

웹 프로그래밍 강의에서 다룰 내용들 요약

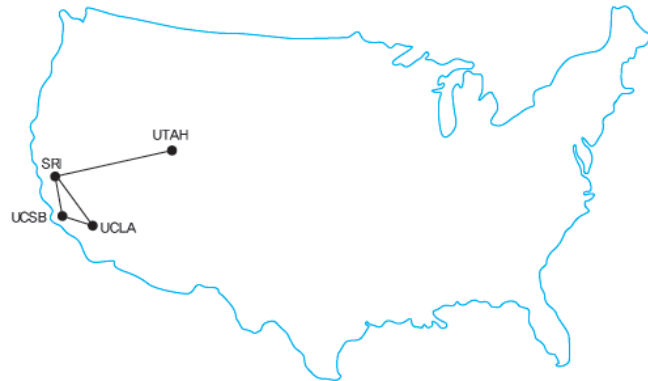
2024년 2학기

인터넷의 발전과정

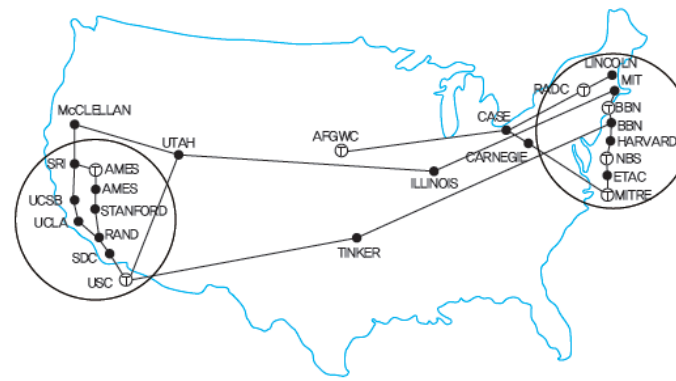
인터넷 용어의 유래 : 인터-네트워크(inter-network)

- ARPANET의 탄생

- 미국 국방성에서 1969년 미국 내 4개 대학 연결
- 데이터 전송에 IP(Internet Protocol) 전송 규약 사용



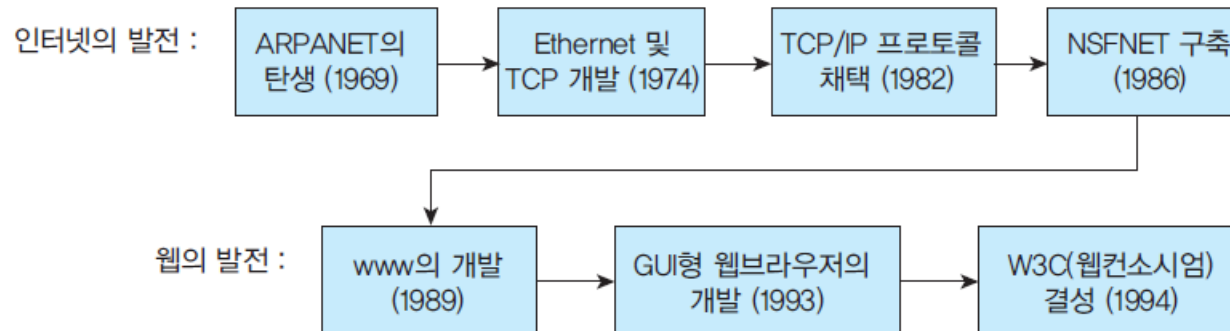
(a) 1969년 4개 사이트로 시작



(b) 1972년 50개 사이트로 확산

• 웹과 브라우저의 출현

- 1989년 유럽입자물리연구소(CERN)의 팀 버너스-리
 - 웹(WWW, World Wide Web)을 처음 제안하고
 - 1994년 웹컨소시엄(W3C, WWW Consortium) 결성
- 1993년 GUI 방식의 모자이크(Mosaic) 웹브라우저 개발,
1994년 상업용 웹브라우저인 넷스케이프 내비게이터 보급



웹의 기본 목적과 구성

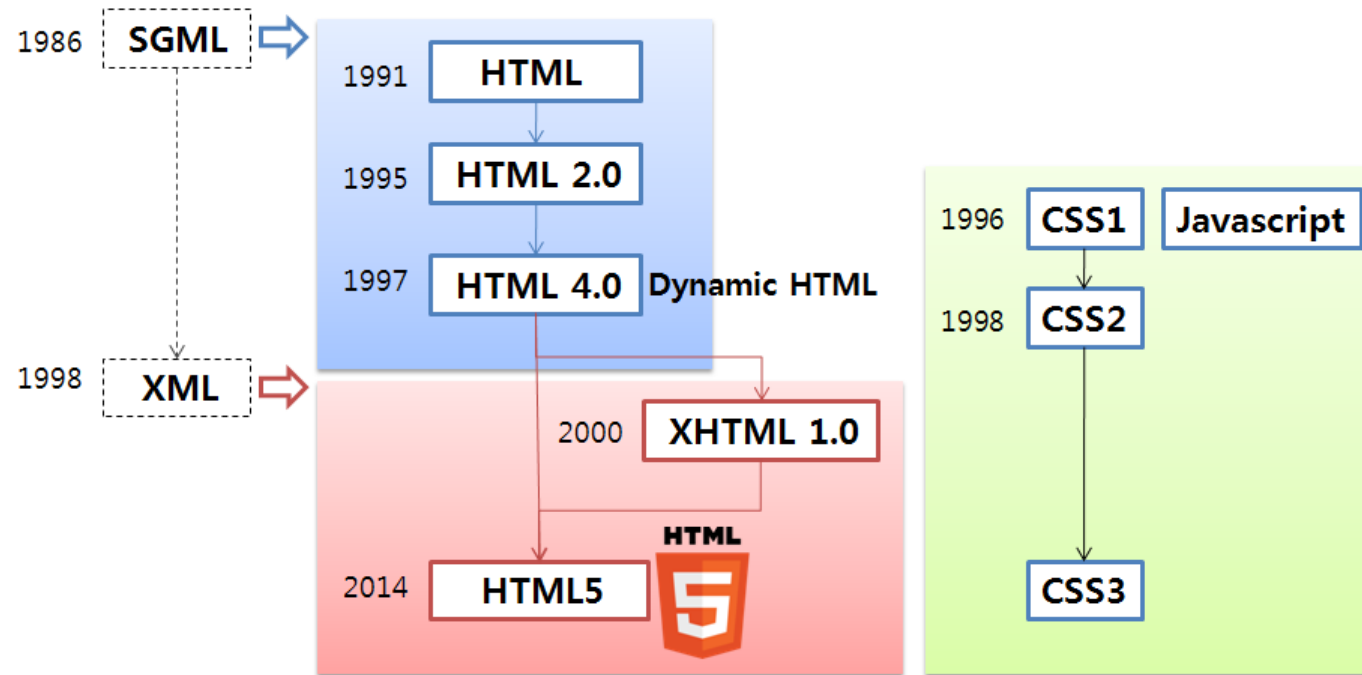
- 웹의 기본 목적
 - 다른 여러 컴퓨터에서 문서를 공유하거나 보는 목적
 - 웹에서 다루는 문서를 웹 문서라고 부름
- 웹의 구조
 - 인터넷을 활용하여 **거미줄처럼 연결된 정보 소통 망**, World Wide Web
 - 웹 문서를 인터넷 상의 컴퓨터들끼리 주고 받는 네트워크 시스템
- 웹의 구성
 - 웹 서버와 웹 클라이언트 컴퓨터들로 구성
 - 웹 서버
 - 웹 사이트를 탑재하는 컴퓨터. 구글(www.google.com), 네이버(www.naver.com) 등
 - 웹 문서, 이미지, 동영상 등의 데이터 저장 관리
 - 웹 클라이언트의 요청을 받아 웹 문서 전송
 - 웹 서버로 작동하도록 하는 소프트웨어 실행
 - 웹 클라이언트
 - 사용자 인터페이스 담당
 - 웹 서버에 웹 문서를 요청하고 받아 사용자에게 출력

WWW 컨소시엄 결성



- World Wide Web Consortium(웹컨소시엄, W3C)
 - 1994년 10월 팀 버너스리를 중심으로 결성
 - 웹 관련 표준안을 제정하고 이를 확산하는 목적
- WWW 운영
 - 미국 MIT 대학교, 유럽 INRIA 연구소, 일본 게이오 대학에 메인 호스트 서버 설치
 - 각 지역에 지역 사무국을 운영
 - 우리나라는 한국전자통신연구원에서 W3C 대한민국 사무국 운영
 - 전 세계의 300여 개 단체가 회원으로 소속
 - 웹 관련된 기술의 발전에 대해 논의하고 기술 표준안을 공동 개발
 - W3C 표준은 국제공인표준과 동등하게 중요한 기준으로 인정
 - 웹 개발자를 위한 정보 공유, 다양한 웹 소프트웨어의 개발 및 교육
 - 매년 봄에 정기적인 WWW 컨퍼런스 개최, 다양한 포럼 운영

• HTML의 발전 역사



CSS3의 완전 지원

- 스타일시트(Stylesheet)
 - 웹 문서의 외형 스타일을 지정하는 언어
 - HTML 문서에는 일반적으로 CSS(Cascading Style Sheet)가 사용
 - 1996 CSS1, 1998 CSS2, 2005년 이후 CSS3는 모듈별로 개발 중
- HTML5에서는 CSS1, CSS2, CSS3까지 완전히 지원
 - 기존 CSS는 주로 텍스트, 배경 및 색상, 목록, 박스모델 등 속성
 - CSS3에는 더욱 다양한 스타일 지정 기능을 포함
 - 예제 화면
 - 박스의 모서리 처리, 그림자 효과, 다단(multi-column) 지정, 텍스트의 그림자, 콘텐츠 위치 지정, 등 다양한 효과
 - 이 외에도 장면 변환(transition), 애니메이션, 사용자 인터페이스에 관련된 속성



다양한 API 지원

- 웹 애플리케이션 개발에 많은 도움을 줄 수 있는 다양한 API 제공
 - 자바스크립트로 구현 가능
 - <video>와 <audio> 요소를 제어하는 API
 - <canvas> 요소에 그림을 그리는 API 등
- 별도의 사양으로 분리된 API
 - 드래그앤 드롭(Drag & Drop), 위치정보(Geo-Location), 웹 스토리지(Web Storage), 웹 소켓(Web Sockets), 파일(File), 애플리케이션 캐시 (Application Cache), 인덱스드 데이터베이스(Indexed Database) 등
 - 이들 모두가 HTML5의 기능
 - 자바스크립트로 매우 다양하고 강력한 기능의 웹애플리케이션 구현 가능해짐 (자바스크립트는 8~10장에서 설명)

