

HTML5 & CSS3 기반의 반응형 모바일 웹 만들기

2일차 강의자료

2일차 목차

HTML5 & CSS3 기반의 반응형 모바일 웹 만들기

1. 멀티미디어 지원 태그

- 1) 이 단원에서 필요한 멀티미디어 파일의 준비
- 2) 음악 재생을 위한 <audio>태그
- 3) 동영상 재생을 위한 <video>태그

2. HTML5 폼 요소

- 1) 기존의 Form요소 살펴보기
- 2) 새롭게 추가된 HTML5 폼 요소
- 3) input태그에 추가된 공통 속성

3. CSS3 시작하기

- 1) CSS의 표현방법
- 2) CSS의 셀렉터와 구체성 단위
- 3) CSS3의 새로운 가상 클래스
- 4) 브라우저마다 다르게 구현되는 프로퍼티

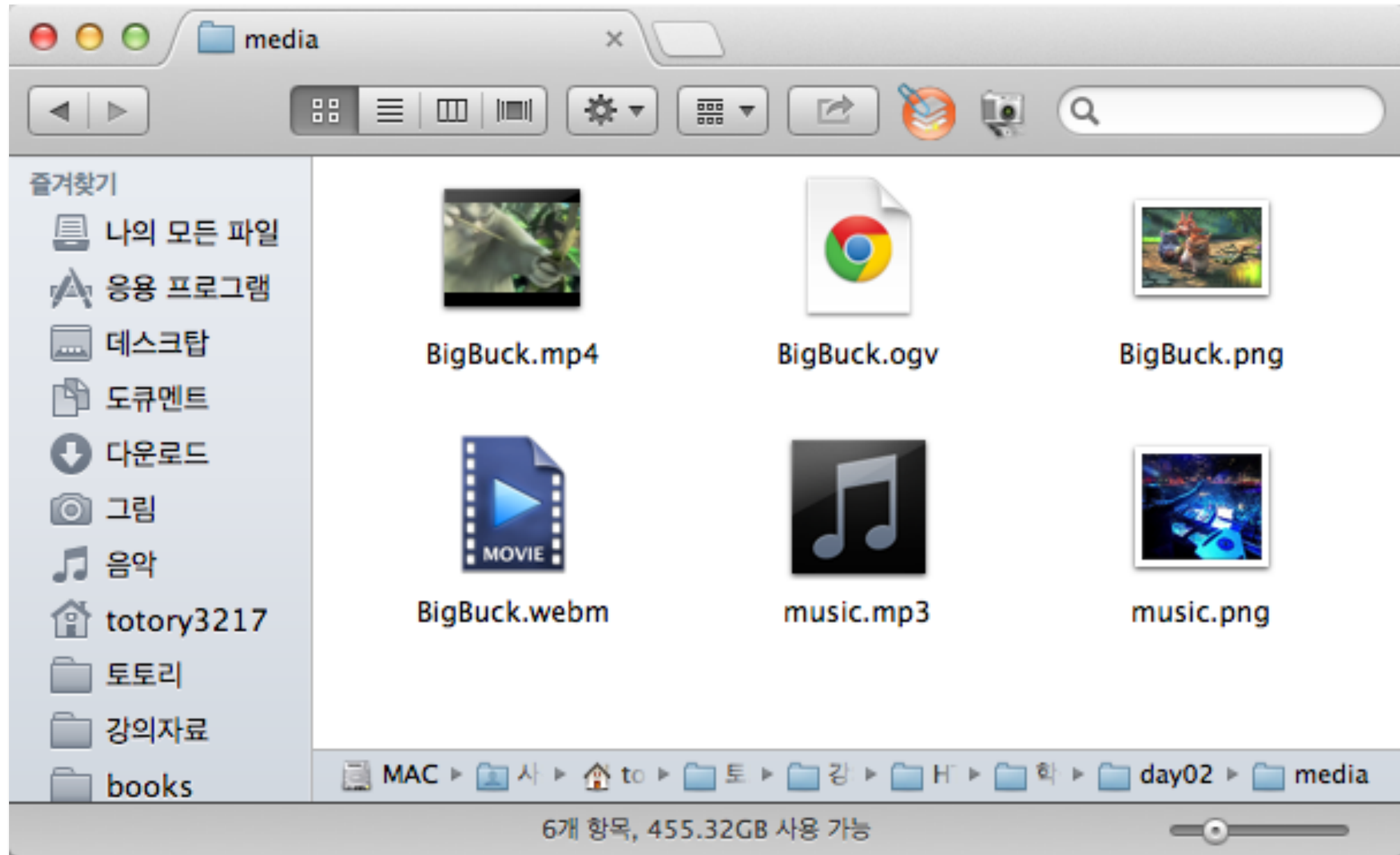
1. 멀티미디어 지원 태그

HTML5 & CSS3 기반의 반응형 모바일 웹 만들기

- 1) 이 단원에서 필요한 멀티미디어 파일의 준비
- 2) 음악 재생을 위한 <audio>태그
- 3) 동영상 재생을 위한 <video>태그

1) 이 단원에서 필요한 멀티미디어 파일의 준비

☑ /day02/media 폴더



2) 음악 재생을 위한 <audio>태그 (1)

```
<audio src="파일경로" [controls] [loop] [autoplay]>  
    [지원하지 않는 브라우저에서 표시할 내용]  
</audio>
```

주요 속성 소개

- src - 파일의 경로를 지정하는 속성. 일반적으로 mp3 파일을 지정하면 브라우저 호환성에 대해서 큰 무리 없이 사용 가능하다.
- controls - 각 브라우저 고유의 컨트롤바를 표시함
- loop - 재생이 끝나고 나면 처음부터 다시 반복하여 재생함
- autoplay - 자동으로 재생 시작

2) 음악 재생을 위한 <audio>태그 (2)

☑ /day02/01-audio.html

```
<div style="width: 300px; margin: auto">  
    
  <audio src="../../media/music.mp3" controls autoplay loop  
  style="width:100%;">  
    이 브라우저는 HTML5를 지원하지 않습니다.  
  </audio>  
</div>
```


2) 음악 재생을 위한 <audio>태그 (3)

☑ /day02/01-audio.html (실행결과)



임의의 이미지

<audio>
태그

3) 동영상 재생을 위한 <video>태그 (1)

```
<video src="파일경로" [controls] [loop] [autoplay] [preload]
[width="폭"] [height="높이"] [poster="미리보기 이미지"]>
[지원하지 않는 브라우저에서 표시할 내용]
</video>
```

주요 속성 소개

- src - 파일의 경로를 지정하는 속성. 브라우저마다 코덱 이슈 있음.
- controls - 각 브라우저 고유의 컨트롤바를 표시함
- loop - 재생이 끝나고 나면 처음부터 다시 반복하여 재생함
- autoplay - 자동으로 재생 시작
- preload - 동영상이 백그라운드에서 다운로드 된다.
- width, height - 동영상의 크기 지정
- poster - 동영상 재생전 보여질 미리보기 이미지

3) 동영상 재생을 위한 <video>태그 (2)

☑ 모든 브라우저를 고려한 Video태그 사용법

- Video태그의 사용 규격은 표준화 되었지만, 각 브라우저별로 지원하는 태그은 아직 표준화되지 않기 때문에 브라우저 별로 인코딩작업을 별도로 해주어야 한다.
- 다음의 코드는 모든 브라우저별로 지원하는 포맷을 분기하는 코드이다.

```
<video [controls] [loop] [autoplay] [preload]
[width="폭"] [height="높이"] [poster="미리보기 이미지"]>
  <!-- mp4: 사파리, IE, iOS, 안드로이드, 윈도우폰 -->
  <source src="*.mp4" type="video/mp4"></source>
  <!-- webm : 파이어폭스와 오페라, 크롬 최신 버전 -->
  <source src="*.webm" type="video/webm"></source>
  <!-- ogg: 파이어폭스 이전 버전과 오페라 -->
  <source src="*.ogg" type="video/ogg"></source>
</video>
```

