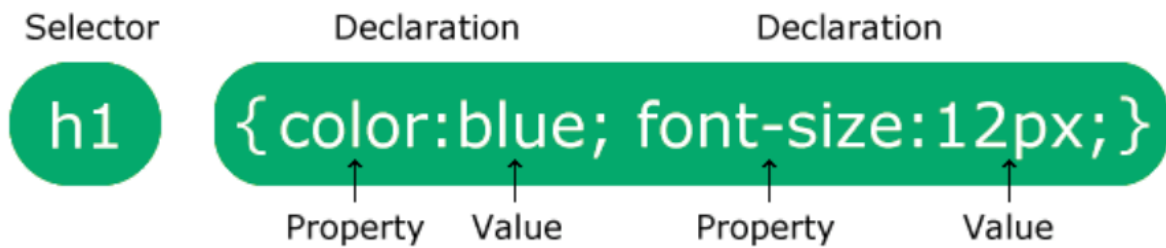


221007 수업 -1

☰ 유형	HTML CSS
☰ 키워드	패딩, 마진, 가상 선택자, 테이블, 리스트, float, semantic 웹

CSS 문법



Padding

- 콘텐츠와 태그와의 간격
- 태그 안 공간

```
p {  
  border: 2px solid powderblue;  
  padding: 30px;  
}
```

Margin

- 태그와 태그 사이의 간격
- 태그 밖 공간

```
p {  
  border: 2px solid powderblue;  
  margin: 50px;  
}
```

Styling Links

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

```

a:link {
  color: red;
}

a:visited {
  color: green;
}

a:hover {
  color: hotpink;
}

a:active {
  color: blue;
}
</style>
</head>
<body>

<h2>Styling a link depending on state</h2>

<p><b><a href="default.asp" target="_blank">This is a link</a></b></p>
<p><b>Note:</b> a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.</p>
<p><b>Note:</b> a:active MUST come after a:hover in the CSS definition in order to be effective.</p>

</body>
</html>

```

가상 선택자 : 의미론적으로만 존재하는

- `a:link` - 기본, 방문하지 않았던 링크
- `a:visited` - 방문 했던 링크
- `a:hover` - 링크 위에 마우스를 올릴 경우
- `a:active` - 마우스를 클릭하는 순간

CSS의 적용은 기존에 존재하는 태그(요소) 기준

ex) p, h3, div, hr

selector (선택자)

- 다중 선택자 : h1, h2, h3 { }

사용자 정의 CSS

- 조건 : id 속성, class 속성 사용해 반영
 - `.` → class
 - `.`으로 만든 건 class 적용 → page에 여러 개 적용
 - `#` → id
 - `#`으로 만든 건 id 적용 → page에 1개의 요소에만 적용
 - 식별자로 사용됨

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Insert title here</title>
  <style type="text/css">

    #park {color: gold}

    .hong{color: blue}

  </style>
</head>
<body>

  <p id="park">id라는 속성을 이용해서 css 적용</p>
  <p class="hong">class 속성을 이용해서 css 적용</p>
  <p class="hong">class 속성을 이용해서 css 적용</p>
  <p class="hong">class 속성을 이용해서 css 적용</p>
  <p class="hong">class 속성을 이용해서 css 적용</p>

</body>
</html>

```

- p 태그 이면서 class가 center인 태그

CSS Universal Selector

```

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

```

테이블

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Insert title here</title>
  <style type="text/css">
    table{
      border-collapse: collapse; /* 붕괴하다 */
      width: 100%;
    }
    th, td{
      border: 1px solid red;
      text-align: left;
      padding: 5px;
    }
    tr:nth-child(even){ /* 짝수 */
      background-color: gray;
    }
  </style>
</head>
<body>

  <table>
    <tr><!-- 한줄 -->

```

```

    <th>Company</th><!-- 한칸 -->
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Centro comercial Moctezuma</td>
    <td>Francisco Chang</td>
    <td>Mexico</td>
  </tr>
</table>

</body>
</html>

```

Table Borders

Table Padding & Spacing

With Padding

hello	hello	hello
hello	hello	hello
hello	hello	hello

With Spacing

hello	hello	hello
hello	hello	hello
hello	hello	hello

Table Colspan & Rowspan : 병합

NAME		

APRIL		

2022		
FIESTA		

<h3>일반 테이블</h3>

```
<table style="height: 200px; width: 500px">
  <tr>
    <td colspan="2">가</td>
    <td>다</td>
  </tr>
  <tr>
    <td>라</td>
    <td>마</td>
    <td>바</td>
  </tr>
  <tr>
    <td colspan="3">사</td>
  </tr>
</table>
```

일반 테이블(병합)

가		다
라	마	바
사		

<h3>일반 테이블(병합)</h3>

```
<table style="height: 200px; width: 500px">
  <tr>
    <td colspan="2" rowspan="2">가, 나, 라, 마</td>
    <td>다</td>
  </tr>
  <tr>
    <td>바</td>
  </tr>
  <tr>
    <td>사</td>
    <td>아</td>
    <td>자</td>
  </tr>
</table>
```