모바일 웹 환경 고려

- 모바일 환경
 - 일찍부터 HTML5를 지원하는 웹 브라우저를 탑재
- 모바일 환경을 위한 HTML5의 특별한 기능
 - 위치정보 API
 - 위치정보는 모바일 애플리케이션 개발에 최적 활용
 - 오프라인 어플리케이션 API
 - 모바일 환경에서 접속이 끊기거나 트래픽 최적화에 긴요하게 활용
 - 오프라인이 될 경우에 로컬 스토리지, 웹 데이터베이스, 어플리케이션 캐시 등이 유용하게 활용
 - 새로운 유형의 다양한 입력 폼 지원
 - 모바일 애플리케이션의 사용자 인터페이스 개발 및 사용이 편리

인터넷의 기본 개념

클라이언트-서버 모델

인터넷의 전송방식

주소체계

클라이언트-서버 모델

- 클라이언트-서버 모델(Client-Server Model)을 기반

- 서버 vs 클라이언트

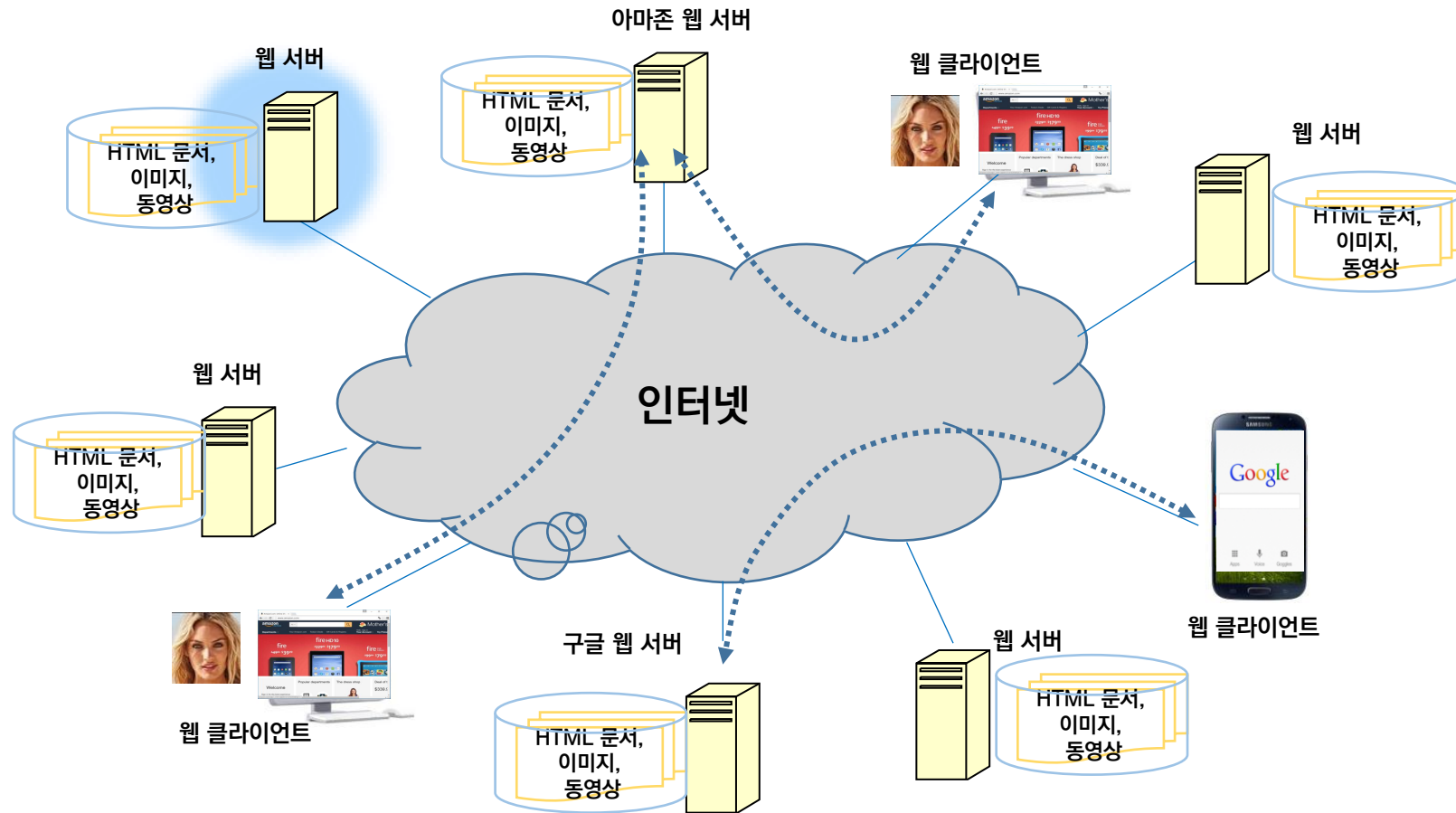
- 서버 : 제공하는 서비스에 적합한 정보들을 보관하고 이를 외부에 공개해주는 컴퓨터
 - 클라이언트 : 서버에서 제공하는 정보를 받기 위해 사용자가 작동시키는 컴퓨터

- 클라이언트-서버에 필요한 프로그램

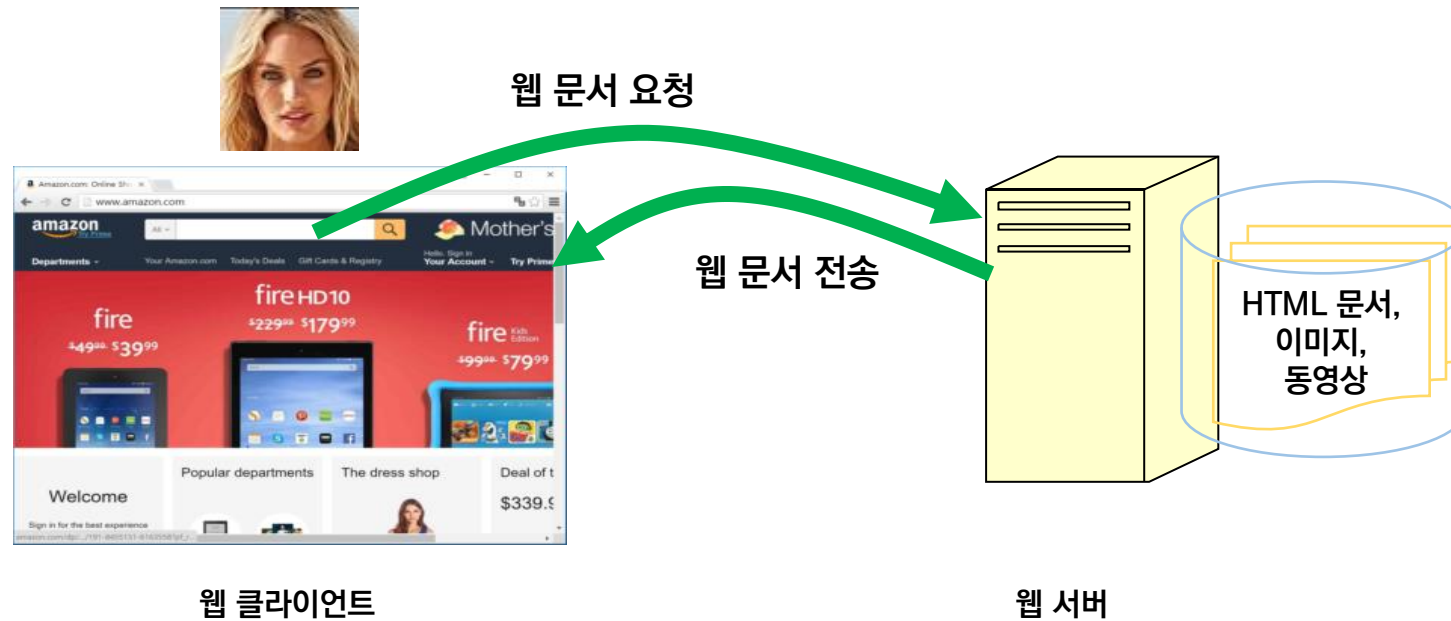
- 서버 역할을 수행하는 서버 프로그램이 필요
 - 웹을 이용하려면 아파치(Apache) 또는 IIS라는 웹 서버 프로그램
 - 서비스를 이용하기 위한 클라이언트 프로그램
 - 크롬(chrome)이나 익스플로러(Explorer)와 같은 웹브라우저



웹 서버와 웹 클라이언트로 이루어진 웹



웹 서버와 웹 클라이언트의 작동



인터넷의 전송방식

- TCP/IP 통신 프로토콜

- 인터넷은 운영체제나 데이터 형식이 다양한 종류의 컴퓨터 연결
 - 모든 컴퓨터에 IP 주소를 할당하고, 데이터를 주고받기 위하여 필요
- 통신 프로토콜 : 컴퓨터간의 통신 규약
 - 컴퓨터 사이에 정보를 전달하기 위하여 필요한 규칙 및 약속의 집합
- TCP(Transmission Control Protocol)
 - 데이터의 흐름을 제어하고 데이터가 정확한지 확인하는 역할
 - 데이터를 여러 개의 작은 조각으로 나누어 패킷(Packet)이란 정보단위를 생성하고 패킷이 제대로 전송되는지 확인
- IP(Internet Protocol)
 - 데이터를 이동시킬 목적지를 지정하는 역할

- HTTP 서비스 프로토콜
 - TCP/IP 전송 프로토콜
 - 인터넷에 연결된 시스템 사이에 데이터를 송수신하는 통신규약
 - 인터넷 서비스 프로토콜
 - 다양한 인터넷 애플리케이션 서비스를 위한 프로토콜 필요
 - 서버와 클라이언트 컴퓨터 사이에 서비스를 위한 데이터 통신 규약
 - HTTP, FTP, Telnet, Mailto 등
 - HTTP(HyperText Transfer Protocol)
 - 웹에서 가장 기본적으로 사용하는 서비스 프로토콜
 - 웹 서버와 웹 브라우저 사이에 하이퍼텍스트 문서를 위한 통신규약

주소체계

- 도메인 이름(Domain Name)

