

# Chapter 03

## HTML5 입력 양식 태그와 구조화 태그

웹프로그래밍

AI컴퓨터공학부  
정경용 교수

# 목차

1. 입력 양식 태그
2. HTML5 문서 구조화

# 학습목표

- 입력 양식별 특징을 이해하고 용도에 맞게 사용할 수 있습니다.
- 블록 형식과 인라인 형식을 구분할 수 있습니다.

# **Section 01**

## **입력 양식 태그**

## 01. 입력 양식 태그

- **입력 양식** : 사용자에게 정보를 입력받는 요소
  - 회원 가입 양식(서버와 통신하기 위함)

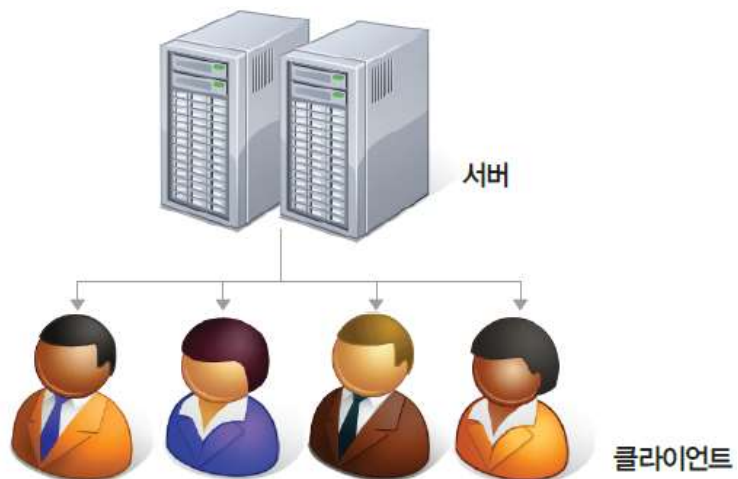
회원정보 (필수) \* 필수 항목에 사합니다.

이름\*

성별\* ☒ 남성 ☐ 여성

생년월일\* -- ☒ 양력 ☐ 음력

그림 4-1 간단한 회원 가입 입력 양식



검색 14:21 192.168.0.9

김밥

완료

김밥  
떡볶이  
순대  
어묵

# 01. 입력 양식 태그

## 1. 입력 양식 개요

- form 태그로 영역을 생성하고, form 태그 내부에 input 태그를 넣음. 데이터를 입력하고 <제출(submit)> 버튼을 누르면 지정된 방식으로 지정된 장소에 데이터를 전달

```
<body>
  <form>
    <input type="text" name="search">
    <input type="submit">
  </form>
</body>
```



그림 4-2 입력 양식 생성 코드와 실행 결과

- method 속성 : 입력 양식에 입력한 데이터를 전달하는 방식
- action 속성 : 입력 양식에 입력한 데이터를 전달하는 장소

# 01. 입력 양식 태그

## 1. 입력 양식 개요

- 전송 방식으로 GET 방식, POST방식을 가장 많이 사용

```
<!-- GET 전송 방식 -->
```

```
<body>
```

```
  <form method="get">
```

```
    <input type="text" name="search">
```

```
    <input type="submit">
```

```
  </form>
```

```
</body>
```

```
<!-- POST 전송 방식 -->
```

```
<body>
```

```
  <form method="post">
```

```
    <input type="text" name="search">
```

```
    <input type="submit">
```

```
  </form>
```

```
</body>
```



rint

제출

그림 4-4 입력 양식을 사용한 데이터 요청 예

- **GET 방식** : 주소에 데이터를 입력해서 전달, 크기가 한정됨
- **POST 방식** : 주소를 변경하지 않고 비밀스럽게 데이터를 전달, 용량 제한X

# 01. 입력 양식 태그

## 1. 입력 양식 개요

### ■ 데이터 전달 방식(GET 방식, POST방식)

**Get 방식:** URL에 데이터정보를 넣어서 전송하는 방식

- 웹브라우저 주소창에 정보가 그대로 보이므로 보안상 취약
- HTTP 요청은 브라우저에 의해 캐시되어 저장, 1024Byte로 크기 제한

**Post 방식:** 데이터를 별도로 첨부하여 전달하는 방식

- 데이터 용량에 제한이 없이 전달, 성능이나 보안적인 측면에서 우수
- HTTP 헤더에 포함되어 전송





속성	설명
action	전송 위치(데이터를 전달할 목적지)
method	전송 방식



# 01. 입력 양식 태그

## 2. 입력 양식 종류

표 4-1 입력 양식 태그

태그	속성	모양	설명
form		보이지 않음	입력 양식의 시작과 끝 표시
input	text		글자 입력 양식 생성
	button		버튼 생성
	checkbox		체크박스 생성
	file		파일 입력 양식 생성
	hidden	보이지 않음	해당 내용 표시 안 함
	image		이미지 형태 생성
	password		비밀번호 입력 양식 생성
	radio		라디오 버튼 생성
	reset		초기화 버튼 생성
	submit		제출 버튼 생성

# 01. 입력 양식 태그

## 2. 입력 양식 종류

태그	속성	모양	설명
textarea	cols/rows		여러 행의 글자 입력 양식 생성. cols는 너비를 지정하고 rows는 높이를 지정
select optgroup option			선택 양식 생성 옵션 그룹화 옵션 생성
fieldset legend			입력 양식의 그룹 지정 입력 양식 그룹의 이름 지정

# 01. 입력 양식 태그

## 2. 입력 양식 종류

기본 예제 4-1

기본 입력 양식 태그

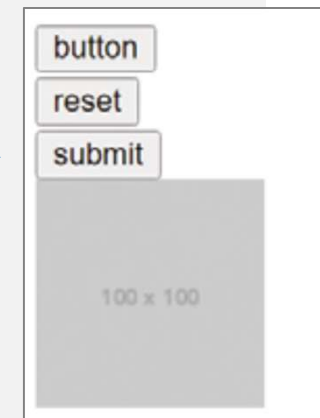
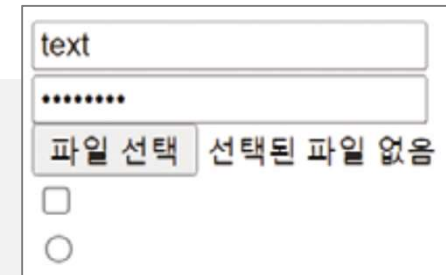
- input 태그에 type 속성으로 입력 양식 생성, value 속성으로 내부 값 지정