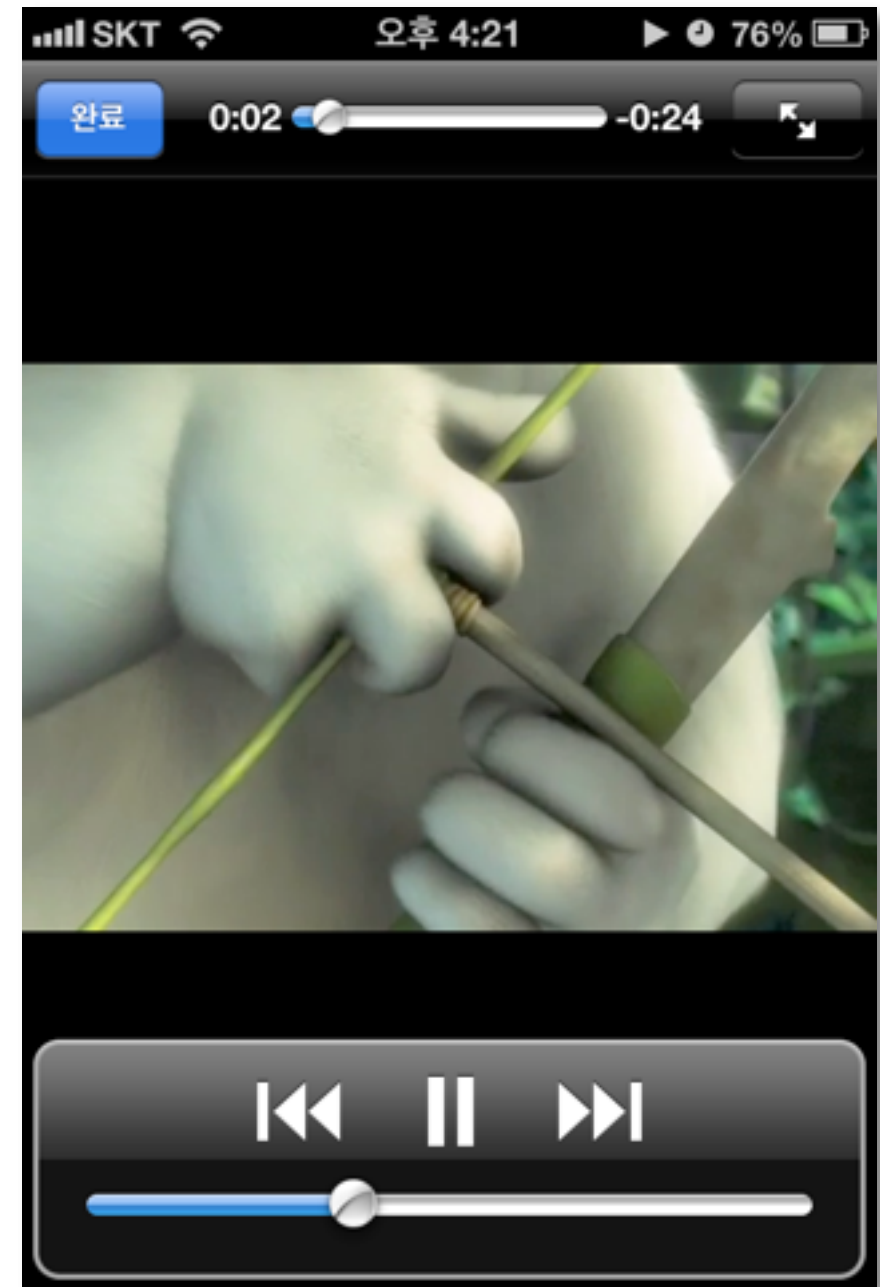
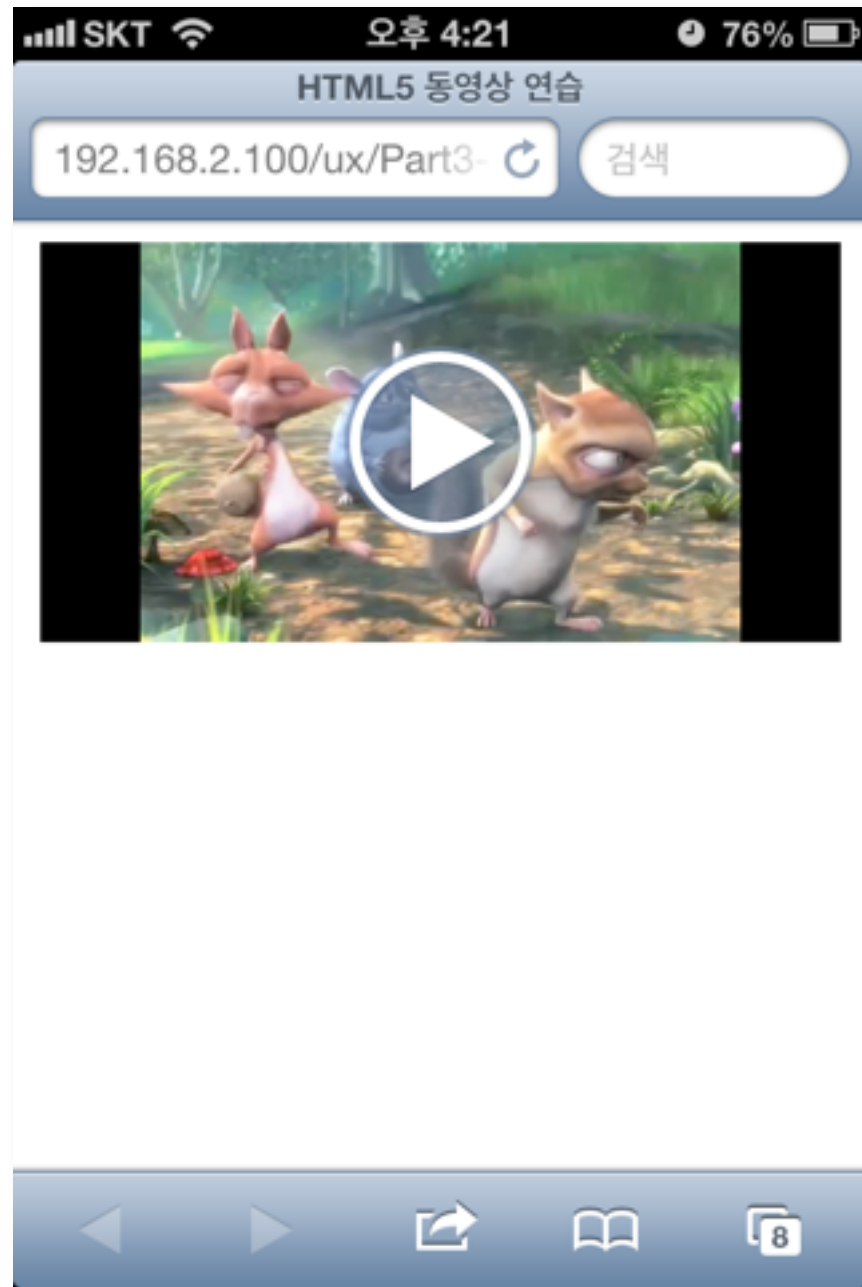
3) 동영상 재생을 위한 <video>태그 (3)

☑ /day02/02-video.html

```
<div style="width: 300px; margin: auto">
  <video style="width: 100%" autoplay controls poster="media/
  BigBuck.png">
    <!-- mp4: 사파리, 크롬, IE, iOS, 안드로이드, 윈도우폰 -->
    <source src="media/BigBuck.mp4" type="video/mp4"></source>
    <!-- webm : 파이어폭스와 오페라, 크롬 최신 버전 -->
    <source src="media/BigBuck.webm" type="video/webm"></source>
    <!-- ogg: 파이어폭스 이전 버전과 오페라 -->
    <source src="media/BigBuck.ogv" type="video/ogg"></source>
  </video>
</div>
```

3) 동영상 재생을 위한 <video>태그 (4)

☑ /day02/02-video.html (실행결과)



2. HTML5 Form 요소

HTML5 & CSS3 기반의 반응형 모바일 웹 만들기

- 1) 기존의 Form요소 살펴보기
- 2) 새롭게 추가된 HTML5 폼 요소
- 3) input태그에 추가된 공통 속성

1) 기존의 Form요소 살펴보기 (1)

☑ Form 태그의 올바른 사용 방법

- 입력양식을 표현하는 <form> 태그는 영역을 의미하는 <fieldset> 태그와 영역의 제목을 의미하는 <legend>태그를 포함한다.
- 입력을 위한 <input>태그에는 id속성이 있어야 하며, <label>태그의 for속성으로 <input>태그의 id속성과 연결하여 타이틀을 설정한다.

