

01 웹 프로그래밍 기초

인터넷

- (QUIZ) 지난 30년 동안 가장 혁신적인 발명품은?

- 인터넷
- PC
- 휴대전화
- E-MAIL
- LOL (리그 오브 레전드)
- ...



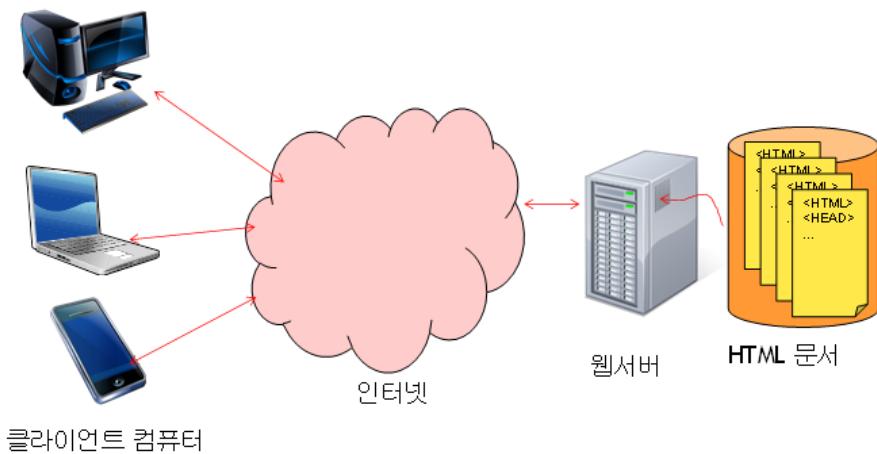
WWW

- WWW(World Wide Web): 세계를 뒤덮는 거미줄
- 초기 인터넷에서는 텔넷, FTP, 전자 메일, 우즈넷 등의 문자 위주 서비스
- WWW은 인터넷을 사용하기 쉽도록 하이퍼 텍스트와 그림을 통하여 모든 서비스를 이용할 수 있도록 만든 것



3

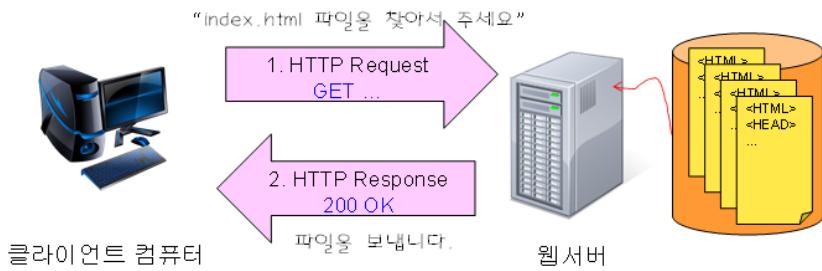
WWW의 동작원리



4

클라이언트와 서버

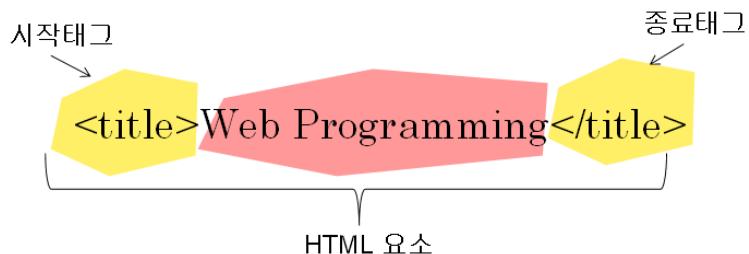
- 2가지의 기본 프로토콜
 - 특정한 파일을 요청하는 HTTP Request
 - 찾은 파일을 돌려주는 HTTP Response



5

HTML

- HTML(Hyper Text Markup Language)은 웹 페이지를 기술하기 위한 마크업(markup) 언어
- 마크업 언어는 텍스트에 태그를 붙여서 텍스트가 문서의 어디에 해당하는지를 기술한 것



6

웹브라우저

- HTML 문서를 읽어서 눈에 보이는 웹 페이지를 만든다.

웹브라우저는 HTML 문서를 해석하여
화면에 웹페이지를 표시한다.



7

HTML의 역사

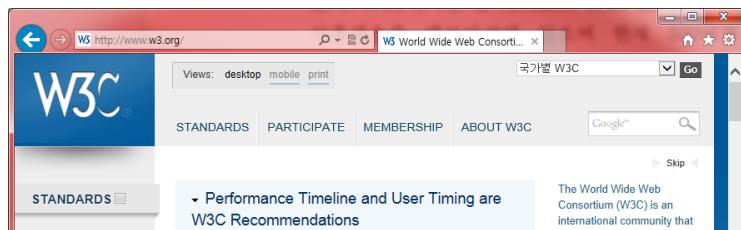


- 팀 버너스리(Tim Berners-Lee)에 의하여 개발
- 인터넷의 아버지
- URL, HTTP, HTML 최초 설계
- 1989년 팀 버너스리는 CERN의 연구자들이 문서를 공유할 수 있는 월드 와이드 웹의 하이퍼텍스트 시스템을 고안하여 개발했다.
- 1990년 최초의 하이퍼텍스트 브라우저와 편집기를 개발
- 차세대 웹 기술인 시맨틱 웹 기술의 표준화 작업중

8

W3C

- W3C란 World Wide Web Consortium의 약자
- 중립적인 기구로서 참여기관들이 협력하여 웹 표준을 개발하는 국제 컨소시엄
- 팀 버너스 리를 중심으로 1994년에 설립
- 웹의 프로토콜과 가이드라인을 개발
- 홈페이지는 <http://www.w3.org>



9

HTML 버전

Version	Year
HTML	1991
HTML+	1993
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML 1.0	2000
HTML5	2012
XHTML5	2013

10

HTML5

- HTML5는 HTML의 새로운 표준
 - 완전한 CSS3 지원
 - 비디오와 오디오 지원
 - 2D/3D 그래픽 지원
 - 로컬 저장소 지원
 - 로컬 SQL 데이터베이스 지원
 - 웹 애플리케이션 지원



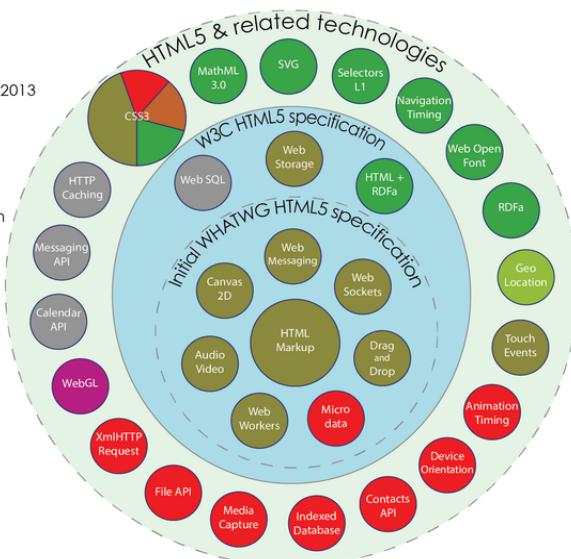
11

HTML5 현황

HTML5

Taxonomy & Status on January 20, 2013

- W3C Recommendation
- Proposed Recommendation
- Candidate Recommendation
- Last Call
- Working Draft
- Non-W3C Specifications
- Deprecated

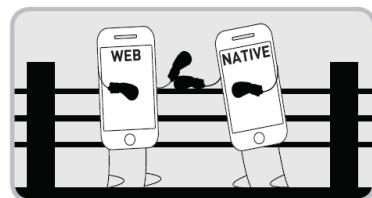


by Sergey Mavrody CC BY · SA

12

웹 앱과 네이티브 앱

	Native App.	Web App.
실행속도	빠름	일반적으로 느림
플랫폼	플랫폼마다 제작하여야 함	하나의 버전으로 모든 플랫폼에서 실행 가능
배포	앱 마켓을 통한 배포	배포할 필요가 없음
버전 업데이트	상당한 시간이 걸림	즉시 반영된다.
오프라인시	사용가능	약간의 기능 사용 가능



13

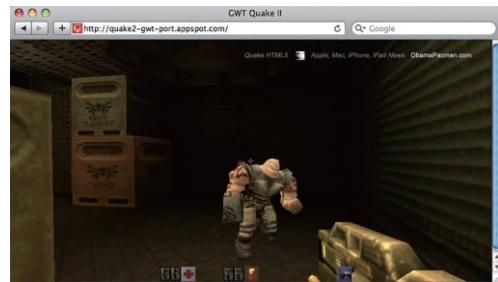
웹 앱과 네이티브 앱



14

멀티미디어 지원

- 웹 브라우저에서 비디오나 오디오를 재생
 - 예전 방법: 어도비(adobe)의 플래시
 - HTML5: <audio>, <video> 태그를 이용해서 지원
- 그래픽을 위한 캔버스 요소 지원
- 벡터 그래픽스를 지원하는 SVG(Scalable Vector Graphics)
- WebGL 3D를 이용하여 3차원 그래픽 지원



15

HTML5의 신기능

- 오프라인 웹 애플리케이션 - 네트워크가 연결되지 않은 상태에서도 실행 가능
- 드래그 앤 드롭(Drag-and-drop) - 요소들을 마우스로 끌어서 넣을 수 있음
- 웹 스토리지(Web Storage) - 쿠키를 대체할 수 있는 웹 저장소 기능 제공
- 위치 정보(Geolocation) 제공 - 지도 기능
- 웹 SQL 데이터베이스(Web SQL Database)
- 파일 API 지원 - 파일 업로드와 파일 관리 기능 제공
- 웹소켓(WebSocket) API 제공 - 서버와 브라우저 간의 양방향 통신 기능 제공

16

HTML5+CSS3+Javascript

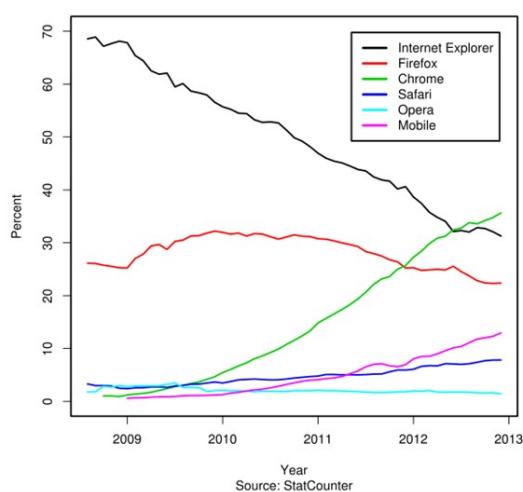
- 웹 페이지의 내용은 HTML5로 작성
- 웹 페이지의 스타일은 CSS3로 지정
- 웹 페이지의 상호작용은 자바스크립트로 작성



17

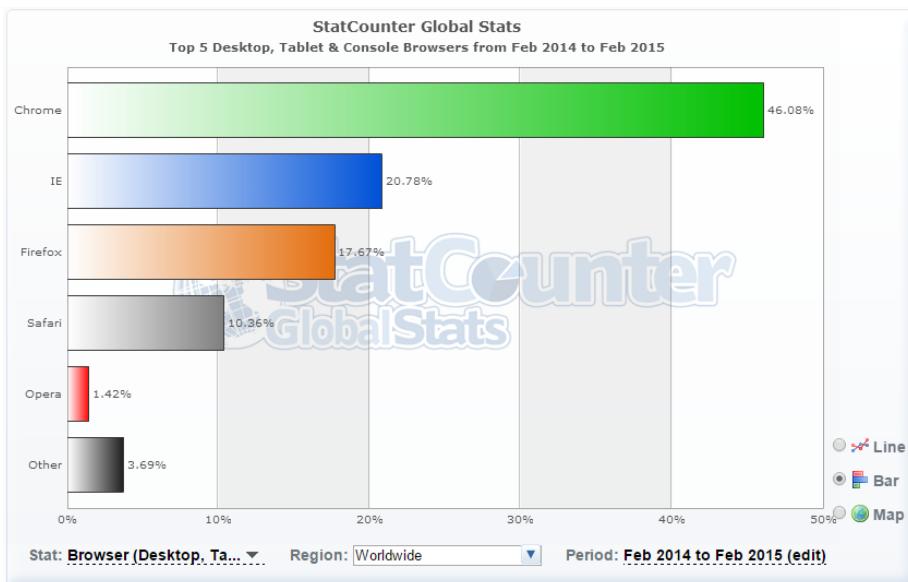
웹브라우저

Usage share of web browsers



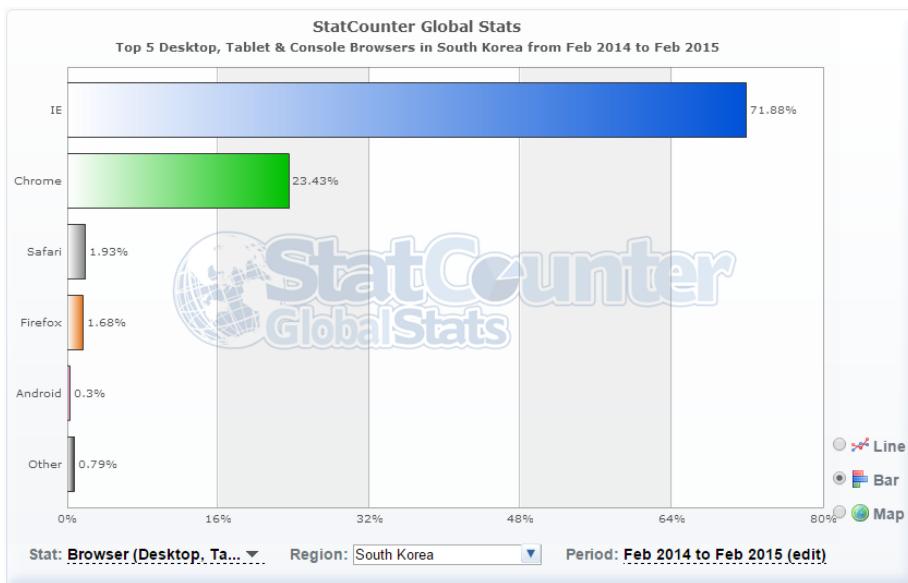
18

웹브라우저 점유율(전세계)



19

웹브라우저 점유율(대한민국)



20

HTML5 지원 여부

웹 브라우저	버전	HTML5 테스트 점수(555만점)
Apple Safari	8.0	396
Google Chrome	40	511
Maxthon	4.4.4.2000	467
Mozilla Firefox	36	449
Opera	26	497
Microsoft Internet Explorer	8	33
Microsoft Internet Explorer	9	113
Microsoft Internet Explorer	10	297
Microsoft Internet Explorer	11	343

21

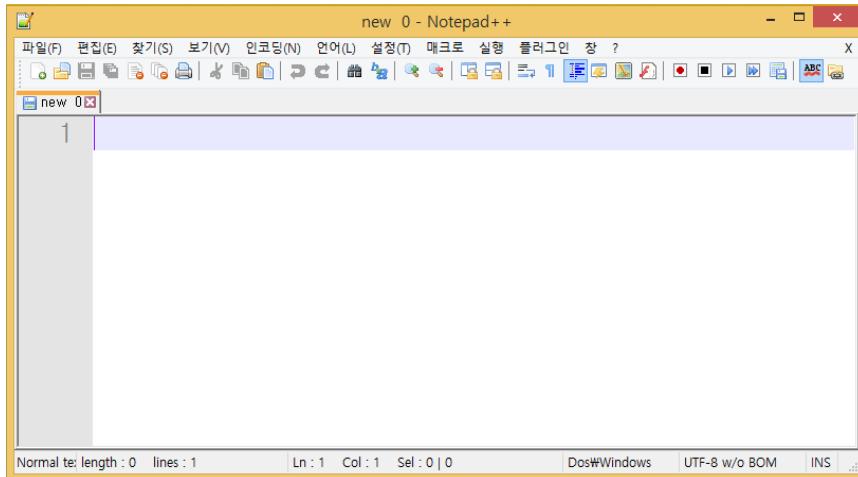
웹 브라우저의 사용

- 다양한 웹 브라우저에서 어떤 브라우저를 사용해야 할까?
- 정답은 없다.
- 사용자로서 개인적으로 좋아하고 편하다고 생각하는 브라우저를 사용하면 된다.
- 다만 HTML과 CSS가 업계 표준이기는 하지만 브라우저마다 지원하는 정도가 조금씩 다르다.
- 따라서 현재 다양한 브라우저가 존재하는 만큼 사용자에게 배포되는 HTML문서를 작성할 때는 여러 브라우저를 사용해 다양한 환경에서 테스트해야 한다.