코드 4-1 form\_basic.html

```
<body>
  <form>
    <!-- 사용자가 입력하는 입력 양식 -->
    → <input type="text" name="text" value="text"><br>
      <input type="password" name="password" value="password"><br>
      <input type="file" name="file" value="file"><br>
      <input type="checkbox" name="checkbox" value="checkbox"><br>
      <input type="radio" name="radio" value="radio"><br>

    <!-- 보이지 않는 입력 양식 -->
    → <input type="hidden" name="hidden" value="hidden"><br>

    <!-- 버튼 -->
    → <input type="button" value="button"><br>
      <input type="reset" value="reset"><br>
      <input type="submit" value="submit"><br>
      <input type="image" src="http://via.placeholder.com/100x100">
  </form>
</body>
```



# 01. 입력 양식 태그

## 2. 입력 양식 종류

### NOTE label 태그

- label 태그의 for 속성에 input 태그의 id 속성을 입력해 input 태그를 설명
- for 속성을 연결하면, label 태그를 클릭했을 때 input 태그에 포커스가 감

코드 4-2 note\_formLabel.html

```
<form>
  <label for="name">이름</label>
  <input id="name" type="text">
  ... (생략) ...
</form>
```

이름 \*

성별 \* ☒ 남성 ☐ 여성

이름 \*

성별 \* ☐ 남성 ☒ 여성

# 01. 입력 양식 태그

## 2. 입력 양식 종류

기본 예제 4-2

간단한 입력 양식 생성

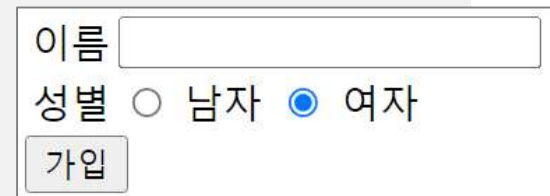
- 라디오 버튼의 name 속성으로 여러 대상 중 하나만 선택하는 형태를 구현

코드 4-3 form\_radio.html

[코드 데모](#)

```
<body>
  <form>
    <table>
      <tr>
        <td><label for="username">이름</label></td>
        <td><input id="username" type="text" name="username"></td>
      </tr>
      <tr>
        <td>성별</td>
        <td>
          <input id="man" type="radio" name="gender" value="m">
          <label for="man">남자</label>
          <input id="woman" type="radio" name="gender" value="w">
          <label for="woman">여자</label>
        </td>
      </tr>
    </table>
    <input type="submit" value="가입">
  </form>
</body>
```

radio 버튼은 name 속성을 같게  
입력해야 여러 항목 중 하나만 선택됨



# 01. 입력 양식 태그

## 2. 입력 양식 종류

[코드 데모](#)

### 기본 예제 4-3 / 선택 가능한 입력 양식

#### 1 한 항목만 선택하기

- select 태그 : 목록의 항목 중 하나 또는 여러 개를 선택할 때 사용
- option 태그 : select 태그 내부에 넣으면 옵션 선택 요소가 생성됨

#### 코드 4-4 form\_selectBasic.html

```
<body>
  <select>
    <option>김밥</option>
    <option>떡볶이</option>
    <option>순대</option>
    <option>어묵</option>
  </select>
</body>
```

# 01. 입력 양식 태그

## 2. 입력 양식 종류

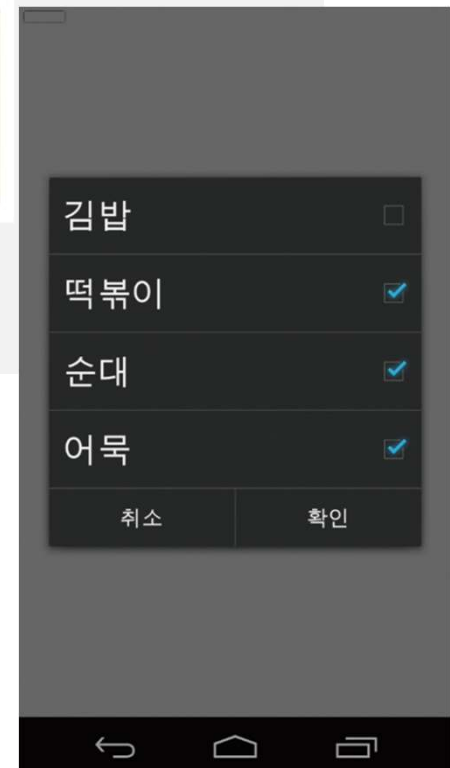
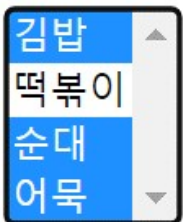
[코드 데모](#)

### 기본 예제 4-3 / 선택 가능한 입력 양식

2 여러 항목 선택하기 select 태그의 multiple 속성으로 여러 항목을 선택

#### 코드 4-5 form\_selectMulti.html

```
<body>
  <select multiple="multiple">
    <option>김밥</option>
    <option>떡볶이</option>
    <option>순대</option>
    <option>어묵</option>
  </select>
</body>
```



# 01. 입력 양식 태그

## 2. 입력 양식 종류

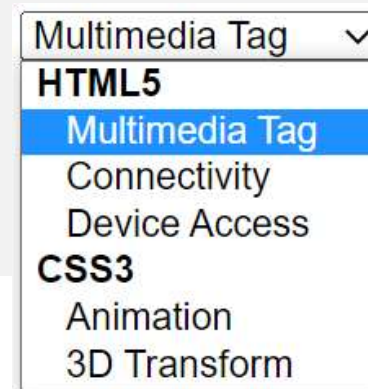
[코드 데모](#)

기본 예제 4-3 / 선택 가능한 입력 양식

3 선택 옵션 묶기 optgroup 태그를 사용해 선택 옵션을 그룹으로 묶음

코드 4-6 form\_selectGroup.html

```
<body>
  <select>
    <optgroup label="HTML5">
      <option>Multimedia Tag</option>
      <option>Connectivity</option>
      <option>Device Access</option>
    </optgroup>
    <optgroup label="CSS3">
      <option>Animation</option>
      <option>3D Transform</option>
    </optgroup>
  </select>
</body>
```





# 01. 입력 양식 태그

## 2. 입력 양식 종류

[코드 데모](#)

### 기본 예제 4-4 / 연관 있는 입력 양식 그룹으로 묶기

- fieldset 태그와 legend 태그로 입력 양식을 그룹으로 묶고 이름을 지정

#### 코드 4-7 form\_group.html

```
<body>
  <form>
    <fieldset>
      <legend>입력 양식</legend>
      <table>
        <tr>
          <td><label for="name">이름</label></td>
          <td><input id="name" type="text"></td>
        </tr>
        <tr>
          <td><label for="mail">이메일</label></td>
          <td><input id="mail" type="email"></td>
        </tr>
      </table>
      <input type="submit">
    </fieldset>
  </form>
</body>
```

