

# 화면 구현

part 2

## HTML5 글자 태그



KOREATECH  
온라인평생교육원



## 학습목표

- HTML5 기본 글자 태그를 알고 사용할 수 있다.
- HTML5 태그로 글자 형태를 자유롭게 적용할 수 있다.

## 학습내용

- 제목, 본문, 앱커, 글자 태그
- 제목, 본문, 앱커, 글자 태그 실습

# 제목, 본문, 앵커, 글자 태그

## 1. 제목 태그

- <hn> … </hn>: 제목을 표시하는 태그
  - h1 ~ h6까지 사용할 수 있으며 h1이 가장 큼



## 2. 본문 태그

- <p> … </p>: 단락을 만드는 태그
- <br>: 줄 바꾸기



# 제목, 본문, 앵커, 글자 태그

## 2. 본문 태그

- <hr>: 수평선을 긋는 태그
- <blockquote> … </blockquote>: 인용문을 넣는 태그

<blockquote cite = “인용 주소”>

cite 속성으로 출처를 남길 수 있음

### 입력

인용할 때<blockquote>활용할 수 있는 태그입니다.</blockquote>



### 결과

인용할 때

활용할 수 있는 태그입니다.

- <pre>: 글자, 글꼴, 줄 바꿈을 입력한 그대로 출력하는 태그

### 입력

<pre>글자, 글꼴  
줄 바꿈 그대로 출력</pre>



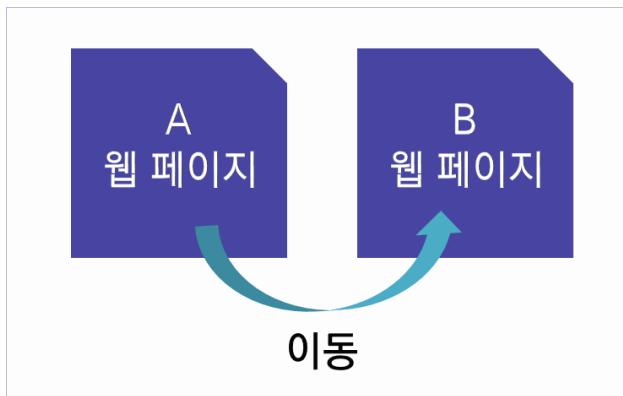
### 결과

글자, 글꼴  
줄 바꿈 그대로 출력

## 제목, 본문, 앵커, 글자 태그

### 3. 앵커(Anchor) 태그

- 서로 다른 웹 페이지 사이를 이동하거나 웹 페이지 내부에서 특정한 위치로 이동할 때 사용하는 태그

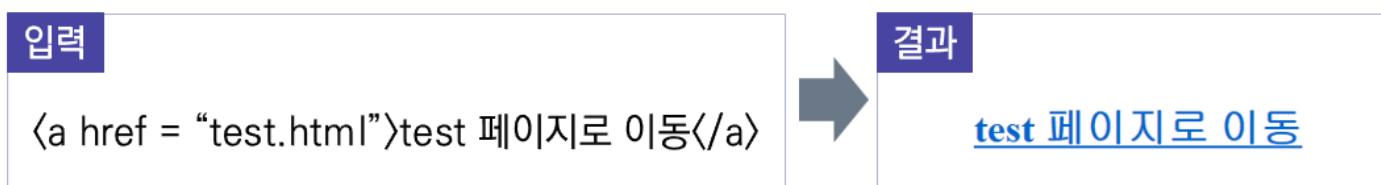


〈웹 페이지 간 이동〉



〈웹 페이지 내부 이동〉

- <a>…</a>로 사용
- a 태그만 사용하여 웹페이지 간 이동은 하지 않으며 href 속성을 함께 사용하면 이동을하게 됨
- <a href = “이동할 웹페이지 주소”>링크 텍스트</a>



## 제목, 본문, 앵커, 글자 태그

### 3. 앵커(Anchor) 태그

- 페이지 내부에서 이동하는 책갈피 기능을 함

```
<a href = “#point 1”>1번 단락</a><br />
<a href = “#point 2”>2번 단락</a><br />
```

```
<p><a id = “point 1”>1번 단락</a></p>
<p><a id = “point 2”>2번 단락</a></p>
```

Tip

HTML5 이전에는 name 속성을 이용했지만,  
HTML5에서는 폐기된 속성으로 id 속성을 이용함

## 제목, 본문, 앵커, 글자 태그

### 3. 앵커(Anchor) 태그

- target 속성: 지정된 href 주소를 보여줄 웹 브라우저의 위치를 지정함

`<a target = “_self”>`

- 속성을 지정하지 않는다면 기본 속성으로 쓰임
- 현재 보고 있는 페이지에서 href로 지정한 사이트로 이동

`<a target = “_blank”>`

클릭했을 때 새로운 창이나 새탭으로 보여줌

`<a target = “_top”>`

클릭했을 때 사이트 상단으로 이동

- download 속성: 지정된 href 주소를 보여주지 않고 파일로 다운받게 해줌

## 제목, 본문, 앵커, 글자 태그

### 4. 글자 태그

- <b>…</b>: 굵은 글자 태그
- <i>…</i>: 기울어진 글자 태그
- <small>…</small>: 작은 글자 태그
- <sub>…</sub>: 아래에 달라붙는 글자 태그
- <ins>…</ins>: 밑줄 글자 태그
- <del>…</del>: 가운뎃줄이 그어진 글자 태그

입력

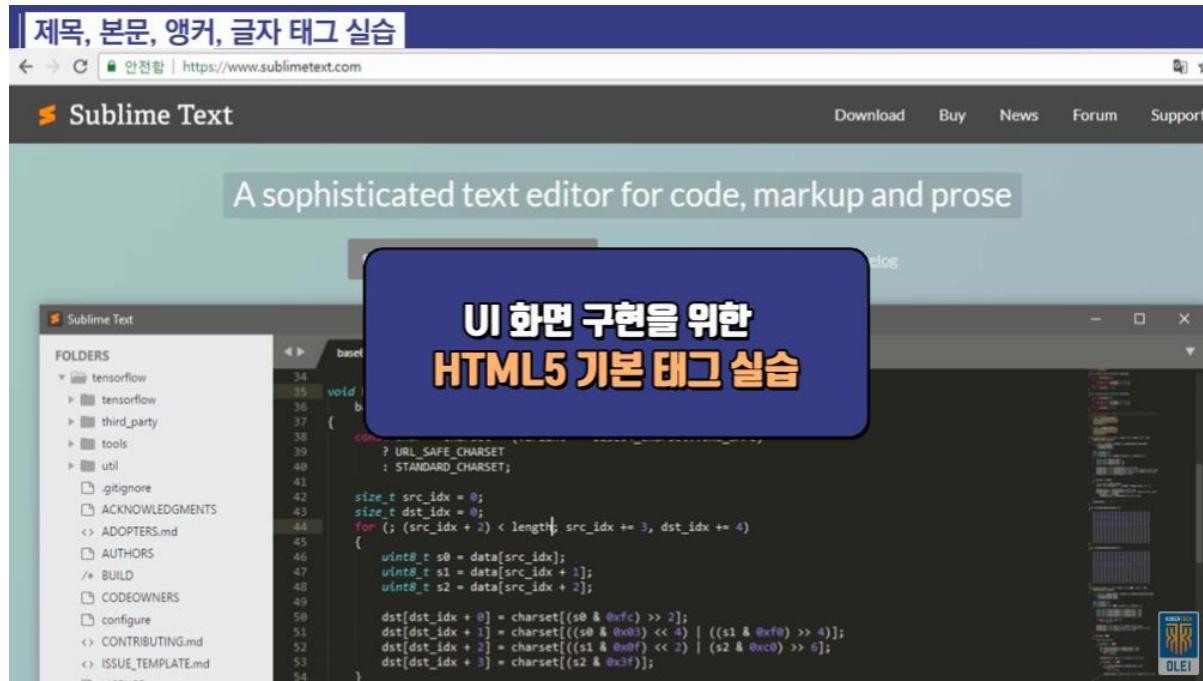
```
<b>굵은 글자 태그</b>
<i>기울어진 글자 태그</i>
<small>작은 글자 태그</small>
<sub>아래에 붙는 글자 태그</sub>
<ins>밑줄 글자 태그</ins>
<del>가운뎃줄 글자 태그</del>
```

결과

굵은 글자 태그  
 기울어진 글자 태그  
 작은 글자 태그  
 아래에 붙는 글자 태그  
밑줄 글자 태그  
~~가운뎃줄 글자 태그~~



# 제목, 본문, 앱커, 글자 태그 실습



※ 실습 영상은 콘텐츠에서 확인할 수 있습니다.

# 핵심정리

## 1. 제목, 본문, 앵커, 글자 태그

- <hn>…</hn>: 제목 태그
  - n은 1부터 6까지 숫자를 사용
  - h1 태그가 가장 큰 글씨 제목으로 지정하는 것이며, h6은 가장 작은 글씨 제목으로 지정
- <p>…</p>: 단락을 지정하는 태그
- <pre>…</pre>: HTML 상에서 사용자가 입력한 대로 출력하는 태그
- <blockquote>…</blockquote>: 인용문을 사용할 때 쓰는 태그
- 앵커 태그: 한 페이지 내의 책갈피 이동이나 다른 페이지로 이동하는 태그
  - href 속성을 함께 사용해야 화면을 이동할 수 있음
- 글자 형태를 표현하는 태그의 종류: <b>, <i>, <small>, <sub>, <ins>, <del>

# 화면 구현

part 2

HTML5 목록 및 테이블 태그



KOREATECH  
온라인평생교육원



## 학습목표

- 기본 목록 및 정의 목록 태그를 이용하여 화면을 구성할 수 있다.
- 테이블 태그를 이용하여 테이블을 생성할 수 있다.

## 학습내용

- 기본 목록 및 정의 목록 태그
- 테이블 태그
- 목록 및 테이블 태그 실습

# 기본 목록 및 정의 목록 태그

## 1. 기본 목록 태그

### Attributes

Attribute	Value	Description
<u>type</u>	1 A a I i disc square circle	<b>Not supported in HTML5.</b> Specifies which kind of bullet point will be used
<u>value</u>	number	Specifies the value of a list item. The following list items will increment from that number (only for <ol> lists)

목록을 만들 때  
**<ul>, <ol>, <dl>, <li> 태그 이용**

# 기본 목록 및 정의 목록 태그

## 1. 기본 목록 태그

### 1) <li> 태그



<li> 태그란?

- List Item의 약자로 ul, ol 태그의 내부에서 사용되는 태그
- 목록의 내용이 되는 실질적인 태그
- 들여쓰기 및 줄 바꿈 기능이 있음
- **type 속성**은 HTML5에서는 지원이 되지 않음 (HTML5는 기존의 모든 문서 타입을 래핑 하기 때문에 모든 유효한 요소들을 포함)
- **value 속성**은 HTML 4.01에서는 사용되지 않지만 HTML5에서는 지원
- **value 속성**은 목록의 번호를 해당 번호부터 증가 시킴(<ol> 태그를 이용할 때 해당)

## 기본 목록 및 정의 목록 태그

### 1. 기본 목록 태그

#### 1) <li> 태그

#### Syntax

```
<li type="1 | a | A | i | I | disc | circle | square">
```

〈 li 태그의 사용 문법 〉

#### Attribute Values

For ordered lists (<ol>):

Value	Description
1	Default. Numerical ordered list (1, 2, 3, 4)
a	Alphabetically ordered list, lowercase (a, b, c, d)
A	Alphabetically ordered list, uppercase (A, B, C, D)
i	Roman numbers, lowercase (i, ii, iii, iv)
I	Roman numbers, uppercase (I, II, III, IV)

〈 li 태그의 속성 설정 〉

## 기본 목록 및 정의 목록 태그

### 1. 기본 목록 태그

#### 1) <li> 태그

#### Attribute Values

For **unordered** lists (<ul>):

Value	Description
disc	Default. A filled circle
circle	An unfilled circle
square	A filled square

< li 태그의 속성 설정 >

## 기본 목록 및 정의 목록 태그

### 1. 기본 목록 태그

#### 1) <li> 태그

#### Syntax

```
<li value="number">
```

#### Attribute Values

Value	Description
<i>number</i>	Specifies the value of the list item

〈 Value 속성 정의 〉

# 기본 목록 및 정의 목록 태그

## 1. 기본 목록 태그

### 1) <li> 태그

#### 입력

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

<ol>
  <li type = "1">list item 1</li>
  <li type = "a">list item 2</li>
  <li type = "A">list item 3</li>
  <li type = "I">list item 4</li>
  <li type = "i">list item 5</li>
  <li type = "disc">list item disc</li>
  <li type = "square">list item square</li>
  <li type = "circle">list item circle</li>
  <li>list item 9</li>
  <li>list item 10</li>
  <li value = "100">list item 100</li>
  <li>list item 101</li>
  <li>list item 102</li>
</ol>

</body>
</html>
```



## 기본 목록 및 정의 목록 태그

### 1. 기본 목록 태그

#### 1) <li> 태그

결과

- 1. list item 1
- b. list item 2
- C. list item 3
- IV. list item 4
- v. list item 5
  - list item disc
  - list item square
  - list item circle
- 9. list item 9
- 10. list item 10
- 100. list item 100
- 101. list item 101
- 102. list item 102

## 기본 목록 및 정의 목록 태그

### 1. 기본 목록 태그

#### 2) <ul> 태그

- Unordered List의 약자로서 순서가 없는 태그 목록에 사용
- <li> 태그와 함께 사용됨
- compact, type 속성은 HTML5에서는 지원 안됨

## Attributes

Attribute	Value	Description
<u>compact</u>	compact	<p><b>Not supported in HTML5.</b></p> <p>Specifies that the list should render smaller than normal</p>
<u>type</u>	disc square circle	<p><b>Not supported in HTML5.</b></p> <p>Specifies the kind of marker to use in the list</p>

〈 Value 속성 정의 〉

## 기본 목록 및 정의 목록 태그

### 1. 기본 목록 태그

#### 2) <ul> 태그

#### Syntax

```
<ul type="disc|circle|square">
```

#### Attribute Values

Value	Description
disc	Default. A filled circle
circle	An unfilled circle
square	A filled square

〈 Value 속성 정의 〉

# 기본 목록 및 정의 목록 태그

## 1. 기본 목록 태그

### 2) <ul> 태그

입력

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

<h3>Type Attribute</h3>
<ul type="square">
    <li>Coffee</li>
    <li>Tea</li>
    <li>Milk</li>
</ul>

<ul type="circle">
    <li>Coffee</li>
    <li>Tea</li>
    <li>Milk</li>
</ul>

<h3>Compact Attribute</h3>

<ul compact>
    <li>Coffee</li>
    <li>Tea</li>
    <li>Milk</li>
</ul>

<p>The type, compact attribute is not
supported in HTML5. Use CSS instead.</p>

</body>
</html>
```



## 기본 목록 및 정의 목록 태그

### 1. 기본 목록 태그

#### 2) <ul> 태그

#### 결과

#### Type Attribute

- Coffee
  - Tea
  - Milk
- 
- Coffee
  - Tea
  - Milk

#### Compact Attribute

- Coffee
- Tea
- Milk

The type, compact attribute is not supported in HTML5. Use CSS instead.

## 기본 목록 및 정의 목록 태그

### 1. 기본 목록 태그

#### 3) <ol> 태그

- Ordered List의 약자로서 순서가 있는 목록을 표현할 때 사용
- 순서가 지정된 목록은 숫자 또는 알파벳 순으로 표시 가능
- start, type 속성은 HTML4.01에서는 사용되지 않지만, HTML5에서는 지원됨
- reversed 속성은 HTML5에 추가됨
- compact 속성은 더 이상 지원되지 않음

## 기본 목록 및 정의 목록 태그

### 1. 기본 목록 태그

#### 3) <ol> 태그

## Attributes

 = New in HTML5.

Attribute	Value	Description
<u>compact</u>	compact	<b>Not supported in HTML5.</b> Specifies that the list should render smaller than normal
<u>reversed</u>	 reversed	Specifies that the list order should be descending (9,8,7...)
<u>start</u>	number	Specifies the start value of an ordered list
<u>type</u>	1 A a I i	Specifies the kind of marker to use in the list

⟨ ol 태그의 속성 ⟩

## 기본 목록 및 정의 목록 태그

### 1. 기본 목록 태그

#### 3) <ol> 태그

#### Syntax

```
<ol reversed>
```

〈 Reversed 속성 정의 〉

#### Syntax

```
<ol start="number">
```

#### Attribute Values

Value	Description
number	Specifies the start value of the first list item in the ordered list

〈 Start 속성 정의 〉

## 기본 목록 및 정의 목록 태그

### 1. 기본 목록 태그

#### 3) <ol> 태그

#### Syntax

```
<ol type="1 | a | A | i | I">
```

#### Attribute Values

Value	Description
1	Default. Decimal numbers (1, 2, 3, 4)
a	Alphabetically ordered list, lowercase (a, b, c, d)
A	Alphabetically ordered list, uppercase (A, B, C, D)
i	Roman numbers, lowercase (i, ii, iii, iv)
I	Roman numbers, uppercase (I, II, III, IV)

〈 Type 속성 정의 〉

# 기본 목록 및 정의 목록 태그

## 1. 기본 목록 태그

### 3) <ol> 태그

입력

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

<h3>reversed Attribute</h3>
<ol reversed>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

<p><strong>Note:</strong> The reversed attribute of the ol tag
is not supported in Internet Explorer and Safari 5.1 (and
earlier versions).</p>

<h3>start Attribute</h3>
<ol start="50">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

<h3>type Attribute</h3>
<ol type="I" start="50">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

<ol type="I">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

</body>
</html>
```



## 기본 목록 및 정의 목록 태그

### 1. 기본 목록 태그

#### 3) <ol> 태그

##### 결과

###### reversed Attribute

- 3. Coffee
- 2. Tea
- 1. Milk

**Note:** The reversed attribute of the ol tag is not supported in Internet and Safari 5.1 (and earlier versions).

###### start Attribute

- 50. Coffee
- 51. Tea
- 52. Milk

###### type Attribute

- L. Coffee
- LI. Tea
- LII. Milk

- I. Coffee
- II. Tea
- III. Milk

## 기본 목록 및 정의 목록 태그

### 2. 정의 목록 태그

#### 1) <dl> 태그



<dl> 태그란?

- Definition List의 약자로 정의 목록을 만드는 데 사용
- <dt> <dd> 태그와 함께 사용

#### 2) <dt> 태그



<dt> 태그란?

- Definition Term의 약자로 용어의 제목을 나타내는 태그
- 줄 바꿈 기능이 있음

## 기본 목록 및 정의 목록 태그

### 2. 정의 목록 태그

#### 3) <dd> 태그



<dd> 태그란?

- Definition Description의 약자로 용어에 대한 설명을 나타내는 태그
- 줄 바꿈 기능이 있음

## 기본 목록 및 정의 목록 태그

### 2. 정의 목록 태그

#### 3) <dd> 태그

##### 입력

```
<!DOCTYPE html>
<html>
<head>
<style>
dd {
    display: block;
    margin-left: 20px;
}
</style>
</head>
<body>

<p>A dd element is displayed like this:</p>

<dl>
    <dt>Coffee</dt>
    <dd>Black hot drink</dd>
    <dt>Milk</dt>
    <dd>White cold drink</dd>
</dl>

<p>Change the default CSS settings to see the effect.</p>

</body>
</html>
```



## 기본 목록 및 정의 목록 태그

### 2. 정의 목록 태그

#### 3) <dd> 태그

#### 결과

A dd element is displayed like this:

Coffee

    Black hot drink

Milk

    White cold drink

Change the default CSS settings to see the effect.

# 테이블 태그

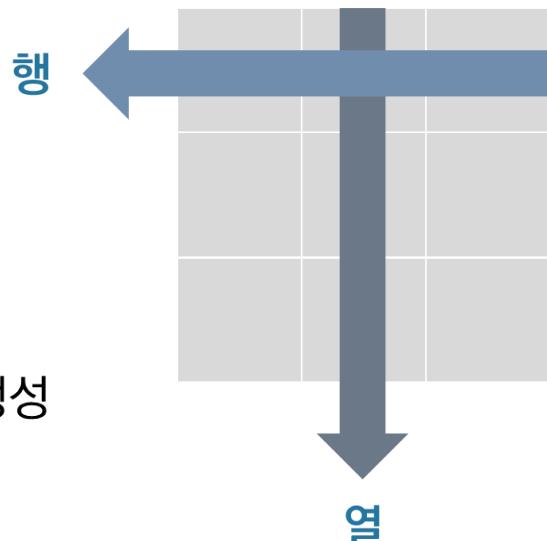
## 1. 테이블 태그 구조

`<table>``<caption>``<colgroup>``<tr>``<th>``<td>`

최근, 표를 만들 때만 쓰이는 추세

### 1) 테이블 태그 용어 정리

- 표: 행(가로)과 열(세로)로 이루어짐
- 행: `<tr>`태그(table row)로 생성
- 열: `<td>`태그(table data)로 생성
- 행의 타이틀: `<th>`태그(table header)로 생성
- `<caption>` 태그: 표 제목을 생성함



# 테이블 태그

## 1. 테이블 태그 구조

### 2) 테이블 태그 종류

- <table>: 표 선언
- <caption>: 표 제목 선언
- <tr>: 표 행 선언
- <th>: 표 해더 선언, 해더는 굵은 글씨로 표현
- <td>: 표 데이터 선언

# 테이블 태그

## 1. 테이블 태그 구조

### 2) 테이블 태그 종류

입력

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
    border: 1px solid black;
}
</style>
</head>
<body>

<table>
    <caption>type1</caption>
    <tr>
        <th>No.</th>
        <th>Name</th>
        <th>age</th>
    </tr>
    <tr>
        <td>1</td>
        <td>kim</td>
        <td>30</td>
    </tr>
    <tr>
        <td>2</td>
        <td>Park</td>
    </tr>

```

```

        <td>27</td>
    </tr>
</table>
<table>
    <caption>type2</caption>
    <tr>
        <th>No.</th>
        <td>1</td>
        <td>2</td>
    </tr>
    <tr>
        <th>Name</th>
        <td>kim</td>
        <td>Park</td>
    </tr>
    <tr>
        <th>age</th>
        <td>30</td>
        <td>27</td>
    </tr>
</table>
</body>
</html>
```



# 테이블 태그

## 1. 테이블 태그 구조

### 2) 테이블 태그 종류

결과

type1

No.	Name	age
1	kim	30
2	Park	27

type2

No.	1	2
Name	kim	Park
age	30	27

## 테이블 태그

### 1. 테이블 태그 구조

#### 2) 테이블 태그 종류

- <colgroup>: 표의 서식 지정을 위하여 하나 이상의 열을 그룹화
- <thead>: 표의 머리글 내용을 그룹화하는데 사용
- <tfoot>: 표의 꼬리말 내용을 그룹화하는데 사용
- <tbody>: 표의 본문 내용을 그룹화하는데 사용

# 테이블 태그

## 1. 테이블 태그 구조

### 2) 테이블 태그 종류

#### 입력

```
<!DOCTYPE html>
<html>
<head>
<style>
thead {color:green;}
tbody {color:blue;}
tfoot {color:red;}

table, th, td {
    border: 1px solid black;
}
</style>
</head>

<body>

<table>
    <caption>Table title</caption>
    <colgroup>
        <col span = "0" style ="background:yellow"/>
    </colgroup>
    <thead style ="background:red">
        <tr>
            <th>No.</th>
            <th>Name</th>
            <th>Age</th>
```

# 테이블 태그

## 1. 테이블 태그 구조

### 2) 테이블 태그 종류

```
</tr>
</thead>
<tfoot style = "background:blue">
<tr>
<th>total</th>
<td colspan = "2">3 person</td>
</tr>
</tfoot>
<tbody>
<tr>
<td>1</td>
<td>Kim</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>Park</td>
<td>27</td>
</tr>
<tr>
<td>3</td>
<td>Lee</td>
<td>14</td>
</tr>
</tbody>
</table>

</body>
</html>
```



# 테이블 태그

## 1. 테이블 태그 구조

### 2) 테이블 태그 종류

결과

Table title

No.	Name	Age
1	Kim	30
2	Park	27
3	Lee	14
total		3 person

# 테이블 태그

## 2. 테이블 태그 속성

### 1) <table> 태그

- HTML5에서는 지원하지 않음
- CSS에서 속성을 지정

## Attributes

Attribute	Value	Description
<u>align</u>	left center right	Not supported in HTML5. Specifies the alignment of a table according to surrounding text
<u>bgcolor</u>	rgb(x,x,x) #xxxxxx colorname	Not supported in HTML5. Specifies the background color for a table
<u>border</u>	1 0	Not supported in HTML5. Specifies whether or not the table is being used for layout purposes
<u>cellpadding</u>	<i>pixels</i>	Not supported in HTML5. Specifies the space between the cell wall and the cell content
<u>cellspacing</u>	<i>pixels</i>	Not supported in HTML5. Specifies the space between cells

〈 table 태그의 속성 정의 〉

# 테이블 태그

## 2. 테이블 태그 속성

### 1) <table> 태그

<u>frame</u>	void above below hsides lhs rhs vsides box border	<b>Not supported in HTML5.</b> Specifies which parts of the outside borders that should be visible
<u>rules</u>	none groups rows cols all	<b>Not supported in HTML5.</b> Specifies which parts of the inside borders that should be visible
<u>summary</u>	text	<b>Not supported in HTML5.</b> Specifies a summary of the content of a table
<u>width</u>	pixels %	<b>Not supported in HTML5.</b> Specifies the width of a table

〈 table 태그의 속성 정의 〉

## 테이블 태그

### 2. 테이블 태그 속성

#### 1) <table> 태그

#### Syntax

```
<table align="left|right|center">
```

#### Attribute Values

Value	Description
left	Left-aligns the table
right	Right-aligns the table
center	Center-aligns the table