22

HTML 편집기

- 메모장
- Notepad++



23

메모장을 이용한 HTML 작성

1. 메모장을 실행하여서 다음과 같이 입력한다.

```
<!DOCTYPE html>
<html>
  <head>
    <title>나의 웹 페이지</title>
  </head>
  <body>
    <p>안녕하세요. 웹프로그래밍기초</p>
  </body>
</html>
```

A screenshot of a Windows Notepad window titled "intro - 메모장". The menu bar includes "파일(F)", "편집(E)", "서식(O)", "보기(V)", and "도움말(H)". The main text area contains the following HTML code:

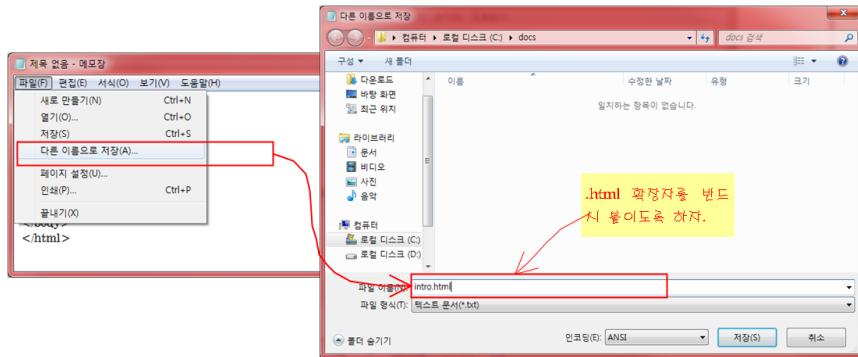
```
<!DOCTYPE html>
<html>
  <head>
    <title>나의 웹 페이지</title>
  </head>
  <body>
    <p>안녕하세요. 웹프로그래밍기초</p>
  </body>
</html>
```

The code is written in a monospaced font, with tags in blue and content in black. The Notepad window has a yellow border and a standard Windows title bar.

24

메모장을 이용한 HTML 작성

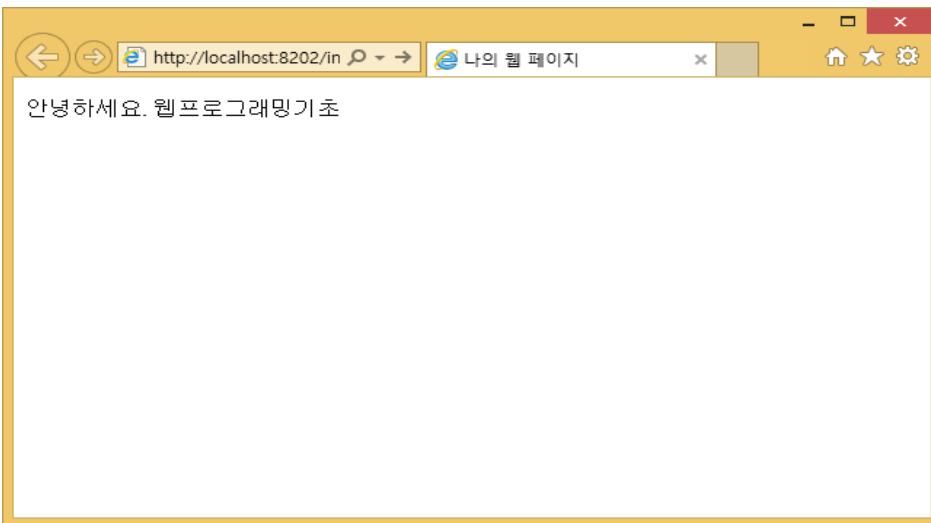
2. 입력된 HTML 코드를 [파일]->[다른 이름으로 저장] 메뉴를 사용하여서 파일에 저장한다.



25

HTML 파일 실행

3. 저장된 HTML 파일을 더블클릭하여 실행한다.



26

HTML 소스 보기

- 마우스 오른쪽 버튼을 누르고 [소스보기] 메뉴를 선택하면 현재 페이지의 HTML 소스를 볼 수 있다.



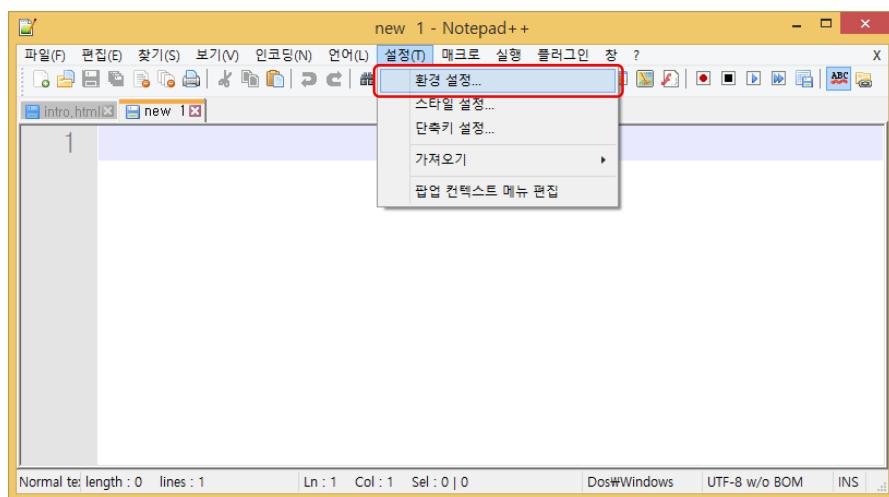
A screenshot of a Windows-style window titled "file:///D:/Library/Desktop/intro.html - 원본". The window contains the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>나의 웹 페이지</title>
5   </head>
6   <body>
7     <p>안녕하세요. 웹프로그래밍기초</p>
8   </body>
9 </html>
```

27

Notepad++

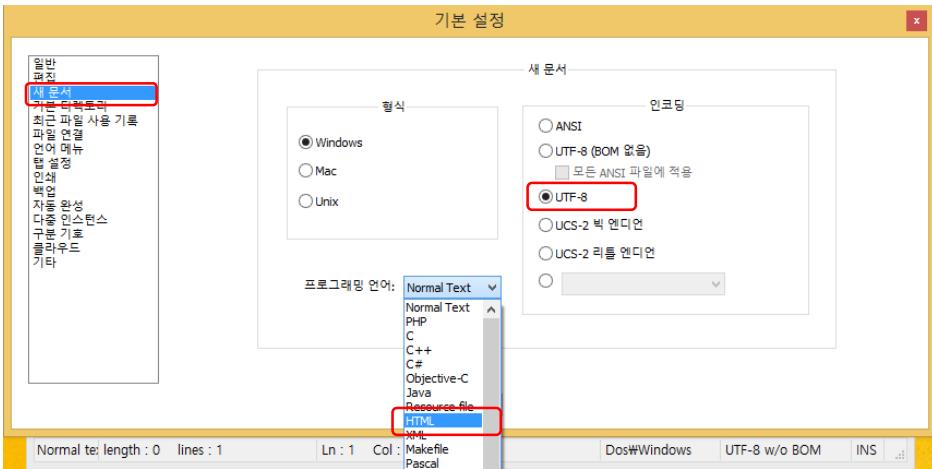
환경설정



28

Notepad++

- 왼쪽에서 새문서를 선택하고 프로그래밍언어는 HTML, 인코딩은 UTF-8을 선택한다.



29

Notepad++

- 소스 변경

The screenshot shows an open file named 'intro.html' in Notepad++. The code is as follows:

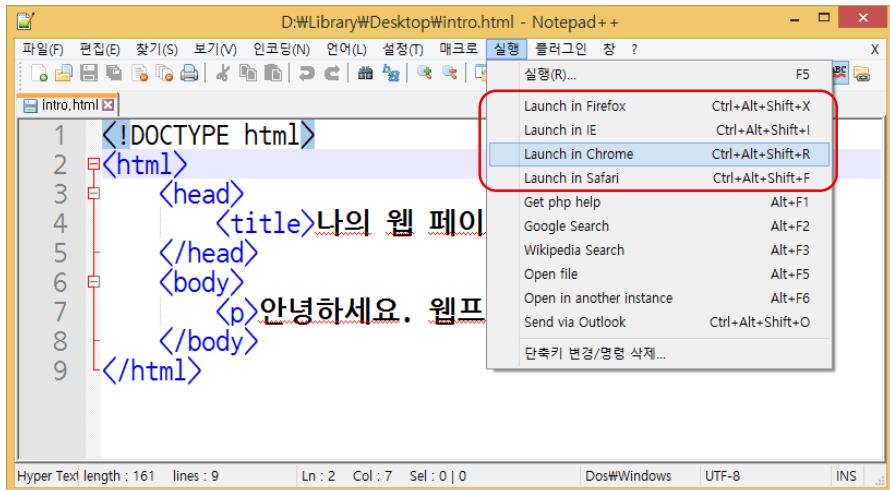
```
<!DOCTYPE html>
<html>
<head>
    <title>나의 웹 페이지</title>
</head>
<body>
    <p>안녕하세요. 웹프로그래밍기초</p>
</body>
</html>
```

The entire code block is circled in red. The code uses syntax highlighting where tags like <html>, <head>, <title>, <body>, and <p> are in blue, and text content is in black. The Notepad++ interface includes a toolbar at the top and status bars at the bottom showing file path, line count, column count, and encoding.

30

Notepad++

- 실행(브라우저가 설치되어 있을 경우 실행된다.)



31

HTML 문서의 기본 구조

```
<!DOCTYPE html>
<html>
  <head>
    <title>나의 웹페이지</title>
  </head>

  <body>
    <p>안녕하세요. 웹프로그래밍 기초</p>
  </body>
</html>
```

32

요소(element)

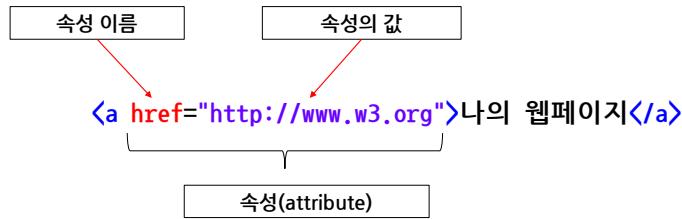
- 시작태그와 종료태그로 이루어진 문서의 구성 요소
- 요소 = (시작 태그 + 콘텐츠 + 종료 태그)



33

속성

- 속성은 요소에 대한 추가적인 정보를 제공
- 속성은 항상 시작태그에 이름=“값” 형태로 기술된다.



34

HTML 주석

- 주석(comment)은 HTML 코드를 설명하는 글

```
<!-- 여기에 주석을 표시합니다. -->
<!DOCTYPE html>
<html>
<...>
```

코드를 설명하는 글

35

HTML 문서 작성시 주의사항

1. HTML 문서는 대소문자를 가리지 않으므로 Head, HEAD, HeaD, head 등 어떠한 형태로 써도 무방하나 되도록 보기 편하고 수정이 용이하도록 소문자로 통일해서 쓰는 것이 좋음
2. 시작태그와 종료태그를 먼저 쓰고 그 안에 내용을 넣는다.
3. HTML 문서를 정의할 때 들여쓰기(indent)에 주의한다.

36

<!DOCTYPE> 선언

- 웹페이지에 사용된 HTML의 종류와 버전을 지정
- HTML5

```
<!DOCTYPE html>
```

- HTML 4.01

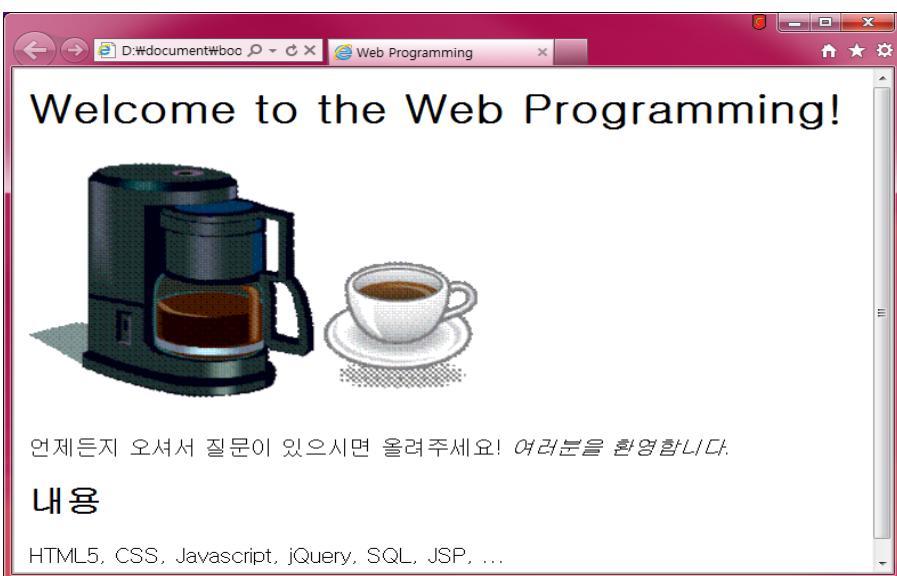
```
<!DOCTYPE html public "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

- XHTML 1.0

```
<!DOCTYPE html public "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

37

HTML 만들기



38

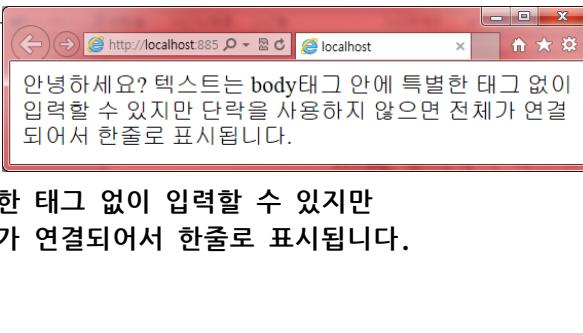
02 HTML 기본 요소

이번 장의 목표



텍스트 표시

- 텍스트는 특별한 태그 없이도 <body>...</body> 안에서 표시할 수 있다.
- 하지만 단락을 생성하지 않으면 모든 텍스트가 연결되어서 하나의 긴 줄로 표시된다.



The screenshot shows a web browser window with the URL `http://localhost:885`. The page content is:

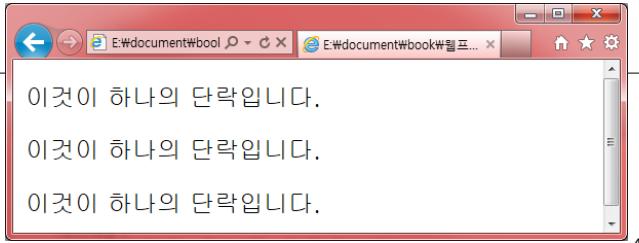
```
<!DOCTYPE html>
<html>
<body>
안녕하세요?  
텍스트는 body태그 안에 특별한 태그 없이 입력할 수 있지만  
단락을 사용하지 않으면 전체가 연결되어서 한줄로 표시됩니다.
</body>
</html>
```

A red box highlights the text "안녕하세요?" and the explanatory text below it. The explanatory text reads: "안녕하세요? 텍스트는 body태그 안에 특별한 태그 없이 입력할 수 있지만 단락을 사용하지 않으면 전체가 연결되어서 한줄로 표시됩니다."

41

<p>

- 단락(Paragraphs)이란 하나하나의 짧은 이야기 토막
- 단락의 전후에 빈 줄이 추가된다.



The screenshot shows a web browser window with the URL `E:\#document#bool`. The page content is:

```
<!DOCTYPE html>
<html>
<body>
<p>이것이 하나의 단락입니다.</p>
<p>이것이 하나의 단락입니다.</p>
<p>이것이 하나의 단락입니다.</p>
</body>
</html>
```

A red box highlights the three paragraphs. The browser window title bar shows "E:\#document#bool" and the status bar shows "E:\#document#bookw 월드...".

42

텍스트 입력시 주의할 점

- HTML 코드에서 엔터키를 눌러서 줄을 바꾸었다고 해서 웹 브라우저에서 줄이 바뀌는 것은 아니다.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

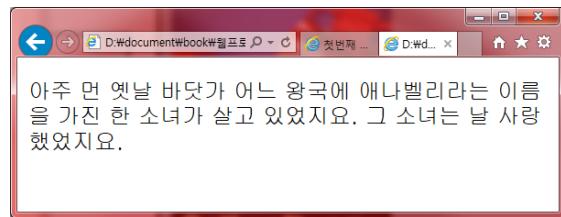
```
<p>
```

아주 먼 옛날 바닷가 어느 왕국에
애나벨리라는 이름을 가진
한 소녀가 살고 있었지요.
그 소녀는 날 사랑했었지요.