입력 양식

이름

이메일

# 01. 입력 양식 태그

## 2. 입력 양식 종류


### NOTE textarea 태그 주의 사항

- textarea 태그 내 들여쓰기가 입력 양식 내부에 출력되므로 주의

코드 4-8 note\_textarea.html

```
<body>
  <textarea>
    Textarea 태그
    Textarea 태그
  </textarea>
</body>
```

#### 잘못된 형태



Textarea 태그  
Textarea 태그

- 올바른 형태

```
<body>
  <textarea>Textarea 태그
  Textarea 태그</textarea>
</body>
```

#### 올바른 형태



Textarea 태그  
Textarea 태그

## 01. 실습문제

- 1. <fieldset>, <legend>, <label>, <input> 태그를 이용하여 다음 폼을 작성하시오.

로그인 폼

---

Login

Username  Password

- 2. 도형 서식을 지정하는 다음 모양의 폼을 작성하시오.

도형 서식 폼 만들기

---

도형 서식 입력

선종류

선두께  선색

투명도(0~100) :

```
<label>선두께 <input type="number" min="1" max="50"
step="1"> </label>
<label>선색 <input type="color"
value="#00bfff"> </label> <br> <br>
<label>투명도(0~100) : <input type="range"
list="transparency" min="0" max="100">
<datalist id="transparency">
    <option value="0" label="투명">
    <option value="50" label="중간">
    <option value="100" label="불투명">
</datalist>
```