〈 table 태그의 align 속성 〉

## 테이블 태그

### 2. 테이블 태그 속성

#### 1) <table> 태그

#### Syntax

```
<table border="1|0">
```

#### Attribute Values

Value	Description
0	No borders around the table cells
1	Add borders around the table cells

⟨ table 태그의 border 속성 ⟩

## 테이블 태그

### 2. 테이블 태그 속성

#### 2) <td>, <tr>, <th> 태그

- rowspan: 셀이 확장되어야 하는 행의 수를 지정
- colspan: 셀이 확장되어야 하는 열의 수를 지정

### Syntax

```
<td rowspan="number">
```

### Attribute Values

#### Value   Description

number Specifies the number of rows a cell should span. **Note:** rowspan="0" tells the browser to span the cell to the last row of the table section (thead, tbody, or tfoot)

#### 〈 table 태그의 border 속성 〉

### Syntax

```
<th colspan="number">
```

### Attribute Values

#### Value   Description

number Sets the number of columns a header cell should span. **Note:** colspan="0" tells the browser to span the cell to the last column of the column group (colgroup)

#### 〈 th 태그의 colspan 속성 〉

# 테이블 태그

## 2. 테이블 태그 속성

### 2) <td>, <tr>, <th> 태그

#### 입력

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

<table border = "1">
  <tr>
    <th colspan="3">Header title</th>
  </tr>
  <tr>
    <td>data1</td>
    <td colspan="2">date2</td>
  </tr>
  <tr>
    <td rowspan="0">data3</td>
    <td>data4</td>
    <td>date5</td>
  </tr>
  <tr>
    <td>date6</td>
    <td>date7</td>
  </tr>
</table>

</body>
</html>
```



## 테이블 태그

### 2. 테이블 태그 속성

#### 2) <td>, <tr>, <th> 태그

결과

Header title		
data1	date2	
data3	data4	date5
	date6	date7

# 목록 및 테이블 태그 실습

The screenshot shows a dual-pane interface. On the left, a code editor displays the following HTML code:

```
1 <html>
2 <body>
3     <h3>Type Attribute</h3>
4     <ul type="square">
5         <li>
6     </ul>
7 </body>
8 </html>
```

The right pane shows a browser preview of the same code, displaying an H3 heading "Type Attribute" followed by an unordered list with square bullet points.

**ul 태그 활용**

데이터의 목록을 각각 하나씩 만들 때 사용

DLEI

※ 실습 영상은 콘텐츠에서 확인할 수 있습니다.

# 핵심정리

## 1. 기본 목록 및 정의 목록 태그

- 기본 목록을 생성할 때는 `<ul>`, `<ol>`, `<li>` 태그를 이용하여 생성
  - ✓ `<li>` 태그는 실질적으로 목록의 내용이 되는 태그
  - ✓ `<ul>` 태그는 순서가 없는 목록을 만들 때 사용
  - ✓ `<ol>` 태그는 순서가 있는 목록을 만들 때 사용하며 start 속성을 이용하여 시작하는 번호 지정 가능
- 정의 목록을 생성할 때는 `<dl>`, `<dt>`, `<dd>` 태그를 이용하여 생성
  - ✓ `<dl>` 태그는 정의 목록을 선언할 때 사용
  - ✓ `<dt>` 태그는 용어의 제목
  - ✓ `<dd>` 태그는 용어에 대한 설명을 나타내는 태그

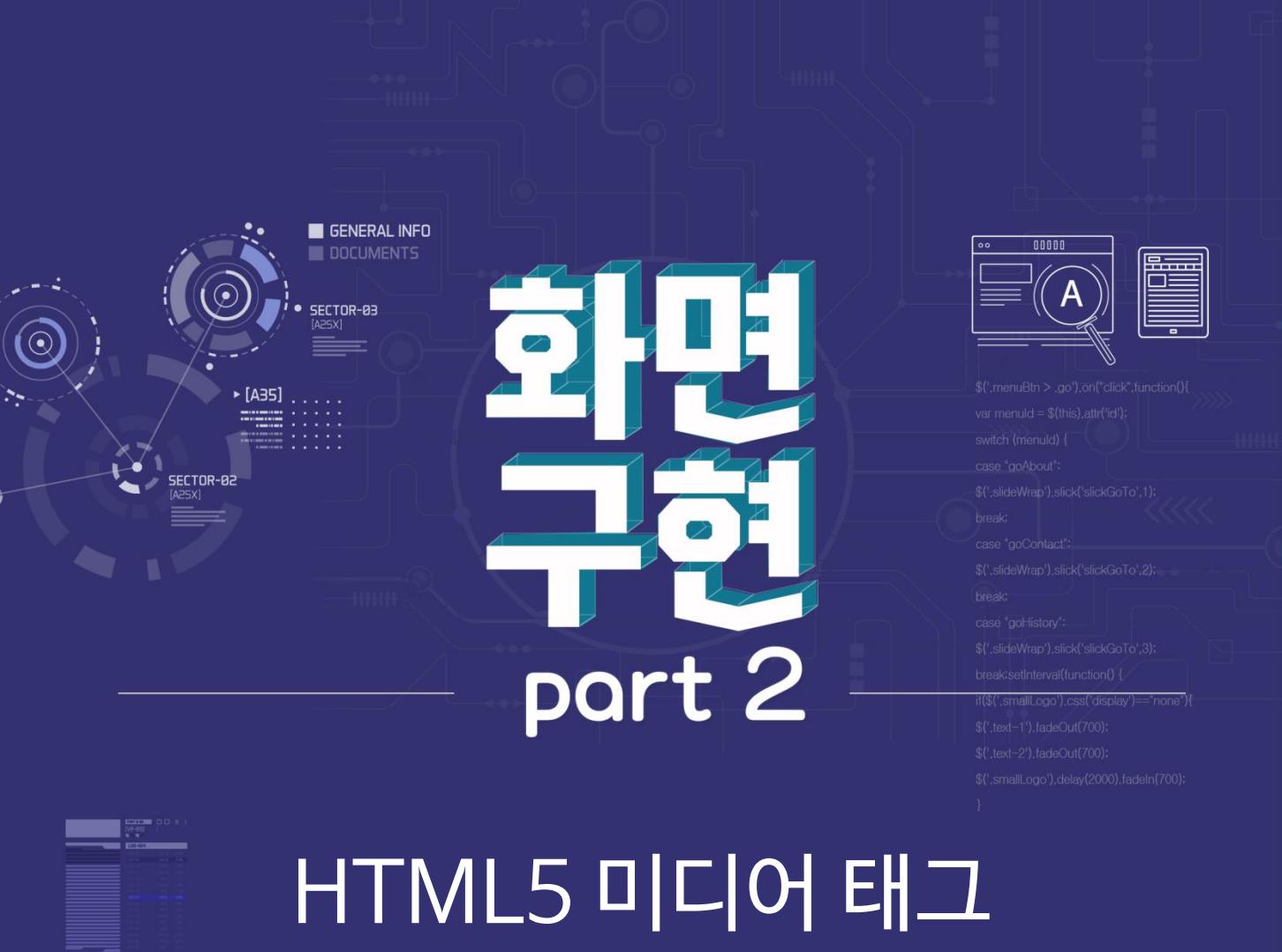
## 2. 테이블 태그

- 테이블 태그는 이전에는 전체 화면을 테이블로 구조화 하는데 사용되었지만 최근에는 실제 표를 만들 때만 주로 쓰임
- `<caption>` 태그는 표의 제목을 나타낼 수 있는 태그
- `<th>`는 테이블 헤더를 나타내고 `<tr>`은 행을 의미하고 `<td>`는 테이블의 데이터를 생성함
- 테이블의 속성들은 HTML5에서는 지원하지 않고 CSS 속성에서 따로 선언함

# 화면 구현

part 2

## HTML5 미디어 태그



KOREATECH  
온라인평생교육원



## 학습목표

- 이미지 태그를 이해하고, 활용할 수 있다.
- 오디오 태그를 이해하고, 활용할 수 있다.
- 비디오 태그를 이해하고, 활용할 수 있다.

## 학습내용

- 이미지 태그
- 오디오 태그
- 비디오 태그

# 이미지 태그

## 1. 이미지 태그 개요

### 1) 정의 및 구조

- HTML 페이지의 이미지를 정의
- <img> 태그는 src와 alt의 두 가지 필수 속성을 가짐
- HTML 페이지에 이미지가 삽입되지 않으면, 이미지는 HTML 페이지에 링크가 되며 참조됨
- 참조된 이미지를 위한 공간을 생성함
- 선언

```
<img src = “imgFile.jpg” alt = “Image is not exist.”/>
```

### 2) 속성

- align, border, hspace, vspace 속성  
: HTML5에서 더 이상 지원하지 않음
- src: 이미지의 경로를 지정
- alt: 이미지가 없을 때 나오는 글자 지정
- width, height  
: 이미지의 너비 높이를 픽셀 단위로 지정

# 이미지 태그

## 1. 이미지 태그 개요

### 2) 속성

## Attributes

 = New in HTML5.

Attribute	Value	Description
<u>align</u>	top bottom middle left right	Not supported in HTML5. Specifies the alignment of an image according to surrounding elements
<u>alt</u>	<i>text</i>	Specifies an alternate text for an image
<u>border</u>	<i>pixels</i>	Not supported in HTML5. Specifies the width of the border around an image
<u>crossorigin</u>	 anonymous use-credentials	Allow images from third-party sites that allow cross-origin access to be used with canvas
<u>height</u>	<i>pixels</i>	Specifies the height of an image

〈 이미지 태그 속성 〉

# 이미지 태그

## 1. 이미지 태그 개요

### 2) 속성

hspace

*pixels*

Not supported in HTML5.

Specifies the whitespace on left and right side of an image

ismap

*ismap*

Specifies an image as a server-side image-map

longdesc

*URL*

Specifies a URL to a detailed description of an image

sizes

Specifies image sizes for different page layouts

src

*URL*

Specifies the URL of an image

srcset

*URL*

Specifies the URL of the image to use in different situations

usemap

*#mapname*

Specifies an image as a client-side image-map

vspace

*pixels*

Not supported in HTML5.

Specifies the whitespace on top and bottom of an image

width

*pixels*

Specifies the width of an image

〈 이미지 태그 속성 〉

# 이미지 태그

## 2. 이미지 태그 활용 예시

### 1) alt 속성 활용

#### Syntax

```
<img alt="text">
```

#### Attribute Values

Value	Description
-------	-------------

text	Specifies an alternate text for an image.
------	---

Guidelines for the alt text:

- The text should describe the image if the image contains information
- The text should explain where the link goes if the image is inside an  element
- Use alt="" if the image is only for decoration

〈 이미지 태그의 alt 속성 〉

# 이미지 태그

## 2. 이미지 태그 활용 예시

### 1) alt 속성 활용

입력

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



</body>
</html>
```



결과



# 이미지 태그

## 2. 이미지 태그 활용 예시

### 1) alt 속성 활용

입력

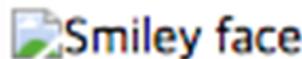
```
<!DOCTYPE html>
<html>
<body>



</body>
</html>
```



결과



# 이미지 태그

## 2. 이미지 태그 활용 예시

### 2) width, height 속성 활용

#### Syntax

```
<img width="pixels">
```

#### Attribute Values

Value	Description
<i>pixels</i>	The <b>width</b> in pixels (e.g. <code>width="100"</code> )

〈 이미지 태그의 width 속성 〉

# 이미지 태그

## 2. 이미지 태그 활용 예시

### 2) width, height 속성 활용

#### 입력

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



</body>
</html>
```



#### 결과



# 이미지 태그

## 2. 이미지 태그 활용 예시

### 3) 기타([placeholder.it](#) - 더미 이미지 제공)

[placeholder.it](#)에서는 이미지가 없는 경우, 준비되지 않은 경우

이미지의 크기에 맞는 이미지를 제공해 줌

#### 입력

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




</body>
</html>
```



#### 결과

300 x 200

200 x 100

# 오디오 태그

## 1. 오디오 태그 개요

### 1) 정의 및 구조

- 음악 또는 기타 오디오 스트림과 같은 사운드를 정의
- MP3, WAV, OGG 3가지 형식의 파일 지원



### 브라우저별 오디오 파일 지원 현황?

Browser	MP3	WAV	OGG
Internet Explorer	YES	NO	NO
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	YES	NO
Opera	YES	YES	YES

- 오디오 포맷별 MIME 타입
  - : MP3(audio/mpeg), OGG(audio/ogg), wav(audio/wav)

# 오디오 태그

## 1. 오디오 태그 개요

### 1) 정의 및 구조

- 선언

- XHTML5

```
<audio src = "audioFile.mp3" controls = "controls"></audio>
```

- HTML5

```
<audio controls>
```

```
  <source src = "audioFile.mp3" type = "audio/mpeg">
```

```
  </audio>
```

### 2) 속성

- autoplay: 음악을 자동으로 재생할지 지정
- controls: 음악 재생 도구를 출력할지 지정
- loop: 음악을 반복할지 지정
- muted: 음소거
- preload: 음악을 재생하기 전에 모두 불러올지 지정
- src: 음악 파일의 경로 지정

## 오디오 태그

### 1. 오디오 태그 개요

#### 2) 속성

## Attributes

Attribute	Value	Description
<u>autoplay</u>	autoplay	Specifies that the audio will start playing as soon as it is ready
<u>controls</u>	controls	Specifies that audio controls should be displayed (such as a play/pause button etc)
<u>loop</u>	loop	Specifies that the audio will start over again, every time it is finished
<u>muted</u>	muted	Specifies that the audio output should be muted
<u>preload</u>	auto metadata none	Specifies if and how the author thinks the audio should be loaded when the page loads
<u>src</u>	URL	Specifies the URL of the audio file

〈 오디오 태그의 속성 〉

## 오디오 태그

### 2. 오디오 태그 활용 예시

#### 1) autoplay 활용

- <source>

- video, audio 및 picture와 같은 미디어 요소에 대한 여러 미디어 리소스를 지정하는 데 사용
- 미디어 유형, 코덱 지원 또는 미디어 큐리를 기반으로 브라우저에서 선택할 수 있는 대체 미디어 파일 지정이 가능함

## Syntax

```
<audio autoplay>
```

입력

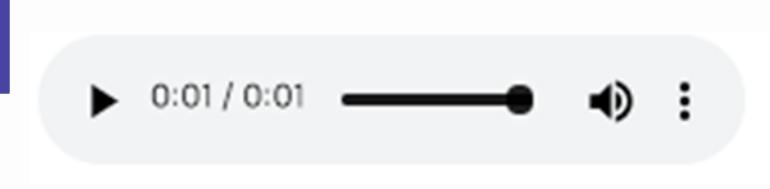
```
<!DOCTYPE html>
<html>
<body>

<audio controls autoplay>
  <source src="horse.ogg" type="audio/ogg">
</audio>

</body>
</html>
```



결과



▶ 0:01 / 0:01 ━━━━ 🔍 ⏰

## 오디오 태그

### 2. 오디오 태그 활용 예시

#### 2) loop 활용

- loop, autoplay: 속성을 사용하여 자동으로 계속 실행되도록 설정

## Syntax

```
<audio loop>
```

### 입력

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

<audio controls loop autoplay>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
</audio>

</body>
</html>
```



### 결과

II 0:01 / 0:01 ━━ ● ⚡ ⋮

## 오디오 태그

### 2. 오디오 태그 활용 예시

#### 3) XHTML5, HTML5 오디오 속성 태그 표시 방법

```
<h3> HTML5 format</h3>
<audio controls autoplay>
  <source src="horse.mp3" type="audio/mpeg">
</audio>

<h3> XHTML5 format</h3>
<audio src = "hores.mp3" contorls="contorls" autoplay = "autoplay"></audio>
```

# 비디오 태그

## 1. 비디오 태그 개요

### 1) 정의 및 구조

- 무비 클립이나 다른 비디오 스트림과 같은 비디오를 지정
- 비디오 타입은 MP4, WebM 및 Ogg와 같이 3가지 형식을 지원



### 브라우저별 비디오 파일 지원 현황?

Browser	MP4	WebM	Ogg
Internet Explorer	YES	NO	NO
Chrome	YES	YES	YES
Firefox	YES from Firefox 21 from Firefox 30 for Linux	YES	YES
Safari	YES	NO	NO
Opera	YES From Opera 25	YES	YES

# 비디오 태그

## 1. 비디오 태그 개요

### 1) 정의 및 구조

- 비디오 포맷별 MIME 타입
  - : MP4(video/mp4), WebM(video/webm), Ogg(video/ogg)

#### ▪ 선언

- XHTML5

```
<video src = “videoFile.mp4” controls = “controls”></video>
```

- HTML5

```
<video controls>
```

```
  <source src = “videoFile.mp4” type = “video/mp4”>
```

```
</video>
```

## 비디오 태그

### 1. 비디오 태그 개요

#### 2) 속성

- autoplay: 비디오를 자동으로 재생할지 지정
- controls: 비디오 재생 도구를 출력할지 지정
- width, height: 비디오의 너비, 높이 지정
- loop: 비디오를 반복할지 지정
- muted: 음소거
- poster: 비디오를 준비 중일 때의 이미지 파일 경로 지정
- preload : 비디오를 재생하기 전에 모두 불러올지 지정
- src : 비디오 파일의 경로 지정

# 비디오 태그

## 1. 비디오 태그 개요

### 2) 속성

## Optional Attributes

Attribute	Value	Description
<u>autoplay</u>	autoplay	Specifies that the video will start playing as soon as it is ready
<u>controls</u>	controls	Specifies that video controls should be displayed (such as a play/pause button etc).
<u>height</u>	<i>pixels</i>	Sets the height of the video player
<u>loop</u>	loop	Specifies that the video will start over again, every time it is finished
<u>muted</u>	muted	Specifies that the audio output of the video should be muted
<u>poster</u>	<i>URL</i>	Specifies an image to be shown while the video is downloading, or until the user hits the play button
<u>preload</u>	auto metadata none	Specifies if and how the author thinks the video should be loaded when the page loads
<u>src</u>	<i>URL</i>	Specifies the URL of the video file
<u>width</u>	<i>pixels</i>	Sets the width of the video player

〈 비디오 태그의 속성 〉

## 비디오 태그

### 2. 비디오 태그 활용 예시

#### 1) autoplay 활용

#### Syntax

```
<video autoplay>
```

입력

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

<video width="320" height="240" controls autoplay>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
</video>

</body>
</html>
```



결과



## 비디오 태그

### 2. 비디오 태그 활용 예시

#### 2) poster 활용

## Syntax

```
<video poster="URL">
```

## Attribute Values

### Value   Description

*URL*   Specifies the URL of the image file.

Possible values:

- An absolute URL - points to another web site (like href="http://www.example.com/poster.jpg")
- A relative URL - points to a file within a web site (like href="poster.jpg")

〈 비디오 태그의 poster 속성 〉

# 비디오 태그

## 2. 비디오 태그 활용 예시

### 2) poster 활용

#### 입력

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

<video width="400" height="300" poster="http://placehold.it/400x300" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
</video>

</body>
</html>
```



#### 결과



## 비디오 태그

### 2. 비디오 태그 활용 예시

#### 3) preload 활용

#### Syntax

```
<video preload="auto|metadata|none">
```

#### Attribute Values

Value	Description
auto	The author thinks that the browser should load the entire video when the page loads
metadata	The author thinks that the browser should load only metadata when the page loads
none	The author thinks that the browser should NOT load the video when the page loads

〈 비디오 태그의 preload 속성 〉

# 비디오 태그

## 2. 비디오 태그 활용 예시

### 3) preload 활용

#### 입력

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

<video width="320" height="240" controls preload="auto">
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
</video>

</body>
</html>
```



#### 결과



# 핵심정리

## 1. 이미지 태그

- <img> 태그는 src, alt 두 가지 필수 속성을 가짐
- HTML 페이지에 이미지가 삽입되지 않으면 이미지는 HTML 페이지에 링크가 되며 참조가 됨
- 기존 HTML4.01 가지는 align, border, hspace, vspace 속성은 지원됐지만 HTML5에서는 정식 지원은 하지 않음
- width, height 속성은 이미지의 너비와 높이는 픽셀 단위로 지정함

## 2. 오디오 태그

- <audio> 태그는 음악 또는 기타 오디오 스트림과 같은 사운드를 정의함
- mp3, wav, ogg 3가지 형식의 파일 포맷을 지원하지만, 브라우저마다 지원되지 않는 형식도 있음
- <source> 태그는 HTML5 선언 포맷에서 사용되며, 같은 미디어 요소에 대한 여러 미디어 리소스(파일형식)를 지정하는데 사용하고, 브라우저에서 선택할 수 있는 대체 미디어 파일을 지정할 수 있음

## 핵심정리

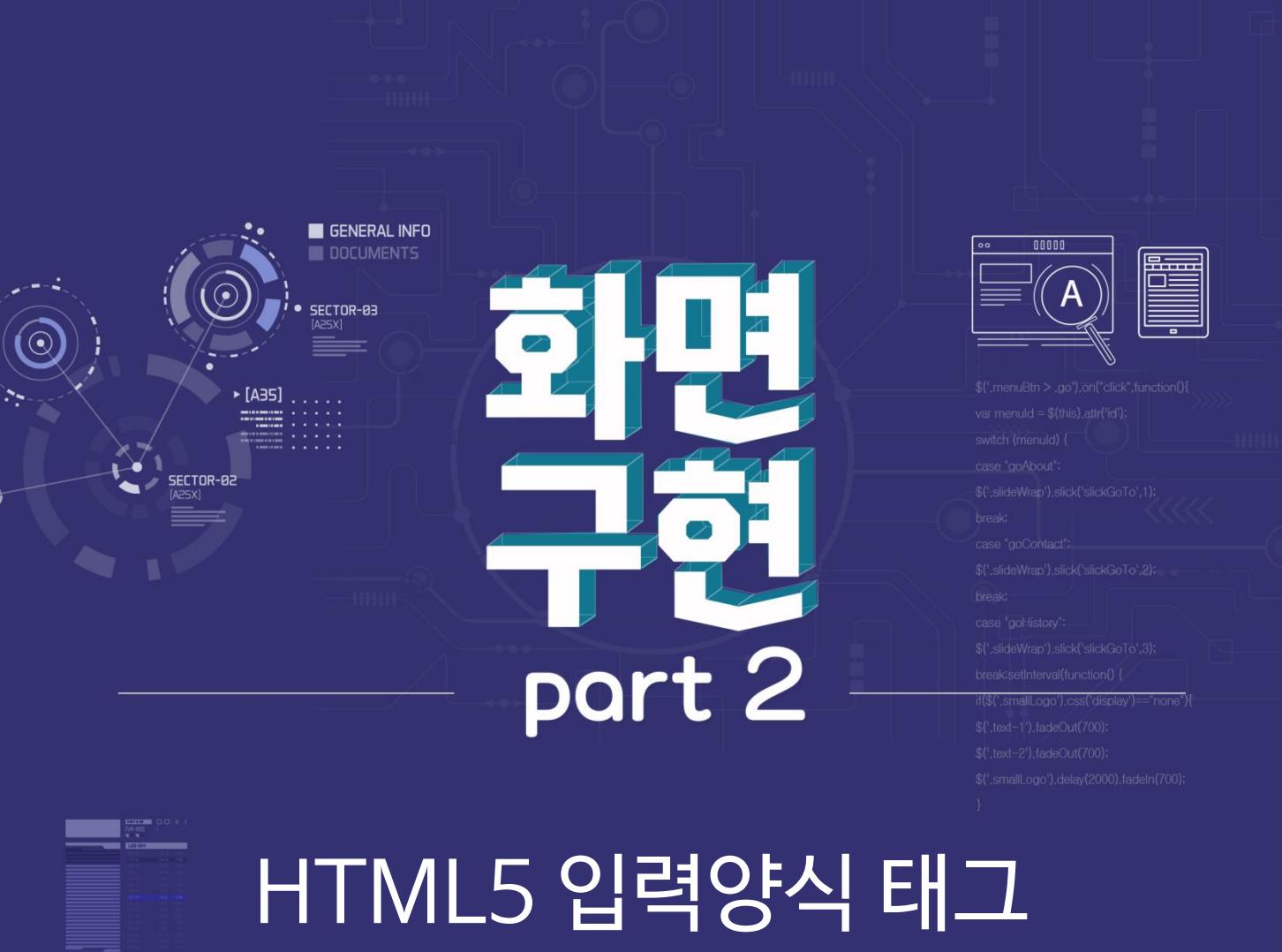
### 3. 비디오 태그

- <video> 태그는 무비 클립이나 다른 비디오 스트림과 같은 비디오를 지정함
- MP4, WebM, Ogg 3가지 형식의 파일 포맷을 지원하지만 브라우저마다 지원되지 않는 형식도 있음
- HTML5에서 새롭게 제공하는 태그임

# 화면 구현

part 2

## HTML5 입력양식 태그



KOREATECH  
온라인평생교육원



## 학습목표

- Input 태그를 이용하여 기본 입력 양식을 작성할 수 있다.
- Textarea, select 태그를 이용하여 입력 양식을 작성할 수 있다.
- Fieldset, legend 태그를 이용하여 입력 양식을 작성할 수 있다.