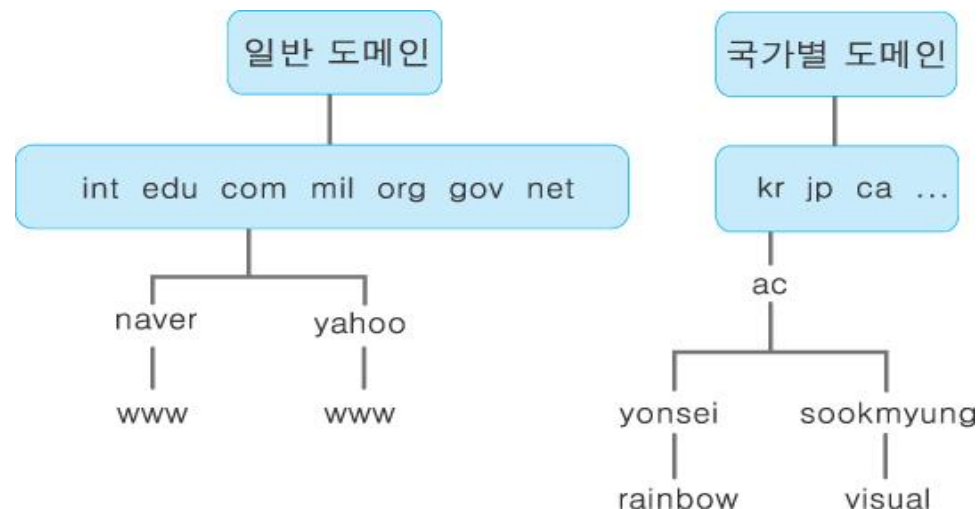
- IP 주소는 숫자로 구성되어 있어서 이용/기억하기 불편
- 도메인 이름 : 쉽게 기억할 수 있도록 IP주소를 문자로 대체
 - 각 호스트 컴퓨터는 하나씩의 IP 주소와 도메인 이름 보유
 - 도메인 이름 서버(Domain Name Server) : 도메인 이름을 IP주소로 변환

- 국가별 도메인 방식

- “호스트이름.소속기관.단체성격.소속국가”의 4단계
- 예, www.daum.co.kr

- 일반 도메인 방식:

- “호스트이름.소속기관.단체성격”의 3단계
- 최상위 도메인이 소속국가가 아닌 단체성격
- 예, www.yahoo.com



- URL(Uniform Resource Locator)
 - 웹에서 각종 파일 등 자원의 위치를 표시하는 표준
 - 서버에서 지원되는 다양한 서비스를 표현하도록
 - HTTP, FTP, Telnet, mail-to 등
 - 구성형식: “서비스프로토콜://도메인주소/폴더경로/파일이름”
 - 예, <http://cafe.naver.com/swimmingtraining/10294>
 - 예, <ftp://it.sookmyung.ac.kr/~sblim/intro.html>

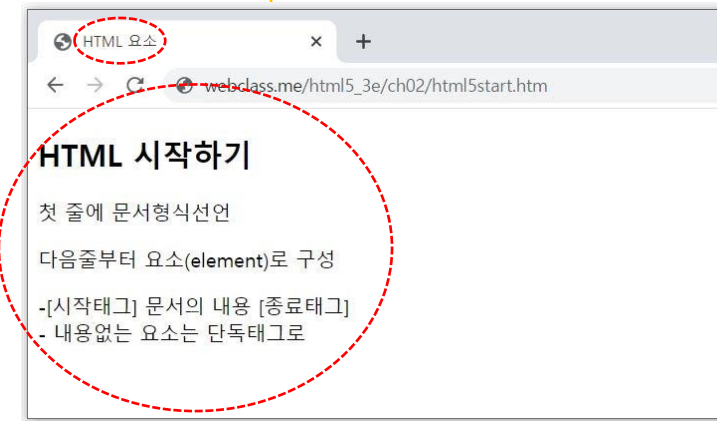
HTML, 웹 환경에서 사용되는 문서

HTML 요소와 속성

- HTML (HyperText Markup Language)
 - 웹(WWW)을 위한 하이퍼텍스트 문서 작성 언어
 - 웹브라우저에서 보여지는 문서의 내부형식을 규정
- HTML 형식의 파일
 - 첫줄에 DOCTYPE 선언, 다음줄부터 요소들(elements)
 - 파일 확장자는 *.htm 혹은 *.html

예제: HTML 시작하기

```
1      <!DOCTYPE html>
2      <html>
3      <head>
4          <title>HTML 요소</title>
5      </head>
6      <body>
7          <h2>HTML 시작하기</h2>
8          <p>첫줄에 문서형식선언 !DOCTYPE html</p>
9          <p>다음줄부터 요소(element)로 구성</p>
10         <p>- [시작태그] 문서의 내용 [종료태그];
11         <br>- 내용 없는 요소는 단독태그로</p>
12     </body>
13 </html>
```



- 문서 형식선언 <!DOCTYPE html>

요소(Element)와 태그(Tag)

- 요소(element)

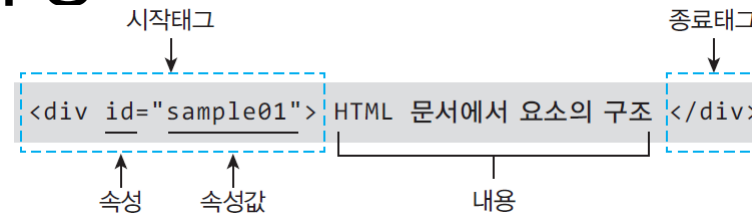
- 문서 일부분의 특징을 지정하는 마크업, 일종의 명령어
- 내용(content)과 이를 둘러싼 태그(tag)로 구성
 - 시작태그 <태그이름 속성들 ...>
 - 종료태그 </태그이름>

- 태그의 이름

- 공백없는 문자열
- 대소문자 구분 없음
- 엇갈려 중첩되면 안된다

- 단독태그

- <태그이름 />



<p> <h2> html 문단 </p> 제목 </h2>
<p> <h2> html 문단 제목 </h2> </p>
<h2> <p> html 문단 제목 </p> </h2>

 <hr />

속성

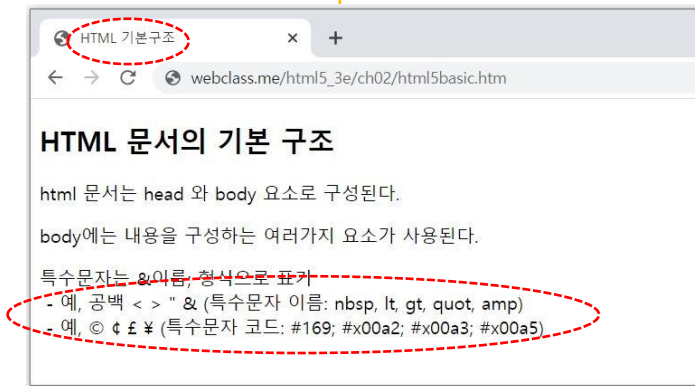
- 요소에 추가정보를 주기 위해서 사용
 - 추가정보 예, 요소의 모양을 나타내는 스타일
 - 하이퍼링크의 링크지점 정보 등
- 속성의 구조
 - 요소의 시작태그 내에 나타남
 - 이름 = “속성값” 혹은 이름 = ‘속성값’
 - 하나의 요소에 여러 개의 속성 가능, 빈 칸으로 구분

```
<table border="1"> ... </table>  
<a href="ch02.htm"> ... </a>  
<a href="ch02.htm" target="blank"> ... </a>
```

HTML 문서의 기본 구조

- 문서형식 선언 다음에 루트요소 <html>
 - <head> 와 <body> 로 구성, <body>에 원하는 요소

```
1      <!DOCTYPE html>
2      <html>
3      <head>
4          <title>HTML 기본구조</title>
5          <meta charset="UTF-8">
6          <meta authors="홍길동">
7      </head>
8      <body>
9          ...
12     <p>특수문자는 &amp;이름; 형식으로 표기<br>
13     &nbsp;- 예, 공백 &lt; &gt; &quot; &amp; (특수문자 이름: nbsp, lt, ..., <br>
14     &nbsp;- 예, &#169; &#x00a2; &#x00a3; &#x00a5; (특수문자 코드: ...
15         <!-- 주석문은 이렇게 작성 -->
16     </body>
17 </html>
```