```
</p>
```

```
</body>
```

```
</html>
```



43

- 강제 줄 바꿈(line break) 태그

```
<!DOCTYPE html>
```

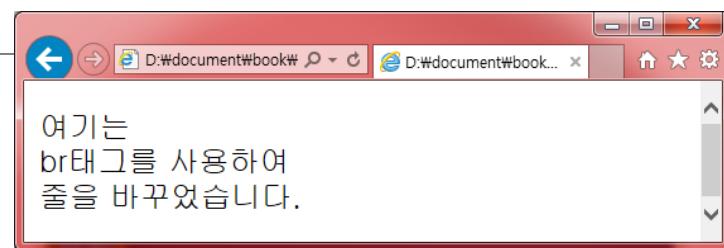
```
<html>
```

```
<body>
```

<p>여기는
br태그를 사용하여
줄을 바꾸었습니다.</p>

```
</body>
```

```
</html>
```



44

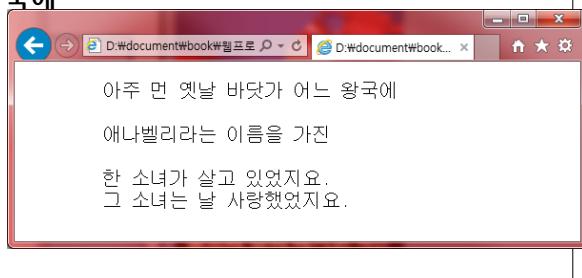
〈pre〉

- 프로그래머가 입력한 그대로 화면에 표시하는 태그
- previously formatted text의 약자

```
<!DOCTYPE html>
<html>
<body>
<pre>
아주 먼 옛날 바닷가 어느 왕국에

애나벨리라는 이름을 가진

한 소녀가 살고 있었지요.
그 소녀는 날 사랑했었지요.
</pre>
</body>
</html>
```

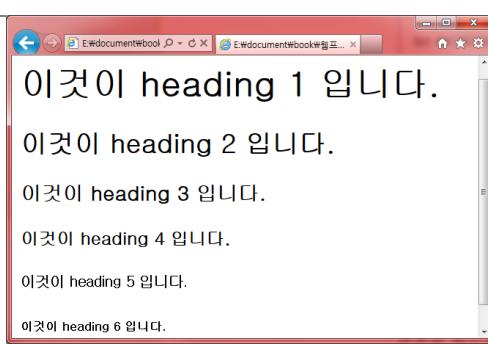


45

〈h1〉 ~ 〈h6〉

- 헤딩(heading): 웹 페이지의 머리기사(headline)

```
<!DOCTYPE html>
<html>
<body>
<h1>이것이 heading 1 입니다.</h1>
<h2>이것이 heading 2 입니다.</h2>
<h3>이것이 heading 3 입니다.</h3>
<h4>이것이 heading 4 입니다.</h4>
<h5>이것이 heading 5 입니다.</h5>
<h6>이것이 heading 6 입니다.</h6>
</body>
</html>
```



46

주석(comment)

- 주석(comment)은 코드를 설명하는 글

```
<!DOCTYPE html>
<html>
<body>
<!-- 참고: 아래 링크는 나의 배너임 -->

<!-- <input type="text" size="12"> -->
</body>
</html>
```

47

텍스트 서식

태그	설명
…	볼드체로 만든다.
<i>…</i>	이탤릭체로 만든다.
…	텍스트를 강하게 표시한다.
…	텍스트를 강조한다.
<code>…</code>	텍스트가 코드임을 표시한다.
[…]	위첨자(superscript)
_…	아래첨자(subscript)

48

텍스트 서식

```
<!DOCTYPE html>
<html>
<body>
<p><b>이 텍스트는 bold입니다.</b></p>
<p><strong>이 텍스트는 strong입니다.</strong></p>
<p><i>이 텍스트는 italic입니다.</i></p>
<p><em>이 텍스트는 emphasized입니다.</em></p>
<p><code>이 텍스트는 code입니다.</code></p>
<p><sub> subscript</sub>이고 <sup>superscript</sup>입니다.</p>
</body>
</html>
```

49

참고

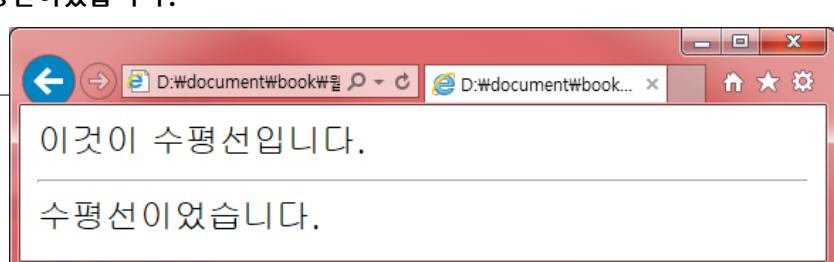
- HTML5 스펙에 따르면 와 <i> 태그는 다른 모든 태그가 적절하지 않는 경우에만 사용되어야 함
- 강조해야 하는 텍스트는 을 사용하는 편이 좋음
- 중요한 텍스트는 태그를 사용
- 하이라이트된 텍스트는 <mark> 태그를 사용
- 모든 텍스트 스타일은 CSS를 이용하는 것이 원칙
- 볼드 텍스트를 만들려면 CSS의 font-weight 속성을 사용

50

<hr>

- <hr> 태그를 사용하면 수평선을 그릴 수 있다.

```
<!DOCTYPE html>
<html>
<body>
    이것이 수평선입니다.
    <hr>
    수평선이었습니다.
</body>
</html>
```



51

특수문자

- ‘<’와 ‘>’ 같은 문자를 화면에 표시하기 위하여 필요하다.
- HTML은 여러 개의 공백이 이웃해 있더라도 하나의 공백으로 간주한다. 따라서 불가피하게 여러 개의 공백을 나타내고자 할 때는 를 여러 개 사용해야 한다.

특수 문자	설명
 	non-breaking space의 약자로 공백 문자 한 개
<	<
>	>
"	"
&	&

52

커피 전문점 웹 페이지



53

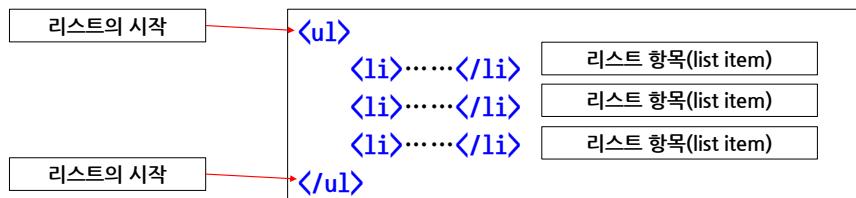
HTML 소스

```
<!DOCTYPE html>
<html>
<head>
    <title>Web Coffee</title>
</head>
<body>
    <h1>Web 커피 메뉴</h1>
    <h2>아메리카노 5000원</h2>
    <p>에스프레소에 물을 추가한 것입니다.</p>
    <h2>카페오레 6000원</h2>
    <p>에스프레소에 우유를 넣은 커피입니다.</p>
    <h2>카푸치노 6000원</h2>
    <p>커피 위에 우유거품을 얹은 커피입니다.</p>
</body>
</html>
```

54

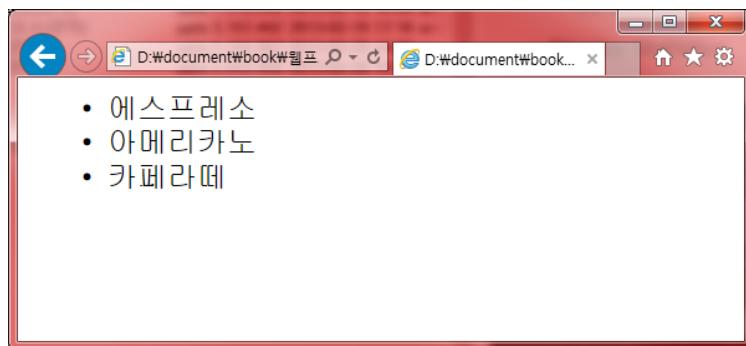
리스트

- 리스트: 항목들을 나열하는데 사용
 - 번호없는 리스트(unordered list) :
 - 번호있는 리스트(ordered list) :
 - 정의 리스트(definition list) : <dl>
 - 리스트 항목 안에도 텍스트, 이미지, 링크, 다른 리스트를 넣을 수 있다.



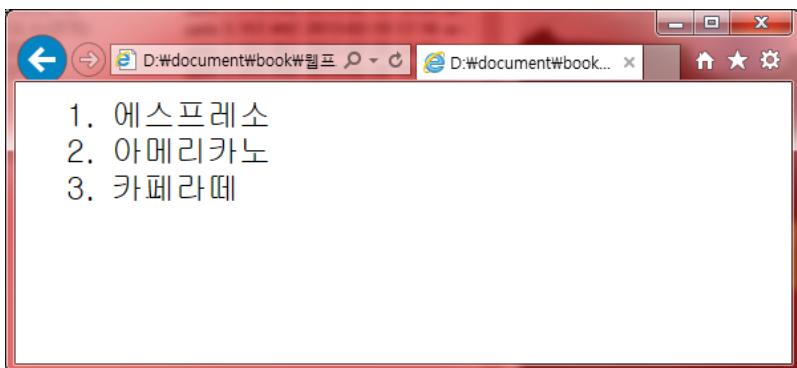
55


```
<ul>
  <li>에스프레소</li>
  <li>아메리카노</li>
  <li>카페라떼</li>
</ul>
```



56

```
<ol>
<li>에스프레소</li>
<li>아메리카노</li>
<li>카페라떼</li>
</ol>
```

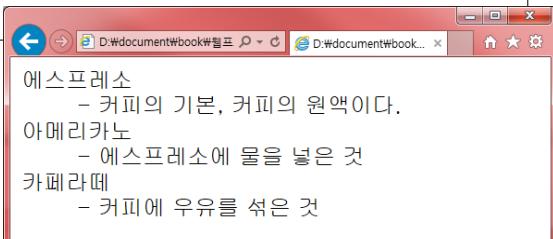


57

<dl>

- 정의 리스트(definition list): 항목들과 함께 항목들의 정의(설명)가 표시되는 리스트

```
<dl>
<dt>에스프레소</dt>
<dd>- 커피의 기본, 커피의 원액이다.</dd>
<dt>아메리카노</dt>
<dd>- 에스프레소에 물을 넣은 것</dd>
<dt>카페라떼</dt>
<dd>- 커피에 우유를 섞은 것</dd>
</dl>
```



58

⟨a⟩

- 하이퍼링크(또는 링크)는 다른 문서로 점프할 수 있는 단어나 이미지

⟨a⟩ 링크는 다른 페이지로 링크를 생성할 때 사용한다.

링크 텍스트이다. 클릭할 수 있다는 것을 나타내기 위하여 화면에서 밑줄이 그려지고 마우스 커서도 손모양으로 바뀐다.

```
<a href="http://www.google.com" target="_blank">구글 방문</a>
```

href 속성은 링크의 목적지를 나타낸다.

59

링크 예제

```
<!DOCTYPE html>
<html>
<body>
    <a href="http://www.google.com" target="_blank">
        google.com 방문</a>
    <p>
        target 속성이 "_blank"이므로 , 링크는 새로운 탭에서 열립니다.
    </p>
</body>
</html>
```

60

target 속성

- target 속성은 각 링크가 클릭되었을 때, 새로운 페이지가 어디에 열리는지를 지정

target	설명
_blank	새로운 윈도우에서 새로운 페이지를 연다.
_self	현재 윈도우에 새로운 페이지를 적재한다.
_parent	부모 프레임에 새로운 페이지를 적재한다.
_top	현재 윈도우에 새로운 페이지를 적재하고 모든 프레임을 취소 한다.

61

예제

```
<!DOCTYPE html>
<html>
<body>
    <a href="http://www.google.com" target="_self">
        google.com 방문1</a>
    <br>
    <a href="http://www.google.com" target="_blank">
        google.com 방문2</a>
</body>
</html>
```

62

id 속성

```
<!DOCTYPE html>
<html>
<body>
  <a href="#section1">참고 사항으로 가려면 여기를 클릭하세요.</a>
  <p> Hello World!</p>
  <p> Hello World!</p>
  <p> Hello World!</p>
  <br><br><br><br><br><br><br><br>
  <br><br><br><br><br><br><br><br><br>
  <hr>
  <a id="section1">참고 사항</a>
  <hr>
  <p> 동일한 페이지 안에서도 점프할 수 있습니다. </p>
</body>
</html>
```

63

이메일 링크

- 이메일 링크는 메일을 보내는 응용프로그램을 실행시켜준다.
- 만약 실행하는 컴퓨터에 메일을 보내는 응용프로그램이 없다면 응용프로그램(예를들어 아웃룩 등)

```
<a href="mailto:hong@gmail.com?subject=Feedback">hong@gmail.com</a>
```

64

- 이미지 : 태그 사용

 태그는 이미지를 웹페이지에 삽입할 때 사용한다.

width, height 속성은 각각 이미지의 가로, 세로 크기를 나타낸다.

src 속성은 이미지 파일 이름을 지정한다.

65

예제

```
<!DOCTYPE html>
<html>
<body>
    <h2>설악산</h2>
    
</body>
</html>
```

66

width와 height 속성

이미지의 이름 width, height 속성은 각각 이미지의 가로, 세로 크기를 나타낸다.