## 학습내용

- input 태그
- textarea, select 태그
- fieldset, legend 태그

# input 태그

## 1. form 태그

### 1) 정의 및 구조

- 사용자 입력을 위한 HTML 양식을 만드는데 사용
- form 태그는 다음 요소 중 하나 이상을 포함 가능

<input>

<option>

<textarea>

<optgroup>

<button>

<fieldset>

<select>

<label>

### 2) 속성

- accept-charset: 입력데이터에 사용될 문자 인코딩을 지정
  - 문자열 인코딩 참고  
[https://www.w3schools.com/tags/ref\\_charactersets.asp](https://www.w3schools.com/tags/ref_charactersets.asp)

# input 태그

## 1. form 태그

### 2) 속성

- action: 입력데이터의 전달 위치를 지정
- method: 입력데이터의 전달 방식을 선택
- enctype: 입력데이터를 서버에 보낼 때 인코딩할 방법을 지정
  - method 속성을 post로 할 때만 사용 가능
- name: 입력데이터의 form양식의 이름을 지정
- novalidate: 입력데이터 전달 시 데이터의 유효성 검사를 하지 않음
- target: 데이터 전달 후 서버 응답을 표시할 위치를 지정

# input 태그

## 1. form 태그

### 2) 속성

## Attributes

 = New in HTML5.

| Attribute             | Value   | Description   |
|-----------------------|---|---|
| <u>accept</u>         | <i>file_type</i>  | Not supported in HTML5.<br>Specifies a comma-separated list of file types that the server accepts (that can be submitted through the file upload) |
| <u>accept-charset</u> | <i>character_set</i>  | Specifies the character encodings that are to be used for the form submission   |
| <u>action</u>         | <i>URL</i>  | Specifies where to send the form-data when a form is submitted  |
| <u>autocomplete</u>   |  on<br>off | Specifies whether a form should have autocomplete on or off   |

〈 form 태그의 속성 설정 〉

# input 태그

## 1. form 태그

### 2) 속성

|                     |  |   |
|---------------------|--|---|
| <u>enctype</u>      | application/x-www-form-urlencoded<br>multipart/form-data<br>text/plain | Specifies how the form-data should be encoded when submitting it to the server (only for method="post") |
| <u>method</u>       | get<br>post  | Specifies the HTTP method to use when sending form-data   |
| <u>name</u>         | text   | Specifies the name of a form  |
| <u>novalidate</u> 5 | novalidate   | Specifies that the form should not be validated when submitted  |
| <u>target</u>       | _blank<br>_self<br>_parent<br>_top                                     | Specifies where to display the response that is received after submitting the form                      |

〈 form 태그의 속성 설정 〉

# input 태그

## 2. input 태그

### 1) 정의 및 구조

- 사용자가 데이터를 입력할 수 있는 입력 필드를 지정
- 입력 컨트롤을 선언하기 위해 form 요소 내에서 사용됨
- input 태그 선언 시 type 속성으로 컨트롤을 지정함
- 선언

〈input type = “value”〉

|             |                |          |        |
|-------------|----------------|----------|--------|
| <b>type</b> | button         | hidden   | search |
|             | checkbox       | image    | submit |
|             | color          | month    | tel    |
|             | date           | number   | text   |
|             | datetime-local | password | time   |
|             | email          | radio    | url    |
|             | file           | reset    | week   |

# input 태그

## 2. input 태그

### 2) type 속성

#### 기존 type 속성

- button: 버튼 생성
- checkbox: 체크박스 생성
- file: 파일 입력 양식 생성
- hidden: 데이터가 보이지 않도록 생성
- image: 이미지 형태 제출 버튼 생성
- password: 비밀번호 입력 양식 생성
- radio: 라디오 버튼을 생성
- reset: 초기화 버튼을 생성
- submit: 제출 버튼을 생성
- text: 글자 입력 양식을 생성

# input 태그

## 2. input 태그

### 2) type 속성

#### HTML5에서 추가된 type 속성

- color: 색상 선택 양식 생성
- date: 일 선택 양식 생성
- datetime: 날짜 선택 양식 생성
- datetime-local: 지역 날짜 선택 양식 생성
- email: 이메일 입력 양식을 생성
- month: 월 선택 양식을 생성
- number: 숫자 생성 양식 생성
- range: 범위 선택 양식 생성
- search: 검색어 입력 양식 생성
- tel: 전화번호 입력 양식 생성
- time: 시간 선택 양식을 생성
- url: url 주소 입력 양식을 생성
- week: 주 선택 양식을 생성

# input 태그

## 2. input 태그

### 2) type 속성

## Attribute Values

| Value                 | Description   |
|-----------------------|---|
| <u>button</u>         | Defines a clickable button (mostly used with a JavaScript to activate a script) |
| <u>checkbox</u>       | Defines a checkbox  |
| <u>color</u>          | Defines a color picker  |
| <u>date</u>           | Defines a date control (year, month, day (no time))                             |
| <u>datetime-local</u> | Defines a date and time control (year, month, day, time (no timezone))          |
| <u>email</u>          | Defines a field for an e-mail address   |
| <u>file</u>           | Defines a file-select field and a "Browse" button (for file uploads)            |
| <u>hidden</u>         | Defines a hidden input field  |
| <u>image</u>          | Defines an image as the submit button   |
| <u>month</u>          | Defines a month and year control (no timezone)                                  |

〈 input 태그의 속성 설정 〉

# input 태그

## 2. input 태그

### 2) type 속성

number



Defines a field for entering a number

password

Defines a password field

radio

Defines a radio button

range



Defines a range control (like a slider control)

reset

Defines a reset button

search



Defines a text field for entering a search string

submit

Defines a submit button

tel



Defines a field for entering a telephone number

text

Default. Defines a single-line text field

time



Defines a control for entering a time (no timezone)

url



Defines a field for entering a URL

week



Defines a week and year control (no timezone)

〈 input 태그의 속성 설정 〉

# input 태그

## 3. input 태그 활용 예시

### 1) 기존 입력 양식

#### 입력

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

<h1>Show input type attribute</h1>

<form action="/action_page.php">
    <input type="button" value="Button" onclick="msg()"><br/><br/>
    <input type="checkbox" name="check" value="1">check value<br/><br/>
    <input type="file" name="myFile"><br/><br/>
    <input type="hidden" id="custId" name="hiddenName" value="1234">
        hidden type<br/><br/>
    <input type="image" src="img_submit.gif" alt="Submit"><br/><br/>
    <input type="password" name="pwd" value = "defaultValue"><br/><br/>
    <input type="radio" name="gender" value="male">Male<br/>
    <input type="radio" name="gender" value="female"> Female<br/><br/>
    First name: <input type="text" name="fname"><br/>
    Last name: <input type="text" name="lname"><br/><br/>
    <input type="reset" value="reset"><br/><br/>
    <input type="submit" value="submit"><br/><br/>
</form>
<script>
function msg() {
    alert("Click Button");
}
</script>

</body>
</html>
```



# input 태그

## 3. input 태그 활용 예시

### 1) 기존 입력 양식

결과

### Show input type attribute

check value

파일 선택 선택된 파일 없음

hidden type



.....

Male  
 Female

First name:

Last name:

# input 태그

## 3. input 태그 활용 예시

### 2) HTML5 입력 양식

#### 입력

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

<h1>Show HTML5 input type attribute</h1>

<form action="/action_page.php">

color : <input type="color" name="favcolor" value="#ff0000"><br/><br/>
date : <input type="date" name="bday"><br/><br/>
datetime : <input type="datetime-local" name="bdaysime"><br/><br/>
month : <input type="month" name="bdaymonth"><br/><br/>
number : <input type="number" name="quantity" min="1" max="5"><br/><br/>
range : <input type="range" name="points" min="0" max="10"><br/><br/>
search : <input type="search" name="searchdata" value="data"><br/><br/>
tel : <input type="tel" name="usrtel"><br/><br/>
time : <input type="time" name="usr_time"><br/><br/>
url : <input type="url" name="homepage"><br/><br/>
week : <input type="week" name="week_year">

</form>

</body>
</html>
```



# input 태그

## 3. input 태그 활용 예시

### 2) HTML5 입력 양식

결과

### Show HTML5 input type attribute

color :

date :

datetime :

month :

number :

range :

search :

tel :

time :

url :

week :

# textarea, select 태그

## 1. textarea 태그

### 1) 정의 및 구조

- 여러 줄 텍스트 입력 컨트롤을 정의
- 텍스트 영역은 무제한의 문자 수를 유지 가능
- 텍스트 영역의 크기는 cols 및 rows 속성으로 지정하거나 CSS를 이용하여 지정 가능
- 선언

```
<textarea rows = “value” cols = “value”>
```

The textarea tag defines a multi-line text input control.

```
</textarea>
```

## textarea, select 태그

### 1. textarea 태그

#### 2) 속성

- cols: 너비 지정
- rows: 높이 지정
- disabled: 텍스트 영역을 비활성화 함
- maxlength: 텍스트 영역의 최대 길이 지정
- placeholder: 텍스트 영역의 예상값을 설명하는 짧은 힌트를 지정하고 사용자가 입력하기 전에 텍스트 영역에 표시됨
- required: 텍스트 영역 필수 입력

# textarea, select 태그

## 1. textarea 태그

### 2) 속성

## Attributes

 = New in HTML5.

Attribute	Value	Description
<u>autofocus</u>	 autofocus	Specifies that a text area should automatically get focus when the page loads
<u>cols</u>	number	Specifies the visible width of a text area
<u>dirname</u>	 textareaname.dir	Specifies that the text direction of the textarea will be submitted
<u>disabled</u>	disabled	Specifies that a text area should be disabled
<u>form</u>	 form_id	Specifies one or more forms the text area belongs to
<u>maxlength</u>	 number	Specifies the maximum number of characters allowed in the text area
<u>name</u>	text	Specifies a name for a text area
<u>placeholder</u>	 text	Specifies a short hint that describes the expected value of a text area
<u>readonly</u>	readonly	Specifies that a text area should be read-only
<u>required</u>	 required	Specifies that a text area is required/must be filled out
<u>rows</u>	number	Specifies the visible number of lines in a text area
<u>wrap</u>	 hard soft	Specifies how the text in a text area is to be wrapped when submitted in a form

〈 textarea 태그의 속성 설정 〉

# textarea, select 태그

## 2. textarea 태그 활용 예시

### 1) 텍스트 영역 지정 선언

#### 입력

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

<textarea rows="3" cols="40">
The <textarea> tag defines a multi-line text
input control.
A text area can hold an unlimited number of
characters, and the text renders in a fixed-
width font (usually Courier).
</textarea>

</body>
</html>
```



#### 결과

The <textarea> tag defines a multi-line  
text input control.  
A text area can hold an unlimited number

# textarea, select 태그

## 2. textarea 태그 활용 예시

### 2) 텍스트 영역에 짧은 힌트 추가

#### 입력

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

Who are you?<br/>
<textarea rows="4" cols="50" placeholder="Describe
yourself here... "></textarea>

</body>
</html>
```



#### 결과

Who are you?

Describe yourself here...

## textarea, select 태그

### 3. select 태그

#### 1) 정의 및 구조

- select 태그: 드랍 다운 목록을 정의
- option 태그: 목록에서 사용 가능한 옵션을 정의
- optgroup 태그: 목록의 옵션 그룹 정의
- 선언

```
<select>
  <optgroup label = “A type”>
    <option value = “A1”>A1 type</option>
    <option value = “A2”>A2 type</option>
  </optgroup>
  <optgroup label = “B type”>
    <option value = “B1”>B1 type</option>
    <option value = “B2”>B2 type</option>
  </optgroup>
</select>
```

# textarea, select 태그

## 3. select 태그

### 2) select 속성

- **autofocus**: 페이지가 로드될 때 드랍 다운 목록에 자동으로 포커스가 됨
- **required**: 선택할 데이터를 필수 입력해야 함
- **size**: 드랍 다운 목록에 표시되는 옵션의 수를 지정

## Attributes

 = New in HTML5.

Attribute	Value	Description
<u>autofocus</u>	 autofocus	Specifies that the drop-down list should automatically get focus when the page loads
<u>disabled</u>	disabled	Specifies that a drop-down list should be disabled
<u>form</u>	 form_id	Defines one or more forms the select field belongs to
<u>multiple</u>	multiple	Specifies that multiple options can be selected at once
<u>name</u>	name	Defines a name for the drop-down list
<u>required</u>	 required	Specifies that the user is required to select a value before submitting the form
<u>size</u>	number	Defines the number of visible options in a drop-down list

〈 select 태그의 속성 설정 〉

## textarea, select 태그

### 3. select 태그

#### 3) option 속성

- **disabled**: 해당 옵션을 사용 못하게 지정
- **selected**: 해당 옵션을 선택된 상태로 드랍 다운 메뉴 지정
- **value**: 선택한 옵션의 정보를 서버로 보낼 때 그 값을 지정

## Attributes

Attribute	Value	Description
<u>disabled</u>	disabled	Specifies that an option should be disabled
<u>label</u>	text	Specifies a shorter label for an option
<u>selected</u>	selected	Specifies that an option should be pre-selected when the page loads
<u>value</u>	text	Specifies the value to be sent to a server

〈 select 태그의 option 속성 설정 〉

## textarea, select 태그

### 4. select 태그 활용 예시

- 옵션 그룹을 이용하여 2개의 그룹으로 분류
- A2옵션을 기본 선택으로 지정

#### 입력

```
<!DOCTYPE html>
<html>
<body>
<form action="/action_page.php">
    <select name="result_type" required>
        <optgroup label = "A type">
            <option value = "A1">A1 type</option>
            <option value = "A2" selected>A2 type</option>
        </optgroup>
        <optgroup label = "B type">
            <option value = "B1">B1 type</option>
            <option value = "B2">B2 type</option>
        </optgroup>
    </select>
    <input type = "submit">
</form>

</body>
</html>
```



결과	<input style="width: 150px; height: 30px; border: none; border-bottom: 1px solid black;" type="text" value="A2 type"/>	<input style="width: 100px; height: 30px; border: none; background-color: #f0f0f0; border-radius: 5px;" type="button" value="제출"/>
----	--	--

# fieldset, legend 태그

## 1. 정의 및 구조

- field set 태그: 양식의 관련 요소를 그룹화하는데 사용
- legend 태그: field set 요소에 대한 캡션을 정의
- 선언

```
<form>
  <fieldset>
    <legend>LOGIN:</legend>
    ID : <input type = “text” name = “id”><br/>
    Password: <input type = “password” name = “pw”><br/>
    <input type = “submit” value = “login”><br/>
  </fieldset>
</form>
```

# fieldset, legend 태그

## 2. 속성

### 1) fieldset

- **disabled**: 관련 요소 그룹을 비활성화 지정
- **form**: field set가 속한 하나 이상의 form을 지정
- **name**: field set의 이름을 지정

## Attributes

 = New in HTML5.

Attribute	Value	Description
<u>disabled</u>	 disabled	Specifies that a group of related form elements should be disabled
<u>form</u>	 form_id	Specifies one or more forms the fieldset belongs to
<u>name</u>	 text	Specifies a name for the fieldset

〈 fieldset 태그의 속성 설정 〉

# fieldset, legend 태그

## 2. 속성

### 2) legend

- align: 정렬 관련 속성으로 HTML5에서는 더 이상 지원 안함

## Attributes

Attribute	Value	Description
<u>align</u>	top bottom left right	<b>Not supported in HTML5.</b> Specifies the alignment of the caption

〈 legend 태그의 속성 설정 〉

# fieldset, legend 태그

## 3. 활용 예시

### 1) 로그인 양식 만들기

#### 입력

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

<form action = "/action_page.php">
  <fieldset>
    <legend>LOGIN:</legend>
    ID      : <input type="text" name="id"><br/>
    Password: <input type="password" name="pw"><br/>
    <input type = "submit" value = "login"><br/>
  </fieldset>
</form>

</body>
</html>
```



#### 결과

LOGIN: -

ID :

Password:

# 핵심정리

## 1. input 태그

- 사용자 입력을 위한 양식을 만들 때 form 태그와 input 태그를 이용하여 생성함
- form 태그는 input, textarea, button, select, option, optgroup, fieldset, label을 하나 이상 포함 가능함
- input 태그의 type 속성을 이용하여 원하는 컨트롤을 선택하여 생성함

## 2. textarea, select 태그

- 여러 줄 텍스트 입력할 때는 textarea 태그를 이용하여 선언하며 텍스트 입력 문자 수는 무제한으로 입력할 수 있지만 maxlength 속성을 이용하여 글자 수를 제한할 수 있음
- select 태그는 드랍다운 목록을 정의함
- option 태그를 이용하여 목록의 항목을 정의함
- optgroup 태그를 이용하여 목록의 항목을 그룹화할 수 있음

## 3. fieldset, legend 태그

- 양식의 관련 요소를 그룹화하는데 사용하며 그룹화한 캡션은 legend 태그를 이용하여 정의함
- legend 태그는 field set 태그와 함께 사용하기도 하지만 독립적으로 사용할 수 있음

# 화면 구현

part 2

## HTML5 영역 태그



KOREATECH  
온라인평생교육원



## 학습목표

- `div`, `span` 태그를 이용하여 영역을 나눌 수 있다.
- 시멘틱 구조 태그를 이해하고, 활용할 수 있다.

## 학습내용

- `div`, `span` 태그
- HTML5 시멘틱 구조 태그

# div, span 태그

## 1. 구조와 속성

### 1) div 태그

- **div**는 HTML 문서에서 부분, 혹은 섹션을 정의
- **div 요소**는 CSS스타일을 지정하거나 Javascript로 특정 작업을 수행하기 위해 다른 HTML 요소의 컨테이너로 사용
- **div 요소**는 CSS와 함께 웹 페이지를 레이아웃 할 때 자주 사용

### 2) span 태그

- **span 태그**는 문서의 인라인 요소를 그룹화함
- **span 태그**는 시각적으로 변경되지 않으며, 텍스트의 일부 혹은 문서의 일부에 후크를 추가하는 방법을 제공

# div, span 태그

## 1. 구조와 속성

### 3) 속성

- align: 영역을 정렬, HTML5에서는 지원하지 않음

## Attributes

Attribute	Value	Description
-----------	-------	-------------

<u>align</u>	left right center justify	<p>Not supported in HTML5.</p> <p>Specifies the alignment of the content inside a &lt;div&gt; element</p>
--------------	------------------------------------	---

# div, span 태그

## 2. 태그 활용 예시

### 1) div 태그

- div 영역을 나타내기 위해 뒤 배경색을 변경함

입력

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

<p>This is some text.</p>

<div style="background-color:lightblue">
    <h3>This is a heading in a div element</h3>
    <p>This is some text in a div element.</p>
</div>

<p>This is some text.</p>

</body>
</html>
```



결과

This is some text.

**This is a heading in a div element**

This is some text in a div element.

This is some text.

# div, span 태그

## 2. 태그 활용 예시

### 2) div 태그

- span 영역을 나타내기 위해 글씨 색과 폰트를 수정함

#### 입력

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

<p>My mother has <span style="color:blue;font-weight:bold">blue</span>
eyes and my father has <span style="color:darkolivegreen;font-
weight:bold">dark green</span> eyes.</p>

</body>
</html>
```



#### 결과

My mother has blue eyes and my father has dark green eyes.

# HTML5 시멘틱 구조 태그

## 1. 시멘틱 웹이란?

### 1) 정의

‘**의미론적인 웹**’이라는 뜻

- 현재의 인터넷과 같은 분산환경에서  
**의미 정보(Semanteme)**를 기계(컴퓨터)가 처리  
할 수 있는 온톨로지 형태로 표현



#### 의미 정보(Semanteme)

웹 문서, 각종 파일, 서비스 등의 리소스에 대한  
정보와 자원 사이의 관계

- 이를 자동화된 기계(컴퓨터)가 처리하도록 하는  
프레임워크이자 기술

# HTML5 시멘틱 구조 태그

## 1. 시멘틱 웹이란?

### 2) 시멘틱 태그의 기본 구성

〈header〉

〈section〉

〈header〉

〈article〉

〈footer〉

〈nav〉

〈aside〉

〈footer〉

## 2. 시멘틱 태그

- header: 헤더를 의미, 문서의 서론이나 섹션의 목차, 검색, 로고 등이 들어가는 곳
- nav: 내비게이션을 의미, 페이지 이동을 위한 메뉴들이 모여 있는 곳

## HTML5 시멘틱 구조 태그

### 2. 시멘틱 태그

- **aside** : 사이드에 위치하는 공간을 의미,  
보조 메뉴나 광고를 배치하는 곳
- **section**: 여러 중심 내용을 감싸는 공간을 의미
- **article**: 글자가 많이 들어가는 부분을 의미,  
콘텐츠의 내용이 들어가는 부분
- **footer**: 푸터를 의미, 저작자의 이름이나 문서의  
날짜, 연락처 등이 들어감

## HTML5 시멘틱 구조 태그

### 2. 시멘틱 태그

#### 1) HTML4와 HTML5 차이

- HTML4

```
<div id = “header”> … </div>
```

- HTML5

```
<header> … </header>
```

Tip

HTML5에서 기존의 표현을 쉽게  
더 의미론적으로 나타낼 수 있게 됨

구조를 쉽게 파악할 수 있기 때문에  
CSS를 적용할 때 시멘틱 태그가 전체적으로 도움이 됨

# HTML5 시멘틱 구조 태그

## 3. 시멘틱 태그 종류

### 1) header



header란?

소개 내용 또는 탐색 링크 모음을 위한  
컨테이너

일반적으로 다음 요소를 포함

하나 이상의  
제목 요소  
(h1~h6)

로고, 아이콘

저자정보

하나의 문서에  
여러 개의  
Header 요소  
포함 가능

footer, address 혹은  
다른 header 요소 내에 배치 불가

## HTML5 시멘틱 구조 태그

### 3. 시멘틱 태그 종류

#### 1) header

입력

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

<article>
  <header>
    <h1>Most important heading here</h1>
    <h3>Less important heading here</h3>
    <p>Some additional information here.</p>
  </header>
  <p>Lorem Ipsum dolor set amet....</p>
</article>

</body>
</html>
```



결과

**Most important heading here**

**Less important heading here**

Some additional information here.

Lorem Ipsum dolor set amet....

# HTML5 시멘틱 구조 태그

## 3. 시멘틱 태그 종류

### 2) nav



nav란?

탐색 링크를 정의하며,  
탐색 링크의 주요 블록에서만 사용

#### 입력

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

<nav>
<a href="/html/">HTML</a><br/>
<a href="/css/">CSS</a><br/>
<a href="/js/">JavaScript</a><br/>
<a href="/jquery/">jQuery</a><br/>
</nav>

</body>
</html>
```

#### 결과

HTML  
CSS  
JavaScript  
jQuery



# HTML5 시멘틱 구조 태그

## 3. 시멘틱 태그 종류

### 3) aside



aside란?

사이드바라고 부르기도 하며,  
본문 이외의 내용을 담음

- 주로 본문 옆에 광고를 달거나 링크들을 이 공간에 넣어 표현
- 배너광고, 위젯 삽입

# HTML5 시멘틱 구조 태그

## 3. 시멘틱 태그 종류

### 3) aside

입력

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

<p>Body area</p>

<aside>
  <h3>aside area</h3>
  <p>The Epcot Center is a theme park in Disney World, Florida.</p>
</aside>

</body>
</html>
```



결과

Body area

**aside area**

The Epcot Center is a theme park in Disney World, Florida.

# HTML5 시멘틱 구조 태그

## 3. 시멘틱 태그 종류

### 4) section, article



section이란?

글을 구분하는 단락의 개념



article이란?

콘텐츠의 내용이 들어가는 부분

- body 영역은 콘텐츠를 <header>, <section>, <footer>의 3가지 공간에 콘텐츠를 저장, 그 중 <section>은 본문 콘텐츠를 담음
- section안에 section을 넣는 것도 가능함
- article 태그 안에는 실질적인 내용을 넣음

## HTML5 시멘틱 구조 태그

### 3. 시멘틱 태그 종류

#### 4) section, article

입력

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

<section>
  <h1>section title</h1>
  <article>
    <p>article contents area</p>
  </article>
</section>
</body>
</html>
```

결과

**section title**

article contents area



#### 5) footer



footer란?

화면의 구조 중  
제일 아래에 위치하는 것

# HTML5 시멘틱 구조 태그

## 3. 시멘틱 태그 종류

### 5) footer

- 회사소개, 저자 정보, 저작권 정보, 연락처 정보, 사이트맵, 위로 가기 링크, 관련된 문서 등이 들어감
  - 연락처 태그
- ⟨address⟩**
- 한 문서 내에서 여러 개의 footer 사용 가능

#### 입력

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

<footer>
  <p>Posted by: Hege Refsnes</p>
  <p>Contact information: <a
  href="mailto:someone@example.com">someone@example.com</a>.</p>
</footer>

</body>
</html>
```



#### 결과

Posted by: Hege Refsnes

Contact information: [someone@example.com](mailto:someone@example.com).