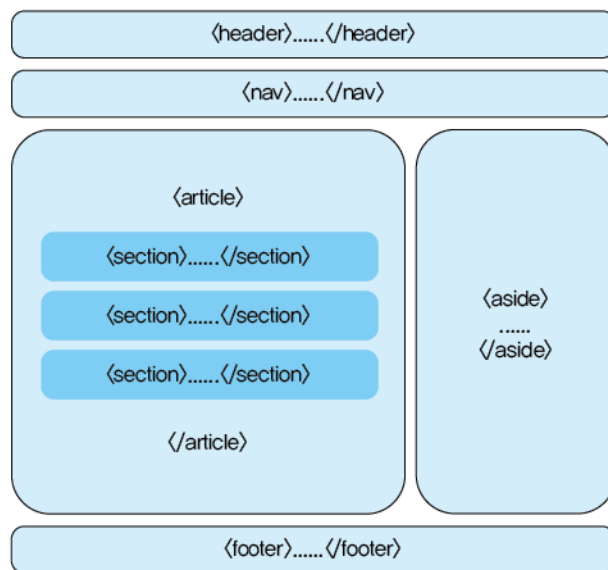


문서 구조화하기

문서 구조화 요소

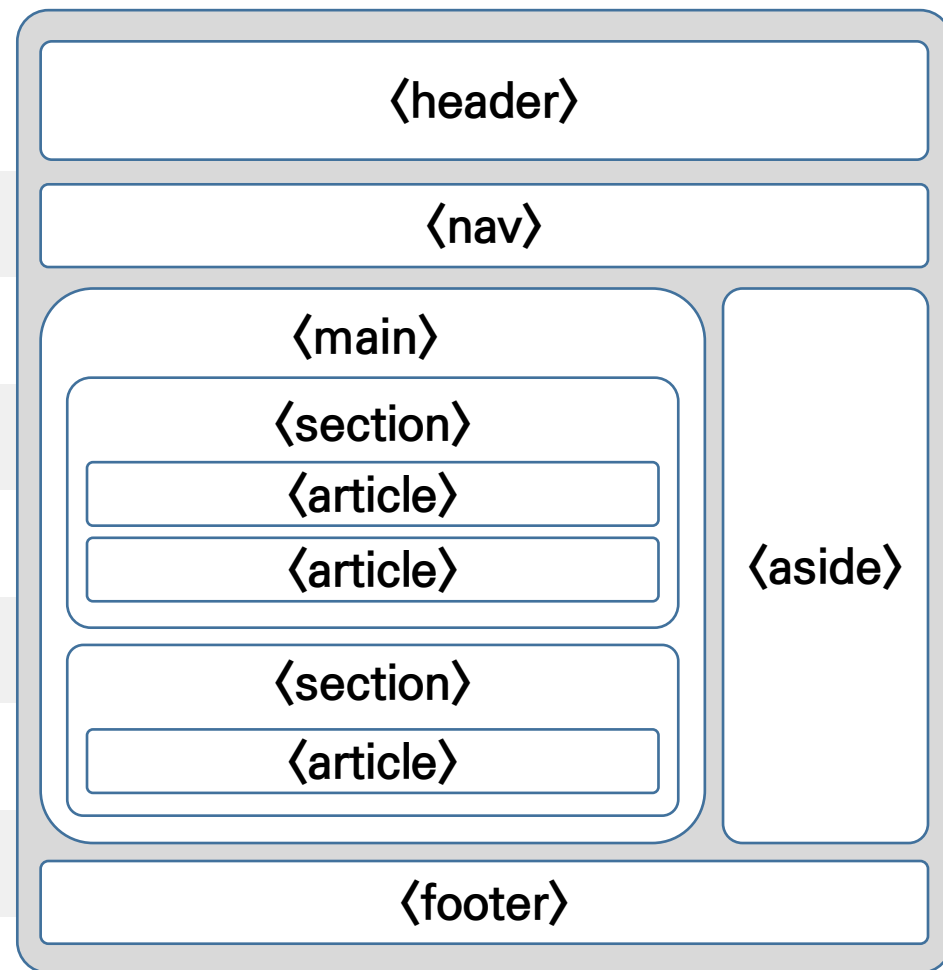
- 구조화 요소

- 머리말, 꼬리말, 탐색 메뉴, 본문, 본문의 섹션, 옆줄 등 문서의 의미적인 구조
 - `<header>` `<footer>` `<nav>` `<article>` `<section>` `<aside>`
 - 의미적인 구분만 할 뿐 모양은 구분하지 않는다.



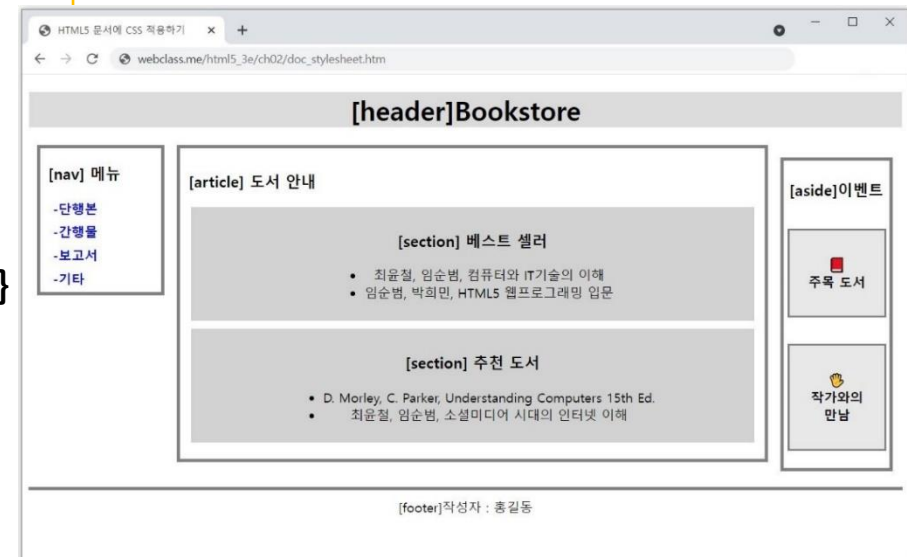
시맨틱 태그의 종류

<header>	사이트의 헤더 정보를 담는 데에 사용
<main>	주요 콘텐츠를 담는 데에 사용
<section>	제목으로 구분할 수 있는 콘텐츠 단위
<article>	각 섹션 내에 담기는 주요 콘텐츠
<aside>	웹 사이트의 측면에 배치될 정보
<footer>	사이트의 하단에 표시도리 내용
<hgroup>	헤더 그룹 - 제목과 부제를 묶어서 담음
<nav>	사이트 내외의 문서들을 찾아갈 수 있도록 하는 내용



예제: 문서에 CSS 적용하기

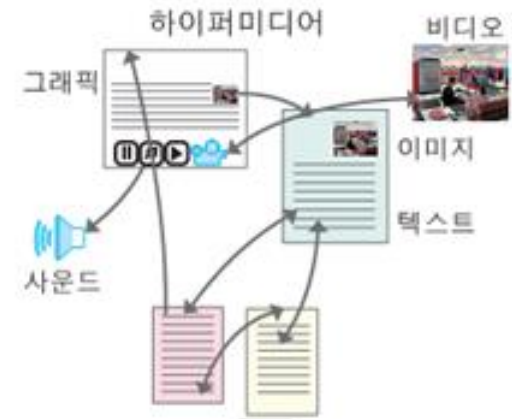
```
<head>
  <title> HTML5 문서에 CSS 적용하기 </title>
  <style>
    header {background-color: #828282; margin: 1px; text-align: center;}
    nav {border: 4px solid gray; width: 12%; float: left; margin-left: 10px; padding: 0px 10px;}
    nav a {display: block; margin-left: 5px; margin-bottom: 5px; padding: 1px;
      text-decoration: none; font-weight: bold;}
    article {border: 4px solid gray; width: 65%; margin: 3px; margin-left: 17%; padding: 10px;}
    section {background-color: #d2d2d2; margin: 3px; margin-bottom: 10px; padding: 10px;
      height: 20vh; text-align: center;}
    section ul {display: table; margin: 0 auto;} /* ul 가운데 정렬 */
    aside {position: absolute; border: 4px solid gray; width: 10%; top: 17vh; right: 20px;
      padding: 5px; text-align: center; min-width: 120px;}
    aside p {display: inline-block; border: 3px solid gray; background-color: #d2d2d280;
      width: 50%; padding: 30px 20px 30px 20px; font-weight: bold; min-width: 75px;}
    footer {border-top: 4px solid gray; margin-top: 30px; padding: 10px; text-align: center;}
  </style>
</head>
<body>
  <header>[header] SM_Bookstore
  ...
```



링크 달기

하이퍼텍스트/하이퍼미디어

- HTML(HyperText Markup Language)
 - 하이퍼텍스트의 마크업 언어
- HTML의 기본 개념
 - 하이퍼텍스트 : 서로 연관된 문서나 텍스트 조각들을 연결
 - 하이퍼미디어 : 텍스트 뿐 아니라 멀티미디어 정보가 서로 연결
- 하이퍼텍스트/하이퍼미디어의 구조
 - 각 정보의 조각은 **링크에 의해 서로 연결**
 - 모든 정보의 접근은 연결 링크를 선택하여 내비게이션



문서간 이동 <a>

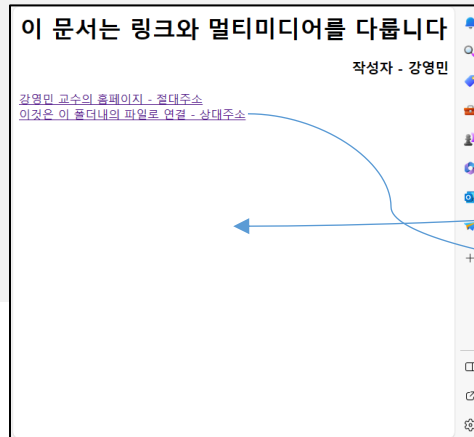
- <a> 요소 : 링크의 시작점 앵커를 표현
 - href 속성 : 이동하고자 하는 목적지 문서의 파일 주소(URL)를 지정
 - title 속성에는 말풍선 창에 나올 설명을 기입
 - target 속성에는 링크 결과가 나타날 장소

링크 텍스트

0301_link.html

```
<html>
  <head>
    <title>Link와 멀티미디어</title>
  </head>
  <body>
    <h1 align="center">이 문서는 링크와 멀티미디어를 다룹니다</h1>
    <h3 align="right">작성자 - 강영민</h3>

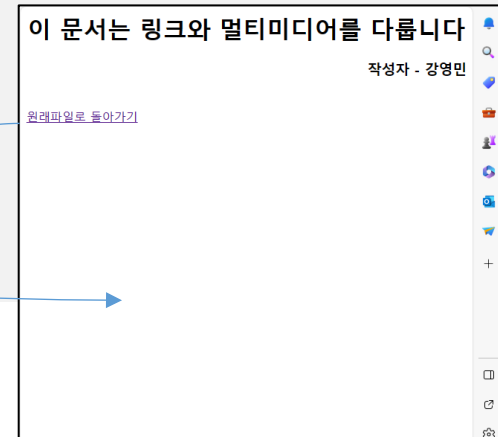
    <a href="https://github.com/dknife/dknife.github.io/wiki" title="Link Test">
      강영민 교수의 홈페이지 - 절대주소
    </a>
    <br>
    <a href="./0301_local.html" title="Link Test">
      이것은 이 폴더내의 파일로 연결 - 상대주소
    </a>
  </body>
</html>
```



0301_local.html

```
<html>
  <head>
    <title>Link와 멀티미디어</title>
  </head>
  <body>
    <h1 align="center">이 문서는 링크와 멀티미디어를 다룹니다</h1>
    <h3 align="right">작성자 - 강영민</h3>

    <br>
    <a href="./0301_link.html" title="Link Test">
      원래파일로 돌아가기
    </a>
  </body>
</html>
```



링크 예제 (2) : 문서 내 위치로 이동

```
...
<ol>
  <li>최윤철, 임순범, 한탁돈 공저, <a href="#book001">컴퓨터와 IT기술의 이해</a></li>
  <li>D. Morley, C. Parker, <a href="#book002">Understanding Computers 15th Ed.</a></li>
  <li>G. Shelly, M. Vermaat, Discovering Computers</li>
</ol></br>
...
<h3>참고도서 소개</h2>
<ol>
  <a id="book001"></a><li>컴퓨터와 I
<table border="1"> <tr><td>
  <ul> <li>저자 : 최윤철, 임순범, 한
    <li>출판사 : 생능출판사</li>
    <li>책소개 : 급변하는 IT기술의
  </ul>
</td></tr> </table>

  <a id="book002"></a><li>Understanding Computers, 15th Ed</li>
<table border="1"> <tr><td>
  <ul> <li>저자 : D. Morley, C. Parker</li>
    <li>출판사 : Cengage Learning</li>
    <li>책소개 : Thoroughly Updated for the Latest Advances in Multimedia ... 생략 ...
  </ul>
</td></tr> </table>
</li>
```

The diagram illustrates the mapping of HTML code to a web page layout. It shows two browser screenshots of a page titled '문서 내 특정 위치로 이동' (Move to specific location within document). The left screenshot shows a table of contents with a link to '컴퓨터와 IT기술의 이해' (Understanding Computers) highlighted. The right screenshot shows the detailed view of this book, with a link to 'Understanding Computers, 15th Ed.' highlighted. Red circles and arrows connect the HTML code on the left to these elements in the browser screenshots, demonstrating how the code controls the document's structure and navigation.

