a paragraph inside a span element, inside a div element.

This is a paragraph, not inside a div element.

This is another paragraph, not inside a div element.

example6-1.html

```
<body>
      <div>
            <p>This is a paragraph inside a div element.</p>
            <p>This is another paragraph inside a div element.</p>
            <span>
                  <p>This is a paragraph inside a span element,
                   inside a div element.</p>
            </span>
      </div>
      <p>This is a paragraph, not inside a div element.</p>
      <p>This is another paragraph, not inside a div element.</p>
</body>
```

```
<style>
      div p {
            color: red;
            background-color: pink;
      }
</style>
```

CSS Combinators (3/5)

Example – child selector

This is a paragraph inside a div element.

This is another paragraph inside a div element.

This is a paragraph inside a span element, inside a div element.

This is a paragraph, not inside a div element.

This is another paragraph, not inside a div element.

This is a paragraph inside a div element.

This is another paragraph inside a div element.

This is a paragraph inside a span element, inside a div element.

This is a paragraph, not inside a div element.

This is another paragraph, not inside a div element.

```
<body>
      <div>
            <p>This is a paragraph inside a div element.</p>
            <p>This is another paragraph inside a div element.</p>
            <span>
                  <p>This is a paragraph inside a span element,
                   inside a div element.</p>
            </span>
      </div>
      <p>This is a paragraph, not inside a div element.</p>
      <p>This is another paragraph, not inside a div element.</p>
</body>
```

```
<style>
      div > p {
            color: red;
            background-color: pink;
      }
</style>
```

CSS Combinators (4/5)

Example – general sibling selector

This is a paragraph inside a div element.

This is another paragraph inside a div element.

This is a paragraph inside a span element, inside a div element.

This is a paragraph, not inside a div element.

This is another paragraph, not inside a div element.

This is a paragraph inside a div element.

This is another paragraph inside a div element.

This is a paragraph inside a span element, inside a div element.

This is a paragraph, not inside a div element.

This is another paragraph, not inside a div element.

```
<body>
      <div>
            <p>This is a paragraph inside a div element.</p>
            <p>This is another paragraph inside a div element.</p>
            <span>
                  <p>This is a paragraph inside a span element,
                   inside a div element.</p>
            </span>
      </div>
      <p>This is a paragraph, not inside a div element.</p>
      <p>This is another paragraph, not inside a div element.</p>
</body>
```

```
<style>
      div ~ p {
            color: red;
            background-color: pink;
      }
</style>
```

CSS Combinators (5/5)

Example – adjacent sibling selector

This is a paragraph inside a div element.

This is another paragraph inside a div element.

This is a paragraph inside a span element, inside a div element.

This is a paragraph, not inside a div element.

This is another paragraph, not inside a div element.

This is a paragraph inside a div element.

This is another paragraph inside a div element.

This is a paragraph inside a span element, inside a div element.

This is a paragraph, not inside a div element.

This is another paragraph, not inside a div element.

```
<body>
      <div>
            <p>This is a paragraph inside a div element.</p>
            <p>This is another paragraph inside a div element.</p>
            <span>
                  <p>This is a paragraph inside a span element,
                   inside a div element.</p>
            </span>
      </div>
      <p>This is a paragraph, not inside a div element.</p>
      <p>This is another paragraph, not inside a div element.</p>
</body>
```

```
<style>
      div + p {
            color: red;
            background-color: pink;
      }
</style>
```

Lost Effect (1/2)

Case of Lost Effect

- Although the intention of these rules is to add emphasis to text by changing its color, the effect will be lost in a case like below

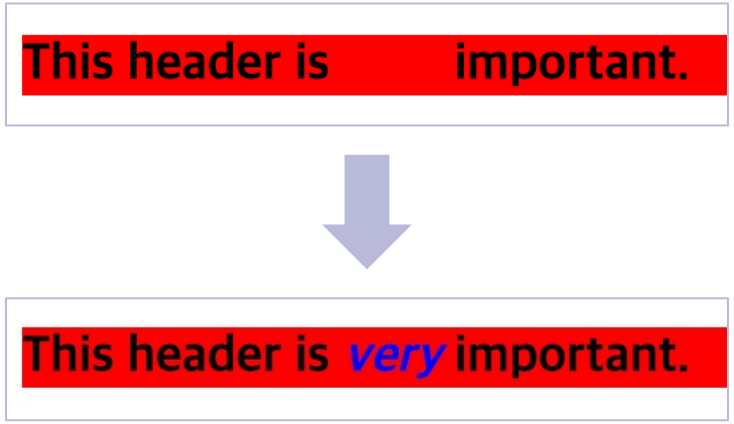
```
<!DOCTYPE html>
<html>
     <head>
          <title>Lost Effect</title>
          <style>
               h3 {
                    background-color: red;
               }
               em {
                    color: red;
               }
          </style>
     </head>
     <body>
          <h3>This header is <em>very</em> important.</h3>
     </body>
</html>
```