일반 테이블(병합)

가, 나, 라, 마		다
		바
사	아	자

HTML Lists

 : 순서가 없는 목록

 : 순서가 있는 목록

 : 리스트 아이템

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

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

- li 태그 안에 다시 ul 태그 적을 수 있음

<ul style="list-style-type:none;"> : 앞에 아무것도 없음

```
<ul style="list-style-type:none;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Horizontal List with CSS : 수평 배치

Nav 태그만들기

Navigation Menu

In this example, we use CSS to style the list horizontally, to create a navigation menu:

Home News Contact About

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333333;
}

li {
  float: left;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 16px;
  text-decoration: none;
}

li a:hover {
  background-color: #111111;
}
</style>
</head>
<body>
```

```

<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>

</body>
</html>

```

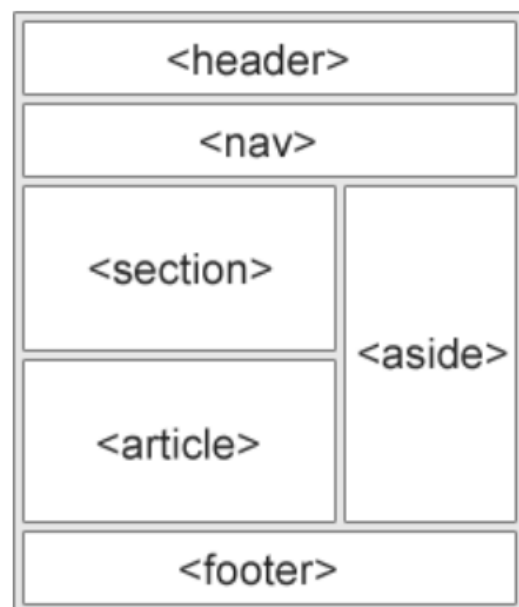
HTML Layout Elements

semantic 웹

- 태그에 의미를 부여해 컴퓨터가 정보를 이해하고, 기계들끼리 정보를 주고 받아 자동으로 처리 할 수 있도록 하는 지능형 웹

시멘틱 웹 구조 태그

- `<header>` - Defines a header for a document or a section
- `<nav>` - Defines a set of navigation links
- `<section>` - Defines a section in a document
- `<article>` - Defines an independent, self-contained content
- `<aside>` - Defines content aside from the content (like a sidebar)
- `<footer>` - Defines a footer for a document or a section
- `<details>` - Defines additional details that the user can open and close on demand



- `<summary>` - Defines a heading for the `<details>` element

<div> Tag : 영역 태그

- 기본 width 값 전체를 다 잡음
- width와 height를 주어도 영역을 잡고 있음

```
<div style="width: 100px; height: 100px; background-color: yellow;"></div>
<div style="width: 100px; height: 100px; background-color: red;"></div>
<div style="width: 100px; height: 100px; background-color: blue;"></div>
```

- 노란색이 float되어 있으므로 빈 공간을 빨간색이 치고 올라감

```
<div style="width: 100px; height: 100px; background-color: yellow; float: left;"></div>
<div style="width: 120px; height: 100px; background-color: red;"></div>
<div style="width: 100px; height: 100px; background-color: blue;"></div>
```



- clear : both → 위치를 고수

```
<div style="width: 100px; height: 100px; background-color: yellow; float: left;"></div>
<div style="width: 100px; height: 100px; background-color: red; float: left;"></div>
<div style="width: 220px; height: 100px; background-color: blue; clear: both;"></div>
```



HTML Forms

- 내용을 서버로 전송하는 태그
- action이 정의하는 경로까지 전송

```
<form action="" method="">
  ID : <input type="text" name="id"><br>
  PWD : <input type="password" name="pwd"><br>
  <input type="submit" value="전송">
  <input type="reset" value="입력값 취소">
</form>
```

1. action="" : 목적지 주소

- JSP : action="loginok.jsp"
- servlet : action="login.do"

2. method="" : 전송 방식

- method = "GET"

ex_ <http://192.168.0.10:8090/WebBasic/loginok.jsp?id=hong&pwd=1004>

- 서버가 정보를 받아서 처리
- 주소에 입력한 정보가 노출됨
- 무조건 GET전송 방식 사용하는 전송 -> 링크를 타고 가는 전송

100번

- method="POST"

ex_ <http://192.168.0.10:8090/WebBasic/loginok.jsp>

- http 프로토콜 내부에 (header) 숨겨서 데이터 전송

3. submit : 서버로 출발

- input type="submit"
- input type="image"
- input type="button" onclick="call()"


```
function call() { document.form.submit() }
```

4. reset : 입력 값 취소

JSP

<% %> : java 코드를 구현하는 영역 (Back단)