```

```

67

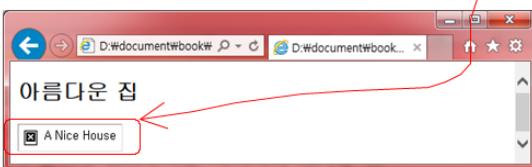
alt 속성

- 브라우저가 어떤 이유로 이미지를 화면에 표시하지 못했을 경우에, 표시되는 대체 텍스트(alternate text)를 지정

이미지가 표시되지 못하는 경우에 화면에 표시되는 텍스트

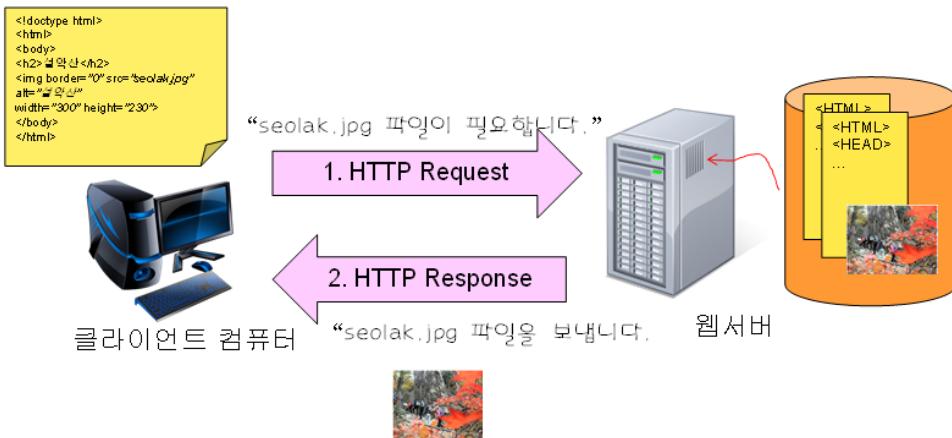
```

```



68

이미지 처리 방법



69

이미지의 종류

- JPEG(JPG)
 - 실사사진과 같이 복잡하고 많은 색상으로 이루어진 이미지에 적합
 - 손실 압축 방식을 사용한다.
- PNG
 - 클립 아트와 같이 적은 수의 색상을 가진 이미지에 적합
 - 무손실 압축 방식
- GIF
 - 로고나 클립아트 형태의 이미지에 적합
 - 256 색상만을 지원
 - 투명 배경과 애니메이션을 지원한다.

70

연습

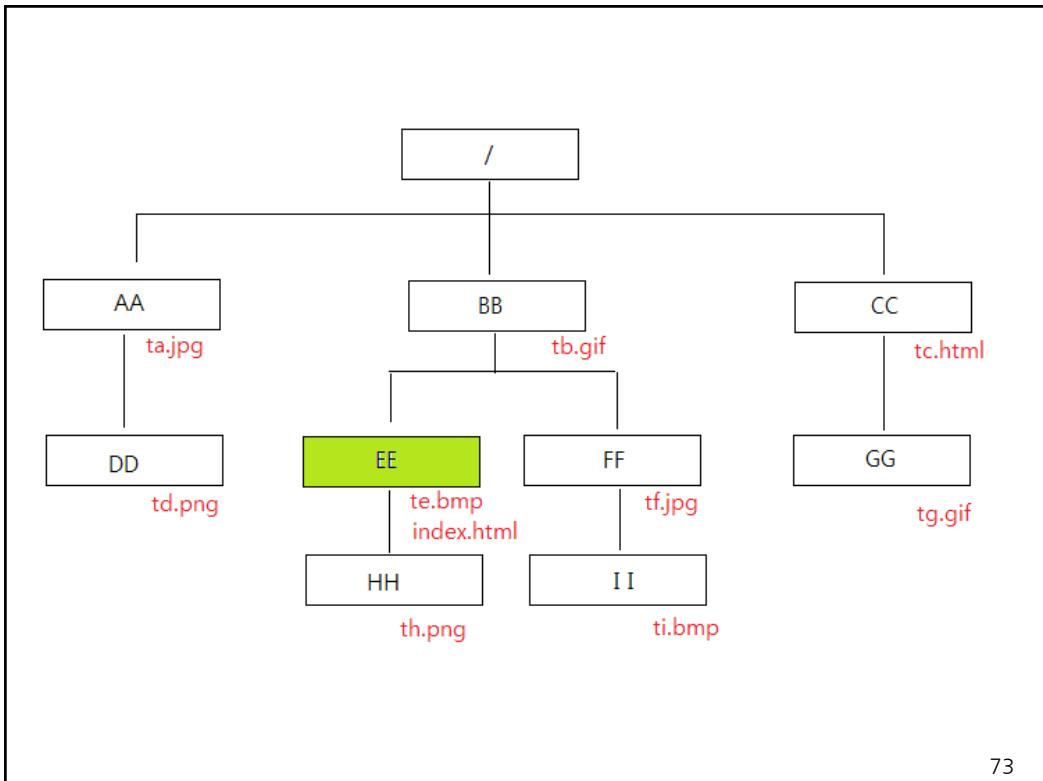


71

패스(경로)지정

- 패스는 디렉토리(폴더)간의 연결 경로를 나타내는 말
- 절대경로와 상대경로 2가지가 있다.
- 절대경로 : 루트를 기준으로 경로를 나타낸다.
 - /로 시작하는 경로로서 루트를 나타낸다.
- 상대경로 : 현재 폴더를 기준으로 경로를 나타낸다.
 - ./로 시작하는 경로로서 생략이 가능하다.
- 현재 폴더 -> ./
- 상위(부모) 폴더 -> ../
- 하위(자식) 폴더 -> 폴더명
- 최상위(루트) 폴더 -> /

72



73

경로 예제

- FF폴더에 있는 tf.jpg파일에 대한 경로
 - 절대경로 -> /BB/FF/tf.jpg
 - 상대경로 -> ../../FF/tf.jpg -> ..//FF/tf.jpg

- HH폴더에 있는 th.png파일에 대한 경로
 - 절대경로 -> /BB/EE/HH/th.png
 - 상대경로 -> ./HH/th.png -> HH/th.png

74

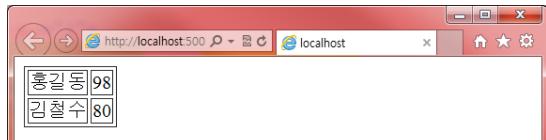
〈table〉

- 표 형태의 데이터를 표시하는 데 사용됨
- 초기의 웹 페이지에서는 전체 페이지의 레이아웃에 사용하였음
- 하나의 행을 <tr> … </tr>로 표현(table row)
- 하나의 데이터는 <td> … </td>로 표현(table data)
- 각 열의 헤더(열의 제목)이 있는 경우 <tr>…</tr>로 행을 만들고 그 안에 <th> … </th>를 사용하여 생성(table header)
- 테이블은 일반 글자 또는 데이터를 표로 표현할 때만 사용하고 화면의 레이아웃은 스타일시트를 이용해서 표현하도록 해야 함

75

〈table〉

```
<table border="1">
  <tr>
    <td>홍길동</td>
    <td>98</td>
  </tr>
  <tr>
    <td>김철수</td>
    <td>80</td>
  </tr>
</table>
```



76

테이블 헤더

```
<table border="1">
  <tr>
    <th>영화제목</th>
    <th>연도</th>
    <th>감독</th>
    <th>평가</th>
  </tr>
  <tr>
    <td>라이프 오브 파이</td>
    <td>2013</td>
    <td>이안</td>
    <td>8.68</td>
  </tr>
...
</table>
```

영화제목	연도	감독	평가
라이프 오브 파이	2013	이안	8.68
레미제라ブル	2012	톰후퍼	8.28
장고: 봉노의 추적자	2012	타란티노	8.29

77

테이블 예제

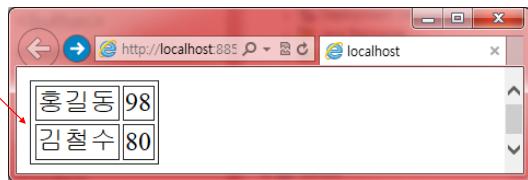
```
<table border="1">
  <tr>
    <td>100</td>
    <td>200</td>
    <td>300</td>
  </tr>
  <tr>
    <td>400</td>
    <td>500</td>
    <td>600</td>
  </tr>
  <tr>
    <td>700</td>
    <td>800</td>
    <td>900</td>
  </tr>
</table>
```

100	200	300
400	500	600
700	800	900

78

테이블 경계

```
<!DOCTYPE html>
<html>
<body>
    <table border="1">
        <tr>
            <td>홍길동</td>
            <td>98</td>
        </tr>
        <tr>
            <td>김철수</td>
            <td>80</td>
        </tr>
    </table>
</body>
</html>
```



79

참고

HTML 4.01의 경우 테이블의 속성으로 border 이외에도 bgcolor, align, cellspacing, frame, rules, cellpadding과 같은 많은 속성이 있었다. 하지만 이는 모두 HTML5에서는 권장하지 않는다. 권장하는 단 하나의 속성은 border이고 border도 값이 "1" 또는 ""만 사용하도록 권장하고 있다. border가 "1"이면 경계선이 있는 것이고 ""이면 경계선이 없다는 것을 의미한다. 테이블의 스타일을 지정하는 작업은 CSS를 사용해야 한다. HTML5에서 요소의 스타일을 지정하려면 반드시 CSS를 사용해야 한다는 것을 잊지 말자.

80

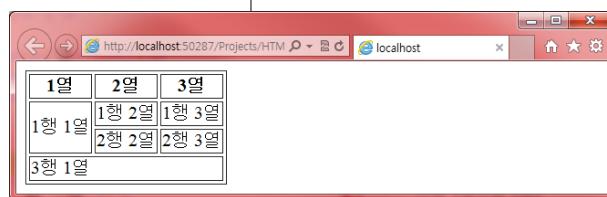
열과 행의 병합

- 행병합(row span) : 행이 병합되는 것 rowspan
- 열병합(column span) : 열이 병합되는 것 colspan
- rowspan을 2라고 지정하면 현재 셀 위치에서 2개의 행을 병합하겠다는 의미
- colspan을 3이라고 지정하면 현재 셀 위치에서 3개의 열을 병합하겠다는 의미

81

테이블 행 열 병합

```
<table border="1">
  <tr>
    <th>1열</th>
    <th>2열</th>
    <th>3열</th>
  </tr>
  <tr>
    <td rowspan="2">1행 1열</td>
    <td>1행 2열</td>
    <td>1행 3열</td>
  </tr>
  <tr>
    <td>2행 2열</td>
    <td>2행 3열</td>
  </tr>
  <tr>
    <td colspan="3">3행 1열</td>
  </tr>
</table>
```



82

테이블 캡션

```
<table border="1">
  <caption>
    최근에 본 영화들
  </caption>
  <tr>
    <th>영화제목</th>
    <th>연도</th>
    <th>감독</th>
    <th>평가</th>
  </tr>
  ...
</table>
```

영화제목	연도	감독	평가
라이프 오브 파이	2013	이안	8.68
레미제라블	2012	톰후퍼	8.28
창고: 분노의 추적자	2012	타란티노	8.29

83

테이블 연습 1

평생교육원 강좌 소개 -- 2015년도 1학기					
강좌 이름	강의자	소개	강좌 코드	실습비	
댄생과 유화	김철수	모든 미술영역의 뼈대를 이루는 댄생의 기본실력을 체계적으로 다루며, 유화에서는 재료를 다루는 방법에서부터 다양한 표현기법에 이르기까지 전문적인 미술인으로 입문할 수 있도록 각 개인의 수준에 맞추어 개인별, 단계별로 지도 합니다.	H27	100,000 원	
바이올린	홍길동	전반적인 현악기의 특성을 파악하며 기초를 다지고, 데워드와 곡을 접하면서 아름다운 음색의 바이올린을 연주할 수 있다.	H28	200,000 원	
성악아카데미	김호성	강의를 통해 배우고 익힌 곡을 음악회를 통하여 기량을 향상하고 무대를 경험 한다.	H30	300,000 원	

84

테이블 연습 2

A screenshot of a web browser window titled "테이블 예제1.html". The page displays a 4x4 grid of table cells. The first row has four equal-width columns. The second row has two columns on the left and two columns on the right. The third row has one column on the left and three columns on the right. The fourth row has four equal-width columns.

85

연습

A screenshot of a web browser window titled "나의 홈페이지". The page title is "나의 홈페이지". The content includes a header "나의 홈페이지", a large image of a person working at a computer in a server room, and a text block: "컴퓨터 프로그래머를 꿈꾸며 열심히 대덕인재개발원에서 공부하고 있는 김철수입니다." Below this is a section titled "현재 학습하고 있는 과목" with a list:

- HTML5 & Script [W3C사이트](#)
- JAVA
- 데이터베이스
- JSP

At the bottom, there is a section titled "시간표" (Schedule) with a table:

	3월	4월	5월	6월	7월	8월	9월
오전	HTML5 & Script	jQuery	고급자바	JSP	Framework	프로젝트	프로젝트
오후	기초자바	Oracle	고급자바	JSP	Framework	프로젝트	프로젝트

86

참고

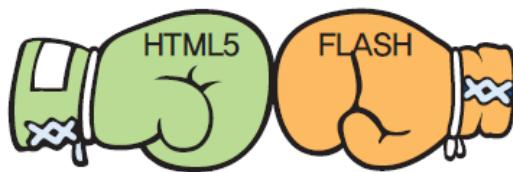
- HTML5에서는 이전에 사용되던 많은 태그가 더 이상 지원되지 않는다.
- 대표적인 것이 , <big>, <basefont>, <center>, <frame>, <frameset>, <noframes>, <strike>, <tt> 등이 있다.
- 이들은 물론 HTML 4.01을 지원하는 브라우저에서 사용할 수는 있겠으나 가급적이면 사용하지 않는 것이 좋다.
- HTML5에서는 태그로 요소의 스타일을 지정하면 안 된다.
- 반드시 CSS로 스타일을 지정해야 한다.

87

03 HTML 멀티미디어와 입력요소

웹브라우저와 멀티미디어

- 예전 방법: HTML 안에서는 <embed>나 <object> 태그를 사용하여야 했고 웹브라우저에는 플래시나 ActiveX를 설치
- HTML5: <audio>와 <video> 태그가 추가



89

<audio>

오디오 삽입 태그

자동 재생

```
<audio src="old_pop.mp3" autoplay controls>  
Your browser does not support the audio element.  
</audio>
```

오디오 소스 파일 경로(URL)

화면에 제어기를 보일 것



90

〈audio〉 요소의 속성

속성	설명
autoplay	이 속성이 존재하면 음악을 자동적으로 재생한다.
controls	이 속성이 존재하면 브라우저가 오디오 재생을 제어하는 제어기를 표시한다.
loop	이 속성이 존재하면 브라우저가 오디오를 반복하여 재생한다.
preload	사용자가 사용할 생각이 없더라도 오디오를 미리 다운로드 한다.
src	재생할 오디오가 존재하는 URL을 지정한다.
volume	오디오의 재생 볼륨을 설정한다.(0.0부터 1.0까지)

91

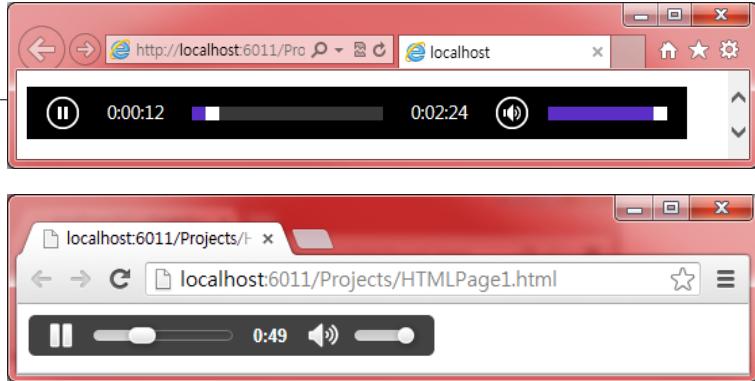
오디오 파일 형식

- MP3 - 'MPEG-1 Audio Layer-3'의 약자로 MPEG기술의 음성 압축 기술
- Wav - 윈도우에서 사용되는 표준 사운드 포맷
- Ogg - 특허권을 반대하고 보다 좋은 음질을 위하여 오픈소스로 개발되었음.

92

오디오 예제

```
<!DOCTYPE html>
<html>
<body>
    <audio src="old_pop.mp3" autoplay controls>
        Your browser does not support the audio element.
    </audio>
</body>
</html>
```



93

<source> 사용

- 모든 브라우저가 지원하는 오디오 형식은 아직까지 없다!
- 호환성을 높이기 위하여 다음과 같이 한다.