# 핵심정리

## 1. div, span 영역 태그

- div, span 태그는 웹 페이지의 영역을 정의하는 태그로서 사용됨
- div 태그는 block 영역을 정의하면 span 태그는 inline block 영역을 나타냄
- CSS 스타일을 지정하거나 JavaScript로 특정 작업을 수행하기 위해 다른 HTML요소의 컨테이너로 사용됨

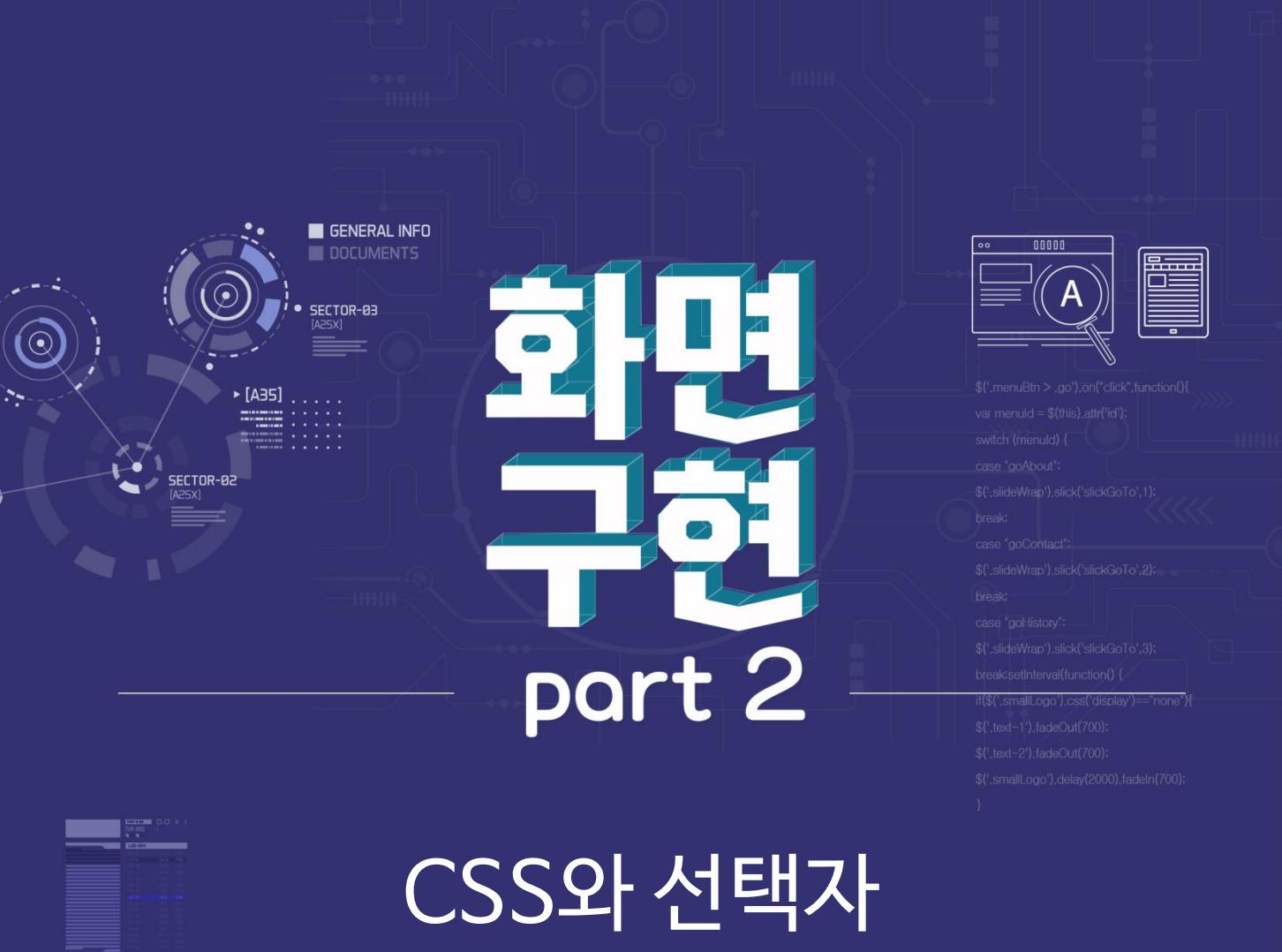
## 2. HTML5 시멘틱 구조 태그

- 시멘틱 웹은 의미론적 웹이라는 뜻으로 리소스에 대한 정보와 자원 사이의 관계-의미 정보를 컴퓨터가 처리할 수 있는 형태로 표현함
- header: 문서의 서론이나 섹션의 목차, 검색, 로고 등이 들어가는 곳
- nav: 페이지 이동을 위한 메뉴들이 모여 있는 곳
- aside: 사이드에 위치하는 공간으로 보조 메뉴나 광고, 위젯 등을 배치하는 곳
- section: 여러 중심 내용을 감싸는 공간
- article: 글자가 많이 들어가는 부분을 의미하며 컨텐츠의 내용이 들어가는 부분
- footer: 저작자의 정보가 들어가는 부분으로 화면의 하단에 주로 위치

# 화면 구현

part 2

## CSS와 선택자



KOREATECH  
온라인평생교육원



## 학습목표

- CSS, 선택자에 대해 설명할 수 있다.
- 선택자의 종류를 이해하고, 이를 활용할 수 있다.

## 학습내용

- CSS, 선택자 개요
- 선택자 종류

# CSS, 선택자 개요

## 1. CSS란?

### 1) 개요

# CSS

Cascading Style Sheets의 약자

HTML, XHTML, XML 같은 문서의 스타일을 꾸밀 때  
사용하는 스타일 시트 언어



CSS는?

- 글꼴이나 배경색, 너비와 높이, 위치 등 레이아웃 지정 가능
- 웹 브라우저, 스크린 크기, 장치에 따라서 화면을 다르게 표시될 수 있도록 지정 가능
- 문서의 내용(Content)과 표현(Presentation)을 분리
- CSS 파일 하나만 수정하면 같은 CSS 파일을 쓰는 HTML 문서가 한 번에 수정됨

# CSS, 선택자 개요

## 1. CSS란?

### 2) CSS블록 구문

- 구성요소: 선택자      스타일 속성      스타일 값
- 속성 선언: 세미콜론(;)으로 구분하여, 하나 이상의 속성을 선언
- <p> 태그에 대한 요소들의 텍스트 색상을 파란색으로 지정

A diagram illustrating the structure of a CSS rule. At the top is the rule: **p { color : blue; }**. Three arrows point downwards from specific parts of the rule to labels below. The first arrow points to the 'p' selector, which is labeled '선택자'. The second arrow points to the 'color' property, which is labeled '스타일 속성'. The third arrow points to the 'blue' value, which is labeled '스타일 값'.

# CSS, 선택자 개요

## 1. CSS란?

### 3) 스타일 꾸미는 방법

#### [ 속성처럼 style 작용(in-line style) ]

- 스타일을 지정할 태그에 선언

style="속성명:값;"

#### 입력

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

<p style="color:blue; text-align:center;">Hello World!</p>
<p>These paragraphs are styled with CSS.</p>

</body>
</html>
```



#### 결과

Hello World!

These paragraphs are styled with CSS.

# CSS, 선택자 개요

## 1. CSS란?

### 3) 스타일 꾸미는 방법

#### [ style 태그를 사용(internal style sheet) ]

- <head> ~ </head> 태그 사이에 <style> ~ </style> 태그 안에 스타일 선언
- 스타일을 변경할 선택자를 선언하여 스타일 지정

# CSS, 선택자 개요

## 1. CSS란?

### 3) 스타일 꾸미는 방법

## [ style 태그를 사용(internal style sheet) ]

### 입력

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

<p>Hello World!</p>
<p>These paragraphs are styled with CSS.</p>

</body>
</html>
```



### 결과

Hello World!

These paragraphs are styled with CSS.

# CSS, 선택자 개요

## 1. CSS란?

### 3) 스타일 꾸미는 방법

#### [ style 태그를 사용(internal style sheet) ]

- <head> ~ </head> 태그 사이에  
<style> ~ </style> 태그 안에 연결할 CSS파일을 선언
- <link rel = “stylesheet” type = “text/css”  
href = “CSS 파일 주소”> 형식으로 선언

# CSS, 선택자 개요

## 1. CSS란?

### 3) 스타일 꾸미는 방법

## [ style 태그를 사용(internal style sheet) ]

- 스타일 정의는 css 파일에서 선언

### 입력

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css"
      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;
}
```

〈 mystyle.css 〉



### 결과

This is a heading

This is a paragraph.

# CSS, 선택자 개요

## 2. 선택자(Selector)란?

### 1) 개요

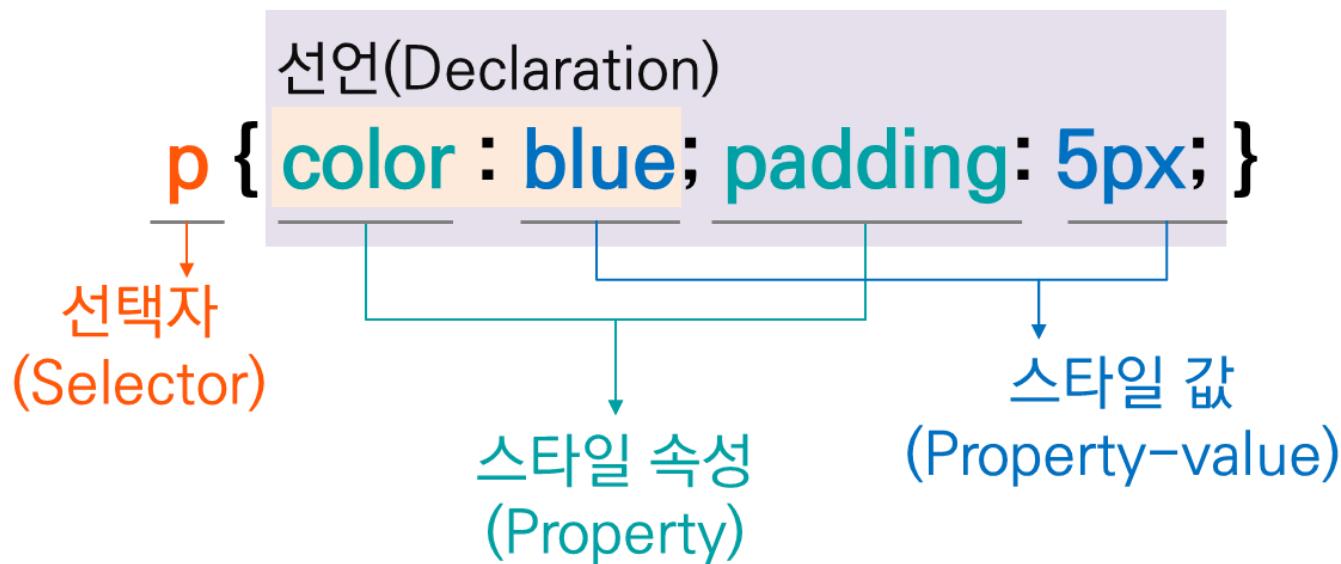


선택자를 사용하는 이유는?

- HTML 요소를 찾기 위해 이름, ID, 클래스, 특성 등을 지정하여 원하는 스타일 기능 적용 가능
- 콤마( , )로 다수의 선택자 사용 가능

```
h1, h2, h3 { font-weight : bold; }
```

선언 블록(Declaration block)



# CSS, 선택자 개요

## 2. 선택자(Selector)란?

### 2) 선택자 우선순위



#### 선택자 우선순위의 개념은?

- 선택자는 기본적으로 가지고 있는 우선순위 규칙을 가지고 있음
  - 높은 우선순위를 가지는 선택자의 속성의 규칙을 따라 감
  - 일반적으로 HTML 문서의 전반적인 스타일을 적용
- 구체적인 선택자는 일반적인 선택자보다 우선순위를 높여서 스타일을 적용해야 함
- 우선순위가 같은 선택자 그룹에 속해 있다면 선택자의 종류와 수에 따라 우선순위가 결정됨

# CSS, 선택자 개요

## 2. 선택자(Selector)란?

### 2) 선택자 우선순위

#### [ 선택자 그룹 ]

- 1순위: !important → 우선순위가 가장 높은 그룹

EX) p { color : blue !important; }

- 2순위: 인라인 스타일 속성 → 다른 규칙 그룹보다 우선시 되지만 인라인 스타일 속성을 쓰면 코드가 어렵고 복잡해짐
- 3순위: 아이디 선택자
- 4순위: 클래스, 속성, 가상선택자
- 5순위: 요소 선택자
- 6순위: 전체 선택자

# CSS, 선택자 개요

## 2. 선택자(Selector)란?

### 2) 선택자 우선순위

#### [ 우선순위 예시 ]

- 아이디 선택자 > 클래스 선택자

##### 입력

```
<!DOCTYPE html>
<html>
<head>
<style>
#classA_id { background : blue; }

.classA { background : red; }

</style>
</head>
<body>
<div class = "classA" id="classA_id">
<h1>CSS Example</h1>
<p>This is a paragraph.</p>
</div>
</body>
</html>
```



##### 결과

### CSS Example

This is a paragraph.

# CSS, 선택자 개요

## 2. 선택자(Selector)란?

### 2) 선택자 우선순위

#### [ 우선순위 예시 ]

- 인라인 스타일 속성 > 아이디 선택자 > 클래스 선택자

##### 입력

```
<!DOCTYPE html>
<html>
<head>
<style>
#classA_id { background : blue; }

.classA { background : red; }

</style>
</head>
<body>
<div class = "classA" id="classA_id"
style = "background:yellow;">
<h1>CSS Example</h1>
<p>This is a paragraph.</p>
</div>
</body>
</html>
```



##### 결과

### CSS Example

This is a paragraph.

# CSS, 선택자 개요

## 2. 선택자(Selector)란?

### 2) 선택자 우선순위

#### [ 우선순위 예시 ]

- !important > 인라인 스타일 속성 > 아이디 선택자 > 클래스 선택자

입력

```
<!DOCTYPE html>
<html>
<head>
<style>
#classA_id { background : blue; }

.classA { background : red !important; }

</style>
</head>
<body>
<div class = "classA" id="classA_id"
style = "background:yellow;">
<h1>CSS Example</h1>
<p>This is a paragraph.</p>
</div>
</body>
</html>
```



결과

CSS Example

This is a paragraph.

# CSS, 선택자 개요

## 2. 선택자(Selector)란?

### 2) 선택자 우선순위

#### [ 우선순위 예시 ]

- 동일한 우선순위 선택자 속성이 중복될 경우 가장 마지막에 정의한 속성이 우선순위가 높음

입력

```
<!DOCTYPE html>
<html>
<head>
<style>
.classA { background : red; }

.classB { background : blue; }
</style>
</head>
<body>
<div class = "classA classB">
<h1>CSS Example</h1>
<p>This is a paragraph.</p>
</div>
</body>
</html>
```



결과

CSS Example

This is a paragraph.

## 선택자 종류

### 1. 전체 선택자(\*)

- HTML 페이지 내부의 모든 태그를 선택
- 선언

```
* {  
    css declarations;  
}
```

〈 모든 태그의 스타일 선언 〉

## 선택자 종류

### 1. 전체 선택자(\*)

- div 태그 모두의 배경색을 yellow로 지정

#### 입력

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

<h3>Welcome to My Homepage</h3>

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

<p>My best friend is Mickey.</p>

</body>
</html>
```



#### 결과

Welcome to My Homepage

My name is Donald.

I live in Duckburg.

My best friend is Mickey.

## 선택자 종류

### 2. 타입 선택자(element)

- 태그명이 element인 특정 태그를 선택
- 선언

```
element {  
    css declarations;  
}
```

〈 element의 스타일 선언 〉

## 선택자 종류

### 2. 타입 선택자(element)

- 태그명이 p인 항목의 배경색을 yellow로 지정

#### 입력

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

    <h3>Welcome to My Homepage</h3>

    <div>
      <p id="firstname">My name is Donald.</p>
      <p id="hometown">I live in Duckburg.</p>
    </div>

    <p>My best friend is Mickey.</p>

  </body>
</html>
```



#### 결과

Welcome to My Homepage

My name is Donald.

I live in Duckburg.

My best friend is Mickey.

## 선택자 종류

### 2. 타입 선택자(element)

- element1, element2: element1과 element2 특정 태그 모두를 선택

```
<style>
  h1, p {
    background-color: yellow;
  }
</style>
```

- element1 element2: element 1 안에 속해있는 elements2 특정 태그 모두를 선택

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

## 선택자 종류

### 2. 타입 선택자(element)

- element1 + element2: element1 요소를 바로 뒤따르는 element2 태그를 선택

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

- element1 > element2: 상위(부모) 요소가 element1인 모든 element2 태그 선택

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

- element1 ~ element2: element1 요소 뒤에 모든 element2 태그를 선택

```
<style>
p ~ ul {
    background: #ff0000;
}
</style>
```

## 선택자 종류

### 3. 클래스 선택자(.className)

- 주어진 값을 class 속성값으로 가진 지정된 태그를 선택
- 선언

```
.class {  
    css declarations;  
}
```

〈 class 속성값의 스타일 선언 〉

## 선택자 종류

### 3. 클래스 선택자(.className)

- p 태그 중 클래스 이름이 hometown인 태그를 선택하여 배경색을 yellow로 지정

#### 입력

```
<!DOCTYPE html>
<html>
<head>
<style>
p.hometown {
    background: yellow;
}
</style>
</head>
<body>

<p>My name is Donald.</p>
<p class="hometown">I live in Ducksburg.</p>

<p>My name is Dolly.</p>
<p class="hometown">I also live in Ducksburg.
</p>

</body>
</html>
```



#### 결과

My name is Donald.

I live in Ducksburg.

My name is Dolly.

I also live in Ducksburg.

## 선택자 종류

### 4. ID 선택자(#id)

- 구체적인 id를 가진 한 요소 선택
- 선언

```
#id{  
    css declarations;  
}
```

〈 id의 스타일 선언 〉

## 선택자 종류

### 4. ID 선택자(#id)

- ‘firstname’의 id를 가진 태그의 배경을 yellow로 변경

입력

```
<!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 Donald.</p>
    <p id="hometown">I live in Duckburg.</p>
</div>

<p>My best friend is Mickey.</p>

</body>
</html>
```



결과

Welcome to My Homepage

My name is Donald.

I live in Duckburg.

My best friend is Mickey.

## 선택자 종류

### 5. 속성 선택자

태그 안에 특정 속성들에 따라 선택

- **element [ attribute ] { }:** element 태그 안에 attribute 속성을 선택

```
[attribute] {  
    css declarations;  
}
```

〈 attribute 속성 스타일 선언 〉

# 선택자 종류

## 5. 속성 선택자

### 입력

```
<!DOCTYPE html>
<html>
<head>
<style>
a[target] {
    background-color: yellow;
}
</style>
</head>
<body>

<p>The links with a target attribute gets a yellow background:</p>

<a href="https://www.w3schools.com">w3schools.com</a>
<a href="http://www.disney.com" target="_blank">disney.com</a>
<a href="http://www.wikipedia.org" target="_top">wikipedia.org</a>

<p><b>Note:</b> For [<i>attribute</i>] to work in IE8 and earlier,
a DOCTYPE must be declared.</p>

</body>
</html>
```



### 결과

The links with a target attribute gets a yellow background:

[w3schools.com](https://www.w3schools.com) [disney.com](http://www.disney.com) [wikipedia.org](http://www.wikipedia.org)

**Note:** For [attribute] to work in IE8 and earlier,  
a DOCTYPE must be declared.

## 선택자 종류

### 5. 속성 선택자

- **element [ attribute = value ] { }**: element 태그 안에 attribute가 value인 속성을 선택

```
[attribute = value] {  
    css declarations;  
}
```

〈 attribute가 value인 속성의 스타일 선언 〉

## 선택자 종류

### 5. 속성 선택자

#### 입력

```
<!DOCTYPE html>
<html>
<head>
<style>
input[type=text] {
    width: 100px;
    -webkit-transition: width .35s ease-in-out;
    transition: width .35s ease-in-out;
}
input[type=text]:focus {
    width: 250px;
}
</style>
</head>
<body>
<h1>The width Property</h1>

<p>Set the width of the input field to 100 pixels. However, when the input field gets focus, make it 250 pixels wide:</p>

Search: <input type="text" name="search">

</body>
</html>
```



#### 결과

## The width Property

Set the width of the input field to 100 pixels. However, when the input field gets focus, make it 250 pixels wide:

Search:

## 선택자 종류

### 5. 속성 선택자

- **element [ attribute ~= value ] { }**

: element 태그 안에 attribute의 value가 속한 요소 모두 선택

```
[attribute ~= value] {  
    css declarations;  
}
```

〈 attribut에 value가 속한 모든 요소의 스타일 선언 〉

# 선택자 종류

## 5. 속성 선택자

### 입력

```
<!DOCTYPE html>
<html>
<head>
<style>
[title~=flower] {
    border: 5px solid yellow;
}
</style>
</head>
<body>
```

`<p>The image with the title attribute containing the word "flower" gets a yellow border.</p>`

```



```

`<p><b>Note:</b> For [<i>attribute</i>~=<i>value</i>] to work in IE8 and earlier, a DOCTYPE must be declared.</p>`

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



### 결과

The image with the title attribute containing the word "flower" gets a yellow border.



`Note: For [attribute~=value] to work in IE8 and earlier, a DOCTYPE must be declared.`

## 선택자 종류

### 5. 속성 선택자

- **element [ attribute |= value ] { }**

: element 태그 안에 attribute가 value를 가지는 요소를 모두 선택

```
[attribute |= value] {  
    css declarations;  
}
```

〈 attribute가 value를 가지는 요소의 스타일 선언 〉

## 선택자 종류

### 5. 속성 선택자

#### 입력

```
<!DOCTYPE html>
<html>
<head>
<style>
[attribute|=value] {
    background: yellow;
}
</style>
</head>
<body>

<h1 class="top-header">Welcome</h1>
<p class="top-text">Hello world!</p>
<p class="content">Are you learning CSS?</p>

<p><b>Note:</b> For [<i>attribute</i>|=<i>value</i>] to work in IE8 and earlier, a DOCTYPE must be declared.</p>

</body>
</html>
```



#### 결과

Welcome

Hello world!

Are you learning CSS?

Note: For [attribute|=value] to work in IE8 and earlier, a DOCTYPE must be declared.

## 선택자 종류

### 5. 속성 선택자

- **element [ attribute ^= value ] { }**

: element 태그 안에 attribute가 value로 시작되는 요소를 모두 선택

```
[attribute ^= value] {  
    css declarations;  
}
```

〈 attribute가 value로 시작되는 요소의 스타일 선언 〉

## 선택자 종류

### 5. 속성 선택자

#### 입력

```
<!DOCTYPE html>
<html>
<head>
<style>
div[class^="test"] {
    background: #ffff00;
}
</style>
</head>
<body>

<div class="first_test">The first div element.</div>
<div class="second">The second div element.</div>
<div class="test">The third div element.</div>
<p class="test">This is some text in a paragraph.</p>

</body>
</html>
```



#### 결과

The first div element.  
The second div element.  
**The third div element.**

This is some text in a paragraph.

## 선택자 종류

### 5. 속성 선택자

- **element [ attribute \*= value ] { }**

: element 태그 안에 attribute가 value를 포함하는 속성 모두를 선택

```
attribute *= value {  
    css declarations;  
}
```

〈 attribute가 value를 포함하는 속성의 스타일 선언 〉

## 선택자 종류

### 5. 속성 선택자

#### 입력

```
<!DOCTYPE html>
<html>
<head>
<style>
div[class*="test"] {
    background: #ffff00;
}
</style>
</head>
<body>

<div class="first_test">The first div element.</div>
<div class="second">The second div element.</div>
<div class="test">The third div element.</div>
<p class="test">This is some text in a paragraph.</p>

</body>
</html>
```



#### 결과

The first div element.

The second div element.

The third div element.

This is some text in a paragraph.

# 핵심정리

## 1. CSS, 선택자 개요

- CSS는 HTML, XHTML, XML 같은 문서의 스타일을 꾸밀 때 사용하는 스타일 시트 언어임
- 글꼴이나, 배경색, 너비와 높이, 위치 등 레이아웃을 지정할 수 있음
- CSS 파일 하나만 수정하면 같은 CSS 파일을 쓰는 HTML 문서가 한번에 수정됨
- CSS 블록은 선택자, 스타일 속성, 스타일 값으로 구성됨
- 선택자를 사용하는 이유는 HTML 요소를 찾기 위해 이름, ID, 클래스, 특성 등을 지정하여 원하는 스타일 기능을 적용할 수 있음
- 선택자는 기본적으로 가지고 있는 우선순위 규칙을 가지고 있고 우선순위가 같은 그룹에 속해 있다면 선택자의 종류와 수에 따라 우선순위가 결정됨
- 일반적으로 HTML 문서의 전반적인 스타일을 적용 후 구체적인 선택자는 일반적인 선택자보다 우선순위를 높여서 스타일을 적용해야 함

# 핵심정리

## 2. 선택자 종류

- 전체 선택자는 \*으로 선언하며 HTML 페이지 내부의 모든 태그를 선택함
- 타입 선택자는 태그명이 element인 특정 태그를 선택함
- 클래스 선택자는 .className으로 선언하며 지정된 요소를 찾아서 선택함
- ID 선택자는 #id로 선언하며 지정된 id 속성값을 찾아서 선택함
- 속성 선택자는 속성값의 조건에 따라 다양한 스타일을 지정할 수 있음

# 화면 구현

part 2

## CSS3 스타일



KOREATECH  
온라인평생교육원



## 학습목표

- CSS3 단위와 그레디언트에 관해 이해하고, 설명할 수 있다.
- CSS3 속성을 이해하고, 이를 활용할 수 있다.

## 학습내용

- CSS3 단위 및 그레디언트
- CSS3 속성
- CSS3 실습

# CSS3 단위 및 그레디언트

## 1. 크기 단위

### 1) 속성과 단위



CSS 속성은?

width

margin

padding

font-size

border-width

… 등과 같이 길이값 사용

- 속성의 값을 지정하기 위하여 단위를 사용

절대 단위

상대 단위

- 일부 속성들은 음수의 크기를 사용할 수 있음

# CSS3 단위 및 그레디언트

## 1. 크기 단위

### 2) 절대 단위



절대 단위란?

- 절대 길이 단위는 고정되어 있고 장치에 따라 크기 조절이 안 됨
- 절대 길이 단위는 화면크기가 다양하기 때문에 사용하지 않는 것을 권장
- 프린트 레이아웃과 같이 출력 매체가 알려진 경우 사용
- 픽셀(px)은 보기 장치와 관련하여
  - 낮은 dpi 장치의 경우 1px은 디스플레이 장치 및 프린터 픽셀
  - 고해상도 스크린의 경우 1px은 여러 장치 픽셀을 의미

# CSS3 단위 및 그레디언트

## 1. 크기 단위

### 2) 절대 단위

Unit	Description
cm	centimeters
mm	millimeters
in	inches ( $1\text{in} = 96\text{px} = 2.54\text{cm}$ )
px *	pixels ( $1\text{px} = 1/96\text{th of 1in}$ )
pt	points ( $1\text{pt} = 1/72 \text{ of 1in}$ )
pc	picas ( $1\text{pc} = 12 \text{ pt}$ )

〈 종류 〉

# CSS3 단위 및 그레디언트

## 1. 크기 단위

### 2) 절대 단위

- mm

#### 입력

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {font-size: 15mm;}
h2 {font-size: 10mm;}
p {
    font-size: 5mm;
    line-height: 10mm;
}
</style>
</head>
<body>

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

</body>
</html>
```

#### 결과

This is heading 1

This is heading 2

This is a paragraph.