## 01. 실습문제

- 3. 9개의 버튼을 가진 다음 웹 페이지를 작성하시오

**버튼을 만들자**

1	2	3
4	5	6
7	8	0

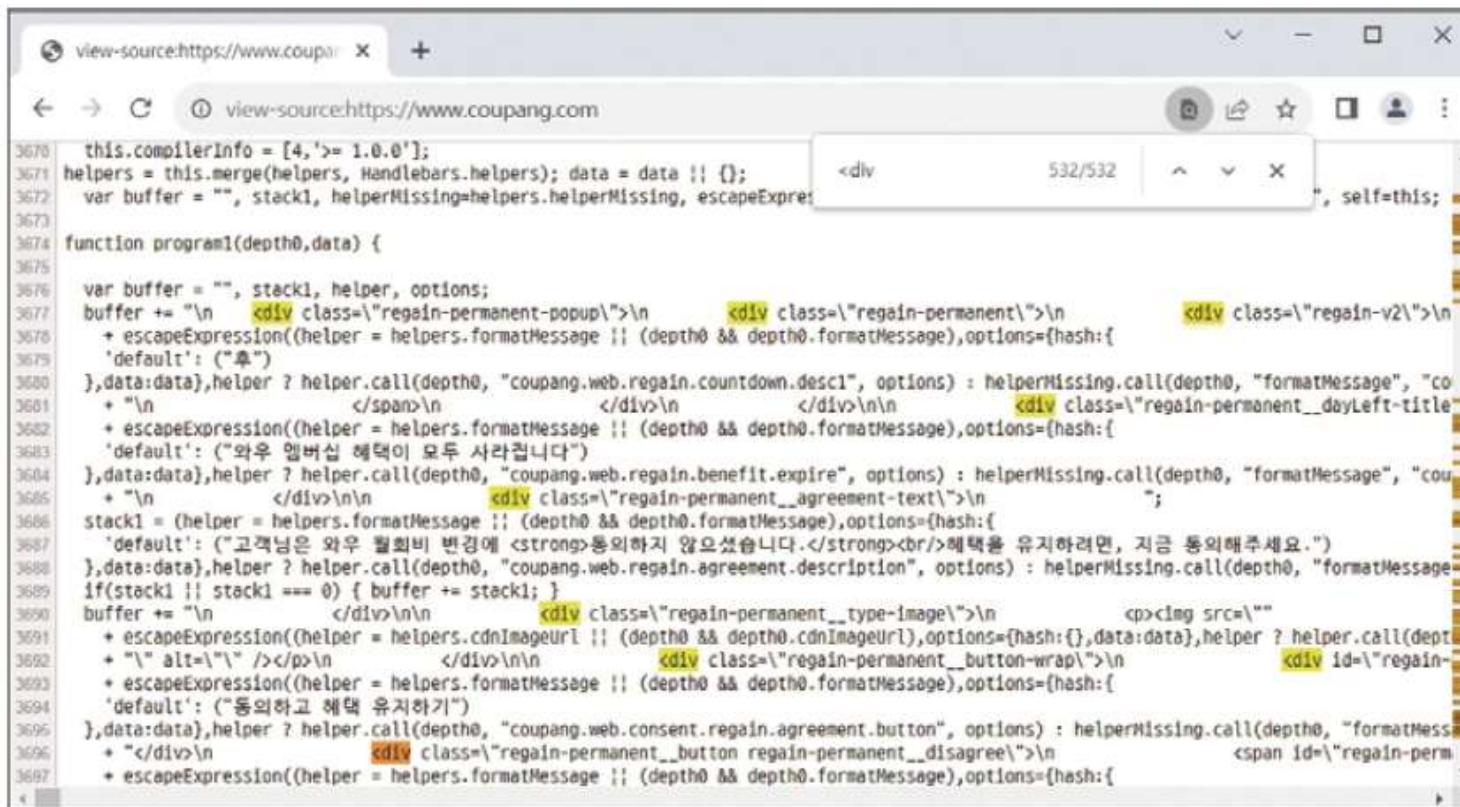
## **Section 02**

# **HTML5 문서 구조화**

## 02 HTML5 문서 구조화

### 1. 공간 분할 태그

- 공간 분할 태그를 사용해 자유롭게 웹 페이지 레이아웃을 구성
  - 쿠팡 메인 페이지는 대표적인 공간 분할 태그인 div를 500개 이상 사용함



```
3670 this.compilerInfo = [4,'>= 1.0.0'];
3671 helpers = this.merge(helpers, Handlebars.helpers); data = data || {};
3672 var buffer = "", stack1, helperMissing=helpers.helperMissing, escapeExpression=helpers.escapeExpression, self=this;
3673
3674 function program1(depth0,data) {
3675
3676   var buffer = "", stack1, helper, options;
3677   buffer += "\n    <div class=\"regain-permanent-popup\">\n      <div class=\"regain-permanent\">\n        <div class=\"regain-v2\">\n          + escapeExpression((helper = helpers.formatMessage || (depth0 && depth0.formatMessage),options={hash:{
3678            'default': ("후")
3679          },data:data},helper ? helper.call(depth0, "coupang.web.regain.countdown.desc1", options) : helperMissing.call(depth0, "formatMessage", "cou
3680            + "\n    </span>\n    </div>\n    </div>\n\n      <div class=\"regain-permanent__dayLeft-title\"
3681            + escapeExpression((helper = helpers.formatMessage || (depth0 && depth0.formatMessage),options={hash:{
3682              'default': ("와우 멤버십 혜택이 모두 사라집니다")
3683            },data:data},helper ? helper.call(depth0, "coupang.web.regain.benefit.expire", options) : helperMissing.call(depth0, "formatMessage", "cou
3684            + "\n    </div>\n\n      <div class=\"regain-permanent__agreement-text\">\n        ";
3685      stack1 = (helper = helpers.formatMessage || (depth0 && depth0.formatMessage),options={hash:{
3686        'default': ("고객님은 와우 월회비 변경에 <strong>통의하지 않으셨습니다.</strong><br>혜택을 유지하려면, 지금 동의해주세요.")
3687      },data:data},helper ? helper.call(depth0, "coupang.web.regain.agreement.description", options) : helperMissing.call(depth0, "formatMessage
3688      if(stack1 || stack1 === 0) { buffer += stack1; }
3689      buffer += "\n    </div>\n\n    <div class=\"regain-permanent__type-image\">\n      <p><img src=\"\"
3690        + escapeExpression((helper = helpers.cdnImageUrl || (depth0 && depth0.cdnImageUrl),options={hash:{},data:data},helper ? helper.call(dept
3691        + "\" alt=\"\" /></p>\n    </div>\n\n    <div class=\"regain-permanent__button-wrap\">\n      <div id=\"regain-
3692        + escapeExpression((helper = helpers.formatMessage || (depth0 && depth0.formatMessage),options={hash:{
3693          'default': ("동의하고 혜택 유지하기")
3694        },data:data},helper ? helper.call(depth0, "coupang.web.consent.regain.agreement.button", options) : helperMissing.call(depth0, "formatMess
3695        + "</div>\n    <div class=\"regain-permanent__button regain-permanent__disagree\">\n      <span id=\"regain-perm
3696        + escapeExpression((helper = helpers.formatMessage || (depth0 && depth0.formatMessage),options={hash:{
```

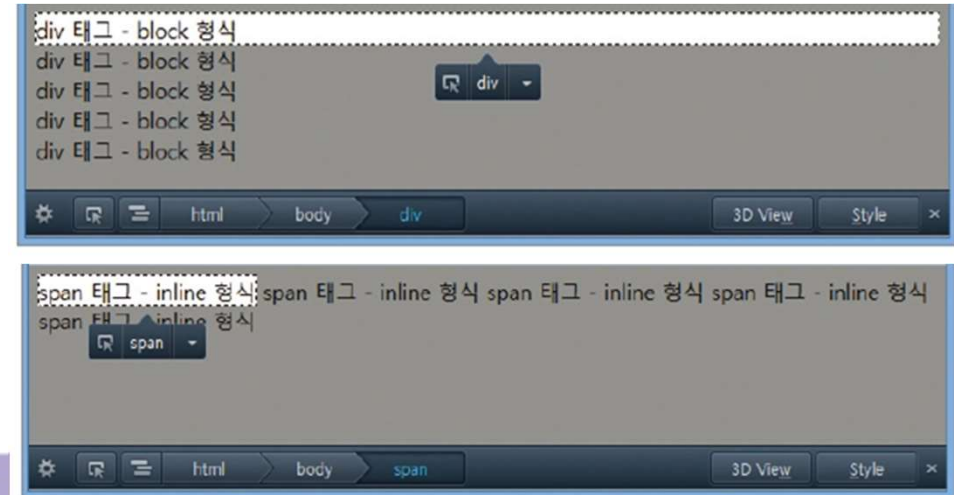
그림 4-5 쿠팡 메인 페이지(<http://www.coupang.com>)의 div 태그

## 02 HTML5 문서 구조화

### 1. 공간 분할 태그

표 4-2 기본 공간 분할 태그

태그	설명
div	블록 형식으로 공간 분할
span	인라인 형식으로 공간 분할



기본 예제 4-5 공간 분할 방법

#### 1 공간을 블록 형식으로 분할하기

- 블록(block) 형식 : 각 태그가 한 행을 모두 차지함

코드 4-9 space\_block.html

```
<body>
  <div>div 태그 - block 형식</div>
  <div>div 태그 - block 형식</div>
  <div>div 태그 - block 형식</div>
  <div>div 태그 - block 형식</div>
  <div>div 태그 - block 형식</div>
</body>
```

## 02 HTML5 문서 구조화

### 1. 공간 분할 태그

#### 기본 예제 4-5 / 공간 분할 방법

#### 2 공간을 인라인 형식으로 분할하기

- **인라인(inline) 형식** : 각 태그가 자신의 글자 크기만큼 영역을 차지하며 왼쪽에서 오른쪽으로 쌓임

#### 코드 4-10 space\_inline.html

```
<body>
  <span>span 태그 - inline 형식</span>
  <span>span 태그 - inline 형식</span>
  <span>span 태그 - inline 형식</span>
  <span>span 태그 - inline 형식</span>
  <span>span 태그 - inline 형식</span>
</body>
```

span 태그 - inline 형식 span 태그 - inline 형식 span 태그 - inline 형  
span 태그 - inline 형식 span 태그 - inline 형식  
span 165.82 x 21.14



## 02 HTML5 문서 구조화

### 1. 공간 분할 태그

#### NOTE 블록 형식 태그와 인라인 형식 태그

- h1 태그 p 태그는 블록 형식으로, a 태그 i 태그는 인라인 형식으로 출력됨

코드 4-11 note\_otherDisplayTag.html

```
<body>
  <h1>h1 태그</h1>
  <p>p 태그</p>
  <hr>
  <a href="#">a 태그</a>
  <i>i 태그</i>
</body>
```

h1 태그

p 태그

a 태그 i 태그

블록 형식 태그	인라인 형식 태그
div 태그	span 태그
h1~h6 태그	a 태그
p 태그	input 태그
목록 태그	글자 형식 태그
테이블 태그	입력 양식 태그