```
<!DOCTYPE html>
<html>
<body>
    <audio controls autoplay>
        <source src="old_pop.ogg" type="audio/ogg">
        <source src="old_pop.mp3" type="audio/mp3">
    Your browser does not support the audio element.
</audio>
</body>
</html>
```

위에서부터 파일 형식을 차례대로 검사한다.

94

<video>

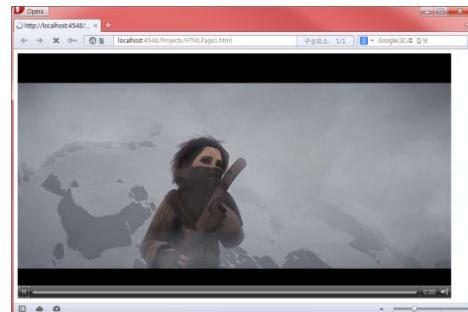
비디오 삽입 태그

자동 재생

```
<video src="movie.mp4" autoplay controls>  
Your browser does not support the audio element.  
</video>
```

비디오 소스 파일 경로(URL)

화면에 제어기를 보일 것



95

<video> 요소의 속성

속성	설명
autoplay	이 속성이 존재하면 비디오를 자동으로 재생한다.
controls	이 속성이 존재하면 브라우저가 비디오 재생을 제어하는 컨트롤을 표시한다.
loop	이 속성이 존재하면 브라우저가 비디오를 반복하여 재생한다.
poster	비디오를 다운로드 하는 중일 때 표시하는 이미지이다.
preload	사용자가 사용할 생각이 없더라도 전체 오디오를 다운로드 한다.
muted	비디오의 오디오 출력을 중지한다.
src	재생할 오디오가 존재하는 URL을 지정한다.
width, height	비디오 재생기의 너비와 높이를 나타낸다.

96

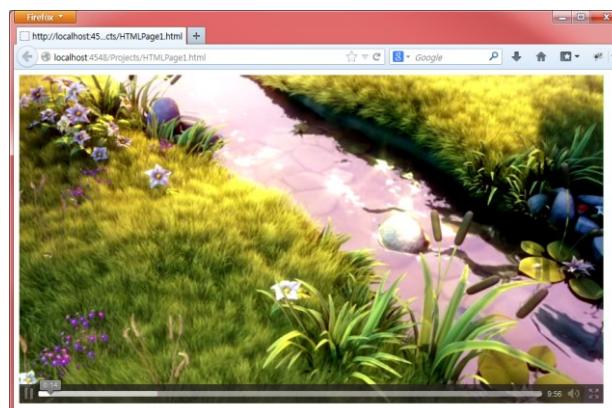
비디오 파일 형식

- MPEG4 - 'MPEG-4' 기술을 사용한다. MPEG-1과 MPEG-2에 비해 적은 용량으로도 고품질의 영상 및 음성을 구현할 수 있다. 코덱은 H.264를 사용한다.
- WebM - 무료로 제공되는 개방형 고화질 압축 형식의 영상 포맷이다. 구글이 지원하고 있다. 코덱은 VP8이라고 불린다.
- Ogg - 역시 무료이고 비디오 압축 형식이다. Ogg Theora 비디오 압축 기술이라 불린다. 확장자가 ogv인 파일에 주로 사용된다.

97

비디오 예제

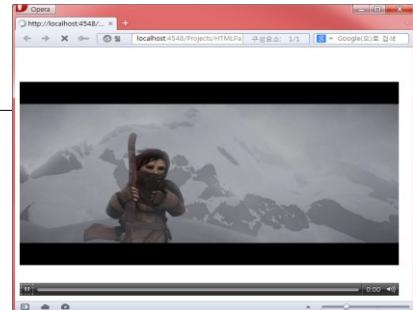
```
<video controls src="movie.ogv">  
Your user agent does not support the HTML5 Video element.  
</video>
```



98

비디오 예제

```
<!DOCTYPE html>
<html>
<body>
    <video width="640" height="480" controls>
        <source src="trailer.mp4" type='video/mp4'>
        <source src="trailer.ogv" type='video/ogg'>
        <p>Your user agent does not support HTML5.</p>
    </video>
</body>
</html>
```



위에서부터 파일 형식을 차례대로 검사한다.

99

<iframe>

- 웹 페이지 안에서 다른 웹 페이지를 표시하고자 할 때 사용
- inline frame
- 익스플로러가 페이지 안에 프레임을 놓기 위해 사용하던 태그였고 w3c는 iframe을 HTML 4.01부터 도입하여 현재는 거의 모든 부라우저가 iframe을 지원
- seamless 속성 : 경계선이 없이 문서의 일부인 것처럼 화면에 그려짐
- 링크의 타겟 프레임으로 사용될 수 있음
- 링크의 타겟 속성은 iframe에서 지정된 이름을 참조

100

<iframe>

The screenshot shows a W3Schools.com "Try it yourself" editor. On the left, the "Source Code" pane contains the following HTML:

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

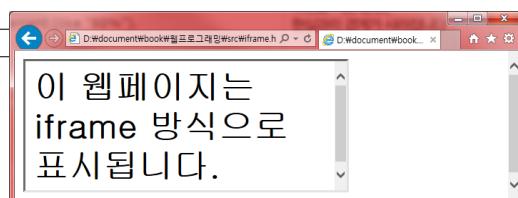
In the center, the "Result" pane displays the output: "My First Heading" followed by "My first paragraph.". A red rounded rectangle highlights the entire result area. A red arrow points from the right side of this highlight to the text "이것이 바로 iframe에 의해서 만들어진 윈도우이다." (This is the window created by the iframe). Below the result area, a green bar says "Edit the code above and click "Submit Code" to see the result." and "(그림 출처: www.w3schools.com)".

101

예제

```
<!DOCTYPE html>
<html>
<body>
<iframe src="inner.html" width="300" height="120"></iframe>
</body>
</html>
```

```
<html>
<head>
<title>INNER</title>
</head>
<body>
<h1>이 웹페이지는 iframe 방식으로 표시됩니다.</h1>
</body>
</html>
```

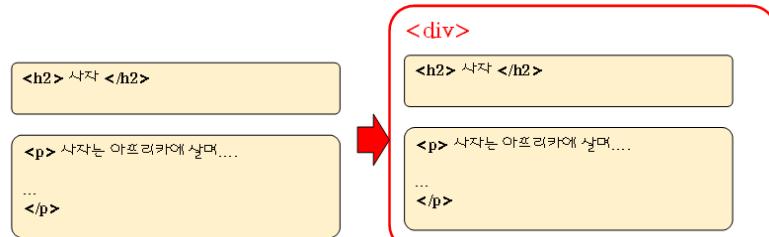


102

<div>와

- <div>은 "divide"의 약자로서 페이지를 논리적인 섹션으로 분리하는데 사용되는 태그

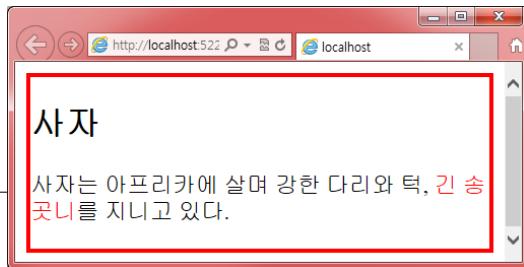
```
<div style="border: 3px solid red">
  <h2>사자</h2>
  <p>사자는 아프리카에 살며 ...</p>
</div>
```



103

예제

```
<!DOCTYPE html>
<html>
<body>
  <div style="border: 3px solid red;">
    <h2>사자</h2>
    <p>
      사자는 아프리카에 살며
      강한 다리와 턱,
      <span style="color: red;">긴 송곳니</span>를
      지니고 있다.
    </p>
  </div>
</body>
</html>
```



104

예제

```
<!DOCTYPE html>
<html>
<body>
  <div style="height:20px; background-color:yellow;"></div>
  <div style="height:20px; background-color:green;"></div>
  <div style="height:20px; background-color:purple;"></div>
</body>
</html>
```



105

<div>와

- HTML 요소는 <div>와 을 이용해 뮤을 수 있음
- <div>는 자체적으로 특별한 의미가 없으며 블록 수준의 요소로서 모든 HTML 요소를 뮤는데 사용함
- <div>는 블록 수준의 요소이기 때문에 하나의 줄을 전부 차지함
- 주로 웹 페이지의 레이아웃을 작성하는데 사용한다.
- 은 자체적으로 특별한 의미가 없으며 인라인 요소로서 텍스트를 뮤어 스타일을 적용할 때 사용함
- 인라인 요소는 자신이 필요한 크기만 차지하는 요소임

106

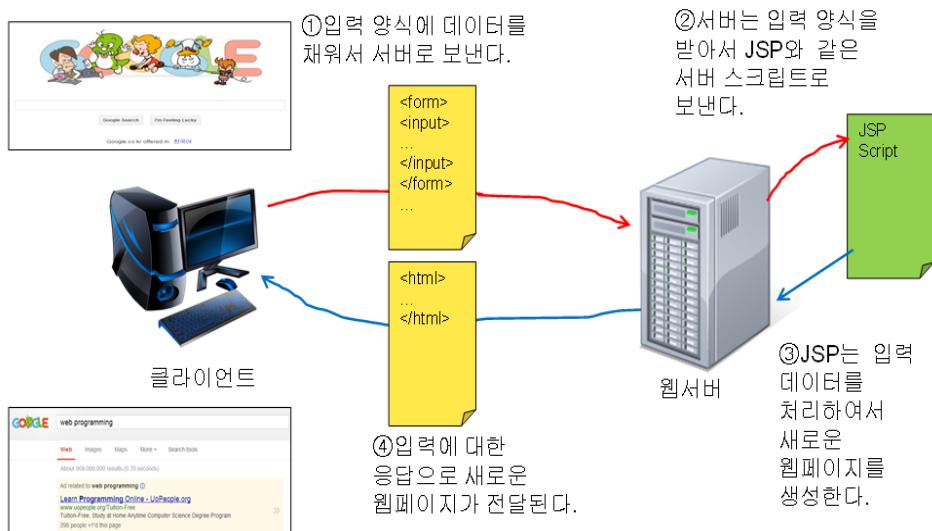
HTML 입력양식

- HTML 문서는 방식에 따라 서버에서 사용자에게 일방적으로 보여주는 방식과 사용자가 서버에 데이터를 보내는 두 가지 방식으로 분류할 수 있음
- 입력양식(form)을 이용하여 서버로 데이터를 전달한다.

The image shows two examples of HTML forms. The left form is a simple contact form with fields for Name, Email, and Place, and a Submit button. The right form is a more complex registration form titled 'Request Information' with fields for First Name, Last Name, Login Email, Password, and Confirm Password, and a Request Information button.

107

입력 양식의 작동 방식



108

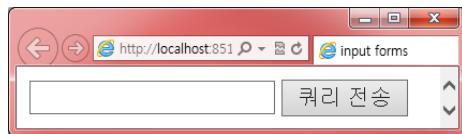
<form>

입력 양식은 항상 <form>으로 시작한다.

입력 데이터가 서버로 보내지는 방법을 기술한다. GET과 POST방식이 있다.

```
<form action="input.jsp" method="post">
  <input type="text" name="input">
  <input type="submit">
</form>
```

여기에 입력을 처리하는 서버스크립트의 주소를 적어준다.

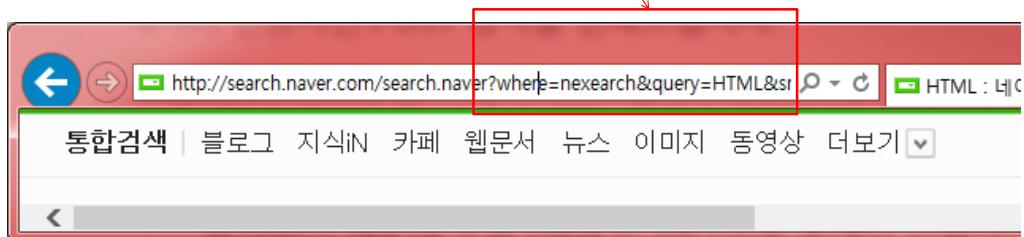


109

GET 방식과 POST 방식

• GET 방식

- GET 방식은 URL 주소 뒤에 파라미터를 붙여서 데이터를 전달하는 방식이다.



110

GET 방식과 POST 방식

- **POST 방식**

- POST 방식은 사용자가 입력한 데이터를 URL 주소에 붙이지 않고 HTTP Request 헤더에 포함시켜서 전송하는 방식
- 길이 제한이 없으며, 보안이 유지된다.

```
POST /test/input.jsp HTTP/1.1
Host: www.naver.com
name1=value1&name2=value2
...
```

111

입력 태그

```
<!DOCTYPE html>
<html>
<body>
<form action="input.jsp" method="post">
    이메일 : <input type="email" name="email"><br>
    URL : <input type="url" name="url"><br>
    전화번호 : <input type="tel" name="tel"><br>
    색상 : <input type="color" name="color"><br>
    월 : <input type="month" name="month"><br>
    날짜 : <input type="date" name="date"><br>
    주 : <input type="week" name="week"><br>
    시간 : <input type="time" name="time"><br>
    지역 시간 : <input type="datetime-local" name="localdatetime"><br>
    숫자 : <input type="number" name="number" min="1" max="10" step="2"><br>
    범위 : <input type="range" name="range" min="1" max="10" step="2"><br>
    <input type="submit" value="제출">
</form>
</body>
</html>
```

112

실행결과

이메일 : hong@google.com

URL : http://www.w3c.org

전화번호 : 010-1111-2222

색상 :

월 : 2013년 08월

날짜 : 2013-08-06

주 : 2013년 08월

시간 : 오전 01:00

지역 시간 : 2013-08-06 -- --:--

숫자 :

범위 :

113

<input> 형식

type 속성은 입력 필드의 종류를 결정한다.

name 속성은 서버로 전달되는 이름이다.(매우 중요)

<input type="button" value="눌러보세요" name="button1">

value 속성은 버튼에 나타내는 텍스트이다.

114

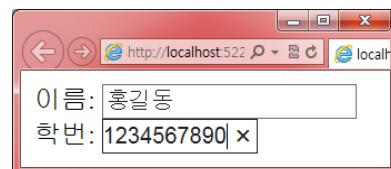
type 속성값

type 속성값	설명
text	텍스트를 입력할 수 있는 한 줄짜리 필드 생성
password	비밀번호를 입력할 수 있는 한 줄짜리 필드 생성
radio	라디오 버튼 생성
checkbox	체크 박스 생성
file	파일 이름을 입력하는 필드 생성
reset	초기화 버튼 생성, 버튼을 누르면 모든 입력 필드가 초기화된다.
image	이미지를 전송 버튼으로 만든다.
hidden	사용자에게는 보이지 않지만 서버로 전송된다.
submit	제출 버튼 생성

115

텍스트 필드

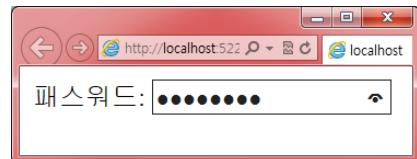
```
<form>
    이름: <input type="text" name="name"><br>
    학번: <input type="text" name="number" size="10">
</form>
```



116

패스워드

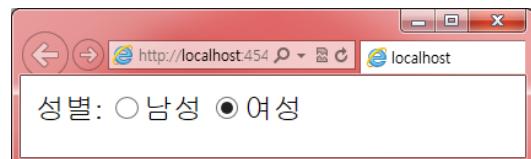
```
<form>
  패스워드: <input type="password" name="pass">
</form>
```



117

라디오 버튼

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



118

체크박스

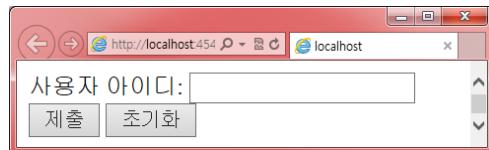
```
<form>
    파일 선택:
    <input type="checkbox" name="fruits" value="apple" checked>Apple
    <input type="checkbox" name="fruits" value="grape">Grape
    <input type="checkbox" name="fruits" value="orange">Orange
</form>
```



119

제출 버튼과 초기화 버튼

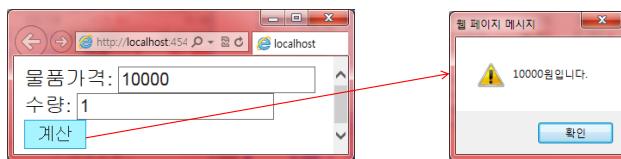
```
<form name="input" action="getid.jsp" method="get">
    사용자 아이디:
    <input type="text" name="user"><br>
    <input type="submit" value="제출">
    <input type="reset" value="초기화">
</form>
```



120

<input> 버튼

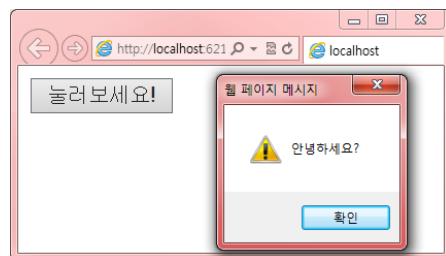
```
<form name="input" action="getid.jsp" method="get">
    물품가격:
    <input type="text" name="user"><br>
    수량:
    <input type="text" name="user"><br>
    <input type="button" value="계산" onclick="alert('10000원입니다.')">
</form>
```



121

<button> 버튼

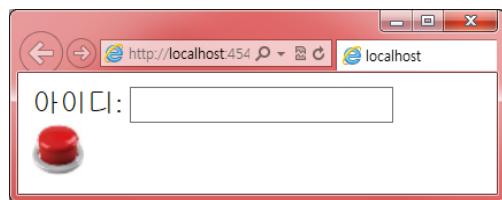
```
<button type="button" onclick="alert('안녕하세요?')>눌러보세요!</button>
```



122

이미지 버튼