This is another paragraph.



# CSS3 단위 및 그레디언트

## 1. 크기 단위

### 2) 절대 단위

- pt

#### 입력

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {font-size: 50pt;}
h2 {font-size: 25pt;}
p {
    font-size: 15pt;
    line-height: 25pt;
}
</style>
</head>
<body>

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

</body>
</html>
```

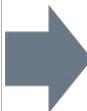
#### 결과

This is heading 1

This is heading 2

This is a paragraph.

This is another paragraph.



# CSS3 단위 및 그레디언트

## 1. 크기 단위

### 3) 상대 단위



상대 단위란?

- 상대 길이 단위는 다른 길이 등록 정보와 관련된 길이를 지정
- em 및 rem 단위는 확장 가능한 레아웃을 만드는데 실용적
- viewport란 브라우저 창 크기를 의미
  - viewport가 50cm이면, 너비는 1vw = 0.5cm

# CSS3 단위 및 그레디언트

## 1. 크기 단위

### 3) 상대 단위

Unit	Description
em	Relative to the font-size of the element (2em means 2 times the size of the current font)
ex	Relative to the x-height of the current font (rarely used)
ch	Relative to width of the "0" (zero)
rem	Relative to font-size of the root element
vw	Relative to 1% of the width of the viewport*
vh	Relative to 1% of the height of the viewport*
vmin	Relative to 1% of viewport's* smaller dimension
vmax	Relative to 1% of viewport's* larger dimension
%	Relative to the parent element

〈 종류 〉

# CSS3 단위 및 그레디언트

## 2. 색상 단위

### 1) 키워드를 이용한 색상 적용

#### ▪ 종류

- [https://www.w3schools.com/cssref/css\\_colors.asp](https://www.w3schools.com/cssref/css_colors.asp)

색상 키워드로 적용 가능한 색 리스트

The screenshot shows the 'Color Names Supported by All Browsers' section of the w3schools website. It lists 140 color names with their corresponding hex codes and color swatches. The colors are arranged in rows, each with a color name, hex code, color swatch, and links for 'Shades' and 'Mix'. A 'COLOR PICKER' icon is visible on the right.

Color Name	HEX	Color	Shades	Mix
AliceBlue	#F0F8FF			
AntiqueWhite	#FAEBD7			
Aqua	#00FFFF			
Aquamarine	#7FFFDD			
Azure	#F0FFFF			
Beige	#F5F5DC			
Bisque	#FFE4C4			
Black	#000000			
BlanchedAlmond	#FFEBCD			
Blue	#0000FF			
BlueViolet	#8A2BE2			
Brown	#A52A2A			
BurlyWood	#DEB887			
CadetBlue	#5F9EA0			

출처: <https://www.w3schools.com>

# CSS3 단위 및 그레디언트

## 2. 색상 단위

### 1) 키워드를 이용한 색상 적용

#### 입력

```
<!DOCTYPE html>
<html>
<head>
<style>
#p1 {background-color:red;}
#p2 {background-color:green;}
#p3 {background-color:blue;}
#p4 {background-color:yellow;}
#p5 {background-color:cyan ;}
</style>
</head>

<body>
<p>keyword colors:</p>
<p id="p1">Red</p>
<p id="p2">Green</p>
<p id="p3">Blue</p>
<p id="p4">Yellow</p>
<p id="p5">Cyan</p>
</body>
</html>
```

#### 결과

keyword colors:

Red

Green

Blue

Yellow

Cyan



# CSS3 단위 및 그레디언트

## 2. 색상 단위

### 2) 색상 단위 종류

- **Hexadecimal colors**: HEX 코드 단위
- **RGB colors**: RGB 색상 단위
- **RGBA colors**: RGBA 색상 단위
- **HSL colors**: HSL 색상 단위
- **HSLA colors**: HSL 색상 단위

## 3. URL 단위

- 이미지 파일이나 폰트 파일을 불러올 때 URL 단위 사용
- url() 함수를 이용하여 지정
- 이미지의 url()을 두 개 이상 지정할 경우로 구분

# CSS3 단위 및 그레디언트

## 4. 그레디언트

- 2가지 이상의 색상을 혼합한 색을 만드는 기능
- CSS3에서부터 그레디언트 기능을 지원  
(이전 버전까지는 이미지로 처리)
- 그레디언트 생성 툴에서 실습:  
<http://www.colorzilla.com/gradient-editor/>

**Ultimate CSS Gradient Generator**  
A powerful Photoshop-like CSS gradient editor from [ColorZilla](#).

For Firefox For Chrome Gradient Generator

— Presets —



Name:



— Stops —

Opacity:	<input type="button" value=""/>	Location:	<input type="text" value="0%"/>	<input type="button" value="delete"/>
Color:	<input type="button" value=""/>	Location:	<input type="text" value="0%"/>	<input type="button" value="delete"/>

— Sponsor —

St
Adobe Creative Cloud for Teams starting at \$29.99 per month.

— Preview —



Orientation:   Size:  x   IE

— CSS — [switch to scss](#)

```
/* Permalink - use to edit and share this gradient:
 http://colorzilla.com/gradient-editor/#1e5799+0,2989d8+50,207cca+51,7db9e8+100;Blue+ */
background: #1e5799; /* Old browsers */
background: -moz-linear-gradient(top, #1e5799 0%, #2989d8 50%, #207cca 51%, #7db9e8 100%); /* FF3.6-15 */
background: -webkit-linear-gradient(top, #1e5799 0%, #2989d8 50%, #207cca 51%, #7db9e8 100%); /* Chrome10-25, Safari5.1-6 */
background: linear-gradient(to bottom, #1e5799 0%, #2989d8 50%, #207cca 51%, #7db9e8 100%); /* W3C, IE10+, FF16+, Chrome26+, Opera12+, Safari7+ */
filter: progid:DXImageTransform.Microsoft.gradient(
 startColorstr='#1e5799',
 endColorstr='#7db9e8',GradientType=0 ); /* IE6-9 */
```

Color format:   Comments

[Link to, save or share the current gradient using its permalink.](#)

[Tweet](#) [좋아요 1.3만개](#) AccessDe [Follow @colorzilla](#)

출처: <http://www.colorzilla.com>

# CSS3 속성

## 1. 가시 속성



가시 속성이란?

- 태그가 화면에 보이는 방식을 지정하는 속성
- display 속성 확인은?  
[https://www.w3schools.com/cssref/playit.asp?filename=playcss\\_display&preval=table-caption](https://www.w3schools.com/cssref/playit.asp?filename=playcss_display&preval=table-caption)

### [ display 속성에 적용 가능한 키워드 ]

- **none**: 태그를 화면에서 보이지 않게 적용
- **block**: 태그를 block 형식으로 지정
- **inline**: 태그를 inline형식으로 지정
- **inline-block**: 태그를 inline-block 형식으로 지정

# CSS3 속성

## 2. 박스 속성



박스 속성이란?

- 웹 페이지의 레이아웃을 구성할 때 가장 중요한 스타일 속성
- margin, border, padding, width, height로 구성
- **width, height 속성**: 글자를 감싸는 영역의 크기를 지정
- **margin, padding**: 너비를 지정
- **border**: 테두리의 너비를 지정

# CSS3 속성

## 3. 배경 속성

- **background 속성**: 특정 태그의 배경 이미지 또는 색상을 지정하는 속성
- **background-image**: 하나 이상의 배경 이미지로 설정
- **background-position**: 배경 이미지의 시작 위치를 설정
- **background-size**: 배경 이미지의 크기를 지정
- **background-repeat**: 배경 이미지가 반복되는 경우를 설정
- **background-attachment**  
: 배경 이미지가 나머지 페이지와 함께 스크롤 되는지 고정되는지 설정
- **background-color**: 배경색을 설정

# CSS3 속성

## 3. 배경 속성

### 1) background-image

입력

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-image: url("paper.gif");
    background-color: #cccccc;
}
</style>
</head>
<body>

<h1>The background-image Property</h1>

<p>Hello World!</p>

</body>
</html>
```



결과

The background-image Property

Hello World!

# CSS3 속성

## 3. 배경 속성

### 2) background-position, attachment, repeat

입력

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-image: url('w3css.gif');
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-position: left;
}
</style>
</head>
<body>
<h3>The background-position Property</h3>
<h3>The background-attachment Property</h3>
<h3>The background-repeat Property</h3>
</body>
</html>
```



# CSS3 속성

## 3. 배경 속성

### 2) background-position, attachment, repeat

결과

The **background-position** Property

The **background-attachment** Property

The **background-repeat** Property



# CSS3 속성

## 4. 폰트 속성



폰트 속성이란?

글자와 관련된 스타일 속성

- **font-size**: 글자의 크기를 지정
- **font-family**: 폰트 종류를 지정
- **font-style**: 폰트 기울기 지정
- **font-weight**: 폰트 두께 지정
- **text-align**: 글자의 정렬 지정

# CSS3 속성

## 5. 위치 속성

절대적  
위치 좌표

요소의 x, y 좌표를 설정해  
절대적으로 위치를 지정

상대적  
위치 좌표

요소를 입력한 순서대로  
상대적으로 위치를 지정

### [ position 속성값 ]

- **static**: 태그가 위에서 아래로 순서대로 배치
- **relative**: 초기 위치 상태에서 상하좌우로 위치를 이동
- **absolute**: 절대적 위치 좌표를 설정
- **fixed**: 화면을 기준으로 절대적 위치 좌표를 설정
- **overflow**: 내부의 요소가 부모의 범위를 벗어날 때  
어떻게 처리할지 지정
- **hidden**: 영역을 벗어나는 부분을 보이지 않게 함
- **scroll**: 영역을 벗어나는 부분을 스크롤로 지정

# CSS3 속성

## 6. float 속성



### float 속성이란?

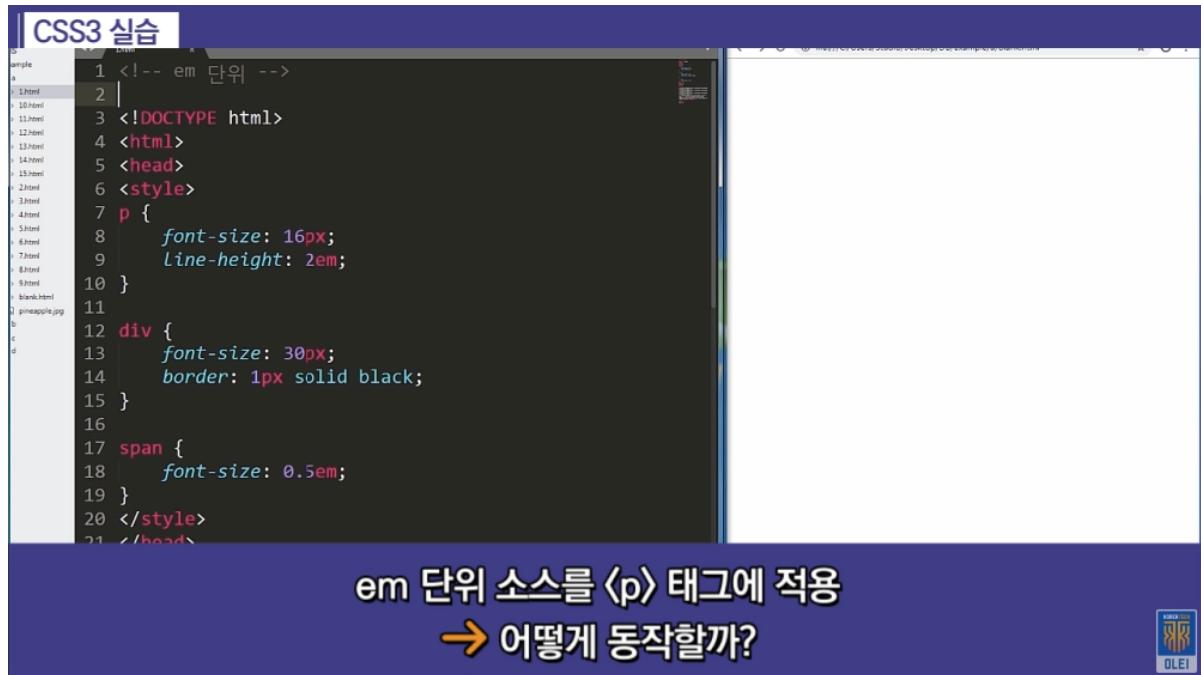
- 태그 요소가 어떻게 떠다니는지 지정
- 절대적 위치를 사용하는 요소는 float 속성을 무시함

### [ float 속성값 ]

- **left**: 태그를 왼쪽에 붙임
- **right**: 태그를 오른쪽에 붙임
- 속성 확인은?

[https://www.w3schools.com/cssref/playit.asp?filename=playcss\\_float&preval=none](https://www.w3schools.com/cssref/playit.asp?filename=playcss_float&preval=none)

# CSS3 실습



The screenshot shows a code editor window titled "CSS3 실습". The code is as follows:

```
1 <!-- em 단위 -->
2 |
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <style>
7 p {
8   font-size: 16px;
9   line-height: 2em;
10 }
11
12 div {
13   font-size: 30px;
14   border: 1px solid black;
15 }
16
17 span {
18   font-size: 0.5em;
19 }
20 </style>
21 </head>
```

Below the code, there is explanatory text in Korean:

em 단위 소스를 <p> 태그에 적용  
→ 어떻게 동작할까?

OLEI

※ 실습 영상은 콘텐츠에서 확인할 수 있습니다.

# 핵심정리

## 1. CSS3 단위 및 그레디언트

- CSS 속성은 width, margin, padding, font-size, border-width 등과 같이 길이값을 사용함
- 단위는 절대 단위와 상대 단위 2가지 유형이 있음
  - ✓ 절대 길이 단위: 화면크기가 다양하기 때문에 사용하지 않는 것을 권장함
  - ✓ 상대 길이 단위: 다른 길이 등록 정보와 관련된 길이를 지정함
- 색상의 단위 종류: HEX 코드, RGB, RGBA, HSL, HSLA
- URL 함수를 이용하여 이미지 파일이나 폰트 파일을 불러올 때 사용함

## 2. CSS3 속성

- 가시 속성은 태그가 화면에 보이는 방식을 지정하는 속성임
- 박스 속성은 웹페이지의 레이아웃을 구성할 때 가장 중요한 스타일 속성임
- margin, border, padding, width, height로 구성됨
- 배경 속성을 이용하여 배경 이미지를 변경하거나 크기 그리고 반복 설정할 수 있음
- 폰트 속성을 이용하여 폰트의 종류, 크기, 기울기, 두께 정렬 방식을 결정할 수 있음
- 위치 속성은 절대적 위치 좌표와 상대적 위치 좌표를 이용하여 요소의 위치를 결정할 수 있음
- float 속성은 태그 요소가 어떻게 떠다니는지 지정할 수 있으며 절대적 위치를 사용하는 요소는 float 속성을 무시함

# 화면 구현

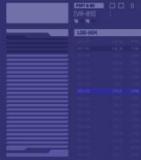
part 2

레이아웃

- GENERAL INFO
- DOCUMENTS
- SECTOR-01 [A25X]
- SECTOR-02 [A25X]
- ▶ [A35]



```
$('.menuBttn > .go').on("click",function(){  
    var menulid = $(this).attr('id');  
    switch (menulid) {  
        case "goAbout":  
            $('.slideWrap').slick('slickGoTo',1);  
            break;  
        case "goContact":  
            $('.slideWrap').slick('slickGoTo',2);  
            break;  
        case "goHistory":  
            $('.slideWrap').slick('slickGoTo',3);  
            break;  
        setInterval(function(){  
            if($('.smallLogo').css('display')=="none"){  
                $('.text-1').fadeOut(700);  
                $('.text-2').fadeOut(700);  
                $('.smallLogo').delay(2000).fadeIn(700);  
            }  
        },1000);  
    }  
});
```



KOREATECH  
온라인평생교육원



## 학습목표

- 웹페이지 레이아웃을 구성하는 방법에 대해 이해하고, 적용할 수 있다.
- 반응형 웹페이지를 이해하고, 적용할 수 있다.

## 학습내용

- 웹페이지 레이아웃
- 반응형 웹페이지 레이아웃
- 레이아웃 실습

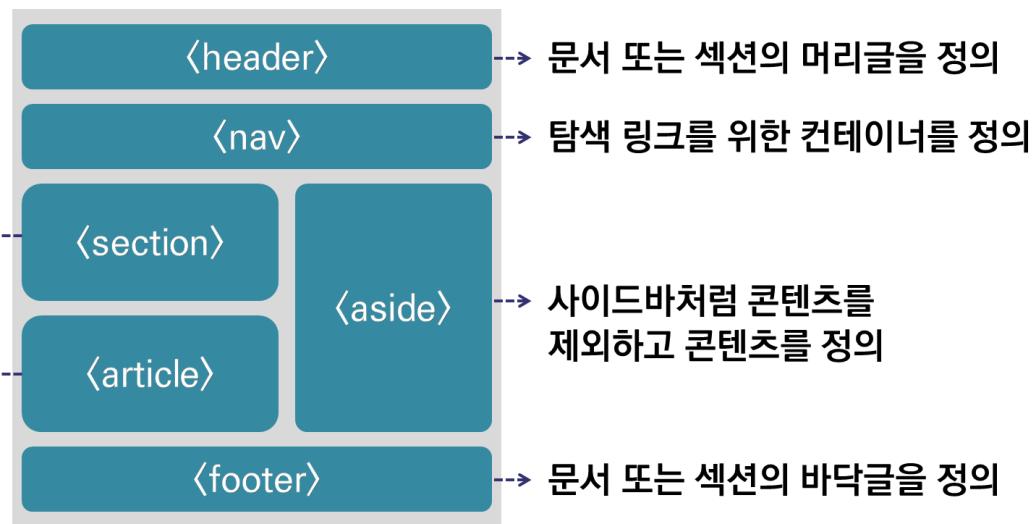
# 웹페이지 레이아웃

## 1. 레이아웃 요소



### 레이아웃 요소는?

- 웹 사이트는 여러 열로 구분되어 내용을 표시함
- HTML5에서는 시멘틱 구조로 영역을 구분함



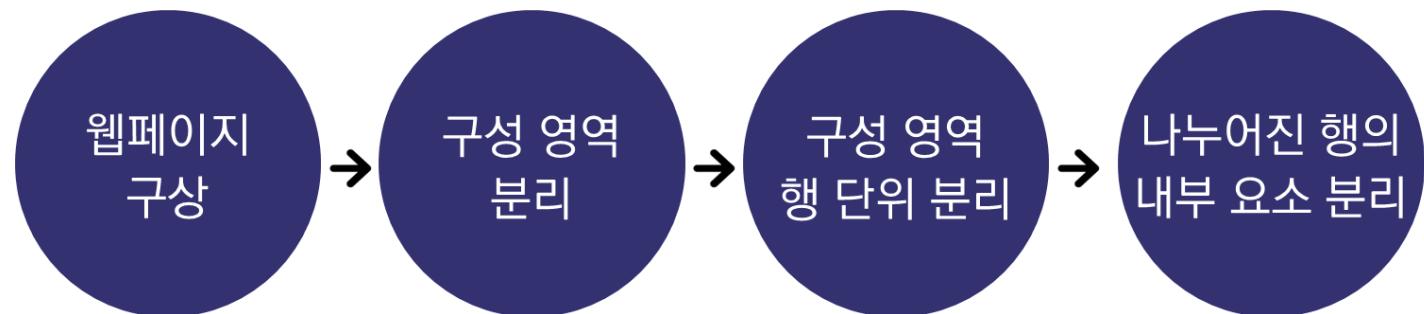
- `<details>`: 추가 세부 정보 정의
- `<summary>`: `<details>` 요소의 제목을 정의

# 웹페이지 레이아웃

## 1. 레이아웃 요소



영역 분할 순서는?



# 웹페이지 레이아웃

## 2. 웹페이지 내용 작성 예시

### 1) 시멘틱 태그를 이용하여 구성 영역을 분리

입력

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

<header>
<h1>header</h1>
</header>

<nav>
nav
</nav>

<section>
<h1>Section</h1>
<p>section content</p>
</section>

<footer>
footer
</footer>

</body>
</html>
```

결과

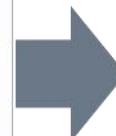
**header**

nav

**Section**

section content

footer



# 웹페이지 레이아웃

## 3. 레이아웃 CSS 적용

### 1) 선택자를 이용하여 각각의 영역을 지정

#### 입력

```
<!DOCTYPE html> <h1>City Gallery</h1>
<html> </header>
<head>
<style>
header {
nav {
}
section {
}
footer {
}
</style>
</head>
<body>
<header>
```

```
<h1>London</h1>
<p>London is the capital city of England. It is the
most populous city in the United Kingdom,
with a metropolitan area of over 13 million
inhabitants.</p>
<p>Standing on the River Thames, London has been a
major settlement for two millennia,
its history going back to its founding by the Romans,
who named it Londinium.</p>
</section>
<footer>
Copyright @ W3Schools.com
</footer>
</body>
</html>
```



# 웹페이지 레이아웃

## 3. 레이아웃 CSS 적용

### 1) 선택자를 이용하여 각각의 영역을 지정

결과

### City Gallery

London  
Paris  
Tokyo

#### London

London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.

Copyright @ W3Schools.com

# 웹페이지 레이아웃

## 3. 레이아웃 CSS 적용

### 2) header 영역 스타일 적용

입력

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

<style>
header {
    background-color:black;
    color:white;
    text-align:center;
    padding:5px;
}
nav {

}
section {

}
footer {

}
</style>
</head>

<body>
<header>
<h1>City Gallery</h1>
</header>
```



## 웹페이지 레이아웃

### 3. 레이아웃 CSS 적용

#### 2) header 영역 스타일 적용

결과

## City Gallery

London  
Paris  
Tokyo

### London

London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.

Copyright @ W3Schools.com

# 웹페이지 레이아웃

## 3. 레이아웃 CSS 적용

### 3) nav 영역 스타일 적용

입력

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

<style>
header {
    background-color:black;
    color:white;
    text-align:center;
    padding:5px;
}
nav {
    line-height:30px;
    background-color:#eeeeee;
    height:300px;
    width:100px;
    float:left;
    padding:5px;
}
section {

}
footer {

}
</style>
</head>
```



## 웹페이지 레이아웃

### 3. 레이아웃 CSS 적용

#### 3) nav 영역 스타일 적용

결과

## City Gallery

London	<b>London</b>  London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.  Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.  Copyright @ W3Schools.com
Paris	
Tokyo	

# 웹페이지 레이아웃

## 3. 레이아웃 CSS 적용

### 4) section 영역 스타일 적용

입력

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

<style>
header {
    background-color:black;
    color:white;
    text-align:center;
    padding:5px;
}

nav {
    line-height:30px;
    background-color:#eeeeee;
    height:300px;
    width:100px;
    float:left;
    padding:5px;
}

section {
    width:350px;
    float:left;
    padding:10px;
}

footer {

}
</style>
</head>
```



## 웹페이지 레이아웃

### 3. 레이아웃 CSS 적용

#### 4) section 영역 스타일 적용

결과

## City Gallery

London

Paris

Tokyo

### London

London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.

Copyright @ W3Schools.com

# 반응형 웹페이지 레이아웃

## 1. 반응형 웹



### 반응형 웹 활용 시 주의사항은?

- 웹사이트 레이아웃을 만들 때 유저가 사용하는 모니터의 화면 해상도 고려
- **해상도를 고려하지 않고 제작할 경우 콘텐츠를 보는 데 불편함**
- 해상도에 따라 가로폭이나 배치를 변경하여 가독성을 높이는 방법
- @media 이용하여 화면 해상도에 따라 다른 레이아웃 지정

## 2. 미디어 쿼리

- @media로 <style> ~ </style>에서 선언하여 미디어 유형·장치에 따른 스타일을 적용함
- 뷰 포트의 너비와 높이
- 장치의 너비와 높이

# 반응형 웹페이지 레이아웃

## 2. 미디어 쿼리

- 방향 결정(가로, 세로 모드의 태블릿 혹은 스마트폰)
- resolution
- 미디어 쿼리를 사용하면 데스크톱, 랩톱, 태블릿 및 스마트폰에 맞춤형 CSS를 사용
- 선언

```
@media not|only mediatype and (media feature and|or|not mediafeature) {  
    CSS-Code;  
}
```

### 〈 미디어 쿼리 선언 〉

- **not**: 전체 미디어 쿼리의 의미를 되돌림
- **only**: 미디어 쿼리를 지원하지 않는 브라우저 지정된 스타일을 적용하지 못하게 함
- **and**: 미디어 기능을 미디어 유형 또는 다른 미디어 기능과 결합

# 반응형 웹페이지 레이아웃

## 2. 미디어 쿼리

- **Media types**

<b>Value</b>	<b>Description</b>
all	Default. Used for all media type devices
print	Used for printers
screen	Used for computer screens, tablets, smart-phones etc.
speech	Used for screenreaders that "reads" the page out loud

### 〈 미디어의 종류 〉

## 3. 반응형 웹 예시

### 1) 뷰포트 meta 태그 추가



**뷰포트 meta 태그란?**

- 웹 브라우저에 특별한 정보를 제공하는 태그

# 반응형 웹페이지 레이아웃

## 3. 반응형 웹 예시

### 1) 뷰포트 meta 태그 추가

```
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
```

〈 뷰포트 meta 태그 〉

브라우저의 화면에 대한 특별한 정보를 제공하기 위하여  
뷰포트 meta 태그 사용

# 반응형 웹페이지 레이아웃

## 3. 반응형 웹 예시

### 2) CSS 스타일 적용

#### 입력

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta name="viewport" content="width=device-width, initial-
scale=1">
<style>
* {
    box-sizing: border-box;
}
header {
    background-color:black;
    color:white;
    text-align:center;
    padding:5px;
}
nav {
    float: left;
    width: 30%;
    height: 300px;
    background-color:#eeeeee;
    padding: 20px;
    line-height:30px;
}
nav ul {
    list-style-type: none;
    padding: 0;
}
article {
    float: left;
    padding: 20px;
    width: 70%;
    height: 300px;
}
footer {
    background-color: black;
    padding: 10px;
    text-align: center;
    clear : both;
    color: white;
}
</style>
</head>
```



# 반응형 웹페이지 레이아웃

## 3. 반응형 웹 예시

### 2) CSS 스타일 적용

결과

## Cities

<u><a href="#">London</a></u>	<b>London</b>
<u><a href="#">Paris</a></u>	London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.
<u><a href="#">Tokyo</a></u>	Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.