Lost Effect (2/2)

Case of Lost Effect - Solution

- We address this case by supplementing the previous rules with a rule that sets the text c olor to blue whenever an occurs anywhere within an <h3>

```
<!DOCTYPE html>
<html>
      <head>
            <title>Lost Effect</title>
            <style>
                  h3 {
                        background-color: red;
                  }
                  em {
                        color: red;
                  }
                  h3 em {
                        color: blue;
                  }
            </style>
      </head>
      <body>
            <h3>This header is <em>very</em> important.</h3>
      </body>
</html>
```



CSS3 Pseudo-classes (1/3)

Pseudo-classes

- Used to define a special state of an element
- Examples
  - Style an element when a user mouse hovers over it
  - Style visited and unvisited links differently
  - Style an element when it gets focus
- Syntax
  - selector:pseudo-class {
    property: value;
    }

```
<style>
    a:link { color: blue; }          /* unvisited link */
</style>
```

CSS3 Pseudo-classes (2/3)

Example

School of Computing Home
Google
W3 Schools

School of Computing Home
Google
W3 Schools

School of Computing Home
Google
W3 Schools

```
<!DOCTYPE html>
<html>
	<head>
		<title>Pseudo-classes</title>
		<style>
			/* unvisited link */
			a:link {
				color: blue;
			}
			/* visited link */
			a:visited {
				color: purple;
			}
			/* mouse over link */
			a:hover {
				color: red;
			}
			/* selected link */
			a:active {
				color: green;
			}
		</style>
	</head>
	<body>
		<a href="https://sw.gachon.ac.kr">School of Computing Home</a><br />
		<a href="https://www.google.com">Google</a><br />
		<a href="https://w3schools.com">W3 Schools</a>
	</body>
</html>
```

CSS3 Pseudo-classes (3/3)

Additional Pseudo-classes

Selector	Example	Example description
:checked	input:checked	Selects every checked <input> element
:disabled	input:disabled	Selects every disabled <input> element
:empty	p:empty	Selects every <p> element that has no children
:enabled	input:enabled	Selects every enabled <input> element
:first-child	p:first-child	Selects every <p> element that is the first child of its parent
:first-of-type	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
:focus	input:focus	Selects the <input> element that has focus
:in-range	input:in-range	Selects <input> elements with a value within a specified range
:invalid	input:invalid	Selects all <input> elements with an invalid value
:lang(language)	p:lang(it)	Selects every <p> element with a lang attribute value starting with “it”
:last-child	p:last-child	Selects every <p> element that is the last child of its parent
:last-of-type	p:last-of-type	Selects every <p> element that is the last <p> element of its parent
:not(selector)	:not(p)	Selects every element that is not a <p> element
:nth-child(n)	p:nth-child(2)	Selects every <p> element that is the second child of its parent
:nth-last-child(n)	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
:nth-last-of-type(n)	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child
:nth-of-type(n)	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent
:only-of-type	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
:only-child	p:only-child	Selects every <p> element that is the only child of its parent
:optional	input:optional	Selects <input> elements with no “required” attribute
:out-of-range	input:out-of-range	Selects <input> elements with a value outside a specified range
:read-only	input:read-only	Selects <input> elements with a “readonly” attribute specified
:read-write	input:read-write	Selects <input> elements with no “readonly” attribute
:required	input:requited	Selects <input> elements with a “required” attribute specified
:root	root	Selects the document’s root element
:target	#news: target	Selects the current active #news element (clicked on an URL containing that anchor name)
:valid	input:valid	Selects all <input> elements with a valid valud

CSS3 Pseudo-elements (1/3)

Pseudo-elements

- Used to style specified parts of an element
- Examples
  - Style the first letter, or line, of an element
  - Insert content before, or after, the content of an element
- Syntax
  - selector::pseudo-element {
    property: value;
    }

```
<style>
    p:first-line {
    color: #ff0000;
    font-variant: small-caps; }
</style>
```

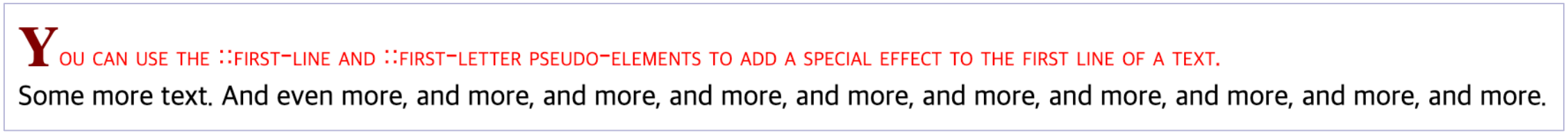
CSS3 Pseudo-elements (2/3)