```
<form name="input" action="getid.jsp" method="get">  
아이디:  
<input type="text" name="name"><br>  
<input type="image" src="submit.png" alt="제출 버튼">  
</form>
```



123

<textarea>

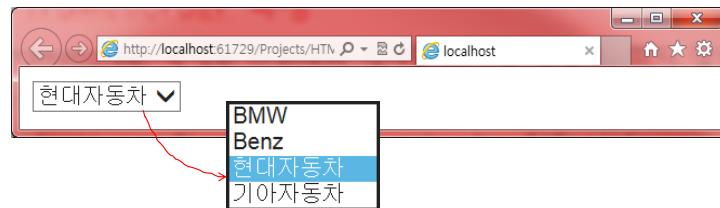
```
<form name="input" action="getfeedback.jsp" method="get">  
고객의 의견<br>  
<textarea name="feedback" rows="5" cols="50"></textarea>  
</form>
```



124

<select>

```
<form action="">
<select name="cars">
    <option value="bmw">BMW</option>
    <option value="benz">Benz</option>
    <option value="hyundai" selected>현대자동차</option>
    <option value="kia">기아자동차</option>
</select>
</form>
```

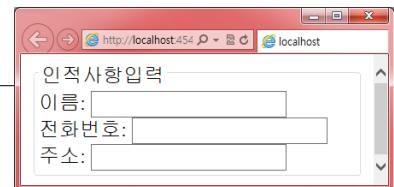


125

<fieldset>

- 입력 요소를 그룹핑하는 데 사용되는 태그
- 그룹의 경계에 선을 그려준다.
- <legend>를 사용하면 그룹에 제목을 붙일 수 있음

```
<form>
    <fieldset>
        <legend>인적사항입력</legend>
        이름: <input type="text"><br>
        전화번호: <input type="text"><br>
        주소: <input type="text">
    </fieldset>
</form>
```

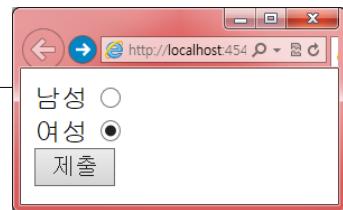


126

<label>

- <input>요소를 위한 레이블(label)을 정의함
- <label> 태그의 속성 for를 사용하면 레이블과 <input>의 id 속성을 통해 서로 연결할 수 있음

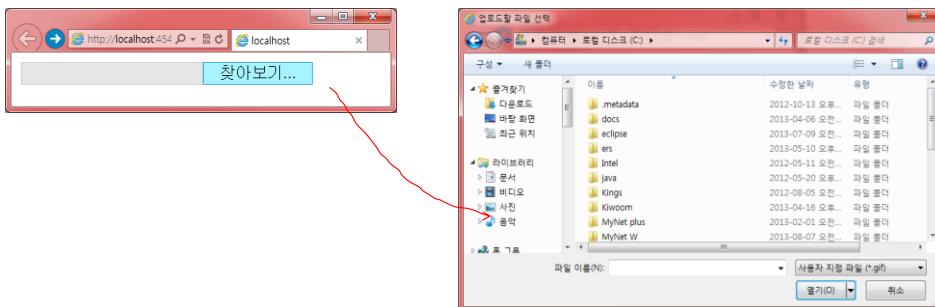
```
<form action="proc_form.jsp">
    <label for="male">남성</label>
    <input type="radio" name="gender" id="male" value="male"><br>
    <label for="female">여성</label>
    <input type="radio" name="gender" id="female" value="female"><br>
    <input type="submit" value="제출">
</form>
```



127

파일 업로드 버튼

```
<form enctype="multipart/form-data">
    <input type="file" accept="image/jpg,image/gif">
</form>
```



128

hidden

- <input type="hidden" name="" value="">
- 사용자가 직접 입력하는 데이터는 아니지만 클라이언트 컴퓨터가 서버 컴퓨터로 특정한 데이터를 전송하고 싶은 경우 많이 사용
- 화면에는 아무것도 나타나지 않고 사용자가 "제출" 버튼을 누를 때 서버로 name과 value가 전송됨

129

HTML 입력 요소

추가된 <input> type	설명
date	날짜를 입력할 수 있는 컨트롤
datetime	UTC 날짜/시각 형식을 이용한 날짜와 시각 표시 컨트롤
datetime-local	현지 날짜/시각
month	월/연도
time	시각
week	주와 연도를 선택할 수 있는 컨트롤
color	색상 코드를 입력할 수 있는 컨트롤
email	표준 이메일 주소를 입력 받아서 검증하는 컨트롤
tel	전화번호를 입력 받아서 검증하는 컨트롤
search	검색어 입력 양식을 생성
range	2개의 숫자 사이의 숫자를 선택할 수 있는 슬라이더 컨트롤
number	숫자만 입력 받는 컨트롤
url	URL만 입력 받는 컨트롤

130

추가된 <input> 의 속성

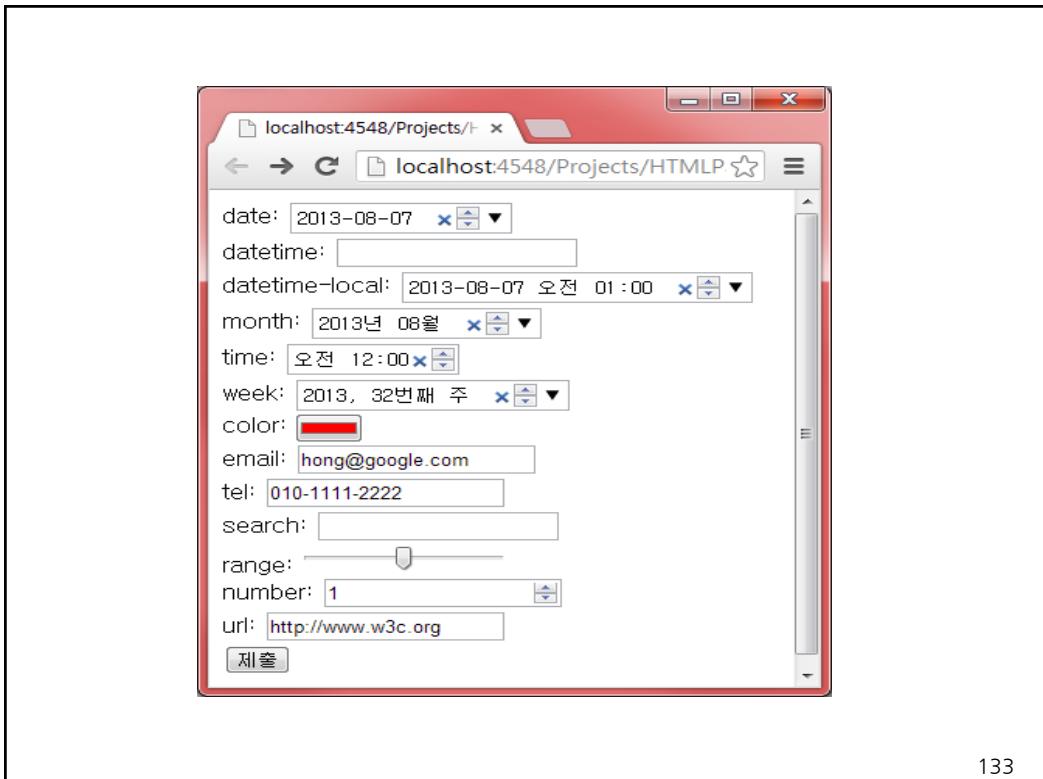
- autocomplete - 자동으로 입력을 완성한다.
- autofocus - 페이지가 로드 되면 자동으로 입력 포커스를 갖는다.
- placeholder - 입력 힌트를 희미하게 보여준다.
- readonly - 읽기 전용 필드
- required - 입력 양식을 제출하기 전에 반드시 채워져 있어야 함을 나타낸다.
- pattern - 허용하는 입력의 형태를 정규식으로 지정한다.

131

예제

```
<body>
  <form>
    date: <input type="date"><br>
    datetime: <input type="datetime"><br>
    datetime-local: <input type="datetime-local"><br>
    month: <input type="month"><br>
    time: <input type="time"><br>
    week: <input type="week"><br>
    color: <input type="color"><br>
    email: <input type="email"><br>
    tel: <input type="tel"><br>
    search: <input type="search"><br>
    range: <input type="range"><br>
    number: <input type="number"><br>
    url: <input type="url"><br>
    <input type="submit">
  </form>
</body>
```

132



133

이메일 입력

```
...
이메일: <input type="email" name="email" required><br>
...

```

A red arrow points from the 'required' attribute in the code above to the 'abc' text in the browser's input field below.

A screenshot of a web browser window titled 'localhost:621'. The page contains a single input field with the placeholder '이메일:' and the value 'abc'. A red box highlights the input field. Below the input field is a message box containing the text '올바른 전자 메일 주소를 입력해야 합니다.' (A valid email address must be entered).

134

전화번호 입력

```
<form>
    전화번호 : <input type="tel" name="tel" required
                  pattern="[0-9]{3}-[0-9]{3,4}-[0-9]{4}"
                  title="010-1234-1234"><br>
    <input type="submit" value="Send">
</form>
```

135

숫자 입력

```
...
신발사이즈
<input type="number" min="230" max="290" step="10" value="260" name="sho">
...
```

신발사이즈 260 □ 제출

구글

신발사이즈 □ 260 □ 보내기

오페라

136

range 입력

```
...  
테니스 스킬 <input type="range" min="1" max="10" value="1">  
...
```

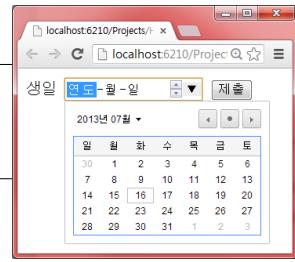


137

날짜 입력

- date - 날짜 입력
- month - 월 입력
- week - 주 입력
- time - 시간 입력
- datetime - 날짜와 시간을 입력할 수 있는 양식 제공, 국제 표준 시간대
- datetime-local - 날짜와 시간을 입력할 수 있는 양식 제공, 지역 표준 시간대

```
...  
생일 <input type="date" name="dob">  
...
```



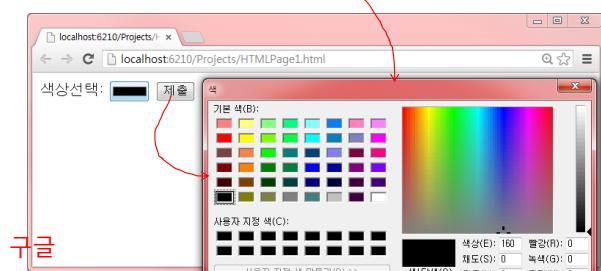
138

색상 입력