- 링크의 목적지 앵커 지정:

- id 속성 : 문서 내의 원하는 위치에 목적지 앵커를 설정
 - 이름을 지정하는 것이므로 <a> 와 사이에 텍스트 필요 없음

예) 컴퓨터와 IT기술의 이해

- 시작점 앵커에서 링크 연결:

- href 속성에 목적지 앵커의 아이디를 지정
 - 목적지 앵커에서 아이디 설정할 때는 이름만 기입
 - 아이디를 이용할 때는 #으로 시작

예) 최윤철, 임순범 공저, 컴퓨터와 IT기술의 이해

이미지 사용하기

이미지 삽입

-
- 요소의 src, width, height 속성
 - src 속성 : 이미지 파일의 이름을 지정
 - 파일의 경로 + 파일 이름
 - width와 height 속성 : 크기를 조정하고 싶을 때 사용
 - 예
 - IT기술의 이해
 - IT기술의 이해
 - IT기술의 이해

〈figure〉, 〈figcaption〉

- 〈figure〉 요소
 - 그림, 사진, 다이어그램과 텍스트 등의 콘텐츠를 함께 묶어서 하나의 독립된 단위로 취급하고 싶을 때 사용
- 〈figcaption〉 요소
 - 〈figure〉 요소를 위한 제목을 표현, 〈figure〉 요소 내에 위치

```
〈figure〉  
  〈table〉 〈tr〉  
    〈td〉〈img src="IT_intro.jpg" alt="책표지" width="75"〉〈/td〉  
    〈td〉IT기술의 이해〈br〉최윤철, 임순범〈br〉생능출판사〈/td〉  
  〈/tr〉 〈/table〉  
  〈figcaption〉[그림 1] 책 소개〈/figcaption〉  
〈/figure〉
```

참고도서 목록



IT기술의 이해
최윤철, 임순범
생능출판사

[그림 1] 책 소개

- 이미지에 하이퍼링크 연결
〈a href = "링크할 곳의 파일이름"〉〈img src="이미지 파일이름"〉〈/a〉

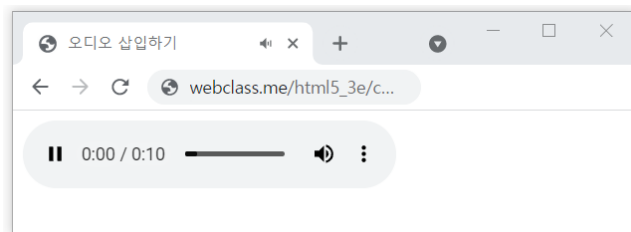
오디오와 비디오 다루기

오디오 삽입하기 <audio>

<audio controls autoplay src="재생할 사운드 파일 이름">

- src 속성 : 사운드 파일 이름
- controls 속성 : 기본적인 미디어 제어기를 표시할 지 여부
- autoplay 속성 : 파일이 로드되자마자 자동으로 재생시킨다는 의미
- loop 속성 : 사운드의 반복 재생

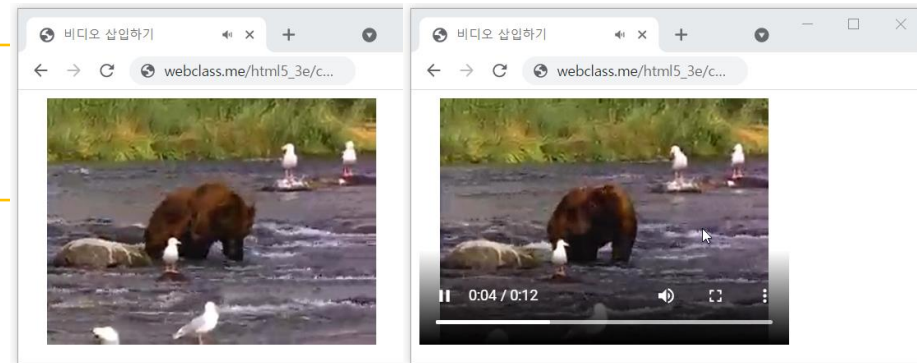
```
<audio controls src="song.mp3" loop>  
브라우저에서 <audio>를 지원하지 않습니다. </br>  
(song.mp3파일이 반복 자동재생 됩니다.)  
</audio>
```



비디오 삽입하기 <video>

- <video controls src="비디오 파일 이름" width="폭" height="높이" >
- <video> 요소의 속성
 - src, controls, loop, autoplay 속성: <audio> 요소의 속성과 동일
 - width, height 속성: 화면에서 비디오가 표시될 영역의 크기
 - videoWidth, videoHeight 속성: 비디오 자체의 너비와 높이
 - poster 속성: 동영상에 로딩되고 있을 때 보여줄 이미지
 - preload 속성: 브라우저가 미리 동영상을 로딩 할 지 여부

```
<video controls autoplay width="360" height="240" src="bear.mp4">  
    비디오를 재생할 수 없습니다.  
</video>
```



객체 포함하기

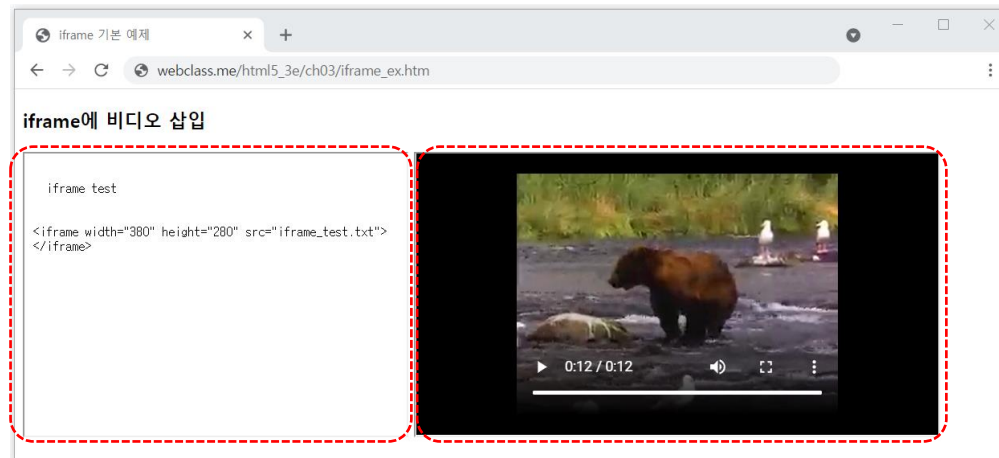
〈iframe〉으로 다른 문서의 내용 표시

- 브라우저 페이지 내에 또 다른 페이지 프레임을 삽입

〈iframe src = "파일주소" width="폭" height="높이" name="이름"〉 〈/iframe〉

- src 속성 : 내부 프레임에 출력할 파일의 url 지정
- width와 height 속성 : 브라우저 프레임의 가로, 높이 크기
- name 속성 : 프레임의 이름
 - 〈a〉 요소의 target 속성에 〈iframe〉 이름을 지정

```
〈iframe width="380" height="280" src="iframe_test.txt"〉 〈/iframe〉  
〈iframe width="520" height="280" src="bear.mp4"〉 〈/iframe〉
```

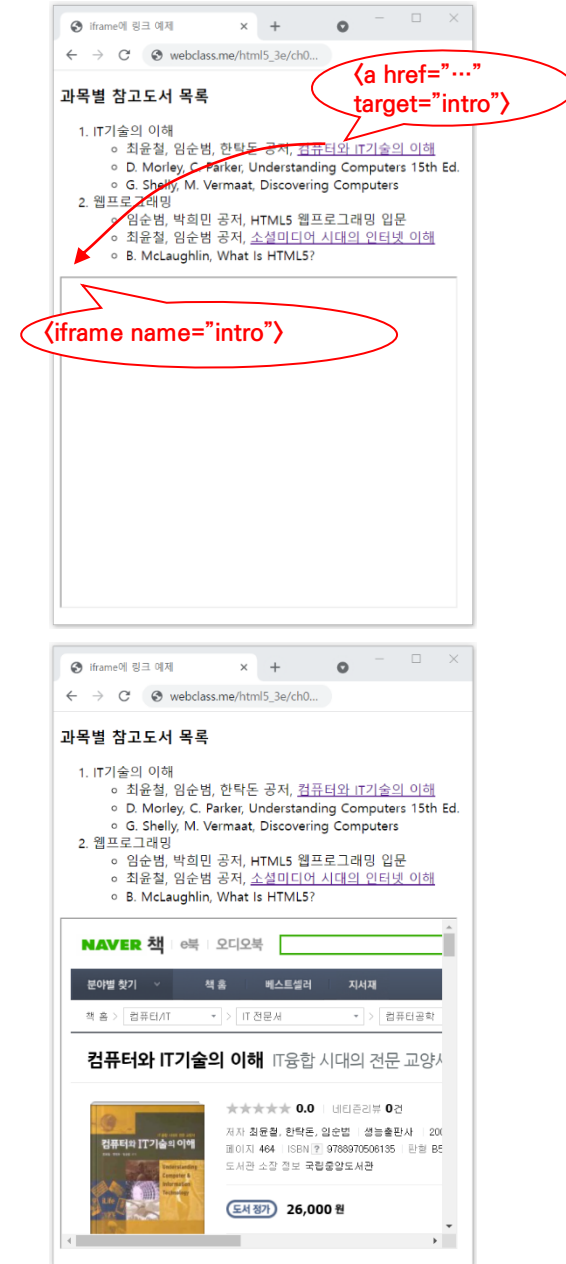


<iframe>으로 다른 문서를 링크

• 한 화면에서 링크로 연결된 내용 보기

```
...
<ul>
  <li><a href="http://book.naver.com/bookdb/book_detail.nhn?bid= 5339292" target="intro">
    컴퓨터와 IT기술의 이해 </a> </li>
  <li>D. Morley, C. Parker, Understanding Computers 15th Ed </li>
  <li>G. Shelly, M. Vermaat, Discovering Computers </li>
</ul>
<li>웹프로그래밍 </li>
<ul>
  <li>임순범, 박희민 공저, HTML5 웹프로그래밍 입문 </li>
  <li><a href="http://book.naver.com/bookdb/book_...?bid=7413144"
    target="intro">최윤철, 임순범 공저, 소셜미디어 시대의 인터넷 이해 </a> </li>
</ul>
</ol>

<iframe src="" name="intro" width="420" height="400"> </iframe>
</body>
```