HTML Input Types

```
<form action="">
  <fieldset>
    <legend>Text: Password</legend>
    Text: <input type="text" name="userid" id="userid"><br>
    Password: <input type="password" name="pwd" maxlength="5"><br>
    <hr>
    Text: <input type="text" maxlength="10" value="기본값" size="100px">
    <hr>
    readonly(읽기전용 : 서버로 전송 가능)<br>
    <input type="text" name="userid" id="userid" value="hong" readonly><br>
    <br>
    disabled(비활성화 : 서버로 전송 불가)<br>
    <input type="text" name="userid" id="userid" value="hong" disabled><br>
  </fieldset>
</form>
```

```
<fieldset>
  <legend>체크박스(다중 선택 가능)</legend>
  당신의 취미는 <br>
  농구 : <input type="checkbox" name="sports_1" value="A" checked>
  야구 : <input type="checkbox" name="sports_2" value="B">
  축구 : <input type="checkbox" name="sports_3" value="C">
</fieldset><fieldset>
  <legend>체크박스(다중 선택 가능)</legend>
  당신의 취미는 <br>
  농구 : <input type="checkbox" name="sports_1" value="A" checked>
  야구 : <input type="checkbox" name="sports_2" value="B">
  축구 : <input type="checkbox" name="sports_3" value="C">
</fieldset>
```

get 방식

register.jsp?sports_1=A&sports_2=B&sports_3=C >>> 서버로 전송

```
String s = request.getParameter("sports_1");
String s1 = request.getParameter("sports_2");
String s2 = request.getParameter("sports_3");
```

```
<fieldset>
  <legend>체크박스(다중 선택 가능)</legend>
  당신의 취미는 <br>
  농구 : <input type="checkbox" name="sports_1" value="A" checked>
  야구 : <input type="checkbox" name="sports_2" value="B">
  축구 : <input type="checkbox" name="sports_3" value="C">
</fieldset>
```

get 방식

register.jsp?sports=A&sports=B&sports=C >> 서버로 전송

```
String[] sportsarr = request.getParameterValues("sports");
sportsarr[0] = "A"
sportsarr[1] = "B"
sportsarr[2] = "C"
```

```
<fieldset>
  <legend>select 태그(단일 선택)</legend>
  <select name="year">
    <option value="2022">2022</option>
    <option value="2021" selected>2021</option>
    <option value="2020">2020</option>
  </select>
</fieldset>
```

- value=""가 서버로 넘어가는 값
- register.jsp?year=2022 서버로 전송 → 서버가 받는 모든 데이터는 문자
 - int year = Integer.parseInt(request.getParameter("year"));

```
<fieldset>
  <legend>라디오 버튼(단일 선택) : name이 같으면 하나의 그룹</legend>
  남<input type="radio" name="gender" value="1"><br>
  여<input type="radio" name="gender" value="2" checked><br>
</fieldset>
```

- name이 같아야 하나의 그룹 인식

```
<fieldset>
  <legend>기타 등등</legend>
  1. 화면에 출력이 안되지만 소스는 가지고 있음
  <input type="hidden" value="1000" name="data">
  <hr>
  2. 파일 업로드 (클라이언트 파일 선택 가능) -> 전송은 따로 구현해야 함
  <input type="file" name="imgupload">
  3. 게시판 글쓰기 (여러 라인) : 초기화면의 크기 (rows, cols 몇줄의 몇자)
  <textarea rows="10" cols="50">기본크기</textarea><br>
  4. 각종 버튼들
  <input type="button" value="나 놀러봐" onclick="alert('쉬는 시간')">
  <br>
  <!--
  type="image"
  type="submit"
```

```

        type="reset"
    -->
</fieldset>

```

HTML5 유용한 태그

Input Type Color

```

<form>
  <label for="favcolor">Select your favorite color:</label>
  <input type="color" id="favcolor" name="favcolor">
</form>

```

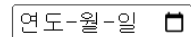


Input Type Date

```

<form>
  <label for="birthday">Birthday:</label>
  <input type="date" id="birthday" name="birthday">
</form>

```



Input Type Email

Enter your email:

❗ 이메일 주소에 '@'를 포함해 주세요. 'sdfsdf'에 '@'가 없습니다.

```

<form>
  <label for="email">Enter your email:</label>
  <input type="email" id="email" name="email">
</form>

```

Input Type Number

```

<form>
  <label for="quantity">Quantity (between 1 and 5):</label>
  <input type="number" id="quantity" name="quantity" min="1" max="5">
</form>

```



Input Type Range

```

<form>
  <label for="vol">Volume (between 0 and 50):</label>
  <input type="range" id="vol" name="vol" min="0" max="50">
</form>

```



Input Type Tel

```
<form>
  <label for="phone">Enter your phone number:</label>
  <input type="tel" id="phone" name="phone" pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}">
</form>
```

list attribute

```
<form>
  <input list="browsers">
  <datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
</form>
```