Copyright @ W3Schools.com

# 반응형 웹페이지 레이아웃

## 3. 반응형 웹 예시

### 3) 화면 사이즈 변경 후 결과 확인

입력

Cities

[London](#)

[Paris](#)

[Tokyo](#)

**London**

London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.

Copyright @ W3Schools.com



# 반응형 웹페이지 레이아웃

## 3. 반응형 웹 예시

### 3) 화면 사이즈 변경 후 결과 확인

결과

width : 600px 이하

Cities

[London](#)

[Paris](#)

[Tokyo](#)

#### London

London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.

Copyright @ W3Schools.com

# 레이아웃 실습

레이아웃 실습

```
1 <!-- Layout -->
2 |
3 <!DOCTYPE html>
4 <html>
5 <body>
6
7 <header>
8 <h1>City Gallery</h1>
9 </header>
10
11 <nav>
12 London<br>
13 Paris<br>
14 Tokyo
15 </nav>
16
17 <section>
18 <h1>London</h1>
19 <p>London is the capital city of England. It is
the most populous city in the United Kingdom, with
a metropolitan area of over 12 million inhabitants.
```

레이아웃 작업이 제일 먼저 되어야  
전체적인 화면 설계가 가능함



※ 실습 영상은 콘텐츠에서 확인할 수 있습니다.

# 핵심정리

## 1. 웹페이지 레이아웃

- 웹사이트는 여러 열로 구분되어 내용을 표시함
- HTML5에서는 시멘틱 구조로 영역을 구분함
- 영역 분할
  - ✓ 웹페이지 구성 → 구성 영역 분리 → 구성 영역을 행 단위로 분리  
→ 나누어진 행의 내부 요소를 분리

## 2. 반응형 웹페이지 레이아웃

- 웹사이트 레이아웃을 만들 때 유저가 사용하는 모니터의 화면 해상도를 고려해야 함
- 고려하지 않고, 제작할 경우 콘텐츠를 보는데 불편함
- 반응형 웹은 사용자의 해상도를 고려하여 화면을 구성함
- @media를 이용하여 해상도에 따라 레이아웃을 지정함
- 브라우저의 화면에 대한 특별한 정보를 제공하기 위하여 뷰포트 meta 태그를 사용함

# 화면 구현

part 2

## 자바스크립트



KOREATECH  
온라인평생교육원



## 학습목표

- 자바스크립트에 대한 개념을 설명할 수 있다.
- 함수와 제어문을 이해하고, 활용할 수 있다.

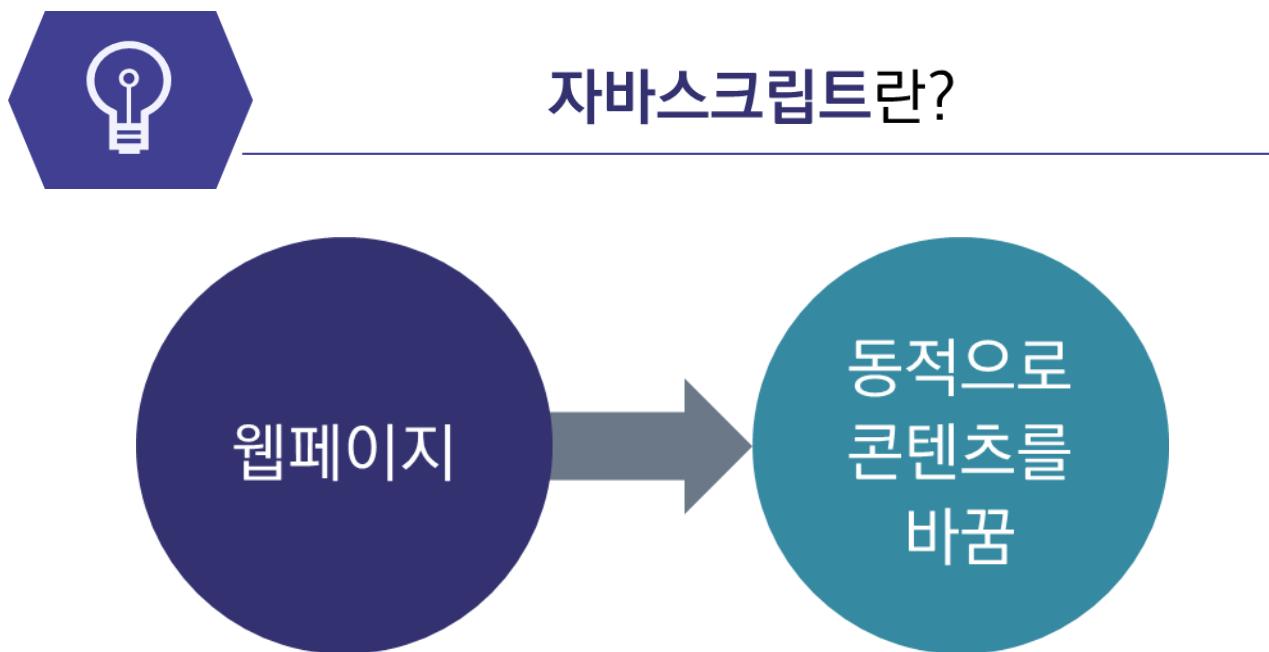
## 학습내용

- 자바스크립트(JavaScript)
- 함수와 제어문

# 자바스크립트 (JavaScript)

## 1. 자바스크립트 (JavaScript) 기초

### 1) 자바스크립트란?



### 프로그래밍적으로 제어하기 위해 고안된 언어

- 객체 기반의 스크립트 언어
- 타입을 명시할 필요가 없는 인터프리터 언어
- 객체 지향형 프로그래밍과 함수형 프로그래밍을 모두 표현 가능

# 자바스크립트(JavaScript)

## 1. 자바스크립트(JavaScript) 기초

### 1) 자바스크립트란?

- 객체 기반의 스크립트 언어
- 타입을 명시할 필요가 없는 인터프리터 언어
- 객체 지향형 프로그래밍과 함수형 프로그래밍을 모두 표현 가능

\* 인터프리터: 프로그램을 해석하는 방법 중 하나로, 사람이 이해할 수 있는 고급언어로 작성된 코드를 한 단계씩 해석하여 실행시키는 방법

# 자바스크립트(JavaScript)

## 1. 자바스크립트(JavaScript) 기초

### 2) 문법

[ 실행문은 세미콜론 ‘;’ 으로 구분됨 ]

```
var x = 10;
```

```
var y = 13;
```

[ 대소문자를 구분함 ]

```
var js = 10;
```

```
var JS = 20;
```

→ js와 JS는 서로 다른 변수로 인식함

[ 상수(Literal)는 직접 표현되는 값 그 자체를 의미함 ]

```
12. // 숫자 상수(Literal)
```

```
“자바script” // 문자열 상수(Literal)
```

```
true // boolean 상수(Literal)
```

# 자바스크립트(JavaScript)

## 1. 자바스크립트(JavaScript) 기초

### 2) 문법

#### [ 식별자 작성 방식 ]

##### Camel Case 방식

- 식별자가 여러 단어로 이루어질 경우에 첫 번째 단어는 모두 소문자로 작성
- 그 다음 단어부터는 첫 문자만 대문자로 작성하는 방식

##### Underscore Case 방식

- 식별자를 이루는 단어들을 소문자로만 작성
- 그 단어들은 언더스코어(\_)로 연결하는 방식

```
var firstVar = 10; // Camel Case
```

```
var second_var = 20; // Undersore Case
```

#### [ 주석: 코드 내에 삽입된 일종의 설명문(실행은 되지 않음) ]

- 한 줄 주석

```
// 주석문
```

- 여러 줄 주석

```
/* 주석문 */
```

## 자바스크립트(JavaScript)

### 1. 자바스크립트(JavaScript) 기초

#### 3) 출력

- HTML 페이지에 출력을 할 수 있음

방법이 무엇인가요?



# 자바스크립트 (JavaScript)

## 1. 자바스크립트 (JavaScript) 기초

### 3) 출력

#### [ window.alert() 메소드 ]

##### 입력

```
<!DOCTYPE html>
<html lang="ko">

<head>
    <meta charset="UTF-8">
    <title>JavaScript Output</title>
</head>

<body>

    <h1>Window 객체의 alert() 메소드</h1>
    <button onclick="alertDialogBox()">alert 대화 상자
    </button>

    <script>
        function alertDialogBox() {
            alert("확인을 누를 때까지 다른 작업을 할 수 없어요!");
        }
    </script>

</body>

</html>
```



# 자바스크립트 (JavaScript)

## 1. 자바스크립트 (JavaScript) 기초

### 3) 출력

#### [ window.alert() 메소드 ]

결과

### Window 객체의 alert() 메소드

alert 대화 상자



이 페이지에 삽입된 페이지 내용:

확인을 누를 때까지 다른 작업을 할 수 없어요!

확인

# 자바스크립트 (JavaScript)

## 1. 자바스크립트 (JavaScript) 기초

### 3) 출력

#### [ HTML DOM 요소를 이용한 innerHTML 프로퍼티 ]

##### 입력

```
<!DOCTYPE html>
<html lang="ko">

<head>
    <meta charset="UTF-8">
    <title>JavaScript Output</title>
</head>

<body>

    <h1>HTML DOM 요소를 이용한 innerHTML 프로퍼티</h1>
    <p id="text">이 문장을 바꿀 것입니다!</p>

    <script>
        var str = document.getElementById("text");
        str.innerHTML = "이 문장으로 바뀌었습니다!";
    </script>

</body>

</html>
```



# 자바스크립트 (JavaScript)

## 1. 자바스크립트 (JavaScript) 기초

### 3) 출력

[ HTML DOM 요소를 이용한 innerHTML 프로퍼티 ]

결과

**HTML DOM 요소를 이용한 innerHTML  
프로퍼티**

이 문장으로 바뀌었습니다!

# 자바스크립트 (JavaScript)

## 1. 자바스크립트 (JavaScript) 기초

### 3) 출력

#### [ document.write() 메소드 ]

입력

```
<!DOCTYPE html>
<html lang="ko">

    <head>
        <meta charset="UTF-8">
        <title>JavaScript Output</title>
    </head>

    <body>

        <h1>Document 객체의 write() 메소드</h1>

        <script>
            document.write(4 * 5);
        </script>

    </body>

</html>
```



결과

Document 객체의 write() 메소드

# 자바스크립트 (JavaScript)

## 1. 자바스크립트 (JavaScript) 기초

### 3) 출력

#### [ document.write() 메소드 ]

##### 입력

```
<!DOCTYPE html>
<html lang="ko">

  <head>
    <meta charset="UTF-8">
    <title>JavaScript Output</title>
  </head>

  <body>

    <h1>Document 객체의 write() 메소드</h1>

    <button onclick="document.write(4 * 5)">버튼을 눌러보세요!</button>

  </body>

</html>
```



##### 결과

## Document 객체의 write() 메소드

버튼을 눌러보세요!



20

# 자바스크립트 (JavaScript)

## 1. 자바스크립트 (JavaScript) 기초

### 4) 적용 방법

#### [ 내부 자바스크립트 코드로 적용 ]

- <head>, <body> 어디에 추가하던 상관없음

### 입력

```
<!DOCTYPE html>
<html lang="ko">

<head>
    <meta charset="UTF-8">
    <title>JavaScript Apply</title>
    <script>
        function printDate() {
            document.getElementById("date").innerHTML = Date();
        }
    </script>
</head>

<body>
    <h3>head 태그 내의 자바스크립트</h3>
    <button onclick="printDate()">현재 날짜와 시간 표시하기!</button>
    <p id="date"></p>
</body>
</html>
```



# 자바스크립트 (JavaScript)

## 1. 자바스크립트 (JavaScript) 기초

### 4) 적용 방법

#### [ 내부 자바스크립트 코드로 적용 ]

- <head>, <body> 어디에 추가하던 상관없음

결과

**head 태그 내의 자바스크립트**

현재 날짜와 시간 표시하기!

Thu Sep 06 2018 16:38:01 GMT+0900 (한국 표준시)

# 자바스크립트 (JavaScript)

## 1. 자바스크립트 (JavaScript) 기초

### 4) 적용 방법

#### [ 외부 자바스크립트 파일로 적용 ]

- HTML 코드와 자바스크립트 코드를 분리하면 유지보수 및 가독성이 좋음

```
<!DOCTYPE html>
<html lang="ko">

<head>
  <meta charset="UTF-8">
  <title>JavaScript Apply</title>
  <script src="example.js"></script>
</head>

<body>
  <h1>외부 자바스크립트 파일</h1>
  <p>자바스크립트를 이용하면 현재 날짜와 시간 정보에도 손쉽게 접근할 수 있어요!</p>
  <button onclick="printDate()">현재 날짜와 시간 표시하기!</button>
  <p id="date"></p>
</body>

</html>
```

# 자바스크립트 (JavaScript)

## 2. 타입과 연산자

### 1) 타입

#### [ 원시 타입 ]

- 숫자 number

```
var num = 10; // 숫자
```

- 문자열 string

```
var hello = “안녕”; // 문자열
```

- 불리언 boolean

```
var bool = false // boolean 타입 false
```

- 심볼 symbol

```
var sym = Symbol(“javascript”); // symbol 타입
```

```
var symObj = Object(sym); // object 타입
```

- undefined / null

```
var str; // undefined
```

```
var str = null; // null
```

# 자바스크립트 (JavaScript)

## 2. 타입과 연산자

### 1) 타입

#### [ 객체 타입 ]

- 객체 object

```
var person = { name: “김씨”, age: 20 }; // 객체의 생성  
// 객체의 프로퍼티 참조  
document.getElementById(“result”).innerHTML =  
“이름: ” + person.name + “나이: ” + person.age ;
```

#### [ 타입 변환 ]

- Number(“10”): 숫자로 타입 변환
- String(true): 문자열로 타입 변환
- Boolean(0): 불리언으로 타입 변환
- Object(2): new Number(3)와 동일한 결과로  
숫자 2 객체 변환
- parseInt(): 정수형 타입으로 변환
- parseFloat(): 부동소수점 타입으로 변환

# 자바스크립트 (JavaScript)

## 2. 타입과 연산자

### 1) 타입

- 변수 선언: var 키워드를 사용하여 변수를 선언

```
var month; // month라는 이름의 변수 선언
```

```
date = 25; // date라는 이름의 변수를 묵시적으로 선언
```

- 변수 초기화

```
var month; // 변수의 선언
```

```
var date = 25; // 변수의 선언과 동시에 초기화
```

```
month = 12; // 변수의 초기화
```

```
var month, date; // 여러 변수를 한 번에 선언
```

```
var hours = 7, minutes = 15; // 여러 변수를 선언과 동시에 초기화
```

```
month = 10, date = 5; // 여러 변수를 한 번에 초기화
```

# 자바스크립트 (JavaScript)

## 2. 타입과 연산자

### 2) 연산자

#### [ 산술 연산자 ]

+

- 왼쪽 피연산자의 값에 오른쪽 피연산자의 값을 더함

-

- 왼쪽 피연산자의 값에서 오른쪽 피연산자의 값을 뺌

\*

- 왼쪽 피연산자의 값에 오른쪽 피연산자의 값을 곱함

/

- 왼쪽 피연산자의 값을 오른쪽 피연산자의 값으로 나눔

%

- 왼쪽 피연산자의 값을 오른쪽 피연산자의 값으로 나눈 후, 그 나머지를 반환함

# 자바스크립트 (JavaScript)

## 2. 타입과 연산자

### 2) 연산자

#### [ 대입 연산자 ] 1/2

=

- 왼쪽 피연산자에 오른쪽 피연산자의 값을 대입함

+=

- 왼쪽 피연산자의 값에 오른쪽 피연산자의 값을 더한 후,  
그 결과값을 왼쪽 피연산자에 대입함

-=

- 왼쪽 피연산자의 값에서 오른쪽 피연산자의 값을 뺀 후, 그 결과값을  
왼쪽 피연산자에 대입함

# 자바스크립트 (JavaScript)

## 2. 타입과 연산자

### 2) 연산자

[ 대입 연산자 ] 2/2

\*=

- 왼쪽 피연산자의 값에 오른쪽  
피연산자의 값을 곱한 후, 그 결과값을  
왼쪽 피연산자에 대입함

/=

- 왼쪽 피연산자의 값을 오른쪽  
피연산자의 값으로 나눈 후,  
그 결과값을 왼쪽 피연산자에 대입함

%=

- 왼쪽 피연산자의 값을 오른쪽  
피연산자의 값으로 나눈 후, 그 나머지를  
왼쪽 피연산자에 대입함

# 자바스크립트 (JavaScript)

## 2. 타입과 연산자

### 2) 연산자

#### [ 증감 연산자 ]

`++x`

- 먼저 피연산자의 값을 1 증가시킨 후에 해당 연산을 진행함

`x++`

- 먼저 해당 연산을 수행하고 나서, 피연산자의 값을 1 증가시킴

`--x`

- 먼저 피연산자의 값을 1 감소시킨 후에 해당 연산을 진행함

`x--`

- 먼저 해당 연산을 수행하고 나서, 피연산자의 값을 1 감소시킴

# 자바스크립트 (JavaScript)

## 2. 타입과 연산자

### 2) 연산자

#### [ 비교 연산자 ] 1/2

==

- 왼쪽 피연산자와 오른쪽 피연산자의 값이 같으면 참을 반환함

====

- 왼쪽 피연산자와 오른쪽 피연산자의 값이 같고, 같은 타입이면 참을 반환함

!=

- 왼쪽 피연산자와 오른쪽 피연산자의 값이 같지 않으면 참을 반환함

!==

- 왼쪽 피연산자와 오른쪽 피연산자의 값이 같지 않거나, 타입이 다르면 참을 반환함

# 자바스크립트 (JavaScript)

## 2. 타입과 연산자

### 2) 연산자

[ 비교 연산자 ] 2/2

>

- 왼쪽 피연산자의 값이 오른쪽 피연산자의 값보다 크면 참을 반환함
- 왼쪽 피연산자의 값이 오른쪽 피연산자의 값보다 크거나 같으면 참을 반환함

>=

- 왼쪽 피연산자의 값이 오른쪽 피연산자의 값보다 작으면 참을 반환함
- 왼쪽 피연산자의 값이 오른쪽 피연산자의 값보다 작거나 같으면 참을 반환함

<

<=

- 왼쪽 피연산자의 값이 오른쪽 피연산자의 값보다 작으면 참을 반환함
- 왼쪽 피연산자의 값이 오른쪽 피연산자의 값보다 작거나 같으면 참을 반환함

# 자바스크립트 (JavaScript)

## 2. 타입과 연산자

### 2) 연산자

#### [ 논리 연산자 ]

&amp;&amp;

- 논리식이 모두 참이면 참을 반환함  
(논리 AND 연산)

||

- 논리식 중에서 하나라도 참이면 참을 반환함(논리 OR 연산)

!

- 논리식의 결과가 참이면 거짓을,  
거짓이면 참을 반환함(논리 NOT 연산)

# 함수와 제어문

## 1. 함수

### 1) 자바스크립트 함수



자바스크립트 함수란?

- 하나의 특별한 목적의 작업을 수행하도록 설계된 독립적인 블록
- 자바스크립트에서는 함수도 하나의 데이터 타입



자바스크립트 함수 특징은?

- 함수를 변수에 대입하거나 함수에 프로퍼티를 지정하는 것도 가능
- 다른 함수 내에 중첩되어 정의 가능

# 함수와 제어문

## 1. 함수

### 1) 자바스크립트 함수

function 키워드로 선언	함수의 이름	구성
괄호 안에 쉼표(,)로 구분되는 함수의 매개변수(parameter)	중괄호({ })로 둘러싸인 자바스크립트 실행문	

```
function 함수이름(매개변수1, 매개변수2,...) {  
    함수가 호출되었을 때 실행하고자 하는 실행문;  
}
```

### 2) 반환문



반환문이란?

- **return 키워드** 사용
- 특정한 목적을 수행하고 그 결과를 반환문을 통해  
실행된 결과를 전달받음
- 반환문은 함수의 실행을 중단할 수 있음
- 배열이나 객체를 포함한 모든 타입의 값을 반환할 수 있음

# 함수와 제어문

## 1. 함수

### 2) 반환문

#### 입력

```
<!DOCTYPE html>
<html>
<body>
    <h1>함수 반환문</h1>
    <script>
        function multiNum(x, y) {
            return x * y;           // x와 y를 곱한 결과를 반환
        }

        var num = multiNum(3, 4);
        // multiNum() 함수가 호출된 후, 그 반환 값이 변수 num에 저장
        document.write(num);
    </script>
</body>
</html>
```



#### 결과

## 함수 반환문

12

# 함수와 제어문

## 1. 함수

### 3) 함수의 유효 범위

#### 입력

```
<!DOCTYPE html>
<html>
<body>
  <h1>함수의 유효 범위</h1>
  <script>
    // x, y를 전역 변수로 선언함.
    var x = 10, y = 20;
    // sub()를 전역 함수로 선언함.
    function sub() {
      return x - y;      // 전역 변수인 x, y에 접근함.
    }
    document.write("전역 함수에서 x - y의 값은 " + sub() + "입니다.<br>");

    // parentFunc()을 전역 함수로 선언함.
    function parentFunc() {
      var x = 1, y = 2; // 전역 변수와 같은 이름으로 선언하여 전역 변수의 범위를 제한함.
      function add() { // add() 함수는 내부 함수로 선언됨.
        return x + y; // 전역 변수가 아닌 지역 변수 x, y에 접근함.
      }

      return add();
    }
    document.write("내부 함수에서 x + y의 값은 " + parentFunc() + "입니다.<br>");
  </script>
</body>
</html>
```

#### 결과

## 함수의 유효 범위

전역 함수에서  $x - y$ 의 값은 -10입니다.  
내부 함수에서  $x + y$ 의 값은 3입니다.

# 함수와 제어문

## 1. 함수

### 4) 함수 호이스팅



#### 함수 호이스팅이란?

- 자바스크립트 함수 안에 있는 모든 변수의 선언은 함수의 맨 처음으로 이동한 것처럼 동작

#### 입력

```
<!DOCTYPE html>
<html>
<body>
  <h1>함수 호이스팅</h1>
  <script>
    var globalNum = 10;      // globalNum을 전역 변수로 선언함.
    function printNum() {
      document.write("지역 변수 globalNum 선언 전의 globalNum의 값은 " + globalNum + "<br>");
      var globalNum = 20;    // globalNum을 지역 변수로 선언함.
      document.write("지역 변수 globalNum 선언 후의 globalNum의 값은 " + globalNum + "<br>");
    }
    printNum();
  </script>
</body>
</html>
```



#### 결과

#### 함수 호이스팅

지역 변수 globalNum 선언 전의 globalNum의 값은 undefined  
 지역 변수 globalNum 선언 후의 globalNum의 값은 20

## 함수와 제어문

### 2. 제어문

#### 1) 조건문



조건문이란  
무엇인가요?