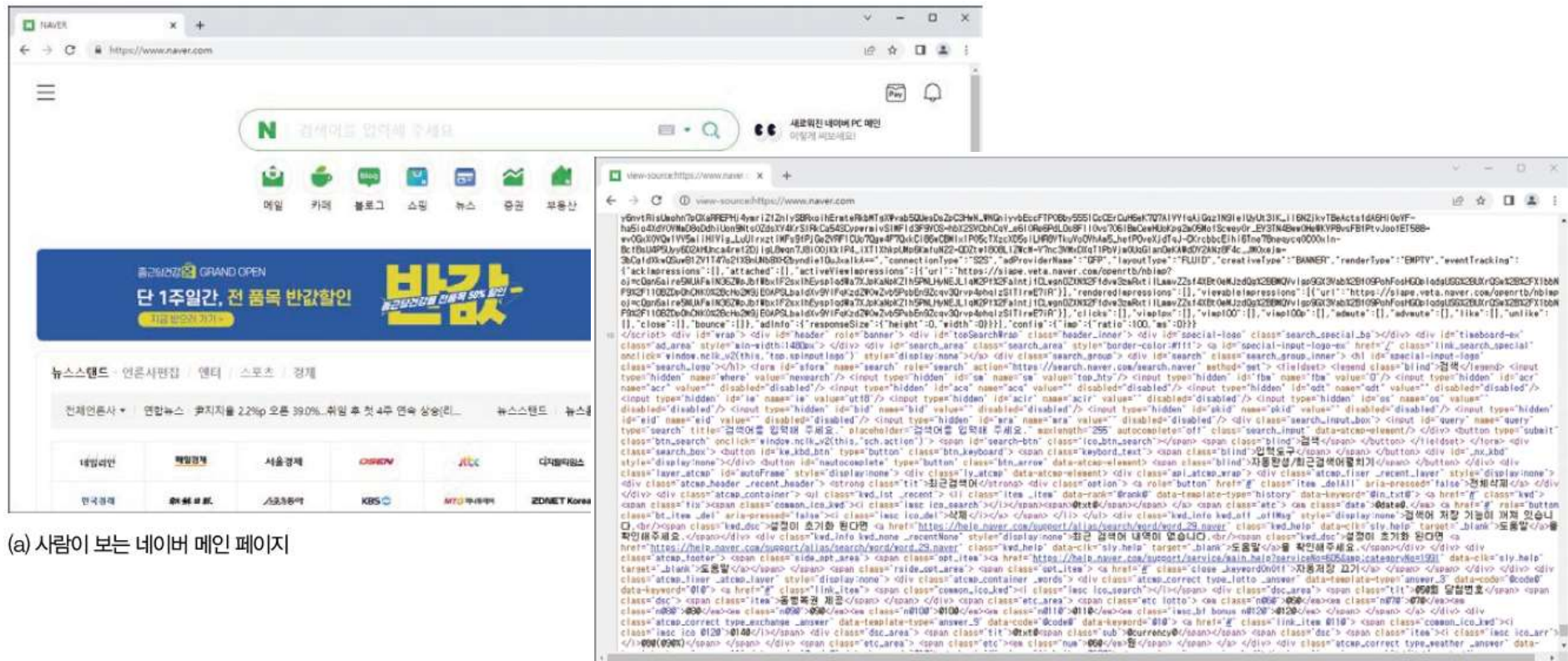
\* 이미지 태그는 인라인-블록 형식

## 02 HTML5 문서 구조화

### 2. 시맨틱 태그

■ **시맨틱(semantic) 웹** : 컴퓨터 프로그램이 코드를 읽고 의미를 인식할 수 있는 지능형 웹

■ **시맨틱 태그** : 웹 페이지의 태그에 의미를 부여하는 태그



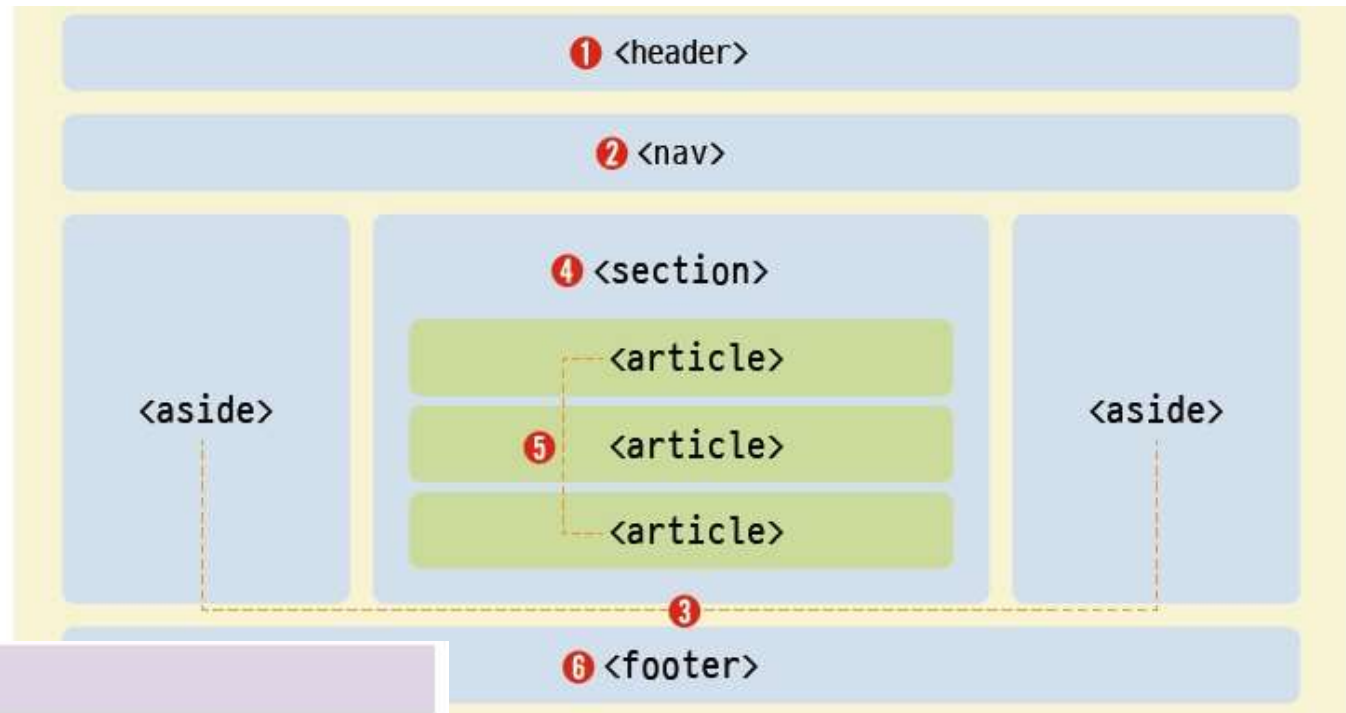
(a) 사람이 보는 네이버 메인 페이지

(b) 컴퓨터가 보는 네이버 메인 페이지

그림 4-6 시맨틱 웹의 필요성

## 02 HTML5 문서 구조화

### 2. 시맨틱 태그



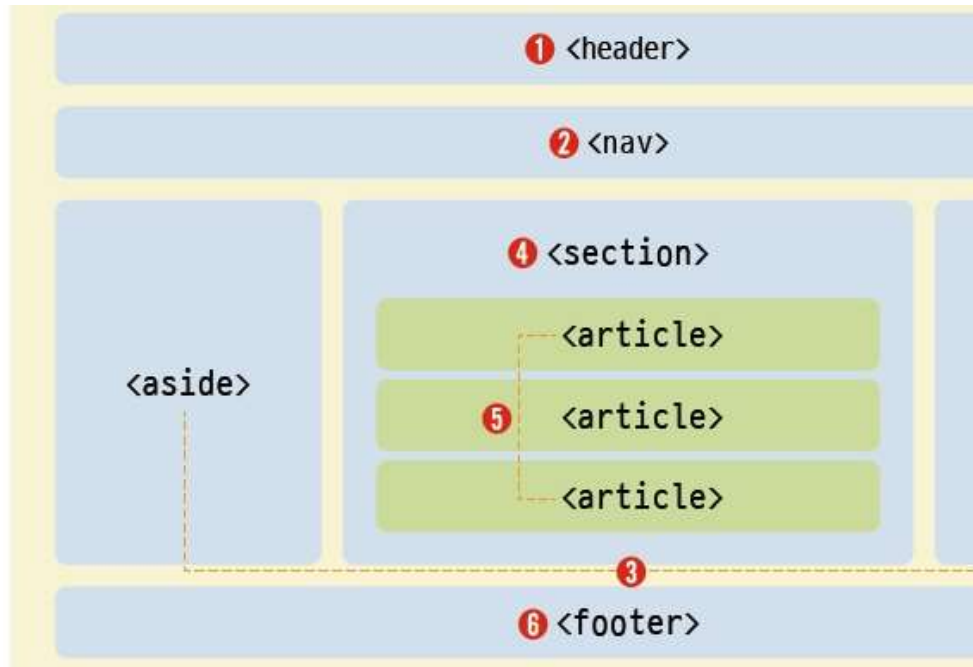
태그	설명
① header	머리말(페이지 제목, 페이지 소개)
② nav	하이퍼링크들을 모아 둔 내비게이션
③ aside	본문 흐름에 벗어나는 노트나 팁
④ section	문서의 장이나 절에 해당하는 내용
⑤ article	본문과 독립적인 콘텐츠 영역
⑥ footer	꼬리말(저자나 저작권 정보)

- 시맨틱 태그를 사용한 시맨틱 웹 구현

그림 4-7 HTML5 시맨틱 태그

## 02 HTML5 문서 구조화

### 2. 시맨틱 태그



```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>HTML5 기본 프로젝트</title>
5  </head>
6  <body>
7      <!--전체를 감싸는 태그-->
8      <div id="page-wrapper">
9          <!--헤더-->
10         <header id="main-header"></header>
11         <!--네비게이션-->
12         <nav id="main-navigation"></nav>
13         <!--본문-->
14         <div id="content">
15             <!--본문 좌측 영역-->
16             <section id="main-section"></section>
17             <!--본문 우측 영역-->
18             <aside id="main-aside"></aside>
19         </div>
20         <!--바닥글-->
21         <footer id="main-footer"></footer>
22     </div>
23 </body>
24 </html>
```