☑ /day02/03-form기본.html (실행결과 미리보기)

The diagram illustrates a search form structure with the following components and their corresponding HTML tags:

- fieldset**: The outer container for the form.
- legend**: The title of the form area, labeled "통합검색".
- label**: The label for the search input, labeled "검색어" with `for="search"`.
- input**: The search input field, labeled with `type="text" id="search"`.
- input**: The submit button, labeled with `type="submit"`.

1) 기존의 Form요소 살펴보기 (2)

☑ /day02/03-form기본.html

```
<form name="폼이름" id="아이디값" action="웹 프로그램 연동파일 지정"
method="웹 프로그램 연동방식 지정">
  <fieldset>
    <legend> 통합검색 </legend>
    <label for="keyword">검색어</label>
    <input type="text" name="keyword" id="keyword" />
    <input type="submit" value="검색" />
  </fieldset>
</form>
```

이 코드는 xhtml에서 부터 지원 되던 기능이므로 수업시간의 실습 대상은 아닙니다.
하지만 XHTML에 대한 연습이 필요하신 분은 꼭 연습해 보시기 바랍니다.

1) 기존의 Form요소 살펴보기 (3)

XHTML의 Form 요소

태그	type	설명
input	text	이름과 같은 일반 텍스트를 입력한다.
	password	비밀번호를 입력한다.
	radio	여러 항목중에서 한 가지만 선택 가능한 라디오 버튼을 표시한다.
	checkbox	다중 선택이 가능한 체크박스를 표시한다.
	hidden	화면상에 보이지 않는 숨김 정보를 처리한다.
	button	버튼을 표시한다. 이 버튼은 클릭했을 경우 아무런 동작도 수행하지 않는다. 자바스크립트로 동작을 구현해야 한다.
	submit	사용자의 입력 내용을 웹 프로그램으로 전송하기 위한 버튼을 표시한다.
	image	이미지 버튼을 표시한다. 이미지 경로를 지정해 주어야 하며, 동작은 submit과 동일하다.
	reset	입력내용을 초기화 하는 버튼을 표시한다.
select	-	드롭다운을 표시한다.
textarea	-	여러 줄을 입력할 수 있는 텍스트 박스를 표시한다.

1) 기존의 Form요소 살펴보기 (4)

☑ /day02/04-xhtml-form.html (step-1)

```
<form name="form1" method="get" action="form_action.html">
```

```
<h1>회원가입</h1>
```

```
<fieldset>
```

```
<legend>기본 정보</legend>
```

```
<label for="user_id">아이디: </label>
```

```
<input type="text" name="user_id" id="user_id" />
```

```
<input type="button" value="중복검사" />
```

```
<br/>
```

```
<label for="user_pw">비밀번호: </label>
```

```
<input type="password" name="user_pw" id="user_pw" />
```

```
<br/>
```

```
<label for="user_job">직업: </label>
```

```
<select name="user_job" id="user_job">
```

```
<option>선택하세요.</option>
```

```
<option value="회사원">회사원</option>
```

이 코드는 xhtml에서 부터 지원 되던 기능이므로 수업시간의 실습 대상은 아닙니다.

하지만 XHTML에 대한 연습이 필요하신 분은 꼭 연습해 보시기 바랍니다.

1) 기존의 Form요소 살펴보기 (5)

/day02/04-xhtml-form.html (step-2)

```
<option value="자영업">자영업</option>
<option value="기타">기타</option>
</select>
<br/>

<label form="user_photo">사진:</label>
<input type="file" name="user_photo" id="user_photo" />
<br />

<label>취미:</label>
<input type="checkbox" name="hobby" id="hobby1" value="독서" />
<label for="hobby1" class="check">독서</label>
<input type="checkbox" name="hobby" id="hobby2" value="영화" />
<label for="hobby2" class="check">영화</label>
<input type="checkbox" name="hobby" id="hobby3" value="음악" />
<label for="hobby3" class="check">음악</label>
<br />
```

1) 기존의 Form요소 살펴보기 (6)

☑ /day02/04-xhtml-form.html (step-3)

```
<label>성별:</label>
```

```
<input type="checkbox" name="gender" id="gender1" value="남자" />
```

```
<label for="gender1" class="check">남자</label>
```

```
<input type="checkbox" name="gender" id="gender2" value="여자" />
```

```
<label for="gender2" class="check">여자</label>
```

```
<br/>
```

```
<label for="introduce">자기소개:</label>
```

```
<textarea name="introduce" /></textarea>
```

```
<br/>
```

```
<input type="reset" value="다시입력" />
```

```
<input type="submit" value="가입하기" />
```

```
<input type="image" src="img/btn_agreement.gif" />
```

```
</fieldset>
```

```
</form>
```

1) 기존의 Form요소 살펴보기 (7)

☑ /day02/04-xhtml-form.html (결과)

회원가입

기본 정보

아이디:

중복검사

비밀번호:

직업:

사진: 선택한 파일 없음

취미: ☐ 독서 ☐ 영화 ☐ 음악

성별: ☐ 남자 ☐ 여자

자기소개:

2) 새롭게 추가된 HTML5 폼 요소 (1)

새로워진 폼 요소 (1)

- HTML5에서는 <input>태그의 type값이 세분화 되었다.
- 이 속성은 PC용 웹 브라우저에서는 영향을 받지 않지만, 스마트 디바이스에서는 키보드에 대한 초기 상태에 대하여 영향을 준다.

type값	유형
email	이메일 주소를 입력한다.
url	웹 URL을 입력한다.
number	숫자값을 입력한다.
tel	전화번호 양식을 입력한다.
range	마우스로 드래그 하여 수치를 입력할 수 있다.
date	날짜를 입력한다.
datetime	날짜와 시간을 입력한다.
search	검색어를 입력한다. 우측에 내용을 삭제하기 위한 버튼이 표시된다.

2) 새롭게 추가된 HTML5 폼 요소 (2)