**프로그램 내에서 주어진 표현식의 결과에 따라  
별도의 명령을 수행하도록 제어하는 실행문**

# 함수와 제어문

## 2. 제어문

### 1) 조건문 | (1) if 문

- 표현식의 결과가 참(True)이면? → 주어진 실행문을 실행
- 표현식의 결과가 거짓(False)이면? → 아무것도 실행하지 않음

```
if (표현식){  
    표현식의 결과가 참일 때 실행하고자 하는 실행문;  
}
```

# 함수와 제어문

## 2. 제어문

### 1) 조건문 | (1) if 문

입력

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

<h3>if 문</h3>

<script>
    var x = 5, y = 4;
    if (x == y) {
        document.write("x와 y는 같습니다.");
    }
    if (x < y) {
        document.write("x가 y보다 작습니다.");
    }
    if (x > y) {
        document.write("x가 y보다 큽니다.");
    }
</script>

</body>
</html>
```



결과

if 문

x가 y보다 큽니다.

# 함수와 제어문

## 2. 제어문

1) 조건문 | (2) If - else 문

- 표현식의 결과가 참(True)이면? → 주어진 실행문을 실행
- 표현식의 결과가 거짓(False)이면? → 주어진 실행문을 실행

if (표현식) {  
 표현식의 결과가 참일 때 실행하고자 하는 실행문;

} else {  
 표현식의 결과가 거짓일 때 실행하고자 하는 실행문;  
}

## 함수와 제어문

### 2. 제어문

1) 조건문 | (2) If - else 문

#### 입력

```
<!DOCTYPE html>
<html>

<body>

<h1>if-else 문</h1>

<script>
    var x = 5, y = 4;
    if (x == y) {
        document.write("x와 y는 같습니다.");
    } else {
        if (x < y)
            document.write("x가 y보다 작습니다.");
        else
            document.write("x가 y보다 큽니다.");
    }
</script>

</body>

</html>
```



#### 결과

### if-else 문

x가 y보다 큽니다.

# 함수와 제어문

## 2. 제어문

### 1) 조건문 | (3) If - else if - else 문

- else if 문을 이용하여 다양한 조건을 설정 가능

if (표현식1){

  표현식1의 결과가 참일 때 실행하고자 하는 실행문;

} else if (표현식2){

  표현식2의 결과가 참일 때 실행하고자 하는 실행문;

} else {

  표현식1의 결과도 거짓이고, 표현식2의 결과도 거짓일 때 실행하고자 하는 실행문;

}

## 함수와 제어문

### 2. 제어문

#### 1) 조건문 | (3) If - else if - else 문

##### 입력

```
<!DOCTYPE html>
<html>

<body>

<h1>if - else if - else 문</h1>

<script>
    var x = 5, y = 4;
    if (x == y) {
        document.write("x와 y는 같습니다.");
    }
    else if (x < y) {
        document.write("x가 y보다 작습니다.");
    }
    else { // x > y인 경우
        document.write("x가 y보다 큽니다.");
    }
</script>

</body>

</html>
```



##### 결과

### if - else if - else 문

x가 y보다 큽니다.

# 함수와 제어문

## 2. 제어문

### 1) 조건문 | (4) 삼항 연산자에 의한 조건문

- If – else 문을 삼항 연산자를 이용하여 간단하게 표현 가능

표현식 ? 반환값1 : 반환값2

#### 입력

```
<!DOCTYPE html>
<html>

<body>
    <h1>삼항 연산자</h1>

    <script>
        var x = 5, y = 4;
        document.write( (x > y) ? "x가 y보다 큽니다" :"y가 x 보다 큽니다" );

    </script>

</body>
</html>
```



#### 결과

## 삼항 연산자

x가 y보다 큽니다

# 함수와 제어문

## 2. 제어문

### 1) 조건문 | (5) switch 문

- If – else 문과 같이 주어진 조건 값에 따라 프로그램이 다른 명령을 수행

```
switch (조건 값) {  
    case 값1:  
        조건 값이 값1일 때 실행하고자 하는 실행문;  
        break;  
  
    case 값2:  
        조건 값이 값2일 때 실행하고자 하는 실행문;  
        break;  
    ...  
    default:  
        조건 값이 어떠한 case 절에도 해당하지 않을 때 실행하고자 하는 실행문;  
        break;  
}
```

# 함수와 제어문

## 2. 제어문

### 2) 반복문

반복문이란  
무엇인가요?



프로그램내에서 같은 명령을 일정 횟수만큼 반복하여  
수행하도록 제어하는 실행문

# 함수와 제어문

## 2. 제어문

### 2) 반복문 | (1) while 문

- 특정 조건을 만족할 때까지 계속해서 주어진 실행문을 반복 실행

`while (표현식) {`

표현식의 결과가 참인 동안 반복적으로 실행하고자 하는 실행문;

}

### 입력

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

    <h3>while 문</h3>

    <script>
        var i = 1;

        while (i < 10) {
            document.write(i + "<br>");
            i++;
        }
    </script>

</body>

</html>
```

### 결과

## while 문

1  
2  
3  
4  
5  
6  
7  
8  
9

# 함수와 제어문

## 2. 제어문

### 2) 반복문 | (2) do - while 문

- 먼저 루프를 한번 실행한 후 표현식을 검사

do {

    표현식의 결과가 참인 동안 반복적으로 실행하고자 하는 실행문;  
} while (표현식);

### 입력

```
<!DOCTYPE html>
<html>

<body>

<h3>do - while 문</h3>

<script>
    var i = 5, j = 1;

    while (i < 10) {
        document.write("i : " + i + "<br>");
        i++;
    }

    do {
        document.write("j : " + (j++) + "<br>");
    } while (j > 3);
</script>

</body>

</html>
```

### 결과

#### do - while 문



```
i : 5
i : 6
i : 7
i : 8
i : 9
j : 1
```

# 함수와 제어문

## 2. 제어문

### 2) 반복문 | (3) for 문

- 초기식, 표현식, 증감식 모두 포함하고 있는 반복문

for (초기식; 표현식; 증감식){

    표현식의 결과가 참인 동안 반복적으로 실행하고자 하는 실행문;

}

### 입력

```
<!DOCTYPE html>
<html>

<body>

    <h3>for 문</h3>

    <script>
        for (var i = 1; i < 10; i++) {
            document.write(i + "<br>");
        }
    </script>

</body>

</html>
```

### 결과

#### for 문

1  
2  
3  
4  
5  
6  
7  
8  
9



# 함수와 제어문

## 2. 제어문

### 2) 반복문 | (4) for - in 문

- 반복문 루프마다 객체의 열거할 수 있는 프로퍼티(enumerable properties)의 이름을 지정한 변수에 대입

```
for (변수 in 객체) {  
    객체의 모든 열거할 수 있는 프로퍼티의 개수만큼 반복적으로  
    실행하고자 하는 실행문;  
}
```

# 함수와 제어문

## 2. 제어문

### 2) 반복문 | (4) for - in 문

#### 입력

```
<!DOCTYPE html>
<html>

<body>

<h3>for - in 문</h3>

<script>
    var arr = [1, 2, 3];

    for (var i = 0; i < arr.length; i++) {
        document.write(i + " ");
    }
    document.write("<br>");

    for (var i in arr) {
        document.write(i + " ");
    }
</script>

</body>

</html>
```

#### 결과

### for - in 문

0 1 2  
0 1 2

# 함수와 제어문

## 2. 제어문

### 2) 반복문 | (5) for - of 문

- 반복할 수 있는 객체(iterable objects – Array, Map, Set, arguments)를 순회

`for (변수 of 객체) {`

  객체의 모든 열거할 수 있는 프로퍼티의 개수만큼 반복적으로  
  실행하고자 하는 실행문;  
}

### 입력

```
<!DOCTYPE html>
<html>

<body>

<h3>for - of 문</h3>

<script>
  var arr = new Set([1, 1, 2, 2, 3, 3]);
  for (var value of arr) {
    document.write(value + " ");
  }
</script>

</body>

</html>
```

### 결과

**for - of 문**

1 2 3

# 핵심정리

## 1. 자바스크립트 (JavaScript)

- 개념
  - ✓ 웹페이지를 동적으로 콘텐츠를 바꾸고, 프로그래밍적으로 제어하기 위해 고안된 언어로서 객체기반의 스크립트 언어

- 자바스크립트의 실행문은 세미콜론(;) 구분하며 대소문자를 구분함
- 자바스크립트로 HTML 페이지에 출력력을 할 수 있음

- 방법
  - ✓ `window.alert()`
  - ✓ HTML DOM 요소를 이용한 `innerHTML`프로퍼티
  - ✓ `document.write(), console.log()`

- HTML 어디서든 자바스크립트를 사용할 수 있지만 외부 파일로 따로 관리하는 것이 유지보수 및 가독성에 좋음

# 핵심정리

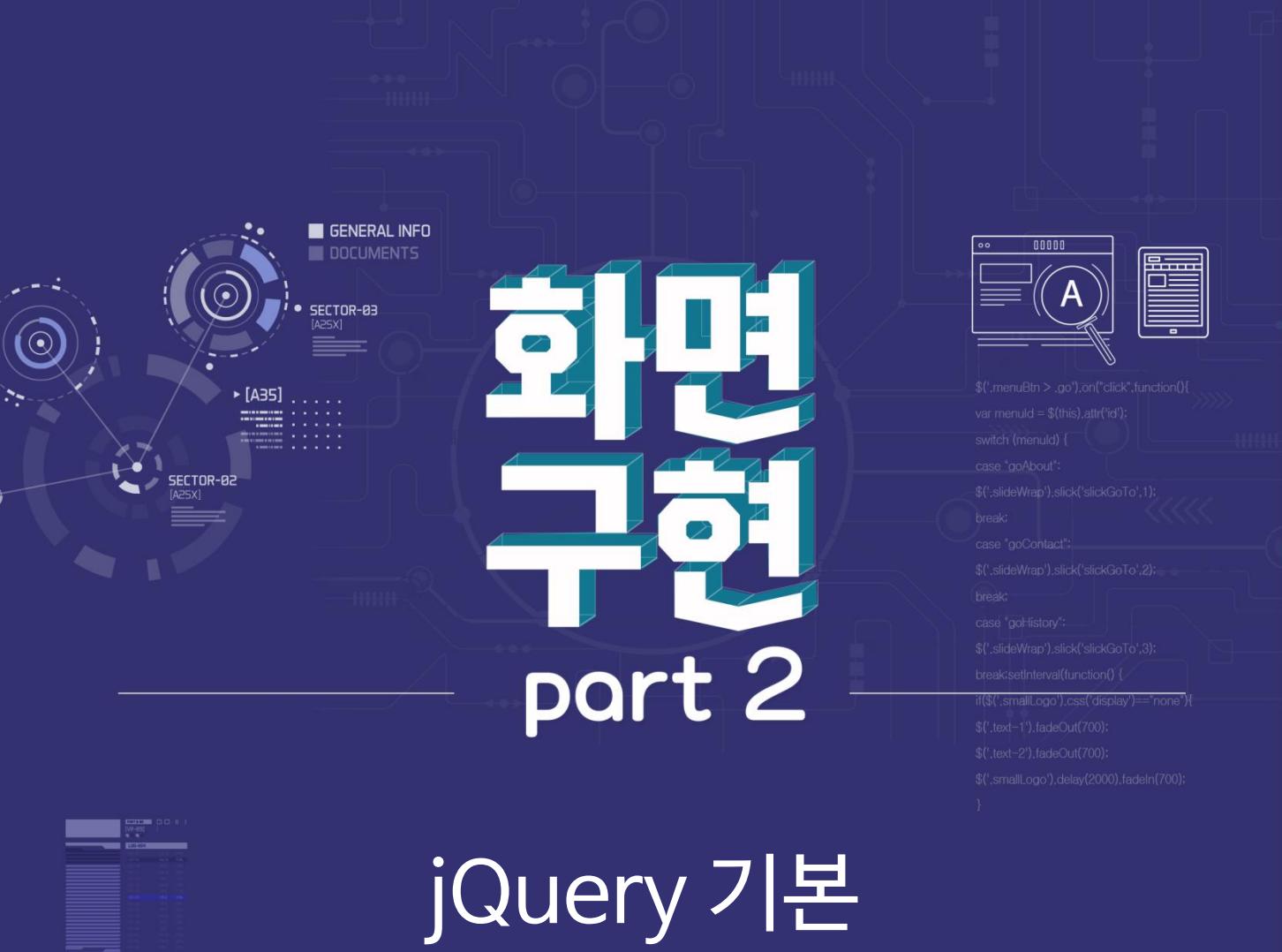
## 2. 함수와 제어문

- 자바스크립트에서는 함수도 하나의 데이터 타입이며 함수를 변수에 대입하거나 함수에 프로퍼티를 지정하는 것도 가능함
- `function` 키워드로 선언하며 중괄호({ }) 안에 자바스크립트 내부 실행문을 넣어서 작성함
- 조건문은 프로그램 내에서 주어진 표현식의 결과에 따라 별도의 명령을 수행하도록 제어함
  - ✓ `If` 문
  - ✓ `else if - else` 문
- 반복문은 특정 조건을 만족할 때까지 계속해서 주어진 실행문을 반복 실행함
  - ✓ `while` 문
  - ✓ `do-while, for` 문

# 화면 구현

part 2

## jQuery 기본



KOREATECH  
온라인평생교육원



## 학습목표

- jQuery에 관해 이해하고, 이를 설명할 수 있다.
- jQuery를 활용할 수 있다.

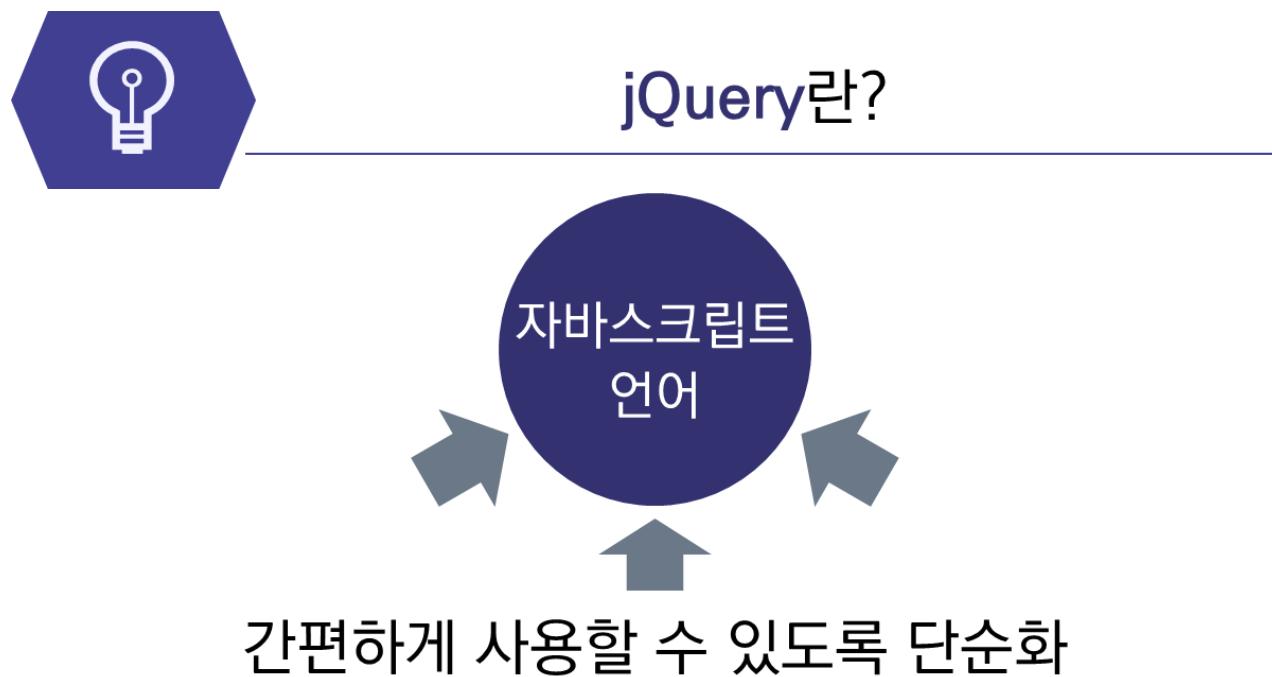
## 학습내용

- jQuery 개요
- jQuery 응용
- jQuery 실습

# jQuery 개요

## 1. jQuery란?

### 1) 개요



## 오픈 소스 기반의 자바스크립트 라이브러리

- HTML의 클라이언트 사이드 조작을 단순화하도록 설계됨
- jQuery 라이브러리는 jQuery라는 이름의 객체로 생성됨

# jQuery 개요

## 1. jQuery란?

### 2) 특징



CSS  
선택자

- HTML 문서의 구조를 명료하고,  
읽기 쉬운 형태로 표현하고 사용 가능



플러그인  
아키텍처

- 중복되는 기능과 코드가 엉키는 것을  
피함
- 이미 개발된 많은 플러그인을 쉽고  
빠르게 이용 가능

# jQuery 개요

## 1. jQuery란?

### 2) 특징



메서드  
체인

- 여러 기능을 한 줄에 나열하고,  
임시 변수의 사용을 최소화  
 불필요한 코드의 반복을 피함



크로스  
브라우저

- 브라우저마다 다르게 발생하는 이벤트,  
객체 등을 처리하고자 여러 개의 메서드를  
사용하지 않음  
 jQuery에서 제공되는 메서드 또는  
문장으로 간단하게 해결 가능

# jQuery 개요

## 1. jQuery란?

### 3) 적용 방법

- jQuery 파일은 자바스크립트 파일(.js) 형태로 존재
- 웹페이지에서 사용하기 위해서는 파일을 먼저 웹페이지에 로드해야 함

방법이 무엇인가요?



# jQuery 개요

## 1. jQuery란?

### 3) 적용 방법

## [ jQuery 파일을 다운받아 로드 ]

```
<head>
  <script src = “/파일경로/제이쿼리파일명.js”></script>
</head>
```

## [ CDN(Content Delivery Network)을 이용하여 로드 ]

- jQuery.com CDN

```
<script src = “https://code.jquery.com/jquery-1.12.4.min.js”></script>
```

- 구글 CDN

```
<script src = “https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js”></script>
```

- MS CDN

```
<script src = “http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.12.4.min.js”></script>
```

- CDNJS CDN

```
<script src = “https://cdnjs.cloudflare.com/ajax/libs/jquery/1.12.4/jquery.min.js”></script>
```

- jsDelivr CDN

```
<script src = “https://cdn.jsdelivr.net/jquery/1.12.4/jquery.min.js”></script>
```

# jQuery 개요

## 2. jQuery 문법

### 1) 기본 문법

- HTML 요소를 선택하고 선택한 요소에 손쉽게 특정 동작을 설정할 수 있음
- \$ 기호는 jQuery를 의미하고 이를 접근할 수 있게 하는 식별자임
- 동작 함수를 정의하여 선택된 요소에 원하는 동작을 설정
- 문법

\$ (선택자).동작함수();

### 2) Document 객체의 ready()

- 자바스크립트는 웹브라우저의 모든 요소를 로드한 뒤 실행되어야 함
- 오류가 발생할 경우

아직 생성되지 않은 HTML  
요소에 속성을 추가할 때

아직 로드되지 않은 이미지의  
크기를 얻으려고 할 때

# jQuery 개요

## 2. jQuery 문법

### 2) Document 객체의 ready()

[ 요소 로딩여부에 따라 자바스크립트를 실행할 수 있음 ]

- 자바스크립트 Window 객체의 onload() 메소드

```
window.onload = function() {  
    자바스크립트 코드;  
};
```

- jQuery에서 Document 객체의 ready() 메소드

```
$(document).ready(function() {  
    제이쿼리 코드;  
});
```

# jQuery 응용

## 1. 태그 요소 조작

### 1) 선택자

- CSS 선택자를 사용하여 HTML 요소를 선택함
- \$( ) 함수에 전달되는 인수는 따옴표(“ ”)를 사용한 문자열 형태로 전달되어야 함

#### [ 기타 선택자 예 ] 1/4

- Selects all elements

`$(*)`

- Selects the current HTML element

`$(this)`

- Selects all <p> elements with class = “intro”

`$("p.intro")`

# jQuery 응용

## 1. 태그 요소 조작

### 1) 선택자

#### [ 기타 선택자 예 ] 2/4

- Selects the first <p> element

`$(“p:first”)`

- Selects the first <li> element of the first <ul>

`$(“ul li:first”)`

- Selects the first <li> element of every <ul>

`$(“ul li:first-child”)`

# jQuery 응용

## 1. 태그 요소 조작

### 1) 선택자

[ 기타 선택자 예 ] 3/4

- Selects all elements with an href attribute

`$(“[href]”)`

- Selects all <a> elements with a target attribute value equal to “\_blank”

`$(“a[target = ‘_blank’]”)`

- Selects all <a> elements with a target attribute value NOT equal to “\_blank”

`$(“a[target! = ‘_blank’]”)`

# jQuery 응용

## 1. 태그 요소 조작

### 1) 선택자

[ 기타 선택자 예 ] 4/4

- Selects all <button> elements and <input> elements of type = “button”

`$(“:button”)`

- Selects all even <tr> elements

`$(“tr:even”)`

- Selects all odd <tr> elements

`$(“tr:odd”)`



jQuery 선택자란?

선택한 요소들을 변수에 저장하여 사용 가능

# jQuery 응용

## 1. 태그 요소 조작

### 1) 선택자

#### [ 필터링에 사용 가능한 선택자 ] 1/2

**:eq(n)**   ▪ 선택한 요소 중에서 인덱스가 n인 요소를 선택함

**:gt(n)**   ▪ 선택한 요소 중에서 인덱스가 n보다 큰 요소를 모두 선택함

**:lt(n)**   ▪ 선택한 요소 중에서 인덱스가 n보다 작은 요소를 모두 선택함

**:even**   ▪ 선택한 요소 중에서 인덱스가 짝수인 요소를 모두 선택함

**:odd**   ▪ 선택한 요소 중에서 인덱스가 홀수인 요소를 모두 선택함

**:first**   ▪ 선택한 요소 중에서 첫 번째 요소를 선택함

**:last**   ▪ 선택한 요소 중에서 마지막 요소를 선택함

**:animated**   ▪ 선택한 요소 중에서 애니메이션 효과가 실행 중인 요소를 모두 선택함

**:header**   ▪ 선택한 요소 중에서 h1부터 h6까지의 요소를 모두 선택함

**:lang(언어)**   ▪ 선택한 요소 중에서 지정한 언어의 요소를 모두 선택함

# jQuery 응용

## 1. 태그 요소 조작

### 1) 선택자

#### [ 필터링에 사용 가능한 선택자 ] 2/2

:not(선택자)

- 선택한 요소 중에서 지정한 선택자와 일치하지 않는 요소를 모두 선택함

:root

- 선택한 요소 중에서 최상위 루트 요소를 선택함

:target

- 선택한 요소 중에서 웹 페이지 URI의 fragment 식별자와 일치하는 요소를 모두 선택함

:contains  
(텍스트)

- 선택한 요소 중에서 지정한 텍스트를 포함하는 요소를 모두 선택함

:has(선택자)

- 선택한 요소 중에서 지정한 선택자와 일치하는 자손 요소를 갖는 요소를 모두 선택함

:empty

- 선택한 요소 중에서 자식 요소를 가지고 있지 않은 요소를 모두 선택함

:parent

- 선택한 요소 중에서 자식 요소를 가지고 있는 요소를 모두 선택함

# jQuery 응용

## 1. 태그 요소 조작

### 1) 선택자

#### [ input 요소를 선택할 수 있는 선택자 ] 1/3

:button

- type 속성값이 “button”인 요소를 모두 선택함

:checkbox

- type 속성값이 “checkbox”인 요소를 모두 선택함

:file

- type 속성값이 “file”인 요소를 모두 선택함

:image

- type 속성값이 “image”인 요소를 모두 선택함

:password

- type 속성값이 “password”인 요소를 모두 선택함

# jQuery 응용

## 1. 태그 요소 조작

### 1) 선택자

#### [ input 요소를 선택할 수 있는 선택자 ] 2/3

:radio

- type 속성값이 “radio”인 요소를 모두 선택함

:reset

- type 속성값이 “reset”인 요소를 모두 선택함

:submit

- type 속성값이 “submit”인 요소를 모두 선택함

:text

- type 속성값이 “text”인 요소를 모두 선택함

:input

- <input>, <textarea>, <select>, <button> 요소를 모두 선택함

# jQuery 응용

## 1. 태그 요소 조작

### 1) 선택자

#### [ input 요소를 선택할 수 있는 선택자 ] 3/3

- |           |  |
|-----------|--|
| :checked  | ▪ type 속성값이 “checkbox” 또는 “radio”인 요소 중에서 체크되어 있는 요소를 모두 선택함 |
| :selected | ▪ <option> 요소 중에서 선택된 요소를 모두 선택함                             |
| :focus    | ▪ 현재 포커스가 가지고 있는 요소를 선택함                                     |
| :disabled | ▪ 비활성화 되어있는 요소를 모두 선택함                                       |
| :enabled  | ▪ 활성화 되어있는 요소를 모두 선택함  |

# jQuery 응용

## 1. 태그 요소 조작

### 2) 탐색

#### [ 조상 요소 탐색 ]

- 선택한 요소의 부모(parent) 요소를 선택함

.parent()

- 선택한 요소의 조상(ancestor) 요소를 모두 선택함

.parents()

- 선택한 요소의 조상 요소 중에서 지정한 선택자에 해당하는 요소 바로 이전까지의 요소를 모두 선택함

.parentsUntil()

- 선택한 요소를 포함한 조상 요소 중에서 지정한 선택자에 해당하는 요소 중 가장 첫 번째 요소를 선택함

.closest()

# jQuery 응용

## 1. 태그 요소 조작

### 2) 탐색

#### [ 형제 요소 탐색 ] 1/2

- 선택한 요소의 형제(sibling) 요소 중에서 지정한 선택자에 해당하는 요소를 모두 선택함

.siblings()

- 선택한 요소의 바로 다음에 위치한 형제 요소를 선택함

.next()

- 선택한 요소의 다음에 위치한 형제 요소를 모두 선택함

.nextAll()

# jQuery 응용

## 1. 태그 요소 조작

### 2) 탐색

#### [ 형제 요소 탐색 ] 2/2

- 선택한 요소의 형제 요소 중에서 지정한 선택자에 해당하는 요소 바로 이전까지의 요소를 모두 선택함

.nextUntil()

- 선택한 요소의 바로 이전에 위치한 형제 요소를 선택함

.prev()

- 선택한 요소의 이전에 위치한 형제 요소를 모두 선택함

.prevAll()

- 선택한 요소의 형제 요소 중에서 지정한 선택자에 해당하는 요소 바로 다음까지의 요소를 모두 선택함

.prevUntil()

# jQuery 응용

## 1. 태그 요소 조작

### 2) 탐색

#### [ 자손 요소 탐색 ]

- 선택한 요소의 자식(child) 요소를 모두 선택함

.children()

- 선택한 요소의 자손(descendant) 요소 중에서 전달받은 선택자에 해당하는 요소를 모두 선택함

.find()

# jQuery 응용

## 1. 태그 요소 조작

### 2) 탐색

#### [ 기타 탐색 ] 1/2

- 선택한 요소의 집합에 전달받은 요소를 추가함

.add()

- 선택한 요소의 집합에 바로 전에 선택했던 요소를 추가함

.addBack()

- 선택한 요소 집합의 요소마다 전달받은 콜백 함수를 실행함

.each()

# jQuery 응용

## 1. 태그 요소 조작

### 2) 탐색

#### [ 기타 탐색 ] 2/2

- 마지막으로 실행한 메소드의 실행 전 상태로 선택한 요소의 집합을 복원함

.end()

- 선택한 요소를 위치시킬 때 기준이 되었던 조상 요소를 선택함

.offsetParent()

- 선택한 요소의 자식(child) 요소를 모두 선택함(텍스트 노드와 주석 노드까지 모두 포함함)

.contents()

# jQuery 응용

## 2. 스타일 및 이벤트

### 1) 스타일

#### [ 프로퍼티 설정 ] 1/2

- `css()` 메소드는 선택한 요소 집합의 첫 번째 요소의 스타일 속성값을 반환하거나, 선택한 요소의 스타일 속성을 인수로 전달받은 값으로 설정함

`.css()`

- 선택한 요소 집합의 첫 번째 요소의 지정된 속성(attribute)값을 반환하거나, 선택한 요소의 지정된 속성을 전달받은 값으로 설정함

`.attr()`

# jQuery 응용

## 2. 스타일 및 이벤트

### 1) 스타일

#### [ 프로퍼티 설정 ] 2/2

- 선택한 요소 집합의 첫 번째 요소의 지정된 프로퍼티(property)값을 반환하거나, 선택한 요소의 지정된 프로퍼티를 전달받은 값으로 설정함

.prop()

- 선택한 요소에서 지정된 속성(attribute)을 제거함

.removeAttr()

- 선택한 요소에서 지정된 프로퍼티(property)를 제거함

.removeProp()

# jQuery 응용

## 2. 스타일 및 이벤트

### 1) 스타일

#### [ 클래스 설정 ]

- 선택한 요소에 인수로 전달받은 클래스를 추가함  
**.addClass()**
- 선택한 요소에서 인수로 전달받은 클래스를 제거함  
**.removeClass()**
- 선택한 요소에 클래스가 없으면 인수로 전달받은 클래스를 추가하고, 전달받은 클래스가 추가되어 있으면 제거함  
**.toggleClass()**
- 인수로 전달받은 값이 선택한 요소의 클래스 이름과 일치하는지를 확인함  
**.hasClass()**

# jQuery 응용

## 2. 스타일 및 이벤트

### 2) 이벤트



이벤트란?