## 02 HTML5 문서 구조화

### 2. 시맨틱 태그

코드 4-12 space\_semanticLess.html

```
<body>
  <div>
    <h1>HTML5 기본</h1>
  </div>
  <div>
    <ul>
      <li><a href="#">메뉴 - 1</a></li>
      <li><a href="#">메뉴 - 2</a></li>
      <li><a href="#">메뉴 - 3</a></li>
    </ul>
  </div>
```

```
    <div>
      <div>
        <h1>Lorem ipsum dolor sit amet</h1>
        <p>Lorem ipsum dolor sit amet.</p>
      </div>
      <div>
        <h1>Lorem ipsum dolor sit amet</h1>
        <p>Lorem ipsum dolor sit amet.</p>
      </div>
    </div>
    <div>
      <span>서울특별시 강서구 내발산동</span>
    </div>
  </body>
```

### 2. 시맨틱 태그

기본 예제 4-6

시맨틱 태그를 이용한 레이아웃 구성

- 시맨틱 태그를 사용해 각 태그에 의미를 부여하여 가독성 ↑

코드 4-13 space\_semantic.html

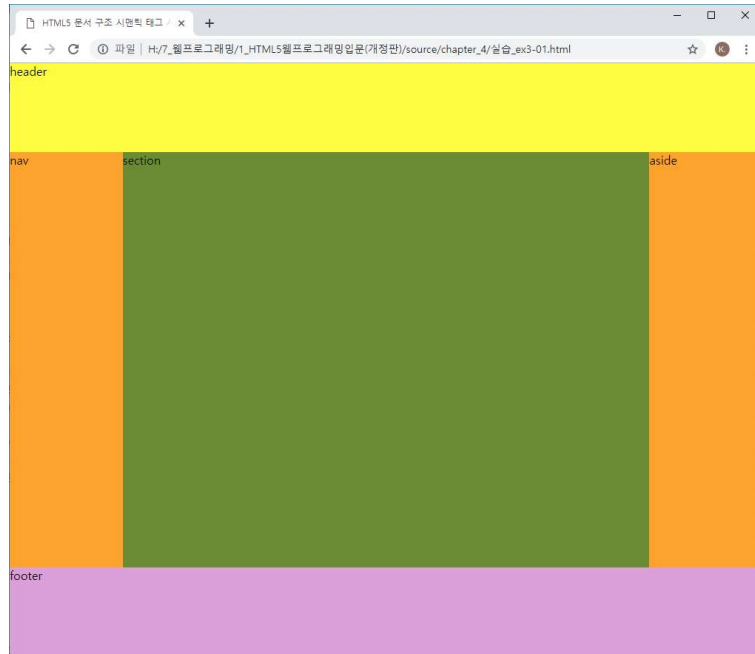
```
<body>
  <header>
    <h1>HTML5 기본</h1>
  </header>
  <nav>
    <ul>
      <li><a href="#">메뉴 - 1</a></li>
      <li>...</li>
      <li>...</li>
    </ul>
  </nav>
```

```
    <section>
      <article>
        <h1>Lorem ipsum dolor sit amet</h1>
        <p>Lorem ipsum dolor sit amet.</p>
      </article>
      <article>
        <h1>Lorem ipsum dolor sit amet</h1>
        <p>Lorem ipsum dolor sit amet.</p>
      </article>
    </section>
    <footer>
      <span>서울특별시 강서구 내발산동</span>
    </footer>
  </body>
```