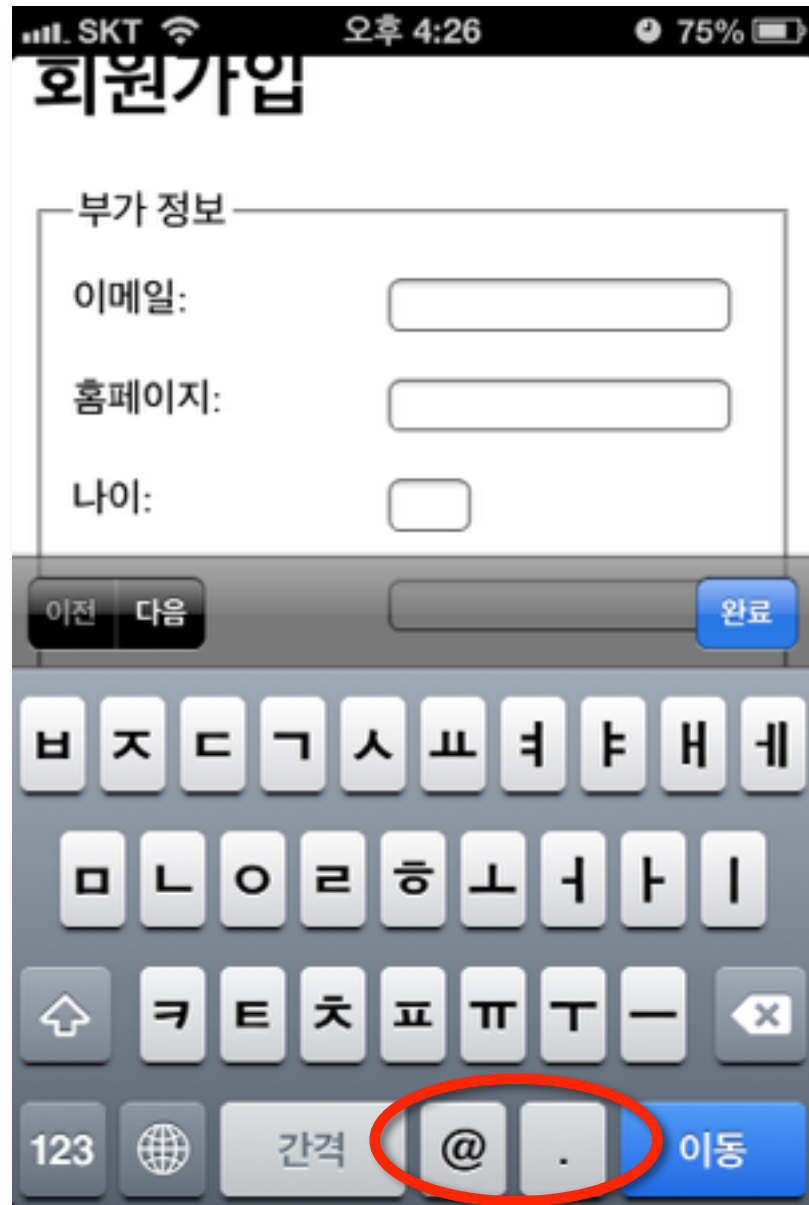
<button> 태그

- <input type="submit | reset | button" />을 대체하기 위한 새로운 태그.
- 오직 <form>태그에만 지정할 수 있던 method, action 속성을 개별적으로 지정할 수 있다.
- 기존에는 <form>하나 당, (버튼의 수에 상관 없이) 한 개의 웹 프로그램만 연동 했지만, 이제는 개별 버튼별로 연동을 지정할 수 있다.

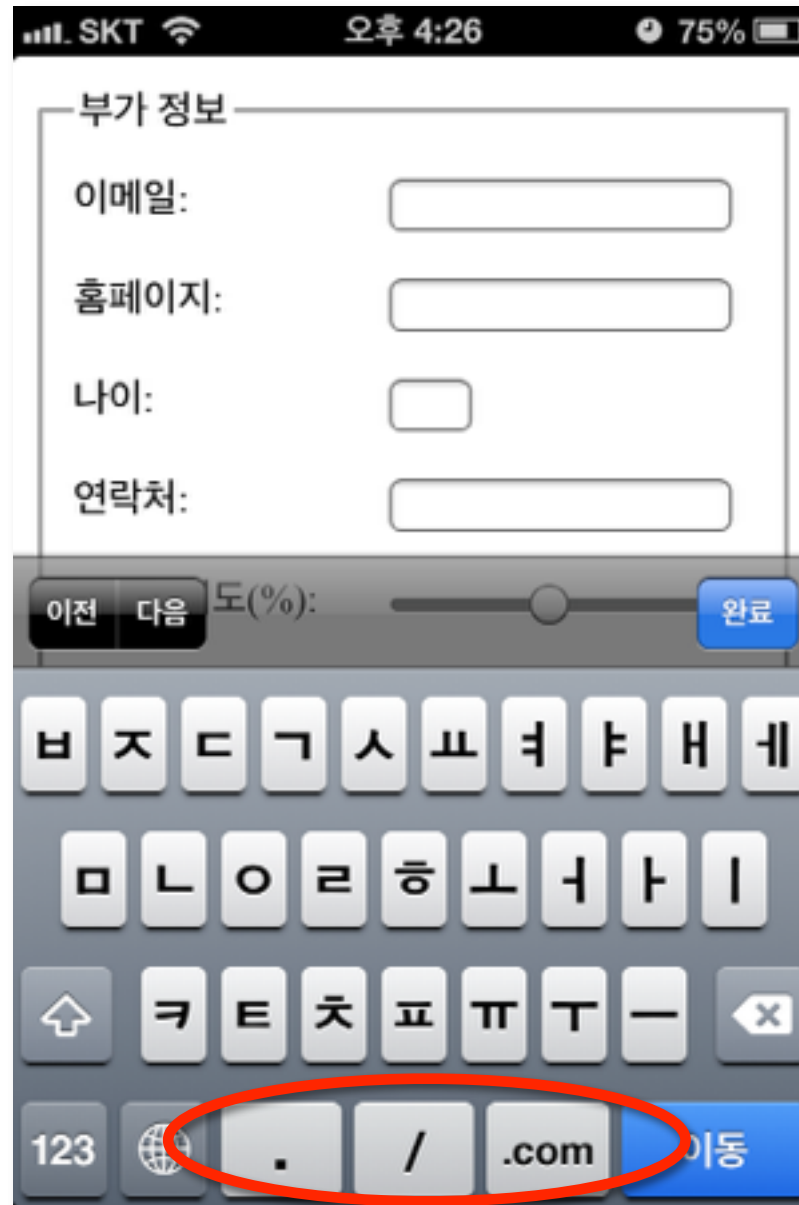
속성	값	설명
type	submit, button, reset	버튼의 종류를 지정한다.
formaction	url	데이터를 전송할 파일을 지정한다.
formmethod	get, post	전송 방식을 지정한다.
formtarget	_blank, _self, _parent, _top	form의 target속성을 덮어 쓴다.

2) 새롭게 추가된 HTML5 폼 요소 (3)