Example

- *::first-line* and *::first-letter* pseudo-elements

```
<body>
      <p>
      You can use the :first-line and :first-letter pseudo-elements to add a
      special effect to the first line of a text. <br />
      Some more text. And even more, and more, and more, and more, and more,
      and more, and more, and more, and more, and more.
      </p>
</body>
```

```
<style>
      p::first-line {
            color: #ff0000;
            font-variant: small-caps;
      }
      p::first-letter {
            color: #800000;
            font: bold xx-large times;
      }
</style>
```



CSS3 Pseudo-elements (3/3)

Additional Pseudo-elements

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

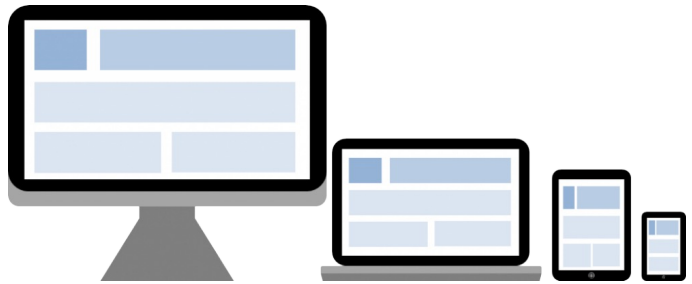
Miscellaneous

CSS3 Media Query (1/2)

Media Queries

- A popular technique for delivering a tailored style sheet to desktops, laptops, tablets, and mobile phones
- Media types

Value	Description
all	Used for all media type devices
print	Used for printers
screen	Used for computer screens
speech	Used for screenreaders that "reads" the page out loud



- Media query syntax
  - <link rel="stylesheet" media="*mediatype* and|not|only (*expressions*)" href="print.css">
  - @media not|only *mediatype* and (*expressions*) { CSS-Code; }

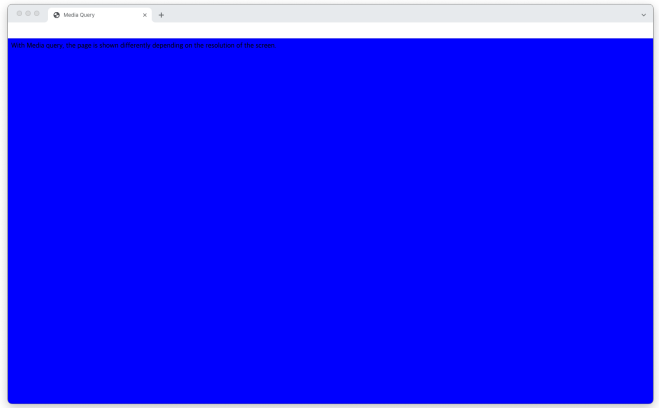
```
<link rel="stylesheet" media="screen and (max-width: 768px)" href="mystyle.css" />
```

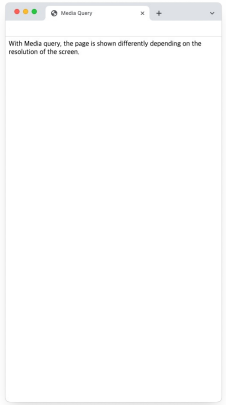
```
<style>
    @media screen and (max-width: 768px) { body { background-color: lightgreen; } }
</style>
```

CSS3 Media Query (2/2)

@media Rules

- With Media query, the page is shown differently depending on the resolution of the screen





```
<style>
      @media (min-width: 600px) {
            body {
                  background-color: green;
            }
      }

      @media (min-width: 768px) {
            body {
                  background-color: blue;
            }
      }
</style>
```

```
<body>
            With Media query, the page is shown differently
            depending on the resolution of the screen.
</body>
```

Which Styles to Apply? (1/3)

!important Rule

- Declarations marked as important carry more weight than unmarked ones

```
<!DOCTYPE html>
<html>
     <head>
          <title>Which Styles to Apply?</title>
          <style>
               p {
                    color: purple !important;
               }
               p {
                    color: blue;
               }
          </style>
     </head>
     <body>
          <p>Which Styles to Apply?</p>
     </body>
</html>
```

Which Styles to Apply?