## 02 실습문제

### ■ 1. 구조화된 HTML5 문서작성



```
30 <body>
31   <header class="header">header</header>
32   <nav class="nav">nav</nav>
33   <section class="section">section</section>
34   <aside class="aside">aside</aside>
35   <footer class="footer">footer</footer>
36 </body>
37 </html>
```

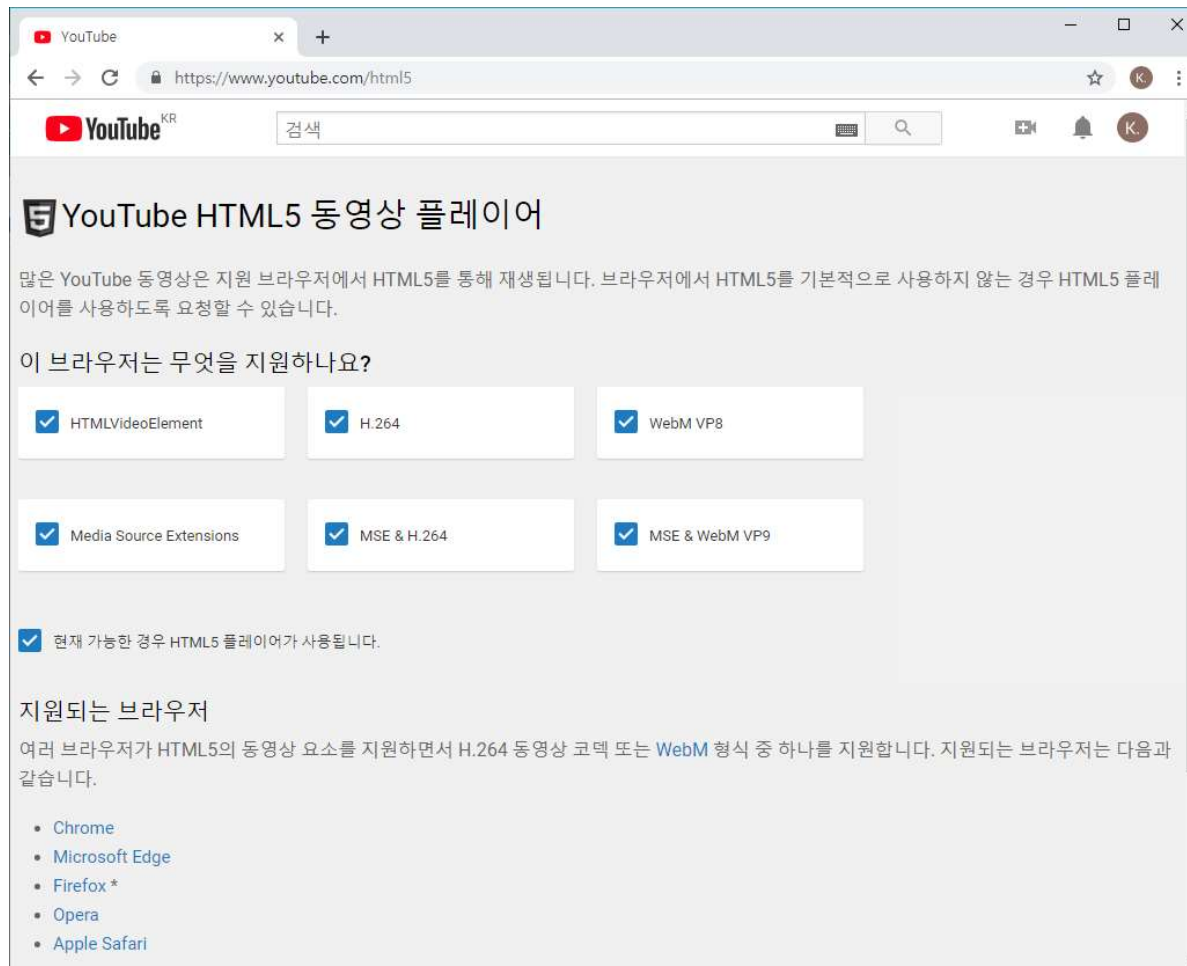
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>HTML5 문서 구조 시맨틱 태그 사용</title>
6 <style>
7 html, body { margin: 0; padding: 0; height: 100%; }
8 .header {
9   width: 100%; height: 15%;
10  background: yellow;
11 }
12 .nav {
13   width: 15%; height: 70%; float: left;
14   background: orange;
15 }
16 .section {
17   width: 70%; height: 70%; float: left;
18   background: olivedrab;
19 }
20 .aside {
21   width: 15%; height: 70%; float: left;
22   background: orange;
23 }
24 .footer {
25   width: 100%; height: 15%; clear: both;
26   background: plum;
27 }
28 </style>
29 </head>
```

## 03 비디오 태그 - 기능추가

### ▪ HTML 5 버전의 유튜브(<https://www.youtube.com/html5>)

유튜브는 플래시를 사용해서 동영상을 재생하게 설계

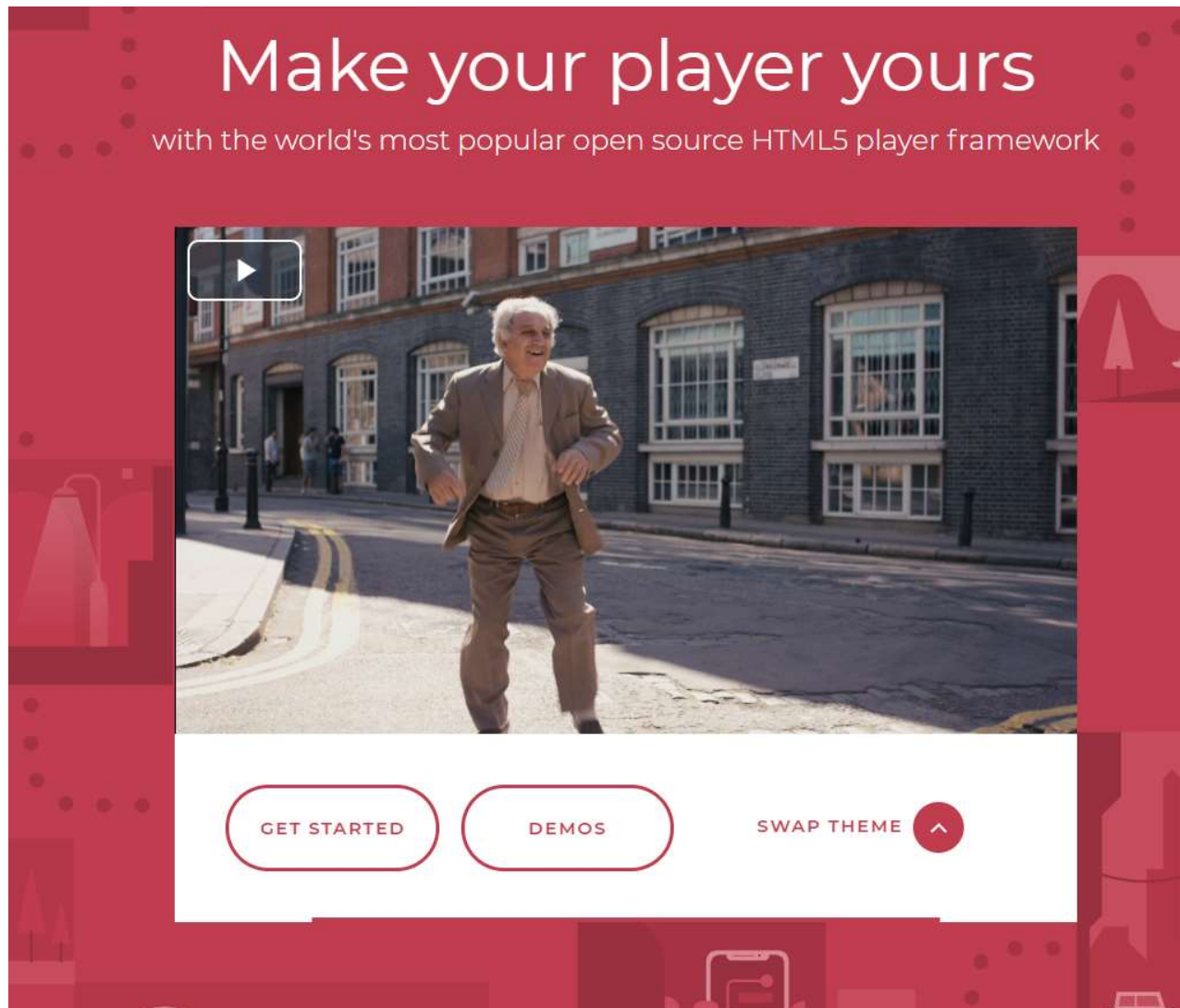
<https://www.youtube.com/html5>에서는 HTML5 기능을 실행해 플래시 없이 동영상 플레이





## 03 비디오 태그 - video.js 플러그인

- **video.js 플러그인** <https://videojs.com/>
- video 태그의 형태를 일관적으로 함



## 03 비디오 태그 - video.js 플러그인

- **video.js 플러그인** <https://videojs.com/getting-started/>  
모든 웹 브라우저에서 video 태그의 코드 실행이 가능함

**Download**

- Video.js CDN
- Install via npm
- Internet Explorer Support
- Google Analytics

**What's in the box?**

- Distributions
- Source Code

**Build Tools**

**Customize**

- Skinning
- Home Page Themes
- Designing your own

Plugins

**Download**

There are a few ways to get started using Video.js (currently v7.18.1), but you should select the one that best fits your particular use case.

**Video.js CDN**

Our friends at **Fastly** are nice enough to provide hosting for all the necessary files for Video.js on their content delivery network. Using these hosted files is probably the easiest way to get started using Video.js, you simply need to include the following links in your page.