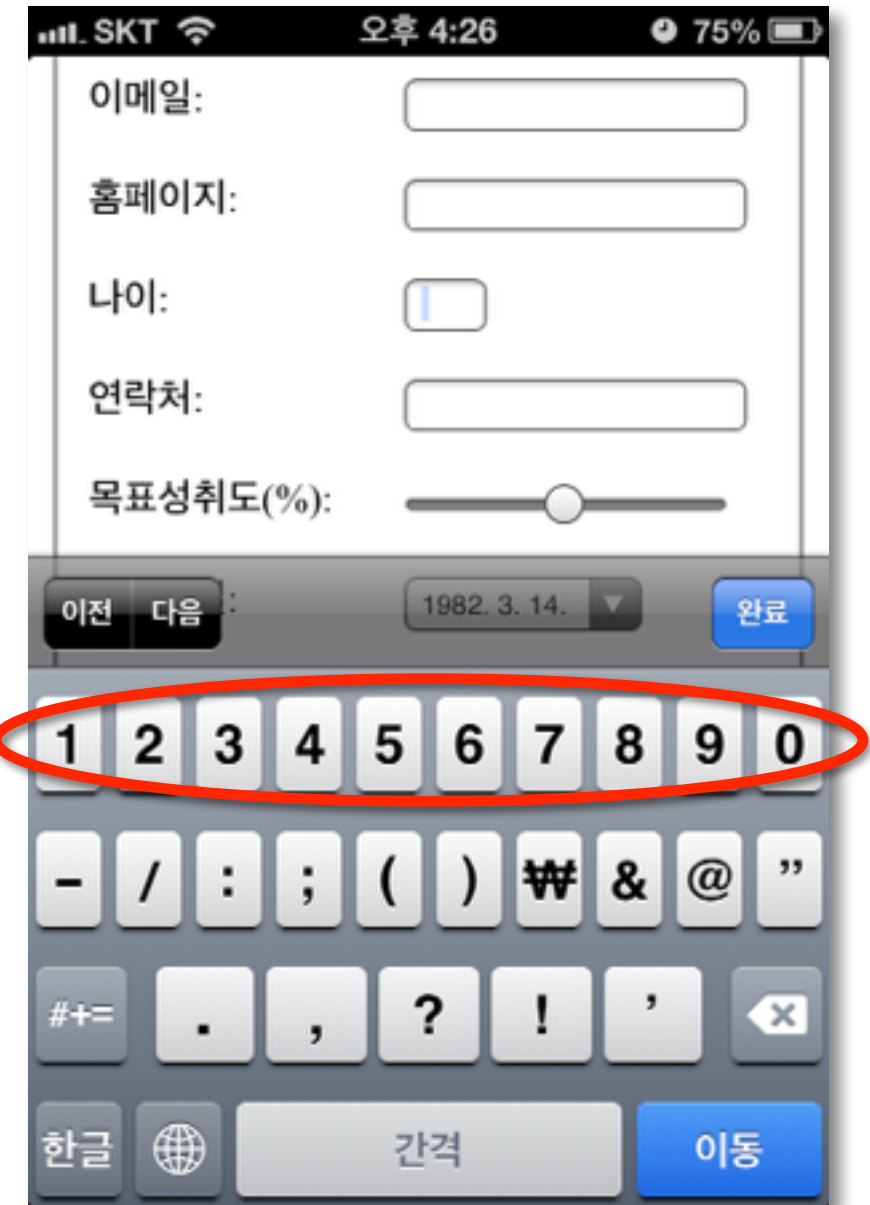
☑ 새로워진 폼 요소 (2) - iOS



input type="email"



input type="url"



input type="number"

2) 새롭게 추가된 HTML5 폼 요소 (4)

☑ 새로워진 폼 요소 (3) - iOS

이 화면은 iOS 기기에서 실행 중인 웹 애플리케이션의 폼을 보여줍니다. 상단 상태 표시줄에는 SKT 통신사, 신호 강도, 와이파이, 시간(오후 4:26), 배터리(75%)가 표시되어 있습니다. 폼에는 다음 요소가 포함되어 있습니다:

- 홈페이지: 텍스트 입력 필드
- 나이: 텍스트 입력 필드
- 연락처: 텍스트 입력 필드
- 목표성취도(%): 슬라이더 컨트롤
- 생년월일: 날짜 선택기 (1982. 3. 14. 선택됨)

폼 하단에는 "이전", "다음", "완료" 버튼이 있습니다. "완료" 버튼이 활성화된 상태입니다. 화면 하단에는 iOS의 표준 키패드가 표시되어 있습니다.

input type="tel"

이 화면은 iOS 기기에서 실행 중인 웹 애플리케이션의 폼을 보여줍니다. 상단 상태 표시줄에는 SKT 통신사, 신호 강도, 와이파이, 시간(오후 4:26), 배터리(75%)가 표시되어 있습니다. 폼에는 다음 요소가 포함되어 있습니다:

- 연락처: 텍스트 입력 필드
- 목표성취도(%): 슬라이더 컨트롤
- 생년월일: 날짜 선택기 (1982. 3. 14. 선택됨)
- 가입일시: 날짜 선택기 (1982. 3. 14. 선택됨)

폼 하단에는 "이전", "다음", "지우기", "완료" 버튼이 있습니다. "완료" 버튼이 활성화된 상태입니다. 화면 하단에는 iOS의 날짜 선택기(달력)가 표시되어 있으며, 1982년 3월 14일이 선택된 상태입니다.

input type="date"

이 화면은 iOS 기기에서 실행 중인 웹 애플리케이션의 폼을 보여줍니다. 상단 상태 표시줄에는 SKT 통신사, 신호 강도, 와이파이, 시간(오후 4:26), 배터리(75%)가 표시되어 있습니다. 폼에는 다음 요소가 포함되어 있습니다:

- 목표성취도(%): 슬라이더 컨트롤
- 생년월일: 날짜 선택기 (1982. 3. 14. 선택됨)
- 가입일시: 날짜 및 시간 선택기 (2012. 10. 23. 오후 4:26 선택됨)
- 추천인 아이디: 텍스트 입력 필드

폼 하단에는 "이전", "다음", "지우기", "완료" 버튼이 있습니다. "완료" 버튼이 활성화된 상태입니다. 화면 하단에는 iOS의 날짜 및 시간 선택기(달력)가 표시되어 있으며, 2012년 10월 23일 오후 4:26이 선택된 상태입니다.

input type="datetime"

2) 새롭게 추가된 HTML5 폼 요소 (5)

☑ 새로워진 폼 요소 (4) - iOS

A screenshot of an iOS mobile web application showing a form with several input fields. The top status bar shows 'SKT', signal strength, time '오후 4:26', and battery '75%'. The form fields are: '목표성취도(%)' with a range slider, '생년월일' with a date picker showing '1982. 3. 14.', '가입일시' with a date-time picker showing '2012. 10. 23. 오후 4:26', and '추천인 아이디' with a text input field. Below the form is a navigation bar with '이전', '다음', and '완료' buttons. At the bottom is a QWERTY keyboard. The '검색' (Search) button on the keyboard is circled in red.

`<input type="range">`

`input type="search"`

2) 새롭게 추가된 HTML5 폼 요소 (6)

☑ 새로워진 폼 요소 (5) - Android

- Android의 경우 type="datetime"은 지원되지 않는다.

A screenshot of an Android browser interface. The address bar shows a URL. Below it, there's a form with labels: "홈페이지:", "나이:", "연락처:", "목표성취도(%)", and "생년월일:". The "목표성취도(%)", "연락처:", and "생년월일:" fields are currently active, and a numeric keypad is displayed on the screen.

input
type="search"

A screenshot of an Android browser interface. The address bar shows a URL. Below it, there's a form with labels: "홈페이지:", "나이:", "연락처:", "목표성취도(%)", and "생년월일:". The "생년월일:" field is active, and a date picker is displayed on the screen, showing the date "1982년 3월 14일 일요일".

input type="date"

A screenshot of an Android browser interface. The address bar shows a URL. Below it, there's a form with labels: "회원가입", "부가 정보", "이메일:", "홈페이지:", and "나이:". The "이메일:" field is active, and a numeric keypad is displayed on the screen.

input type="url"

A screenshot of an Android browser interface. The address bar shows a URL. Below it, there's a form with labels: "회원가입", "부가 정보", "이메일:", "홈페이지:", and "나이:". The "이메일:" field is active, and a numeric keypad is displayed on the screen.

input
type="number"

iOS와 중복되므로, 대표 속성 몇가지만 제시합니다.

2) 새롭게 추가된 HTML5 폼 요소 (7)

number 타입과 range 타입의 추가 속성

- input type="number"와 input type="range"는 공통적으로 수치값을 입력하기 위한 컴포넌트이기 때문에, 수치값의 최대값, 최소값, 입력가능한 값의 단계에 대한 추가적인 속성들이 필요하다.

속성명	설명
max	컴포넌트에 입력 가능한 최대값을 숫자로 설정한다.
min	컴포넌트에 입력 가능한 최소값을 숫자로 설정한다.
step	컴포넌트에 입력 가능한 값의 단계를 숫자로 설정한다. 예를 들어서 5라고 입력하면 input type="number"는 5의 배수만 입력가능하다. 또한 input type="range"는 슬라이드를 왼쪽이나 오른쪽으로 드래그 할 때 마다 5 단위로 값이 변하게 된다.