CSS3 시작하기

스타일시트와 CSS3 기본 개념

CSS 속성선언

문서 일부분에 CSS 속성 설정

CSS3의 기능

- CSS(Cascading Style Sheet)
 - HTML 문서의 색이나 모양 등 외관을 꾸미는 언어
 - CSS3의 기능
 - 색상과 배경
 - 텍스트
 - 폰트
 - 박스 모델(Box Model)
 - 비주얼 포맷 및 효과
 - 리스트
 - 테이블
 - 사용자 인터페이스

CSS 속성선언

- CSS 기본 문법
 - 특정 엘리먼트 혹은 그 일부분에 대해 CSS 속성(Property) 설정
- CSS 스타일시트 구성요소
 - 선택자(Selector)
 - 스타일을 설정할 대상이 되는 태그
 - 여러 개의 경우 콤마(,)로 구분
 - 속성 선언(Property Declaration)
 - 속성(Property)과 속성값(value)
 - 콜론(:)으로 구분, 세미콜론(;)으로 종료

선택자(Selector) 속성선언(Declaration)

선택자 { 속성:값; 속성:값; ... }

예, h3 {color:red; font-style:italic; }

예제 4-2 CSS3 스타일 시트로 꾸민 웹 페이지

```
<!DOCTYPE html>
<html>
<head><title>스타일을 가진 웹 페이지</title>
<style>
  /* CSS 스타일 시트 작성 */
  body { background-color : mistyrose; }
  h3 { color : purple; }
  hr { border : 5px solid yellowgreen; }
  span { color : blue; font-size : 20px; }
</style>
</head>
<body>
<h3>CSS 스타일 맛보기</h3>
<hr>
<p>나는 <span>웹 프로그래밍</span>을 좋아합니다.</p>
</body>
</html>
```

〈body〉요소의 배경색 mistyrose

〈span〉의 글자는 blue에 20픽셀

CSS 스타일 맛보기

나는 웹 프로그래밍을 좋아합니다.

CSS3 스타일 시트 구성

- 예) 텍스트를 20픽셀 blue로 출력하는 CSS3 스타일시트

A diagram illustrating the components of a CSS selector and declaration block. The text `span { color : blue; font-size : 20px; } /* span 태그 스타일 선언 */` is shown with brackets and arrows identifying its parts:

- selector** (선택터): A bracket above `span`.
- property** (프로퍼티): Brackets above `color` and `font-size`.
- value** (값): Brackets above `blue` and `20px`.
- colon** (콜론): An arrow points to the colon after `color`.
- semicolon** (세미콜론): An arrow points to the semicolon after `blue`.
- rule** (규칙): A bracket above the entire declaration block `{ color : blue; font-size : 20px; }`.
- comment** (주석문): A bracket above the comment `/* span 태그 스타일 선언 */`.

- 셀렉터
 - CSS3 스타일 시트를 HTML 페이지에 적용하도록 만든 이름
- 프로퍼티
 - 스타일 속성 이름. 약 200개 정도의 프로퍼티 있음
- 값
 - 프로퍼티의 값
- 주석문
 - 스타일 시트 내에 붙이는 설명문으로 /* ... */. 여러 줄, 아무 위치에나 사용 가능
- 대소문자 구분 없음

```
body { background-color : mistyrose; }
BODY { Background-Color : Mistyrose; }
```

HTML문서에 CSS3 스타일 시트 만들기

- HTML 문서에 CSS3 스타일 시트 만드는 방법 3 가지
 - `<style></style>` 태그에 스타일 시트 작성
 - 지금까지 사용한 방법
- 다른 방법은?
 - style 속성에 스타일 시트 작성
 - 스타일 시트를 별도 파일로 작성
 - `<link>` 태그나 `@import`로 불러 사용

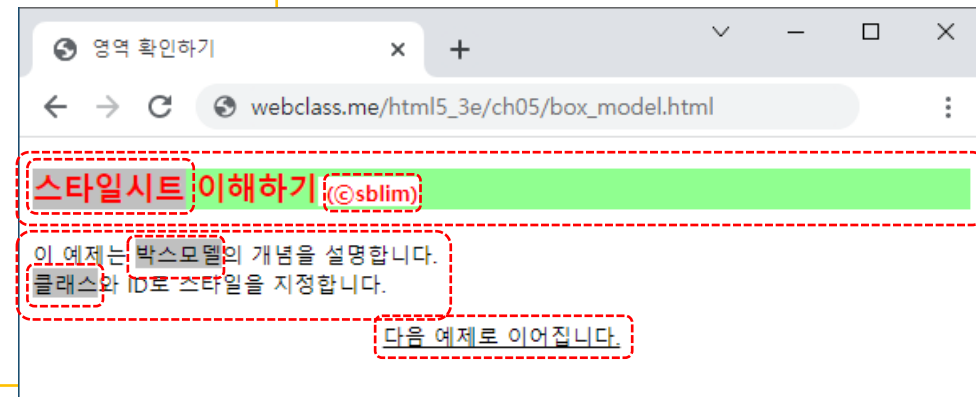
박스모델 설정하기

영역설정을 위한 박스모델

• 배경 영역

- <h3>, <p>, <div> : 해당하는 줄만큼 배경
- , : 해당하는 글자들만
- <table>, : 자신의 영역이 미리 결정

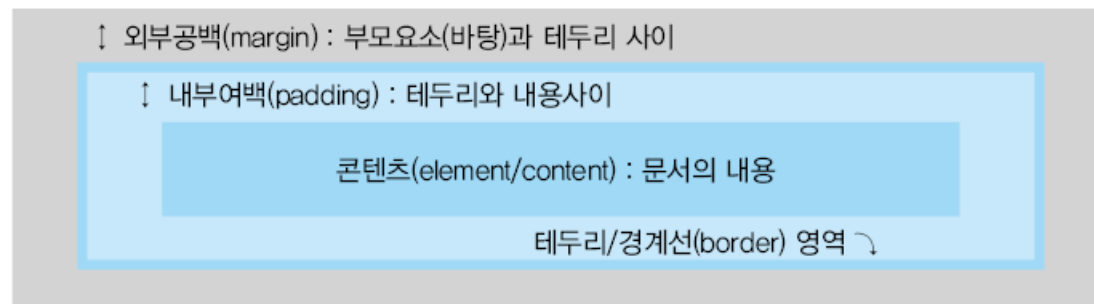
```
<head> <style type="text/css">
  p {font-size: 10pt}
  h3 { text-decoration: underline; background-color: #90ff90 }
  h3:after { content: " (©sblim)"; font-size:10pt; background-color: white }
  strong, .under1 { background-color: silver }
  #next { text-align: center; text-decoration: underline; }
</style> </head>
<body>
  <h3> <strong>스타일시트</strong> 이해하기</h3>
  <p>이 예제는 <span class="under1">박스모델</span>의 개념을 설명합니다.
  <br><span class="under1">클래스</span>와 ID로 스타일을 지정합니다.</p>
  <p id="next">다음 예제로 이어집니다.</p>
</body>
```



박스 공간

- 박스공간의 구성

- HTML의 모든 요소들은 네모 박스 모양의 공간을 차지
- 요소가 차지하는 공간 개념 => 박스모델(box model)
- 내부여백(padding), 테두리(border), 외부공백(margin) 지정

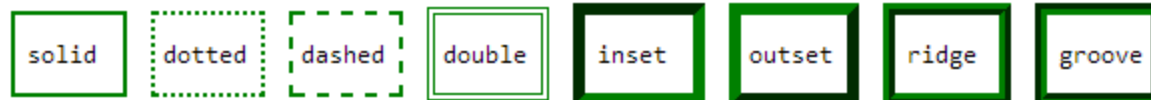


박스 속성

- 박스공간을 위한 속성

- 외부공백 : `margin`, `margin-top`, `margin-right`, `margin-left`, ...
- 내부여백 : `padding`, `padding-top`, ...
- 테두리/경계선의 두께 : `border-width`, `border-top-width`, ...
- 테두리의 모양 : `border-style` (실선, 점선, 이중선 등)
- 테두리의 색상 : `border-color`
- 테두리 지정 줄여쓰기(shorthand) :
`border: <width> <style> <color>`

`border-style`



레이아웃 설정하기

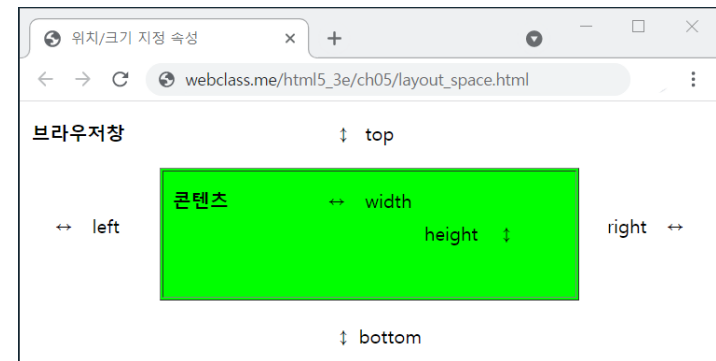
콘텐츠의 위치 지정 방법

플로팅 박스 배치하기

콘텐츠 박스의 크기 조정하기

콘텐츠의 위치 지정 방법

- 위치 및 크기 지정
 - top, right, bottom, left, width, height 속성
- 위치값의 유형 지정 : position 속성
 - static (기본값) : 순서대로
 - absolute : 문서(상위 요소) 내에서 절대위치
 - relative : 직전 요소에 이어서 상대위치
 - fixed : 현재 브라우저 화면 내에서 절대위치