```
...  
색상선택: <input type="color" name="color">  
...
```



오페라



139

연습 1

A screenshot of a web browser window titled 'D:\Whhtml\#품연습2.html'. The page content is titled '이메일 전송화면' (Email Send Form). It contains three input fields: '이름:' with a text input box, '이메일 주소:' with a text input box, and '내용:' with a large text area. At the bottom are two buttons: '전송' (Send) and '재설정' (Reset).

140

연습 2

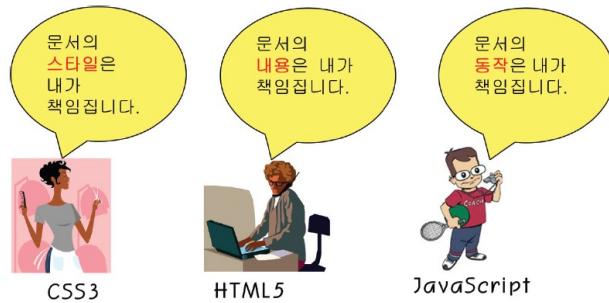
The screenshot shows a web browser window with a yellow header bar. The title bar reads '회원 가입 화면.html'. The main content area is titled '회원 가입 화면'. It contains three text input fields labeled '이름:', '주소:', and 'email:'. Below these are two radio buttons: 'Male' and 'Female'. At the bottom are two buttons: 'Send' and '원래대로'.

141

04 CSS 스타일시트 기초

CSS의 개념

- 문서의 구조-> HTML
- 문서의 스타일 -> ?



143

CSS의 역할

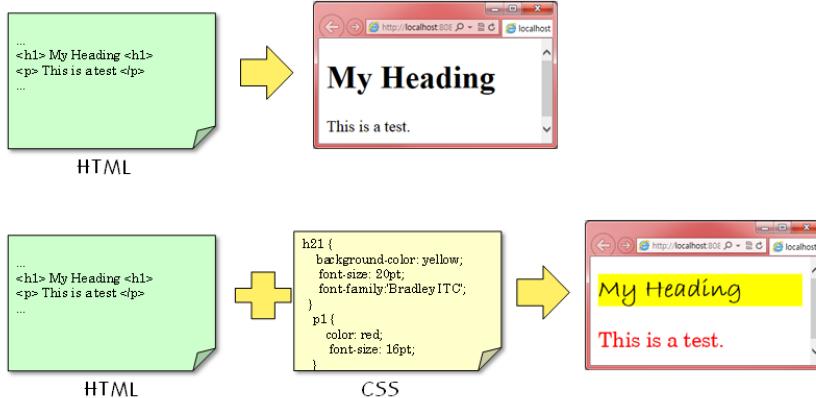
- 만약 CSS가 없다면



144

CSS

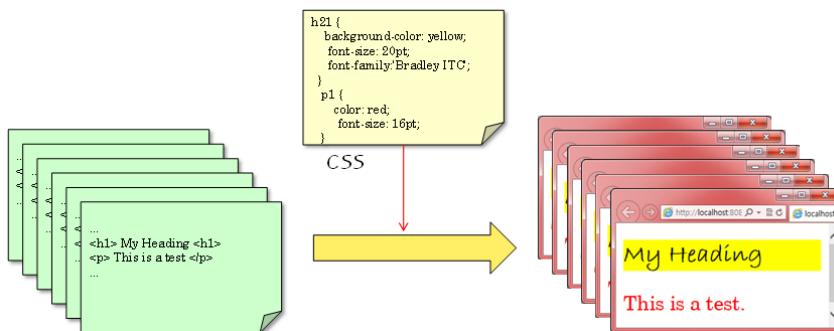
- CSS(Cascading Style Sheets): 문서의 스타일을 지정한다.



145

CSS의 장점

- 거대하고 복잡한 사이트를 관리할 때에 필요
- 모든 페이지들이 동일한 CSS를 공유
- CSS에서 어떤 요소의 스타일을 변경하면 관련되는 전체 페이지의 내용이 한꺼번에 변경



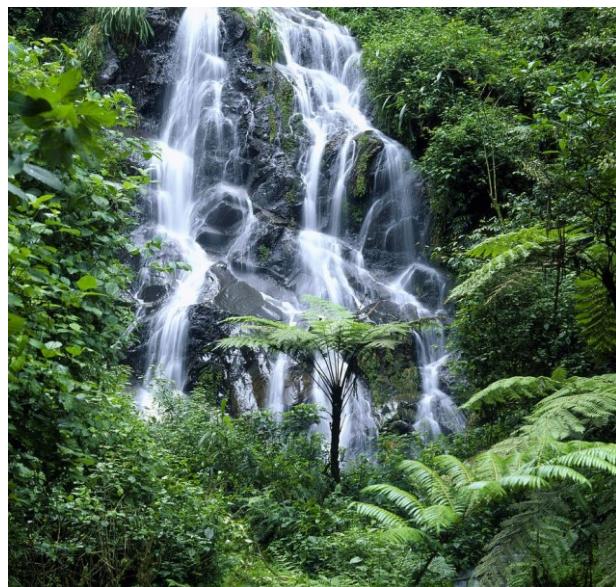
146

CSS3의 기능

- 선택자(selectors)
- 박스 모델(Box Model)
- 배경 및 경계선(Backgrounds and Borders)
- 텍스트 효과(Text Effects)
- 2차원 및 3차원 변환(2D/3D Transformations)
- 애니메이션(Animations)
- 다중 컬럼 레이아웃(Multiple Column Layout)
- 사용자 인터페이스(User Interface)

147

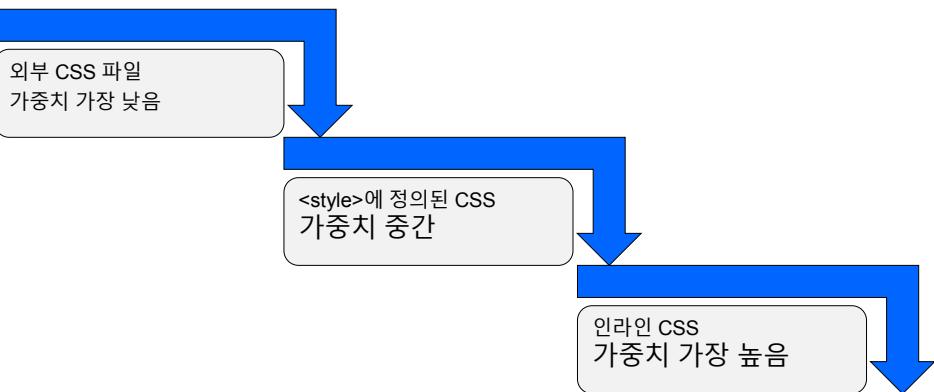
cascading



148

왜 cascading인가요?

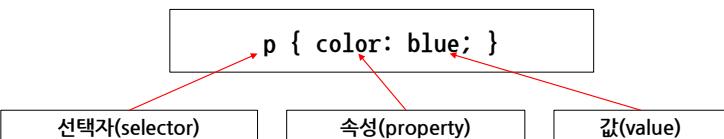
- 캐스케이딩은 폭포같은, 연속적인, 계속되는 이라는 뜻
- 연속되는 작은 폭포들처럼 위에서 아래로 순차적으로 적용됨
- 하나의 요소에 여러 개의 CSS가 충돌할 경우 우선 순위(가중치)가 계산되고 계산 결과에 따라 CSS 충돌이 처리된다.



149

CSS3의 문법

- 선택자(selector) { 속성: 값; }
- 끝에 반드시 ;을 적어 준다.

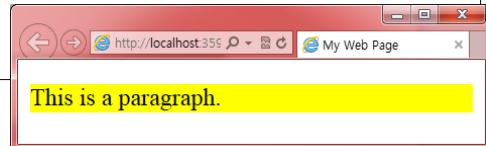


- 주석: /* ... */

150

CSS의 위치

```
<!doctype html>
<html>
<head>
    <title>My Web Page</title>
    <style>
        p { background-color: yellow; }
    </style>
</head>
<body>
    <p>This is a paragraph.</p>
</body>
</html>
```



151

예제

```
<!DOCTYPE html>
<html>
<head>
    <title>My Web Page</title>
    <style>
        h1 {
            background-color: yellow;
            border: 2px solid red;
        }
    </style>
</head>
<body>
    <h1>This is a heading.</h1>
</body>
</html>
```



152

선택자

- 선택자(selector): HTML 요소를 선택하는 부분



- 선택자는 jQuery에서도 사용
- 가장 많이 사용되는 것은 6가지 정도
- 선택자에 대한 W3C의 문서는 <http://www.w3.org/TR/css3-selectors/>

153

선택자의 종류

- 타입 선택자(type selector)
- 전체 선택자(universal selector)
- 클래스 선택자(class selector)
- 아이디 선택자(ID selector)
- 속성 선택자(attribute selector)
- 의사 선택자(pseudo-class)

154

타입 선택자

- 타입 선택자(type selector) : HTML 요소 이름을 사용

`h1 { color: green; }`

모든 h1 요소를 선택한다.

155

전체 선택자

- 전체 선택자(universal selector): 페이지 안의 모든 요소를 선택

`* { color: green; }`

전체 요소를 선택한다.

156

아이디 선택자

- 아이디 선택자(id selector): 특정한 요소를 선택

```
#target { color: red; }
```

id가 target인 요소를 선택한다.

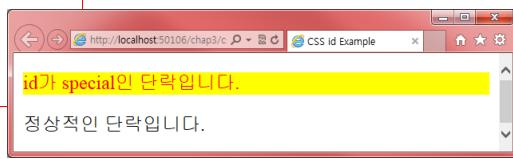
```
<p id="target">Hello World!</p>
```

<p>요소의 id를 "target"로 지정한다.

157

예제

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS id Example</title>
    <style>
        #special {
            background-color: yellow;
            color: red;
        }
    </style>
</head>
<body>
    <p id="special">id가 special인 단락입니다.</p>
    <p>정상적인 단락입니다.</p>
</body>
</html>
```



158

클래스 선택자

- 클래스 선택자(class selector)는 클래스가 부여된 요소를 선택한다.

```
.target { color: red; }
```

클래스가 target인 요소를 선택한다.

```
<p class="target">Hello World!</p>
```

<p>요소의 클래스를 "target"로 지정한다.

159

예제

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS class Example</title>
    <style>
        .type1 {
            text-align: center;
        }
    </style>
</head>
<body>
    <h1 class="type1">class가 type1인 헤딩입니다.</h1>
    <p class="type1">class가 type1인 단락입니다</p>
</body>
</html>
```



160

선택자 그룹

- 선택자를 콤마(,)로 분리하여 나열할 수 있다.

```
h1, h2, h3 { font-family: sans-serif; }
```

<h1>, <h2>, <h3>요소를 선택한다.

161

예제

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS selector Example</title>
    <style>
        h1, p {
            font-family: sans-serif;
            color: red;
        }
    </style>
</head>
<body>
    <h1>This is a heading1.</h1>
    <p>This is a paragraph.</p>
</body>
</html>
```



162

자손, 자식, 형제 결합자

선택자	설명
s1 s2	s1 요소에 포함된 s2 요소를 선택한다. (후손 관계)
s1 > s2	s1 요소의 직계 자식 요소인 s2를 선택한다.(자식 관계)

```
body em { color: red; } /* body 안의 em 요소 */  
body > h1 { color: blue; } /* body 안의 h1 요소 */
```

163

예제

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
    body em { color: red; } /* body 안의 em 요소 */  
    body > h1 { color: blue; } /* body 안의 h1 요소 */  
</style>  
</head>  
<body>  
    <h1>This headline is <em>very</em> important</h1>  
</body>  
</html>
```



164

의사 클래스(pseudo-class)

- 의사 클래스(pseudo-class): 클래스가 정의된 것처럼 간주

- a:link { color: blue; }
- a:visited { color: green; }
- a:hover { color: red; }

165

예제

```
a:link {  
    text-decoration: none;  
    color: blue;  
    background-color: white;  
}  
  
a:visited {  
    text-decoration: none;  
    color: green;  
    background-color: silver;  
}  
  
a:hover {  
    text-decoration: none;  
    color: white;  
    background-color: blue;  
}  
...
```

The diagram illustrates the effect of the three pseudo-classes on different links. Three arrows point from colored boxes to specific parts of the CSS code:

- A red arrow points from a red box containing the text "겨울 왕국" to the "a:link" rule.
- A green arrow points from a green box containing the text "눈의 여왕" to the "a:visited" rule.
- A blue arrow points from a blue box containing the text "라푼젤" to the "a:hover" rule.

166

속성 선택자

- 속성 선택자: 특정한 속성을 가지는 요소를 선택한다.
 - h1[title] { color: blue; }
 - p[class="example"] { color: blue; }
 - p.example {color: blue; }

167

CSS 삽입 위치

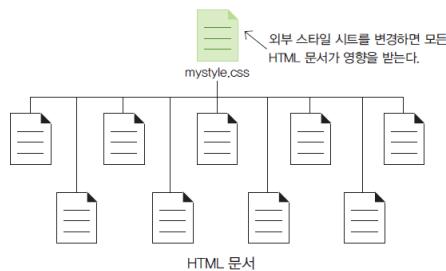
- 외부 스타일 시트(external style sheet)
- 내부 스타일 시트(internal style sheet)
- 인라인 스타일 시트.inline

168

외부 스타일 시트

- 외부 스타일 시트는 스타일 시트를 외부에 파일로 저장하는 것
- 많은 페이지에 동일한 스타일을 적용하려고 할 때 좋은 방법

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



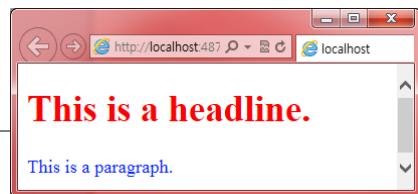
169

예제

mystyle.css

```
h1 { color: red; }  
p { color:#0026ff; }
```

```
<!DOCTYPE html>  
<html>  
<head>  
    <link type="text/css" rel="stylesheet" href="mystyle.css">  
</head>  
<body>  
    <h1>This is a headline.</h1>  
    <p>This is a paragraph.</p>  
</body>  
</html>
```



170

내부 스타일 시트

- 내부 스타일 시트는 HTML 안에 CSS를 정의하는 것이다.

```
<!DOCTYPE html>
<html>
<head>
<style>
    h1 { color: red; }
    p { color: #0026ff; }
</style>
</head>
<body>
    <h1>This is a headline.</h1>
    <p>This is a paragraph.</p>
</body>
</html>
```



171

인라인 스타일 시트

- 각각의 HTML 요소마다 스타일을 지정하는 것
- 2개 이상의 선언이 있다면 반드시 끝에 ;을 적어 준다.

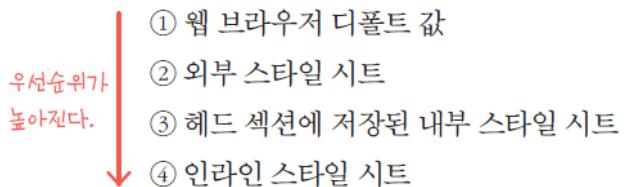
```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
    <h1 style="color: red">This is a headline.</h1>
    <p style="color: #0026ff">This is a paragraph.</p>
</body>
</html>
```



172

다중 스타일 시트

- 하나의 요소에 대하여 외부, 내부, 인라인 스타일이 서로 다르게 지정하고 있다면 어떤 스타일이 사용될까?



- 공통적으로 사용되는 스타일은 <body>요소의 스타일에 정의하는 것이 편리하다.

173

예제

coffee.css

```
h1, p {  
    font-family: serif;  
    color: black;  
}  
h1 {  
    border-bottom: 1px solid gray;  
    color: red;  
}  
body {  
    background-color: yellow;  
}
```

174

```

<!DOCTYPE html>
<html>
<head>
    <title>Web Programming</title>
    <link type="text/css" rel="stylesheet" href="coffee.css">
</head>
<body>
    <h1>Welcome to Web Coffee!</h1>
    
    <p>
        하우스 로스팅 원두의 신선한 커피를 맛보세요!
        <em>공인 1급 Barista</em>가
        최고급 원두만을 직접 엄선하여 사용합니다.
    </p>
    <h2>메뉴</h2>
    <p>
        아메리카노, 카페라떼, 카푸치노, 카페모카, ...
    </p>
</body>
</html>

```

175

CSS의 속성들

속성	설명
color	텍스트 색상
font-weight	볼드체 설정
padding	요소의 가장자리와 내용간의 간격
font-size	폰트의 크기
background-color	배경색
border	요소를 감싸는 경계선
font-style	이탤릭체 설정
background-image	배경 이미지
text-align	텍스트 정렬
list-style	리스트 스타일

176

색상

방법	설명
이름으로 표현	"red"
16진수로 표현	#FF0000
10진수로 표현	rgb(255, 0, 0)
퍼센트로 표현	rgb(100%, 0%, 0%)



177

16진수로 색상 나타내기

- 16진수 코드는 빨간색, 녹색, 청색 값을 각각 2자리의 16진수로 표시한 것

16진수 코드는 항상 #으로 시작한다.
f f 0 0 0 0
red green blue

빨간색 성분이 ff, 녹색 성분이 00, 청색 성분이 00이므로 이것을 합친 색상은 빨간색이다.