2) 새롭게 추가된 HTML5 폼 요소 (8)

☑ /day02/05-html5-form.html (step-1)

실행결과는 앞에서 설명을 위해
미리 제시하였으므로 생략합니다.

```
<form name="form1" method="get" action="form_action.html">
  <h1>회원가입</h1>
  <fieldset>
    <legend> 부가 정보 </legend>

    <label for="user_mail">이메일: </label>
    <input type="email" name="user_mail" id="user_mail" />
    <br/>

    <label for="user_homepage">홈페이지: </label>
    <input type="url" name="user_homepage" id="user_homepage" />
    <br/>

    <label for="user_age">나이: </label>
    <input type="number" name="user_age" id="user_age" min="19"
    max="100" step="1"/>
    <br />
```

2) 새롭게 추가된 HTML5 폼 요소 (9)

☑ /day02/05-html5-form.html (step-2)

```
<label>목표성취도(%):</label>
```

```
<input type="range" name="point" id="point" value="50" min="0"  
max="100"/>
```

```
<br />
```

```
<label for="birthday">생년월일:</label>
```

```
<input type="date" name="birthday" id="birthday"  
value="1982-03-14" />
```

```
<br/>
```

```
<label for="worktime">출근시간:</label>
```

```
<input type="time" name="worktime" id="worktime" value="09:00" />
```

```
<br/>
```

```
<label for="join_datetime">가입일시:</label>
```

```
<input type="datetime" name="join_datetime" id="join_datetime"  
value="1982-03-14 13:15:30" />
```

```
<br/>
```

2) 새롭게 추가된 HTML5 폼 요소 (10)

☑ /day02/05-html5-form.html (step-3)

```
<label for="friend">추천인 아이디:</label>  
<input type="search" name="friend" id="friend" />  
<br/>
```

```
<button type="submit" formaction="action1.html" formmethod="get"  
formtarget="_blank">14세 이상 회원가입</button>  
<button type="submit" formaction="action2.html" formmethod="post"  
formtarget="_self">14세 미만 회원가입</button>
```

```
</fieldset>  
</form>
```

3) input태그에 추가된 공통 속성 (11)

☑ placeholder 속성

- placeholder 속성은 사용자가 직접 값을 입력하는 종류의 양식에 지정하여 값이 입력되기 전에 보여질 설명글을 표시한다.
- 그 외에 알려진 추가 속성들은 iOS에서 현재 지원하지 않고 있다.

☑ /day02/06-placeholder.html

```
<form name="form1" id="myform">  
  <label for="keyword">검색어를 입력하세요.</label>  
  <input type="search" id="keyword" name="keyword" placeholder="검색  
어를 입력하세요." />  
</form>
```

검색어를 입력하세요.

검색어를 입력하세요.

3. CSS3 시작하기

HTML5 & CSS3 기반의 반응형 모바일 웹 만들기

- 1) CSS의 표현방법
- 2) CSS의 셀렉터와 구체성 단위
- 3) CSS3의 새로운 가상 클래스
- 4) 브라우저마다 다르게 구현되는 프로퍼티

1) CSS의 표현 방법 (1)

CSS란?

- Cascading StyleSheet의 약자로서 HTML구문이 정의한 문서의 구조에 그래픽적인 요소를 입히는 스킨의 역할을 하는 언어.
- 글자의 모양, 배경 색상, 태그 요소의 위치 등을 지정할 수 있다.
- HTML태그를 지시하는 이름안에 “속성: 값”의 형식으로 기술된다.
- HTML태그를 지시하는 이름을 Selector라고 한다.

```
selector { 속성: 값; 속성: 값; ... 속성: 값; }
```


1) CSS의 표현 방법 (2)

☑ CSS를 HTML에 적용하는 방법

- <head> ~ </head> 사이에 다음과 같이 처리.

```
<style type="text/css">  
  selector { 속성: 값; 속성: 값; ... 속성: 값; }  
</style>
```

- 별도의 파일에 *.css 형식의 파일을 구성하고, <head>~</head> 사이에 해당 파일을 참조하도록 처리

```
selector { 속성: 값; 속성: 값; ... 속성: 값; } /css/style.css
```

```
<link rel="stylesheet" type="text/css" href="/css/style.css" />
```

2) CSS의 셀렉터와 CSS 구체성 단위 (1)

CSS 셀렉터 표현의 종류

이름	표현방법	설명
태그방식	selector { ... }	특정 태그를 가리킨다. HTML내에 동일 태그가 존재할 경우 모든 태그 요소를 일괄 처리한다.
Class방식	.selector { ... }	셀렉터이름 앞에 점(.)을 붙여서 표시하고 HTML태그의 class 속성에 점을 제외한 이름을 명시하여 지정한다. 여러 요소에 태그의 종류를 가리지 않고 여러번 지정 가능하다. (재사용 목적) 이 경우 selector의 이름은 마음대로 지정 가능하다. 사용예 : <div class="selector">
ID방식	#selector { ... }	셀렉터 이름 앞이 (#)을 붙여서 표시하고 HTML태그의 id속성에 섹트 (#)을 제외한 이름을 명시하여 지정한다. id속성의 값은 HTML페이지 안에서 중복 사용될 수 없다. (고유특성 명시) 이 경우 selector의 이름은 마음대로 지정 가능하다. 사용예 : <div id="selector">

2) CSS의 셀렉터와 CSS 구체성 단위 (2)

CSS 셀렉터의 조합 방법

- CSS 셀렉터의 조합 방법은 CSS2에서 30가지가 있으며, CSS3에서 30가지가 새롭게 추가되었다. 자주 사용되는 기법은 알아 두면 HTML 태그를 제어하기가 수월해 진다.

콤마로 구분하기 --> And의 의미를 갖는다.

- 한번의 명시로 여러개의 selector에 적용하기 위해서 콤마(,)로 구분할 수 있다.

```
selector1, selector2, selector3 ... , selector(n) {  
    속성1: 값1;  
    속성2: 값2;  
    ...  
    속성n: 값n;  
}
```

2) CSS의 셀렉터와 CSS 구체성 단위 (2)

☑ 공백으로 구분하여 태그의 구조를 표현 --> (자손 셀렉터)

- 태그의 계층 관계를 나타낸다. A안의 B라는 형식이다. **p div #pwd**라고 명시할 경우 “p태그 안의 div태그 안의 pwd라는 id값을 갖는 요소”라는 의미가 되어 두번째 textfield를 가리킨다.
- 명시 과정에서 중간단계를 생략해도 된다. 몇 depth를 더 내려가던, 지정된 태그의 범위 안에 존재하기만 하면 된다.

```
<p>
  <div>
    <fieldset>
      <input type="text" />
      <input type="password" id="pwd"/>
      <input type="checkbox" />
    </fieldset>
  </div>
</p>
```

p div #pwd { ... }

2) CSS의 셀렉터와 CSS 구체성 단위 (3)

☑ 공백 없이 명시하여 구체적 서술

- 주로 HTML태그 이름과 Class 혹은 id속성과 함께 명시된다. 특정 요소를 더욱 정교하게 명시하고자 할 경우 사용되는데 A#B일 경우 B라는 아이디가 적용된 A태그, A.B일 경우 B라는 클래스가 적용된 A태그의 의미가 된다.

```
<p>
  <div>
    <fieldset>
      <input type="text" />
      <input type="password" id="pwd"/>
      <input type="checkbox" />
    </fieldset>
  </div>
</p>
```

p div input#pwd { ... }

2) CSS의 셀렉터와 CSS 구체성 단위 (4)

☑ “>”로 구분하여 구조적 표현

- 공백으로 지정한 방식이 태그의 계층에 상관 없이 범위 안에 존재하기만 하면 요소를 가리킬 수 있었다면 “>”는 반드시 1depth 안에 존재해야만 요소를 지정할 수 있다.

```
<p>
  <div>
    <fieldset>
      <input type="text" />
      <input type="password" id="pwd"/>
      <input type="checkbox" />
    </fieldset>
  </div>
</p>
```

p > div > fieldset > input#pwd { ... }

2) CSS의 셀렉터와 CSS 구체성 단위 (5)

☑ 태그의 속성에 따른 표현

- 스타일시트의 셀렉터가 적용되는 대상에게 특정 속성이 있는지 여부와 특정 속성의 값이 적용되어 있는지 여부에 따라서 태그 요소를 좀 더 구체적으로 가리킬 수 있다.

```
<p>
```

```
<div>
```

```
<fieldset>
```

```
<input type="text" />
```

```
<input type="password" id="pwd"/>
```

```
<input type="checkbox" />
```

```
</fieldset>
```

```
</div>
```

```
</p>
```

`input[type='password'] { ... }`

`input[type] { ... }`

2) CSS의 셀렉터와 CSS 구체성 단위 (4)

가상클래스

- HTML태그에서 특정 상황이 발생했을 경우에만 적용되는 CSS셀렉터도 있다. 이를 가상 클래스라고 한다. 가상클래스는 CSS의 셀렉터 이름 뒤에 “: 상황”의 형식으로 명시한다.