```
<head>
  <link href="https://vjs.zencdn.net/7.18.1/video-js.css" rel="stylesheet" />

  <!-- If you'd like to support IE8 (for Video.js versions prior to v7) -->
  <!-- <script src="https://vjs.zencdn.net/ie8/1.1.2/videojs-ie8.min.js"></script> -->
</head>

<body>
  <video
    id="my-video"
    class="video-js"
    controls
    preload="auto"
    width="640"
    height="264"
    poster="MY_VIDEO_POSTER.jpg"
    data-setup="{}"
  >
    <source src="MY_VIDEO.mp4" type="video/mp4" />
    <source src="MY_VIDEO.webm" type="video/webm" />
    <p class="vjs-no-js">
      To view this video please enable JavaScript, and consider upgrading to a
      web browser that
      <a href="https://videojs.com/html5-video-support/" target="_blank"
        >supports HTML5 video</a>
    </p>
  </video>

  <script src="https://vjs.zencdn.net/7.18.1/video.min.js"></script>
</body>
```

## 03 CDN, Content Delivery Network

### ■ CDN 서비스



- ✓ 서버에 저장할 수 없는 동적 콘텐츠(HTML, JSP, PHP 등)를 미들 마일 구간에서 가속 전송

정적 콘텐츠(JPG, GIF, TXT 등)는 미리 저장해놓은 엣지 서버에서 바로 전송하고, 캐싱이 어려운 동적 콘텐츠(HTML, JSP, PHP 등)는 미들 마일(Middle mile)에서 가속합니다.



- ✓ SSL로 전송되는 CDN 콘텐츠를 암호화해 보안

CDN은 SSL 인증 암호화 프로세스를 서버에 별도로 저장해, SSL 인증서 사용으로 인한 고객 오리진 서버(원본 서버)의 과부하를 방지하고 안정적인 서비스 성능을 제공합니다.



## 03 비디오 태그 - video.js 플러그인

```
<!DOCTYPE html>
<html>
<head>
  <title>Video.js Basic</title>
  <link href="http://vjs.zencdn.net/c/video-js.css" rel="stylesheet">
  <script src="http://vjs.zencdn.net/c/video.js"></script>
</head>
<body>
  <video controls="controls" width="640" height="360"
    class="video-js vjs-default-skin" data-setup="{}">
    <source src="Wildlife.mp4" type="video/mp4" />
    <source src="Wildlife.webm" type="video/webm" />
  </video>
</body>
</html>
```

video.js 플러그인 웹 사이트에서  
얻을 수 있습니다.