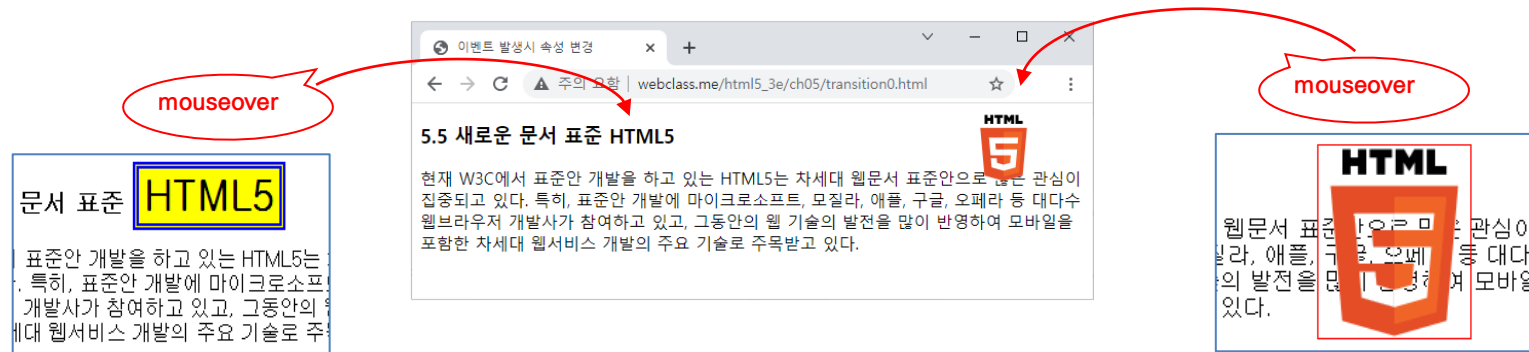
전환 효과 설정하기

속성값 변경으로 전환 효과
점진적으로 변하는 전환 효과

전환효과

- 이벤트 : <style attribute>
 - 객체 모양의 변화는 CSS 속성값의 변화: 예, width/height 값

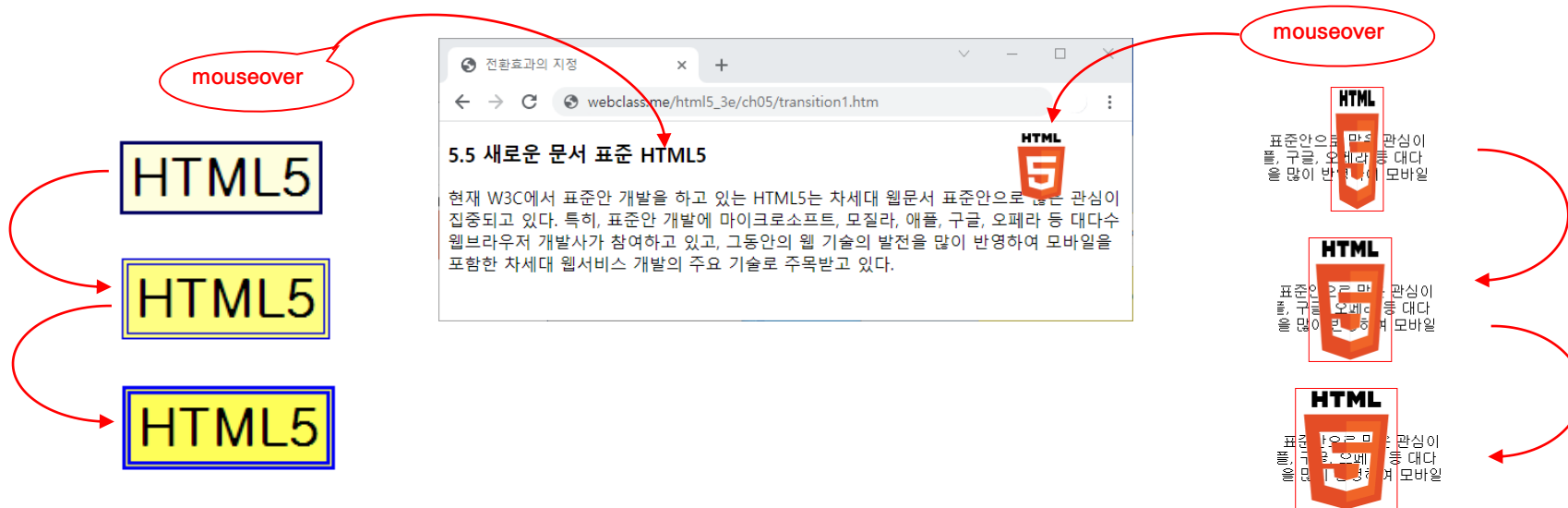
```
<head> <style type="text/css">
  #title:hover { border: thick double blue; padding:4px;
                background-color: yellow; font-size: xx-large; }
  #h5_logo { position: absolute; top: 10px; right: 60px; }
  #h5_logo:hover { border: thin solid red; width: 108px; height: 132px; }
</style> </head>
<body>
  <h3>5.5 새로운 문서 표준 <span id="title">HTML5</span></h3>
  
  <p>현재 W3C에서 표준안 개발을 하고 있는 HTML5는 차세대 웹문서 ...(중략)... </p>
</body>
```



점진적으로 변하는 전환효과

- transition: <style attribute> <duration>
 - 변화될 속성이름과 전환시간을 지정

```
<style type="text/css">  
#title:hover { border: thick double blue; padding:4px; background-color: yellow; font-size:  
xx-large;  
transition: border 4s, background-color 8s; }  
#h5_logo { position: absolute; top: 10px; right: 60px; }  
#h5_logo:hover { border: thin solid red; width: 108px; height: 132px; transition: width 4s; }  
</style>
```



폼 이해하기

〈form〉 요소의 사용

- 폼 요소의 사용
 - 회원가입, 상품구매, 키워드 검색 등 사용자로부터 정보를 받을 때
 - 사용자와 애플리케이션이 상호작용
 - 사용자 입력 ⇒ 전송버튼 ⇒ 애플리케이션에 전달 ⇒ 실행결과 반환
- 〈form〉 요소의 역할
 - 사용자가 입력하는 정보를 하나로 묶어서 애플리케이션에 전달
 - 다양한 입력 양식을 그룹핑하고 전송방법을 설정
 - 〈form〉 요소 내 사용자의 정보 입력 양식
 - 〈input〉, 〈textarea〉, 〈select〉, 〈button〉 등의 입력 요소를 이용

다양한 입력 폼 예제

<h3>다양한 입력 폼</h3>

<form method="get" action="form_app.js">

성명: <input type="text" name="person"/>

성별: <input type="radio" name="gender" value="male"/>남성
<input type="radio" name="gender" value="female"/>여성

직업: <select name="job" size="1">
<option>학생</option> <option>회사원</option>
<option>공무원</option> <option>기타</option>
</select>

<p> 구입희망분야(복수선택 가능)

- 분야: <input type="checkbox" name="books" value="computer"/>컴퓨터
<input type="checkbox" name="books" value="economy"/>경제
<input type="checkbox" name="books" value="common"/>상식

비고:

<textarea name="comments" rows="4" cols="40"/></textarea>

</p> <hr>

<input type="submit" value="신청"/>

<input type="reset" value="취소"/>

</form>

다양한 입력 폼

성명:

성별: ☐ 남성 ☐ 여성

직업:

구입희망분야(복수선택 가능)

분야: ☐ 컴퓨터 ☐ 경제 ☐ 상식

비고:

다양한 입력 폼

성명: 홍길동

성별: ☒ 남성 ☐ 여성

직업:

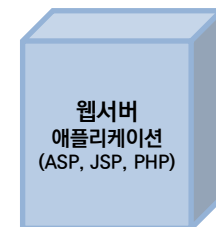
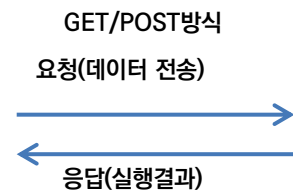
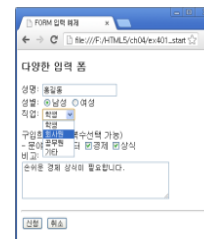
구입희망분야(복수선택 가능)

- 분야: ☒ 컴퓨터 ☒ 경제 ☐ 상식

비고:

〈form〉 요소의 주요 속성

- 〈form name="이름" method="get/post" action="애플리케이션 주소"〉
 - method 속성 : 데이터 전송 방식 지정
 - get방식: 전송할 데이터를 URL 주소 뒤에 포함,
 - 예, `http://... ch04/app.js?person=%C8%AB%B1%E6&sex=male& jobs=...`
 - 간단한 데이터에는 편리, 그러나 보안에 취약
 - post방식: 프로그램의 입출력 방식을 사용, 데이터 양에 제한 없다
 - action 속성 : 데이터 처리할 애플리케이션 주소 (웹서버 경우 URL)
 - name속성 : 폼 요소에 대한 이름
- HTML 문서에서 〈form〉의 역할
 - 다양한 〈input〉 요소에 대해
 - 데이터 입력 ⇒ 전달 ⇒ 반환



자바스크립트 시작하기

자바스크립트 개요

- 개요 및 특징

- 동적인 웹 문서 제작과 웹 응용프로그램 개발을 위해 필수적으로 사용됨
 - 자바 애플릿, CGI 스크립트 대체 가능
- C/C++이나 자바 언어 등에 비해서 작성 및 실행이 매우 간편함
- 인터프리터 (interpreter) 방식

- 자바스크립트 연혁

- 라이브스크립트라는 이름으로 넷스케이프사에서 개발 시작
- 1995년 썬 (Sun, 현재 오라클)사와 공동 개발 → 자바스크립트 (JavaScript)로 개명
- 현재 표준
 - ECMA(European Computer Manufacturers Association): ECMA-262 혹은 ISO 16262
 - ECMAScript라고도 불리움