명시방법	의미
:link	링크의 기본 상태
:active	요소 위에서 마우스가 눌러져 있는 상태
:hover	요소에 마우스가 올라가 있는 상태
:visited	방문했던 경험이 있는 링크
:focus	요소에 포커스가 지정된 상태
:checked	체크박스나 라디오버튼등이 체크된 상태
:selected	드롭다운에서 선택된 요소

2) CSS의 셀렉터와 CSS 구체성 단위 (5)

CSS 구체성 단위

- 여러가지 형태로 조합된 CSS 셀렉터는 태그, CLASS, ID 형식의 요소가 몇 번 지정되었는가에 따라서 각각 점수를 부여받는다.
- 부여된 점수가 높은 형식이 우선적으로 적용된다.

명시방법	점수
TAG	1점
CLASS	10점
ID	100점
HTML태그의 style속성 사용	1000점

- * div.bbs_content : TAG방식1회 + CLASS방식1회 = 11점
- * body > div#container > div#header : TAG 3회 + ID2회 = 203점
- * body, p, span, input, textarea : TAG 4회 = 4점

2) CSS의 셀렉터와 CSS 구체성 단위 (6)

☑ /day02/07-selector.html (step-1)

- 다음의 HTML 태그를 <body>~</body> 사이에 작성한다.

```
<h1>셀렉터 예제</h1>
<div id="menu">
<div>메뉴1</div>
<div>메뉴2</div>
<div>메뉴3</div>
<div>메뉴4</div>
</div>
```

2) CSS의 셀렉터와 CSS 구체성 단위 (7)

☑ /day02/07-selector.html (step-2)

- <head>~</head> 사이에 다음의 CSS를 작성하고 결과를 확인한다.

```
<style type="text/css">
  body {
    color: #26102E;
  }

  #menu > div {
    color: #0D88C1;
  }

  #menu > div:last-child {
    color: #CC00CC;
  }
</style>
```

← color: 글자색상;

2) CSS의 셀렉터와 CSS 구체성 단위 (8)

☑ /day02/07-selector.html (결과)

iOS7 화면
입니다.



3) CSS3의 새로운 가상 클래스 (1)

☑ nth-child 가상 클래스 (1)

- HTML5의 CSS3에서는 태그 요소를 좀 더 구조적으로 명시할 수 있도록 하기 위해서, 여러개의 요소중에서 n번째 요소라는 의미로 :nth-child(n) 형식이 추가되었다.
- 요소를 뒤에서부터 카운트 할 경우 nth-last-child(n)을 사용한다.

```
<ul id="menu">
  <li>Item 01</li>
  <li>Item 02</li>
  <li>Item 03</li>
  <li>Item 04</li>
</ul>
```

ul > li:nth-child(2)

ul > li:nth-last-child(2)

3) CSS3의 새로운 가상 클래스 (2)

☑ nth-child (2)

- 주어진 요소에 대해서 특정 수의 배수에 해당하는 요소만 적용할 경우 `li:nth-child(2n)`과 같은 형식을 사용한다.
- 홀수번째 요소만을 지정할 경우 `nth-child(odd)`라고 지정한다.

```
<ul id="menu">  
  <li>Item 01</li>  
  <li>Item 02</li>  
  <li>Item 03</li>  
  <li>Item 04</li>  
</ul>
```

`ul > li:nth-child(2n)`

`ul > li:nth-child(odd)`

3) CSS3의 새로운 가상 클래스 (3)

/day02/08-nth-child.html (step-1)

- 각 step 별로 결과를 확인하면서 변경사항을 체크하세요.
- <body>안에 다음의 HTML태그를 작성.

```
<ul>
  <li>1</li>
  <li>2</li>
  <li>3</li>
  <li>4</li>
  <li>5</li>
  <li>6</li>
  <li>7</li>
  <li>8</li>
  <li>9</li>
  <li>10</li>
</ul>
```

3) CSS3의 새로운 가상 클래스 (4)

✓ /day02/08-nth-child.html (step-2)

● <head> 태그 안에 다음의 CSS구문을 작성.

* 목록 정의에 대한 정렬과 크기 및 기본 글자 형태를 결정하는 CSS2 구문입니다.

```
<style type="text/css">
  ul {
    list-style: none;
    margin: 0; padding: 0;
    text-align: center;
  }

  li {
    width: 10%; padding: 5px 0;
    display: block;
    float: left;
    font-size: 30px; font-weight: bold;
  }
```

3) CSS3의 새로운 가상 클래스 (5)

☑ /day02/08-nth-child.html (step-3)

● <style> 태그 안에 다음의 CSS구문을 추가

* 셀렉터는 CSS3에서 추가된 형식이지만, 내용은 CSS2의 기능입니다.

```
li:nth-child(odd) {  
    color: red; background-color: green;  
}  
  
li:nth-child(2) {  
    color: green; background-color: red;  
}  
  
li:nth-child(3n) {  
    color: blue; background-color: yellow;  
}  
</style>
```

3) CSS3의 새로운 가상 클래스 (6)

☑ /day02/08-nth-child.html (결과)

- nth-child(odd) - 홀수 번째 요소에 대해서 글자색 빨강, 배경색 초록
- nth-child(2) - 두 번째 요소에 대해서 글자색 초록, 배경색 빨강
- nth-child(3n) - 3의 배수 번째 요소 (3, 6, 9번째 요소) 에 대해서 글자색 파랑, 배경색 노랑



3) CSS3의 새로운 가상 클래스 (7)

target 가상 클래스 (1)

- 현재의 HTML 페이지 안에서 링크를 통하여 활성화 되는 가상 클래스.
- 링크에서 # 기호의 의미
 - * HTML의 링크에서 “#” 기호는 현재 페이지를 의미한다.
 - * 아래의 링크는 클릭시, 페이지 이동이 아니라 현재 페이지의 맨 처음으로 이동한다.

```
<a href="#">클릭하세요</a>
```

● CSS에서 # 기호의 의미

- * HTML태그에 지정된 id속성값을 의미한다.

```
<div id="#foo">여기입니다.</div>
```

```
#foo { ... }
```

3) CSS3의 새로운 가상 클래스 (8)

target 가상 클래스 (2)

● 링크의 #과 CSS의 #기호는 같은 것이다.

- * 다음의 HTML 태그가 한 페이지 상에서 동시에 존재하고 있을 경우, 링크를 클릭하면 현재 페이지(#)에서 foo라는 아이디를 갖는 요소를 찾아간다.
- * 즉, 스크롤 이동이 발생한다.

```
<a href="#foo">클릭하세요</a>  
... 생략 ...  
<div id="#foo">여기입니다.</div>
```

● 가상 클래스 :target

- * 다음의 CSS 블록은 평상시에는 동작하지 않다가, 해당 요소가 링크에 의해서 이동된 경우에만 동작한다.

```
#foo:target { ... }
```

3) CSS3의 새로운 가상 클래스 (9)

☑ /day02/09-target.html (step-1) - 문서 구조 표현

- <body>태그 안에 작성하세요.

```
<section>
  <nav>
    <ul>
      <li> <a href="#html5">HTML5</a> </li>
      <li> <a href="#css3">CSS3</a> </li>
      <li> <a href="#js">Javascript</a> </li>
    </ul>
  </nav>
  <article>
    /day03/txt/03-target.txt 파일에 전문이 있습니다.
    <p id="html5">HTML5는 HTML 4.01, XHTML 1.0, ... </p>
    <p id="css3">CSS 또는 캐스케이딩 스타일 시트...</p>
    <p id="js">자바스크립트(Javascript)는 ...</p>
  </article>
</section>
```