Which Styles to Apply? (2/3)

ID and Class Selector

- An id is more specific than a class
- A class is more specific than an element

```
<!DOCTYPE html>
<html>
	<head>
		<title>Which Styles to Apply?</title>
		<style>
			h1#top {
				color: blue;
			}
			h1.top {
				color: red;
			}
			h1 {
				color: green;
			}
		</style>
	</head>
	<body>
		<h1 id="top">Which Styles to Apply?</h1>
		<h1 class="top">Which Styles to Apply?</h1>
		<h1>Which Styles to Apply?</h1>
	</body>
</html>
```

Which Styles to Apply?

Which Styles to Apply?

Which Styles to Apply?

Which Styles to Apply? (3/3)

Declaration Order

- If two declarations are otherwise equal, the later declaration applies

```
<!DOCTYPE html>
<html>
     <head>
          <title>Which Styles to Apply?</title>
          <style>
               h1 {
                    color: red;
               }
               h1 {
                    color: blue;
               }
          </style>
     </head>
     <body>
          <h1>Which Styles to Apply?</h1>
     </body>
</html>
```

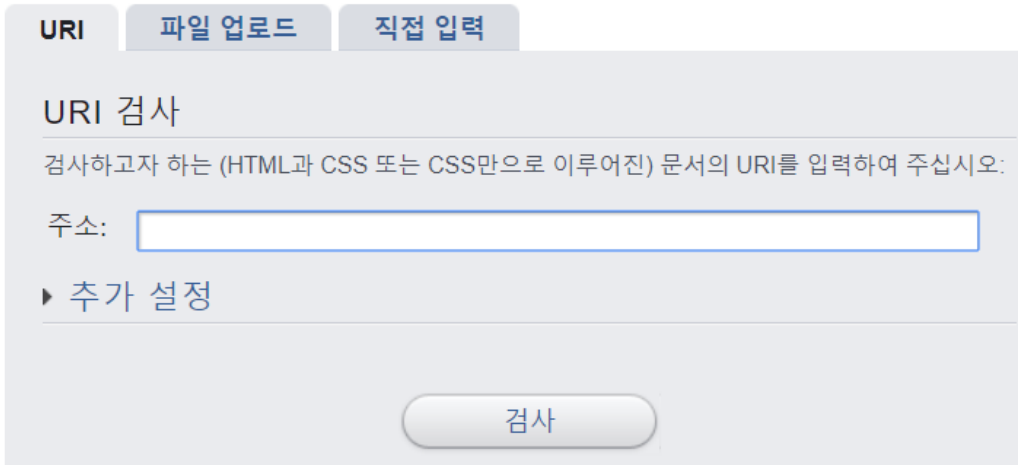
Which Styles to Apply?

CSS Validation

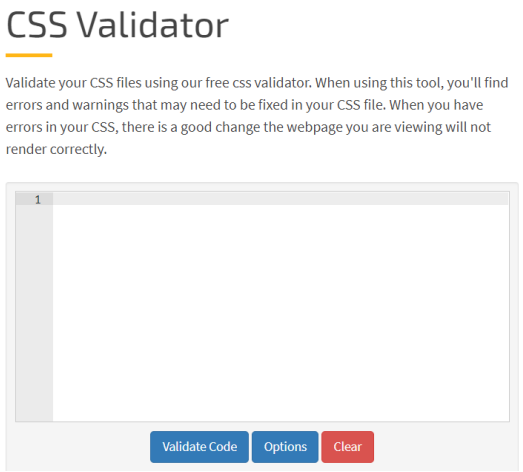
Validators

- Check the syntax of CSS to identify any violation of the language rules as defined by W3C

http://jigsaw.w3.org/css-validator/



http://www.cssportal.com/css-validator/



Exercise

Exercise 1

Result Example

My Home Page

Home
Education
Service
Publication

My Favorite Sites:

- School of Computing Home Page
- Naver
- Movie Database
- W3 Schools

Helpful Resources:

- HTML Reference
- CSS Reference

Last Modified on Apr. 08, 2022.
Managed by SoYeop

Exercise 1 (cont'd)

Requirements 1

- Use external CSS
  - 1 HTML file & 1 CSS file & other resources
- Change font style: "Times New Roman"
- Top section
  - Image and title are middle-aligned (use *vertical-align* property)
  - Title
    - Padding: left (10px)
    - Font: bold, xx-large, #003399

Exercise 1 (cont'd)

Requirements 2

- Middle section
  - Borders
    - top & bottom (2px solid gray)
  - Margin
    - top (10px), right (0px), bottom (10px), left (0px)
  - Left navigation
    - Background color: #CFCFFF
    - Links are not underlined
    - Unvisited links are navy, and mouse-over links are red
    - Margin: top (10px), right (20px), bottom (5px), left (0px)
- Footer section
  - Font: italic, small, right-aligned
  - Change the date and author name

End of Class