객체 기반의 자바스크립트

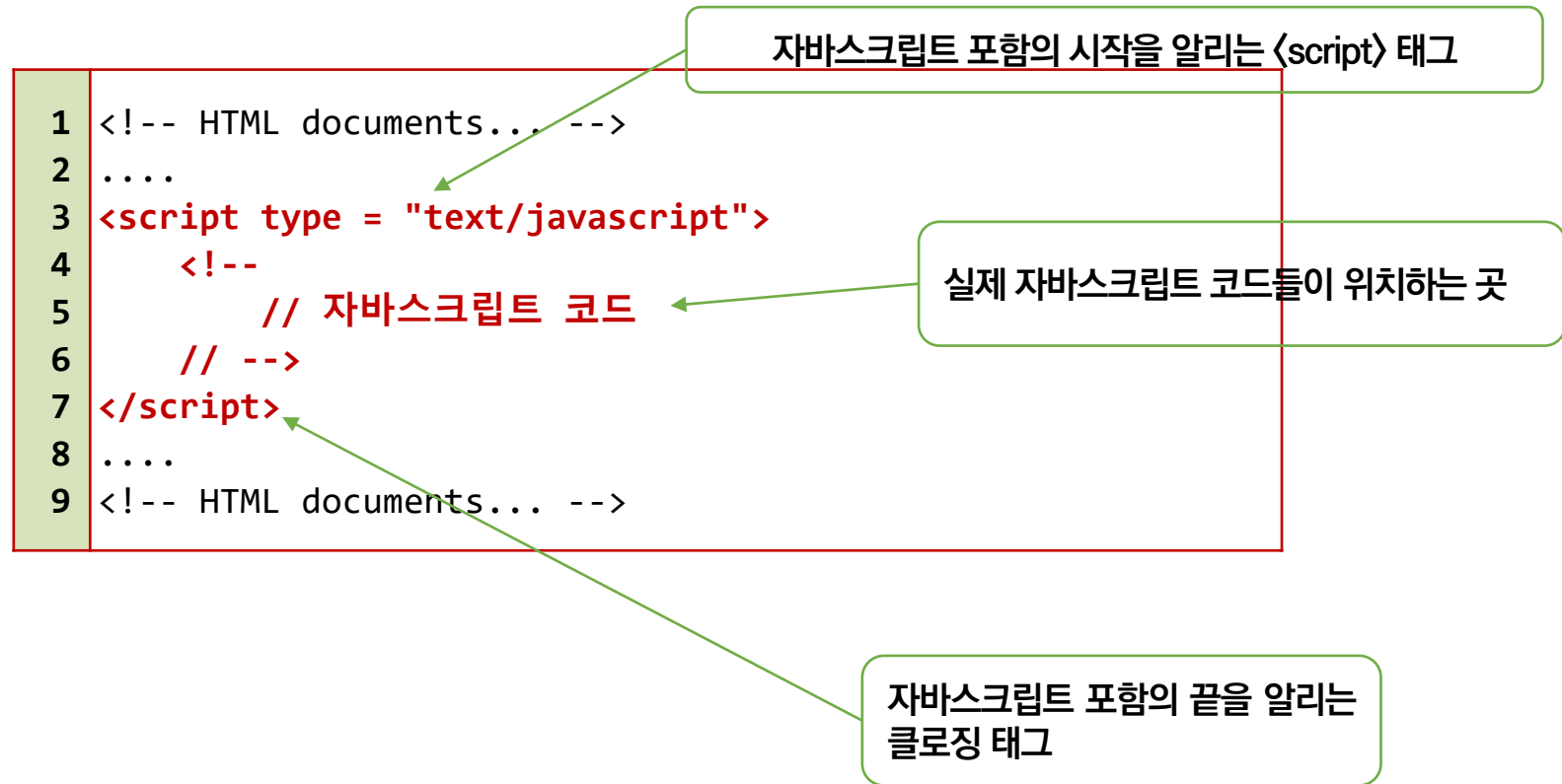
- 자바 언어의 영향을 받아서 문법적으로 비슷한 형태를 가지는 공통점이 있으나 자바 언어와는 다음과 같이 차이점을 가진다

	자바스크립트	자바 언어
실행 방식	웹 브라우저에서 바로 자바스크립트 코드를 해석하고 바로 실행 (스크립트/인터프리터 기반 언어)	자바 프로그램을 컴파일 후 변환된 object code를 자바가상머신에서 실행하는 방식 (컴파일 기반 언어)
성격	객체기반(object-based)	객체지향(object-oriented)
작성 형태	HTML 파일 내에 포함되어 작성됨	별도의 자바 프로그램 파일로 작성
변수형 선언 및 타입 검사	변수의 선언이 따로 필요 없으며 타입 검사도 매우 느슨함	변수의 선언이 필요하며 변수 타입의 검사가 매우 엄격함

자바스크립트 작성하기

- 자바스크립트 코드
 - HTML 파일 없이 웹브라우저에서 독립적으로 실행 될 수 없음
 - 반드시 HTML 파일 내에 포함되어 있어야 한다
- HTML 파일 내에 포함 시키는 두가지 방식
 - 웹문서 내장 방식
 - 외부 파일 참조 방식
 - CSS를 다룰 때도 내장하거나 외부 파일을 사용할 수 있었음

웹문서 내장 방식



외부 파일 참조 방식

- `<script>` 태그의 `src` 속성의 값으로 자바스크립트 파일의 경로를 지정

```
1 <!-- HTML documents... -->
2 ....
3 <script type="text/javascript" src="myscript.js">
4 </script>
5 ....
6 <!-- HTML documents... -->
```

`<script>` 요소의 `src` 속성의 값으로 실제 자바스크립트 파일의 경로를 지정할 수 있다.

- 자바스크립트 파일의 URL 경로 지정 가능

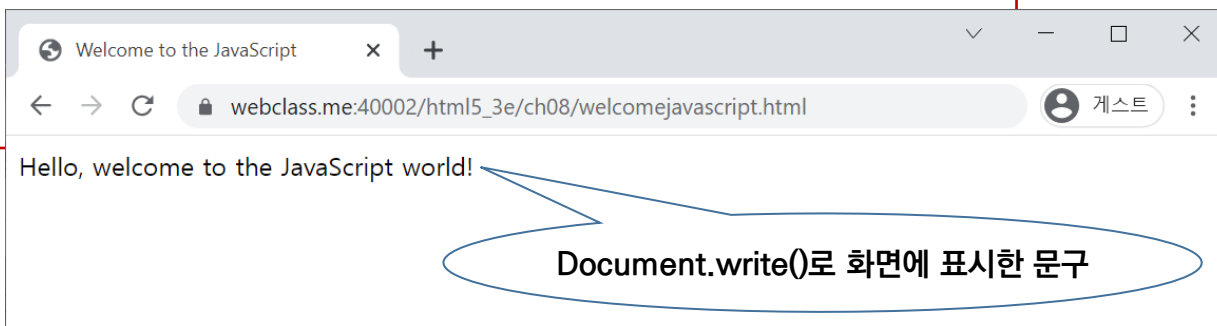
```
1 <!-- HTML documents... -->
2 ....
3 <script type="text/javascript"
4         src="http://webclass.me/html5/ch08/remove_script.js">
5 </script>
6 ....
7 <!-- HTML documents... -->
```

`<script>` 요소의 `src` 속성의 값으로 자바스크립트 파일이 위치한 URL 경로를 지정할 수도 있다.

간단한 자바스크립트 예제

```
1 <!DOCTYPE html>
2 <!-- hello.html
3   간단한 인사말을 화면에 표시하는 HTML/자바스크립트 기본 예제
4 -->
5 <html>
6 <head>
7 <title> Welcome to the JavaScript </title>
8 </head>
9 <body>
10  <script type="text/javascript">
11    <!--
12      document.write("Hello, welcome to the JavaScript world!");
13    // -->
14  </script>
15 </body>
16 </html>
```

웹브라우저에서 보여지는 문서상에 Hello, welcome to the JavaScript world!라는 문구를 출력하는 명령문



- “Hello, welcome to the JavaScript world!”라는 문구를 출력

자바스크립트 기본 문법

자바스크립트 기본 변수
기본 연산자와 변수 형 변환
화면 출력 및 키보드 입력

자바스크립트 기본 연산자

종류	연산자	설명	비고
사칙연산	+ - * / %	더하기, 빼기, 곱하기, 나누기, 나머지	*, /, %가 +, - 보다 연산 우선순위가 높다
대입 연산자	+ = - = * = / = % =	왼편 변수에 우측 값을 연산 후 왼편 변수에 대입	
증감 연산자	++ --	기존 변수 값에 +1 혹은 -1 연산 수행	
논리연산자	> <	왼편의 값이 크다, 작다	결과값이 true 혹은 false이다.
	>= <=	왼편의 값이 크거나 같다, 작거나 같다	
	== !=	양편이 같다, 다르다 (값만 비교)	
	=== !==	양편의 같다, 다르다 (값과 타입 모두 비교)	
	! &&	논리부정 (NOT), 논리합 (OR), 논리곱 (AND) 연산	

자바스크립트 제어문

자바스크립트 제어문

- 제어문으로 if 문과 switch 문을 제공

자바스크립트 제어문	문법 및 사용 형식	비고
if-then-else	<pre>if (조건식) { // 조건식의 값이 true일 때 실행될 문장 } else { // 조건식의 값이 false일 때 실행될 문장 }</pre>	실행될 문장이 한 개인 경우에는 { }를 생략할 수 있다.
switch	<pre>switch (expression) { case value_1: // expression값이 value_1일 때 실행될 문장 break; case value_2: // expression값이 value_2일 때 실행될 문장 break; case value_3: // expression값이 value_3일 때 실행될 문장 break; ... default: // case문에서 찾을 수 없을 때 실행될 문장 } </pre>	C/C++, 자바 언어와는 달리 (expression)에 정수형 이외의 타입도 사용할 수 있다. 예를 들면 문자열 형식의 값을 사용할 수도 있다.

자바스크립트 반복문

- while, for, do-while 문을 제공

자바스크립트 반복문	문법 및 사용 형식	비고
while	<pre>while (조건식) { // 조건식의 값이 true일 동안 반복해서 실행될 문장 }</pre>	실행될 문장의 개수가 하나인 경우에는 { }를 생략할 수 있다.
for	<pre>for (초기화 문장; 조건식; 증감문) { // 조건식의 값이 true일 동안 반복해서 실행될 문장 }</pre>	
do-while	<pre>do { // 조건식의 값이 true일 동안 반복해서 실행될 문장 } while (조건식)</pre>	

DOM 모델

DOM의 정의 및 문서 구조

- DOM (Document Object Model)
 - 웹 문서를 자바스크립트 입장에서 구조적 문서 (document) 객체 형태로 다루는 모델
 - 자바스크립트는 HTML 문서를 객체(Object)로 바라보고 다룬다
 - HTML 문서 뿐만 아니라 CSS 속성도 변경 가능
- DOM 표준
 - 2000년에 DOM2 제정
 - 대부분의 웹브라우저가 지원
 - 2004년에 DOM3 까지 제정된 상태
- 자바스크립트 활용
 - 자바스크립트를 이용하여 HTML 문서의 내용을 변경
 - 사용자 입력을 받아 처리

DOM과 HTML 웹 문서의 관계

- DOM을 이용해 웹문서를 프로그램 언어에서의 변수나 구조체와 같은 데이터로 처리할 수 있도록 한다

자바스크립트를 이용해 DOM의 내용을 추가/변경



HTML 문서의 태그/콘텐츠가 변경되는 효과

화면에 디스플레이 되는 내용도 변경

HTML 문서의 내용을 동적으로 변경

HTML 문서와 DOM

HTML 태그 요소와 DOM

- 태그 요소는 DOM의 객체로 표현 됨
 - 태그 요소에 포함된 다른 요소는 객체내에 소속된 객체 형태로 표현 (하위객체)
- 태그 속성은 DOM 객체의 속성으로 표현 됨
- 요소 전체가 하나의 객체

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

- type과 name은 속성
- "text"와 "username"은 type과 name의 속성값

DOM을 통한 HTML 문서 접근

- DOM 접근 방법

1. document의 forms나 images 속성을 이용해서 접근하는 방법
2. document 객체가 제공하는 getElementByTagName(), getElementByClassName() 메소드를 이용해 태그나 클래스 이름으로 접근하는 방법
3. document 객체가 제공하는 getElementById() 등의 메소드를 이용해서 접근하는 방법
 - 가장 사용이 쉽고 많이 사용되고 방법인 getElementById() 메소드 방법을 중심으로 설명