3) CSS3의 새로운 가상 클래스 (10)

☑ /day02/09-target.html (step-2) - 목록의 박스 모양과 배치

- <head>태그 안에 <style>블록을 추가하고 작성하세요.

```
ul {  
  list-style: none;  
  width: auto;  
  padding: 0;  
  margin: 0;  
}  
  
ul > li {  
  width: 33%;  
  float: left;  
  border: 1px solid #ccc;  
  border-left: 0;  
}  
  
ul > li:first-child {  
  border-left: 1px solid #ccc;  
}
```

3) CSS3의 새로운 가상 클래스 (11)

☑ /day02/09-target.html (step-3) - float 사용에 대한 마감 처리

- <head>태그 안에 <style>블록을 추가하고 작성하세요.

```
ul:after {  
    content: ''; display: block;  
    width: 0; height: 0;  
    float: none; clear: both;  
}
```

```
ul > li > a {  
    display: block;  
    padding: 10px 0;  
    width: auto;  
    text-align: center;  
    text-decoration: none;  
    font: bold 16px "Helvetica";  
    color: #aa0000;  
    background: #eee;  
}
```

3) CSS3의 새로운 가상 클래스 (12)

☑ /day02/09-target.html (step-4) - 본문을 숨김

- style 블록 안에 추가하세요.

```
article > p { display: none; }
```

☑ /day02/09-target.html (step-5) - target 을 적용한 탭 기능 구현

```
article > p:target {  
    display: block;  
    border: 3px solid #aa0000;  
    padding: 5px;  
    background: #FFFB3;  
}
```


3) CSS3의 새로운 가상 클래스 (13)

✓ /day02/09-target.html (결과 미리보기)

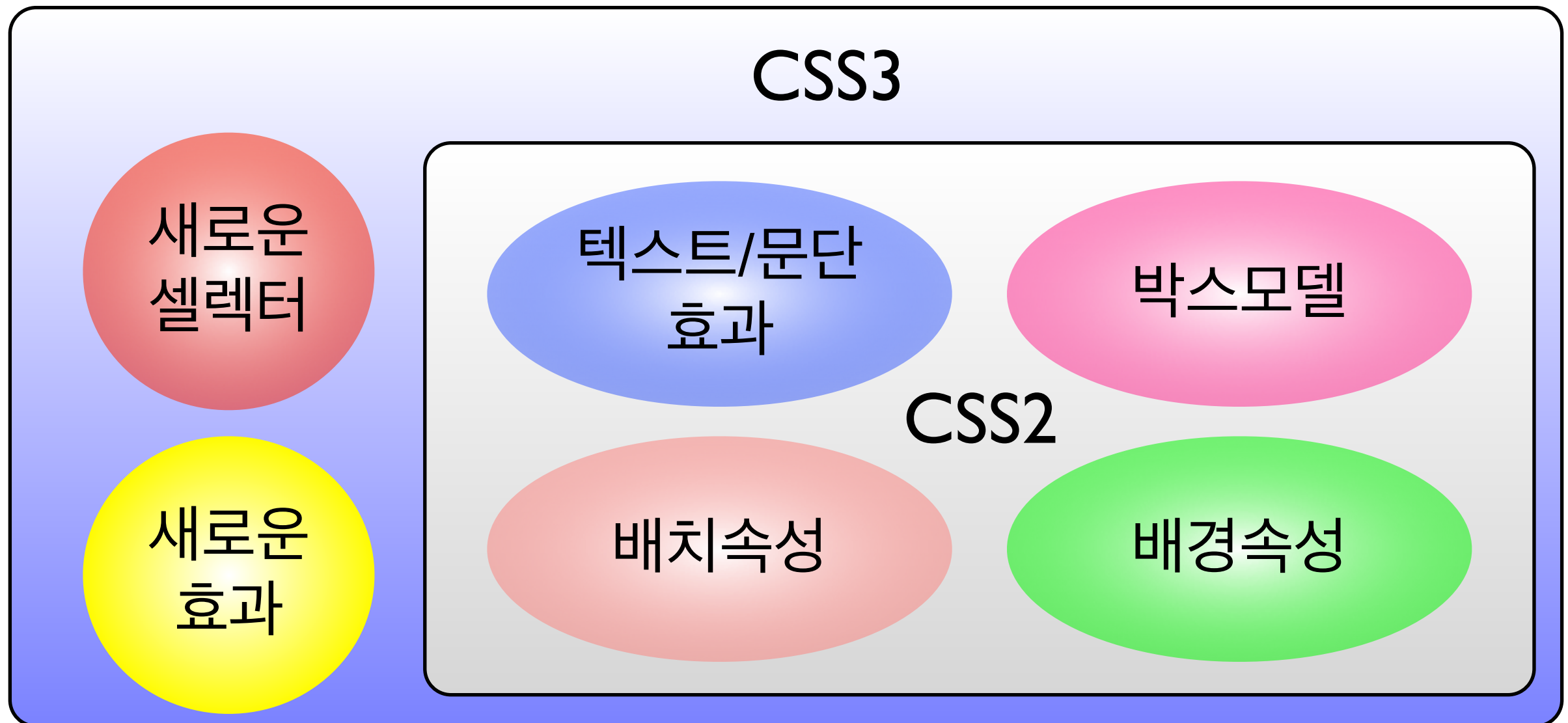


4) 브라우저마다 다르게 구현되는 프로퍼티 (1)

☑ CSS2와 CSS3의 관계

- CSS3는 CSS2의 모든 속성들을 그대로 지원하면서, 새로운 효과들이 추가되어 있다.

* 기존의 CSS2를 이해하고 있는 것은 CSS3를 파악하는데 매우 중요하다.



4) 브라우저마다 다르게 구현되는 프로퍼티 (2)

아직 발전 중인 CSS3

- **프로퍼티 = 속성**
- CSS3의 새로운 속성들에 대해서는 아직까지 각 브라우저 업체들 간의 규약만 책정되어 있고, 각각의 브라우저 개발사들이 부분적으로 구현중인 상태이다.
- 브라우저의 종류별로 구현에 대한 진행 상태가 서로 다르다.
- 그러므로 대부분의 CSS3 속성들은 속성의 이름 앞에 브라우저를 식별할 수 있는 접두사를 붙여서 사용해야 한다.
- 이러한 방법은 CSS 코드를 지저분하게 만들고 디자인에 일관성이 없어지게도 하겠지만, 현재로서는 모든 브라우저에서 동일한 결과를 만들어 내기 위한 최선의 방법이다.

4) 브라우저마다 다르게 구현되는 프로퍼티 (3)

브라우저별 접두사의 종류

- 지금은 상당수의 속성들이 표준화 되어 있어서 적용할 필요가 없어진 속성들이 많아졌지만, CSS3 초창기에는 거의 대부분의 속성을 구현하는데 있어서 브라우저 접두사를 적용하여 4번씩 코드를 기술해야 했었다.

접두사	설명
-webkit-	웹킷 계열 브라우저 (사파리나 구글 크롬)
-moz-	모질라 계열 브라우저 (파이어폭스)
-o-	오페라 브라우저
-ms-	마이크로소프트 인터넷 익스플로러 9 이상

4) 브라우저마다 다르게 구현되는 프로퍼티 (4)

☑ 브라우저별 프로퍼티의 사용 예

- CSS2에서는 특정 속성에 대해서 다음과 같이 한 번만 명시하였다.

```
selector1 {  
    속성이름: 적용할 값;  
}
```

- CSS3에서는 몇몇 속성들에 대해서 다음과 같이 브라우저별로 동일한 내용을 기술하는 경우가 있다.

```
selector1 {  
    -webkit-속성이름: 적용할 값;  
    -moz-속성이름: 적용할 값;  
    -ms-속성이름: 적용할 값;  
    -o-속성이름: 적용할 값;  
    속성이름: 적용할 값;  
}
```

수고하셨습니다.

HTML5 & CSS3 기반의 반응형 모바일 웹 만들기