사용자의  
마우스  
움직임  
(클릭 등)



사용자가  
텍스트 박스에  
글 쓰는  
행위 등

모든 동작이 이벤트를 발생

- 이벤트가 발생하면 웹페이지에서 특정 동작이 발생하여 웹 브라우저가 그 사실을 알려줌

# jQuery 응용

## 2. 스타일 및 이벤트

### 2) 이벤트



이벤트 핸들러란?

- 특정 요소에 발생하는 이벤트를 처리하기 위해서 이벤트 핸들러라는 함수 작성
- 이벤트 핸들러가 연결된 특정 요소에서 지정된 타입의 이벤트가 발생하면 웹 브라우저는 연결된 이벤트 핸들러를 실행



이벤트 객체란?

- 이벤트 핸들러는 이벤트 객체를 인수로 전달받을 수 있음
- 전달받은 이벤트 객체를 이용하여 이벤트의 성질을 결정하거나 이벤트의 기본 동작을 막을 수 있음

# jQuery 응용

## 2. 스타일 및 이벤트

### 2) 이벤트



이벤트 연결이란?

이벤트와 이벤트 핸들러를  
특정 요소에 연결하는 것

#### [ 이벤트 연결 ]

.on() : 특정 요소에 이벤트 연결을 위해 사용

- 선택한 요소에 어떤 타입의 이벤트라도 연결할 수 있음
- 하나의 이벤트 핸들러에 여러 개의 이벤트를 동시에 연결할 수 있음
- 선택한 요소에 여러 개의 이벤트 핸들러와 여러 개의 이벤트를 같이 연결할 수 있음
- 사용자 지정 이벤트를 위해 이벤트 핸들러로 데이터를 넘길 수 있음

# jQuery 응용

## 2. 스타일 및 이벤트

### 2) 이벤트

#### [ 이벤트 연결 – 마우스 이벤트 메소드 ] 1/4

- 자바스크립트의 “click” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “click” 이벤트를 발생시킴

.click()

- 자바스크립트의 “dblclick” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “dblclick” 이벤트를 발생시킴

.dblclick()

- 자바스크립트의 “mouseenter”와 “mouseleave” 이벤트를 같이 인벤트 핸들러와 연결함

.hover()

# jQuery 응용

## 2. 스타일 및 이벤트

### 2) 이벤트

#### [ 이벤트 연결 – 마우스 이벤트 메소드 ] 2/4

- 자바스크립트의 “mousedown” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “mousedown” 이벤트를 발생시킴

.mousedown()

- 해당 요소 위로 마우스가 진입할 때 발생하는 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 마우스 진입 이벤트를 발생시킴

.mouseenter()

- 해당 요소에서 마우스가 빠져나갈 때 발생하는 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 마우스가 빠져나가는 이벤트를 발생시킴

.mouseleave()

# jQuery 응용

## 2. 스타일 및 이벤트

### 2) 이벤트

#### [ 이벤트 연결 – 마우스 이벤트 메소드 ] 3/4

- 자바스크립트의 “mousemove” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “mousemove” 이벤트를 발생시킴

.mousemove()

- 자바스크립트의 “mouseout” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “mouseout” 이벤트를 발생시킴

.mouseout()

- 자바스크립트의 “mouseover” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “mouseover” 이벤트를 발생시킴

.mouseover()

# jQuery 응용

## 2. 스타일 및 이벤트

### 2) 이벤트

#### [ 이벤트 연결 – 마우스 이벤트 메소드 ] 4/4

- 자바스크립트의 “mouseup” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “mouseup” 이벤트를 발생시킴

.mouseup()

#### [ 이벤트 연결 – 키보드 이벤트 메소드 ] 1/2

- 자바스크립트의 “keydown” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “keydown” 이벤트를 발생시킴

.keydown()

- 자바스크립트의 “keyup” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “keyup” 이벤트를 발생시킴

.keyup()

# jQuery 응용

## 2. 스타일 및 이벤트

### 2) 이벤트

#### [ 이벤트 연결 – 키보드 이벤트 메소드 ] 2/2

- 자바스크립트의 “keypress” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “keypress” 이벤트를 발생시킴

.keypress()

#### [ 이벤트 연결 – 입력 양식 이벤트 메소드 ] 1/3

- 자바스크립트의 “blur” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “blur” 이벤트를 발생시킴

.blur()

- 자바스크립트의 “change” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “change” 이벤트를 발생시킴

.change()

# jQuery 응용

## 2. 스타일 및 이벤트

### 2) 이벤트

#### [ 이벤트 연결 – 입력 양식 이벤트 메소드 ] 2/3

- 자바스크립트의 “select” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “select” 이벤트를 발생시킴

.select()

- 자바스크립트의 “submit” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “submit” 이벤트를 발생시킴

.submit()

# jQuery 응용

## 2. 스타일 및 이벤트

### 2) 이벤트

#### [ 이벤트 연결 – 입력 양식 이벤트 메소드 ] 3/3

- 자바스크립트의 “focus” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “focus” 이벤트를 발생시킴

.focus()

- “focusin” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “focusin” 이벤트를 발생시킴

.focusin()

- “focusout” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “focusout” 이벤트를 발생시킴

.focusout()

# jQuery 실습

```
jQuery 실습
4 <html>
5 <head>
6 <script src="https://ajax.googleapis.com/ajax/libs/
jquery/3.3.1/jquery.min.js"></script>
7 <script>
8     function func() {
9         addAttribute();
10        createElement();
11    }
12    function createElement() {
13        var criteriaNode = document.getElementById(
14            "text");
15        var newNode = document.createElement("p");
16        newNode.innerHTML = "새로운 단락";
17        newNode.setAttribute("id", "para");
18        document.body.insertBefore(newNode,
19            criteriaNode);
20    }
21    function addAttribtue() {
22        document.getElementById("para").set
23            tyle("style", "color:red");
}
jQuery 사용을 위해
라이브러리 URL을 입력하여 활용
```

※ 실습 영상은 콘텐츠에서 확인할 수 있습니다.

# 핵심정리

## 1. jQuery 개요

- 자바스크립트 언어를 간편하게 사용할 수 있도록 단순화시킨 오픈 소스 기반의 자바스크립트 라이브러리로 HTML의 클라이언트 사이드 조작을 단순화 하도록 설계됨
- jQuery 라이브러리는 jQuery라는 이름의 객체로 생성됨
- 중복되는 기능과 코드가 엉키는 것을 피하고 이미 개발된 많은 플로그인을 쉽고 빠르게 이용할 수 있음
- 브라우저마다 다르게 발생하는 이벤트, 객체 등을 처리하고자 여러 개의 메서드를 사용하지 않고 jQuery에서 제공되는 메서드 또는 문장으로 간단하게 해결 가능함
- jQuery 파일은 자바스크립트 파일(.js)형태로 존재하며 사용하기 위하여 웹페이지에 로드해야 함
- \$ 기호는 jQuery를 의미하고 이를 접근할 수 있게 하는 식별자이고 동작 함수를 정의하여 선택된 요소에 원하는 동작을 설정함

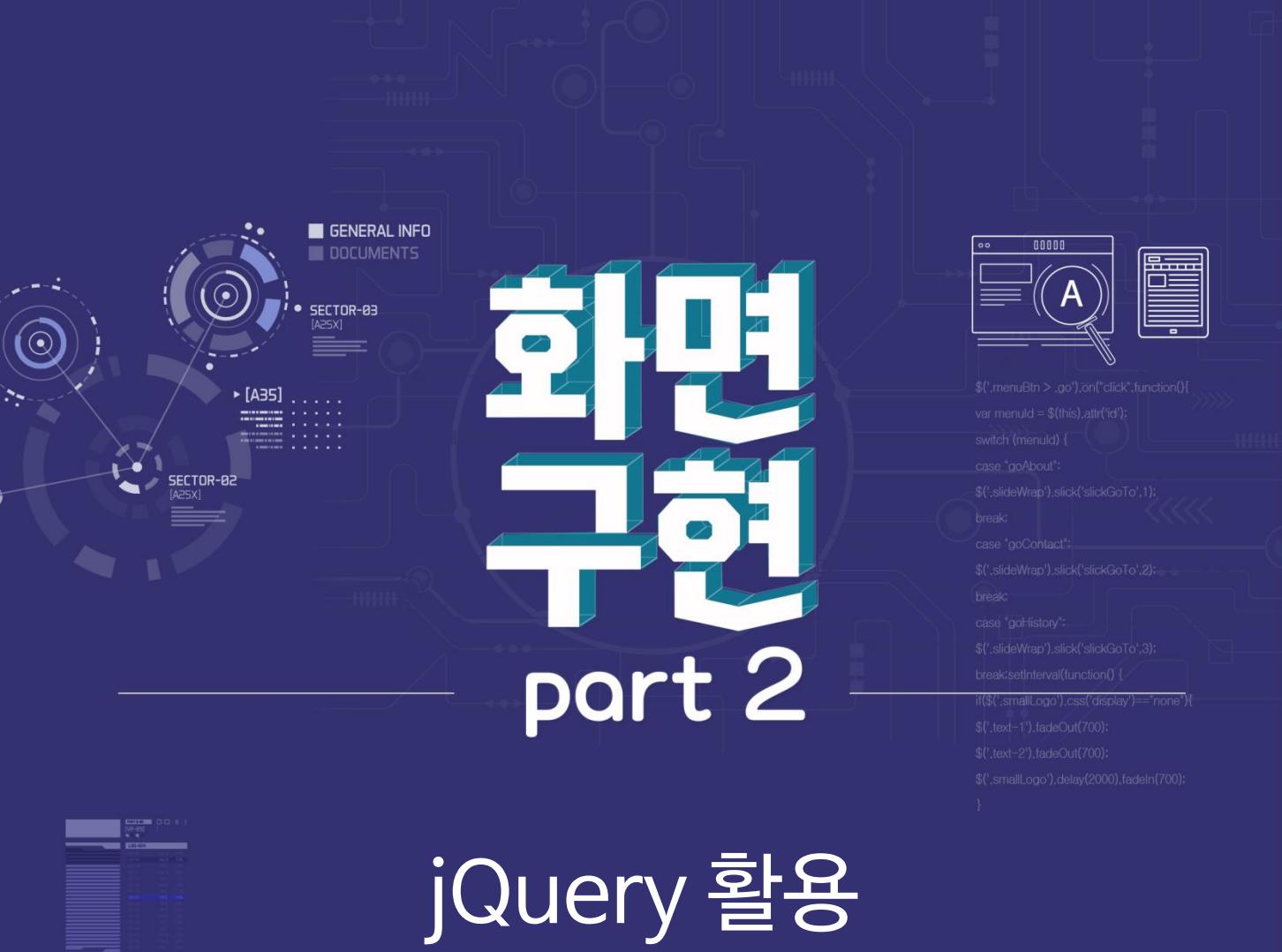
## 2. jQuery 응용

- 선택자를 이용하여 적용할 요소를 선택하고 선택자는 CSS선택자 및 jQuery 선택자를 이용함
- HTML 요소를 탐색할 수 있는 기능을 제공하며 요소의 스타일(프로퍼티)를 설정할 수 있음
- 사용자가 마우스를 움직이거나 클릭하거나 텍스트 박스에 글을 쓰는 등 모든 동작들을 이벤트라고 하며 특정 요소에 이벤트 핸들러를 연결하여 지정된 이벤트를 웹브라우저에서 동작하게 함

# 화면 구현

part 2

## jQuery 활용



KOREATECH  
온라인평생교육원



## 학습목표

- jQuery UI 위젯에 관해 이해하고, 활용할 수 있다.
- 나에게 맞는 jQuery 플러그인을 사용할 수 있다.

## 학습내용

- jQuery UI 위젯
- jQuery UI 위젯 및 플러그인 실습

# jQuery UI 위젯

## 1. jQuery UI 인터랙션

### 1) 인터랙션

- **Draggable**: 마우스 드래그로 엘리먼트를 이동함
- **Droppable**: 엘리먼트 위에 드래그한 엘리먼트를 드롭할 수 있도록 이벤트 감지
- **Resizable**: 엘리먼트의 크기를 변경할 수 있는 UI 제공
- **Selectable**: 엘리먼트를 선택할 수 있는 UI를 제공
- **Sortable**: 엘리먼트를 정렬할 수 있는 UI를 제공

### 2) 위젯

- **Accordion**: 아코디언 형태의 UI를 제공
- **Autocomplete**: 자동완성 기능과 UI를 제공
- **Button**: 엘리먼트 버튼처럼 보이도록 하는 UI를 제공
- **Datepicker**: 날짜 선택 도구 UI를 제공

# jQuery UI 위젯

## 1. jQuery UI 인터랙션

### 2) 위젯

- **Dialog**: 알림창 레이어를 띄울 수 있는 UI를 제공
- **Progressbar**: 진행 막대 UI를 제공
- **Slider**: 슬라이더 막대 형식의 입력도구를 제공
- **Tabs**: 탭 메뉴 형식의 UI를 제공

# jQuery UI 위젯

## 1. jQuery UI 인터랙션

### 2) 위젯

#### 입력

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>jQuery UI Accordion - Default functionality</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
  <link rel="stylesheet" href="/resources/demos/style.css">
  <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
  <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
  <script>
    $( function() {
      $( "#accordion" ).accordion();
    } );
  </script>
</head>
<body>

<div id="accordion">
  <h3>Section 1</h3>
  <div>
    <p>
      Mauris mauris ante, blandit et, ultrices a, suscipit eget, quam.
    </p>
  </div>
</div>
```

# jQuery UI 위젯

## 1. jQuery UI 인터랙션

### 2) 위젯

입력

```
<h3>Section 2</h3>
<div>
  <p>
    Sed non urna. Donec et ante. Phasellus eu ligula. Vestibulum sit amet
    purus.
  </p>
</div>
<h3>Section 3</h3>
<div>
  <p>
    Nam enim risus, molestie et, porta ac, aliquam ac, risus. Quisque lobortis.
  </p>
  <ul>
    <li>List item one</li>
    <li>List item two</li>
  </ul>
</div>
<h3>Section 4</h3>
<div>
  <p>
    Cras dictum. Pellentesque habitant morbi tristique senectus et netus
    et malesuada fames ac turpis egestas.
  </p>
</div>
</div>
</body>
</html>
```



# jQuery UI 위젯

## 1. jQuery UI 인터랙션

### 2) 위젯

결과

› Section 1

Mauris mauris ante, blandit et, ultrices a, suscipit eget, quam.

› Section 2

› Section 3

› Section 4

# jQuery UI 위젯

## 1. jQuery UI 인터랙션

### 2) 위젯

#### [ Autocomplete ]

##### 입력

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>jQuery UI Autocomplete - Default functionality</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
  <link rel="stylesheet" href="/resources/demos/style.css">
  <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
  <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
  <script>
$( function() {
  var availableTags = [
    "ActionScript",
    "AppleScript",
    "Asp",
    "BASIC",
    "C",
    "C++",
    "Clojure",
    "COBOL",
    "ColdFusion",
    "Erlang",
    "Fortran",
    "Groovy",
    "Java",
    "JavaScript",
    "Lisp",
    "Perl",
    "PHP",
    "Python",
    "Ruby",
    "Visual Basic"
  ];
  $( "#tags" ).autocomplete({
    source: availableTags
  });
})</script>
```

# jQuery UI 위젯

## 1. jQuery UI 인터랙션

### 2) 위젯

#### [ Autocomplete ]

입력

```
"Haskell",
"Java",
"JavaScript",
"Lisp",
"Perl",
"PHP",
"Python",
"Ruby",
"Scala",
"Scheme"

];
$( "#tags" ).autocomplete({
    source: availableTags
});
}
</script>
</head>
<body>

<div class="ui-widget">
    <label for="tags">Tags: </label>
    <input id="tags">
</div>
</body>
</html>
```



# jQuery UI 위젯

## 1. jQuery UI 인터랙션

### 2) 위젯

#### [ Autocomplete ]

결과

Tags: S

# jQuery UI 위젯

## 1. jQuery UI 인터랙션

### 2) 위젯

#### [ Datepicker ]

##### 입력

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>jQuery UI Datepicker - Default functionality</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
  <link rel="stylesheet" href="/resources/demos/style.css">
  <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
  <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
  <script>
    $( function() {
      $( "#datepicker" ).datepicker();
    });
  </script>
</head>
<body>

<p>Date: <input type="text" id="datepicker"></p>

</body>
</html>
```

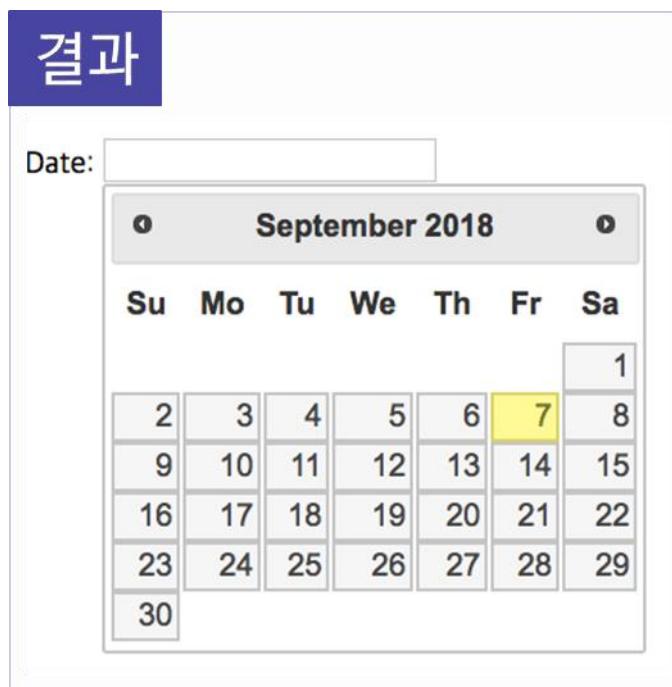


# jQuery UI 위젯

## 1. jQuery UI 인터랙션

### 2) 위젯

#### [ Datepicker ]



# jQuery UI 위젯

## 1. jQuery UI 인터랙션

### 2) 위젯

#### [ Progressbar ]

##### 입력

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>jQuery UI Progressbar - Default functionality</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
  <link rel="stylesheet" href="/resources/demos/style.css">
  <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
  <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
  <script>
$( function() {
  $( "#progressbar" ).progressbar({
    value: 29
  });
} );
</script>
</head>
<body>

<div id="progressbar"></div>

</body>
</html>
```



##### 결과



# jQuery UI 위젯

## 1. jQuery UI 인터랙션

### 2) 위젯

#### [ Slider ]

##### 입력

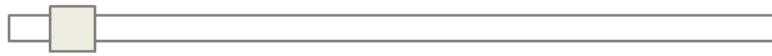
```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>jQuery UI Slider - Default functionality</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
  <link rel="stylesheet" href="/resources/demos/style.css">
  <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
  <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
  <script>
    $( function() {
      $( "#slider" ).slider();
    } );
  </script>
</head>
<body>

<div id="slider"></div>

</body>
</html>
```



##### 결과



# jQuery UI 위젯

## 1. jQuery UI 인터랙션

### 2) 위젯

#### [ Tabs ]

입력

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>jQuery UI Tabs - Default functionality</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
  <link rel="stylesheet" href="/resources/demos/style.css">
  <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
  <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
  <script>
    $( function() {
      $( "#tabs" ).tabs();
    } );
  </script>
</head>
<body>

<div id="tabs">
  <ul>
    <li><a href="#tabs-1">Nunc tincidunt</a></li>
    <li><a href="#tabs-2">Proin dolor</a></li>
    <li><a href="#tabs-3">Aenean lacinia</a></li>
  </ul>
</div>
```

# jQuery UI 위젯

## 1. jQuery UI 인터랙션

### 2) 위젯

#### [ Tabs ]

입력

```
<div id="tabs-1">
    <p>Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus. Curabitur nec
arcu. Donec sollicitudin mi sit amet mauris. </p>
</div>
<div id="tabs-2">
    <p>Morbi tincidunt, dui sit amet facilisis feugiat, odio metus gravida ante, ut pharetra
massa metus id nunc. Duis scelerisque molestie turpis. </p>
</div>
<div id="tabs-3">
    <p>Mauris eleifend est et turpis. Duis id erat. Suspendisse potenti. Aliquam vulputate,
pede vel vehicula accumsan, mi neque rutrum erat, eu congue orci lorem eget lorem. </p>
    <p>Duis cursus. Maecenas ligula eros, blandit nec, pharetra at, semper at, magna. Nullam
ac lacus. Nulla facilisi.</p>
</div>
</div>

</body>
</html>
```



결과

Nunc tincidunt      Proin dolor      Aenean lacinia

Mauris eleifend est et turpis. Duis id erat. Suspendisse potenti. Aliquam vulputate, pede vel vehicula accumsan, mi neque rutrum erat, eu congue orci lorem eget lorem.

Duis cursus. Maecenas ligula eros, blandit nec, pharetra at, semper at, magna. Nullam ac lacus. Nulla facilisi.

# jQuery UI 위젯 및 플러그인 실습

The screenshot shows a Windows desktop environment. On the left, there is a Sublime Text editor window titled "jQuery 실습". The file path is "C:\Users\Studio\Desktop\03\example\index.html". The code in the editor is:

```
4 <html lang="en">
5   <head>
6     <meta charset="utf-8">
7     <meta name="viewport" content="width=device-width, initial-scale=1">
8     <title>jQuery UI Draggable - Default
9       functionality</title>
10    <link rel="stylesheet" href="http://code.jquery.com/ui/1.12.1/themes/base/
11      jquery-ui.css">
12    <link rel="stylesheet" href="http://code.jquery.com/ui/1.12.1/themes/base/
13      resources/demos/style.css">
14    <style>
15      #draggable { width: 150px; height: 150px; padding
16        : 0.5em; }
17    </style>
18    <script src="https://code.jquery.com/
19      js"></script>
```

The word "jQuery UI" is highlighted in yellow. On the right, there is a web browser window titled "Blank.html" showing the rendered HTML. The browser's address bar shows "file:///C:/Users/Studio/Desktop/03/example/index.html". The page displays a single blue square element.

다양한 화면 구성을 할 수 있도록 여러 가지 기능 제공

 OLEI

※ 실습 영상은 콘텐츠에서 확인할 수 있습니다.

# 핵심정리

## 1. jQuery UI 위젯

- jQuery UI 인터랙션 지원 종류
  - ✓ Draggable, Droppable, Resizable, Selectable, Sortable
- jQuery UI 위젯 지원 종류
  - ✓ Accordion, Autocomplete, Button, Datapicker, Dialog, Progressbar, Slider, Tabs UI

## 2. jQuery UI 위젯 및 플러그인 실습

- jQuery 공식 홈페이지에서는 유저들이 jQuery를 이용하여 만든 플러그인을 등록하고 이를 검증하여 많은 사람들이 사용할 수 있도록 제공함(<https://plugins.jquery.com>)
- UI를 구성할 때 직접 만들어 사용할 수 있지만 이미 공개되어 있는 플러그인을 활용하여 작성하면 개발 시간을 단축 시킬 수 있음