```
body
{
    background-color: #ffd800;
```

178

색상의 이름으로 나타내기



```
body
{
    background-color: aqua;
}
```

179

RGB 값으로 표시하기

```
body {
    background-color: rgb(60%, 40%, 10%);
}

body {
    background-color: rgb(153, 102, 25);
}
```

180

예제

```
<!DOCTYPE html>
<html>
<head>
    <style>
        h1 {background-color: #6495ed;}
        p.a {background-color: #ff0000;}
        p.b {background-color: #00ff00;}
        p.c {background-color: #0000ff;}
        p.d {background-color: #888888;}
    </style>
</head>
<body>
    <h1>CSS Color Chart</h1>
    <p class="a">Color #1</p>
    <p class="b">Color #2</p>
    <p class="c">Color #3</p>
    <p class="d">Color #4</p>
</body>
</html>
```



181

폰트

속성	설명
font	한줄에서 모든 폰트 속성을 설정할 때 사용
font-family	폰트 패밀리 설정
font-size	폰트의 크기 설정
font-style	폰트 스타일 설정
font-weight	폰트의 볼드체 여부 설정

182

폰트 지정

```
body {  
    font-family: "Times New Roman", Times, serif;  
}
```

폰트의 종류 속성

가장 선호하는 폰트

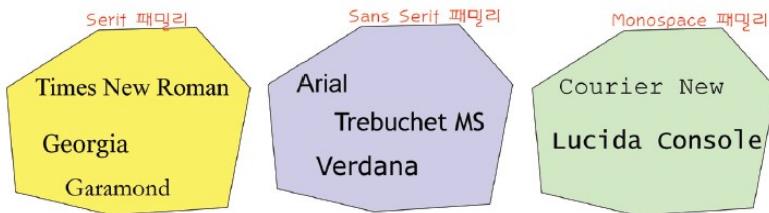
마지막에는 항상 일반
서인 폰트를 지정해야
한다.

Arial
Arial (Bold)
Arial (Italic)
Arial (Bold Italic)
Arial Black
Comic Sans MS
Comic Sans MS (Bold)
Courier New
Courier New (Bold)
Courier New (Italic)
Courier New (Bold Italic)
Georgia
Georgia (Bold)
Georgia (Italic)
Georgia (Bold Italic)
Impact
Times New Roman
Times New Roman (Bold)
Times New Roman (Italic)
Times New Roman (Bold Italic)

183

폰트 패밀리

- serif 폰트는 우아하고 전통적인 느낌
- sans-serif은 깔끔하고 가독성이 좋다.
- monospace는 타자기 서체
- cursive와 fantasy 폰트는 장난스러우며 스타일리쉬한 느낌



184

웹폰트

```
<html>
<head>
<title>Web Font Test</title>
<style>
@font-face {
    font-family: "Vera Serif Bold";
    src:
    url("http://developer.mozilla.org@api/deki/files/2934/=VeraSeBd.ttf");
}
body { font-family: "Vera Serif Bold", serif }
</style>
</head>
<body>
    이것이 모질라에서 제공하는 Vera Serif Bold입니다.
</body>
</html>
```

185

폰트 크기 설정

- 폰트의 단위
 - pt - 포인트
 - px - 픽셀
 - % - 퍼센트
 - em - 배수(scale factor)
 - 키워드 - xx-small, x-small, small, medium, large, x-large, xx-large

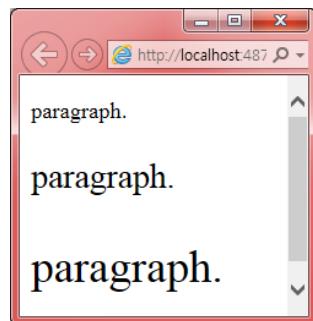
```
body {
    font-size: 20pt;
}
```

폰트 크기를 나타내는 속성 20포인트를 의미한다.

186

폰트 크기 예제

```
<!DOCTYPE html>
<html>
<head>
    <style>
        body {font-size: medium;}
        p#t1 {font-size: 1.0em;}
        p#t2 {font-size: 1.5em;}
        p#t3 {font-size: 2.0em;}
    </style>
</head>
<body>
    <p id="t1">paragraph.</p>
    <p id="t2">paragraph.</p>
    <p id="t3">paragraph.</p>
</body>
</html>
```



187

폰트 속성

- font-weight - 볼드체 여부(normal, bold)
- font-style - 이탤릭체 여부(normal, italic, oblique)

```
h1#s1 { font-weight: normal; }
h1#s2 { font-weight: bold; }
```

↑
볼드체로 설정

heading 1
heading 1

```
h1#s1 { font-style: normal; }
h1#s2 { font-style: italic; }
```

↑
이탤릭체로 설정

heading 1
heading 1

188

폰트 축약 기법

```
<!DOCTYPE html>
<html>
<head>
<style>
    p.style1 {font: italic 30px arial,sans-serif;}
    p.style2 {font: bold 40px Georgia,serif;}
</style>
</head>
<body>
    <p class="style1">font: italic 30px arial,sans-serif</p>
    <p class="style2">font: bold 40px Georgia,serif</p>
</body>
</html>
```



189

텍스트 스타일

속성	설명
color	텍스트의 색상을 지정한다.
direction	텍스트 작성 방향을 지정한다. (가로쓰기, 세로쓰기)
letter-spacing	글자간 간격을 지정한다.
line-height	텍스트 줄의 높이를 지정한다.
text-align	텍스트의 수평 정렬을 지정한다.
text-decoration	텍스트 장식을 지정한다.
text-indent	텍스트의 들여쓰기를 지정하낟.
text-shadow	그림자 효과를 지정한다.
text-transform	대소문자 변환을 지정한다.

190

텍스트 정렬

```
<!DOCTYPE html>
<html>
<head>
<style>
    h1 {text-align: center; color: red;} // 중앙정렬
    p.date {text-align: right; color: indigo;} // 오른쪽정렬
    p.poet {text-align: justify; color: blue;} // 양쪽정렬
</style>
</head>
<body>
    <h1>CSS 텍스트 정렬 예제</h1>
    <p class="date">2013년 9월 1일</p>
    <p class="poet">삶이 그대를 속일지라도 슬퍼하거나 노여워하지 말라
    ...
    </p>
    <p><em>참고 푸시킨의 시</em></p>
</body>
</html>
```

191

텍스트 장식

```
<!DOCTYPE html>
<html>
<head>
<style>
    h1 { text-decoration:overline; }
    h2 { text-decoration:line-through; }
    h3 { text-decoration:underline; }
</style>
</head>
<body>
    <h1>텍스트 장식의 예입니다.</h1>
    <h2>텍스트 장식의 예입니다.</h2>
    <h3>텍스트 장식의 예입니다.</h3>
</body>
</html>
```

192

텍스트 변환

```
<!DOCTYPE html>
<html>
<head>
<style>
    p.upper { text-transform:uppercase; }
    p.lower { text-transform:lowercase; }
    p.capit { text-transform:capitalize; }
</style>
</head>
<body>
    <p class="upper">text_transform is uppercase.</p>
    <p class="lower">text_transform is lowercase.</p>
    <p class="capit">text_transform is capitalize.</p>
</body>
</html>
```

193

텍스트 그림자

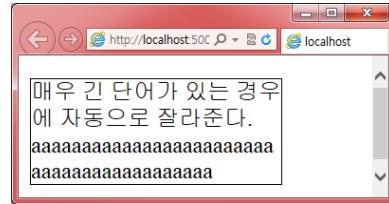
```
<!DOCTYPE html>
<html>
<head>
    <style>
        h1 {text-shadow: 5px 5px 5px #FF0000;}
    </style>
</head>
<body>
    <h1>Text-shadow 처리!</h1>
</body>
</html>
```



194

Word Wrapping

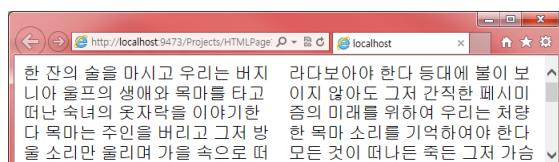
```
<!DOCTYPE html>
<html>
<head>
<style>
    p.test {
        width: 11em;
        border: 1px solid #000000;
        word-wrap: break-word;
    }
</style>
</head>
<body>
    <p class="test">
        매우 긴 단어가 있는 경우에 자동으로 잘라준다.
       aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
    </p>
</body>
</html>
```



195

다중 컬럼

```
<!DOCTYPE html>
<html>
<head>
<style>
    .poet {
        column-count: 2;
    }
</style>
</head>
<body>
    <div class="poet">
        한 잔의 술을 마시고 우리는 버지니아 울프의 생애와 목마를 타고 떠난 숙녀의 옷자락을 이야기한다
        ...
        가을바람 소리는 내 쓰러진 술병 속에서 목메어 우는데
    </div>
</body>
</html>
```



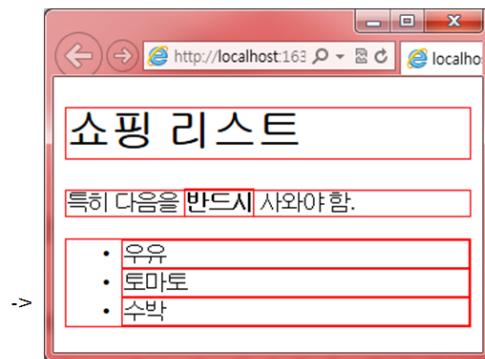
196

05 CSS 박스모델과 응용

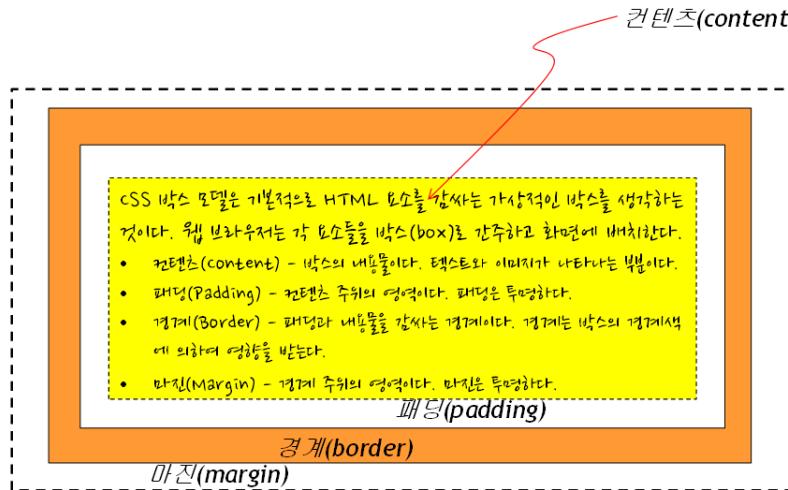
박스모델

- HTML 요소들을 박스(사각형) 형태로 그리는 것
- 박스는 배치, 색상, 경계 등의 속성을 가진다.

```
<!DOCTYPE html>
<html>
<body>
  <h1>쇼핑 리스트</h1>
  <p>특히 다음을 <strong>반드시</strong> 사와야 함.</p>
  <ul>
    <li>우유</li>
    <li>토마토</li>
    <li>수박</li>
  </ul>
</body>
</html>
```



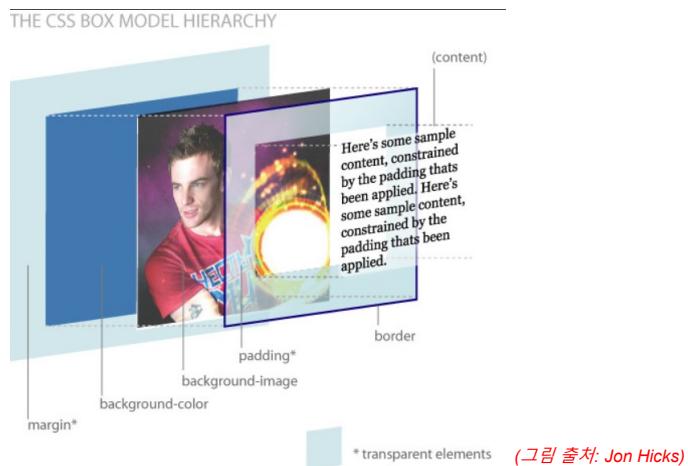
박스모델의 속성



199

배경색과 배경 이미지

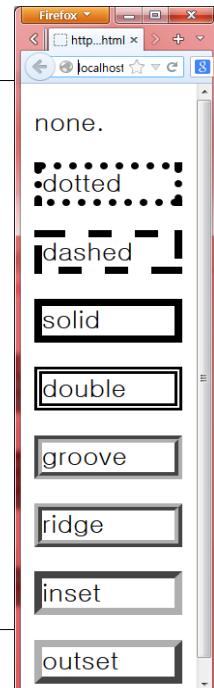
- HTML 요소에 배경색과 배경 이미지가 설정되어 있는 경우에, 패딩은 투명하므로 배경 이미지와 배경색이 보이게 된다



200

경계선 스타일

```
<!DOCTYPE html>
<html>
<body>
    <p style="border-style: none;">none.</p>
    <p style="border-style: dotted;">dotted</p>
    <p style="border-style: dashed;">dashed</p>
    <p style="border-style: solid;">solid</p>
    <p style="border-style: double;">double</p>
    <p style="border-style: groove;">groove</p>
    <p style="border-style: ridge;">ridge</p>
    <p style="border-style: inset;">inset</p>
    <p style="border-style: outset;">outset</p>
</body>
</html>
```



201

경계선의 폭

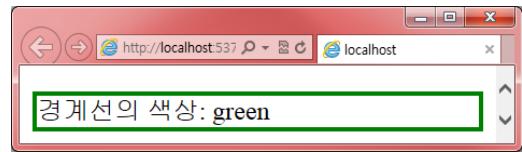
```
...
<head>
    <style>
        p.thick {border-style: solid; border-width: thick;}
        p.medium {border-style: solid; border-width: medium;}
        p.thin {border-style: solid; border-width: 1px;}
    </style>
</head>
<body>
    <p class="thick">경계선이 thick으로 설정되었음</p>
    <p class="medium">경계선이 medium으로 설정되었음</p>
    <p class="thin">경계선이 1px으로 설정되었음</p>
</body>
</html>
```



202

경계선의 색상

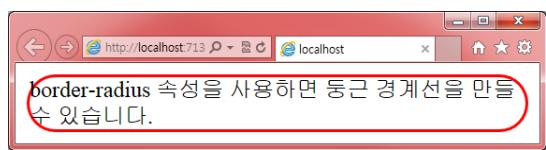
```
<!DOCTYPE html>
<html>
<head>
<style>
    p.green {
        border-style: solid;
        border-color: green;
    }
</style>
</head>
<body>
    <p class="green">경계선의 색상: green</p>
</body>
</html>
```



203

동근 경계선

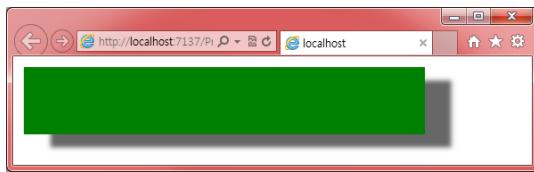
```
<!DOCTYPE html>
<html>
<head>
<style>
    div {
        border: 2px solid red;
        border-radius: 25px;
    }
</style>
</head>
<body>
    <div>border-radius 속성을 사용하면 둥근 경계선을 만들 수 있습니다.</div>
</body>
</html>
```



204

경계선 그림자

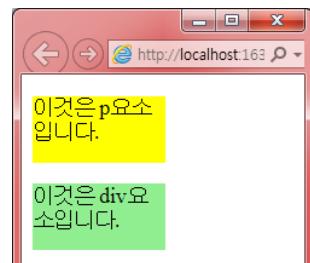
```
<!DOCTYPE html>
<html>
<head>
    <style>
        div {
            width: 300px;
            height: 50px;
            background-color: green;
            box-shadow: 20px 10px 5px #666666;
        }
    </style>
</head>
<body>
    <div></div>
</body>
</html>
```



205

요소 크기 설정

```
<!DOCTYPE html>
<html>
<head>
    <style>
        #target1 {
            width: 100px;
            height: 50px;
            background-color: yellow;
        }
        #target2 {
            width: 100px;
            height: 50px;
            background-color: lightgreen;
        }
    </style>
</head>
<body>
    <p id="target1">이것은 p요소입니다. </p>
    <div id="target2">이것은 div요소입니다.</div>
</body>
</html>
```



206

마진과 패딩 설정하기

값	설명
auto	브라우저가 마진을 계산한다.
length	마진을 px, pt, cm 단위로 지정할 수 있다. 디폴트는 0px이다.
%	마진을 요소 폭의 퍼센트로 지정한다.
inherit	마진이 부모 요소로부터 상속된다.

- 마진의 각 변을 별도로 지정하는 방법.

```
margin-top:10px;  
margin-right:20px;  
margin-bottom:30px;  
margin-left:40px;
```

- 한 줄에서 모든 변에 대한 값을 지정하는 방법.

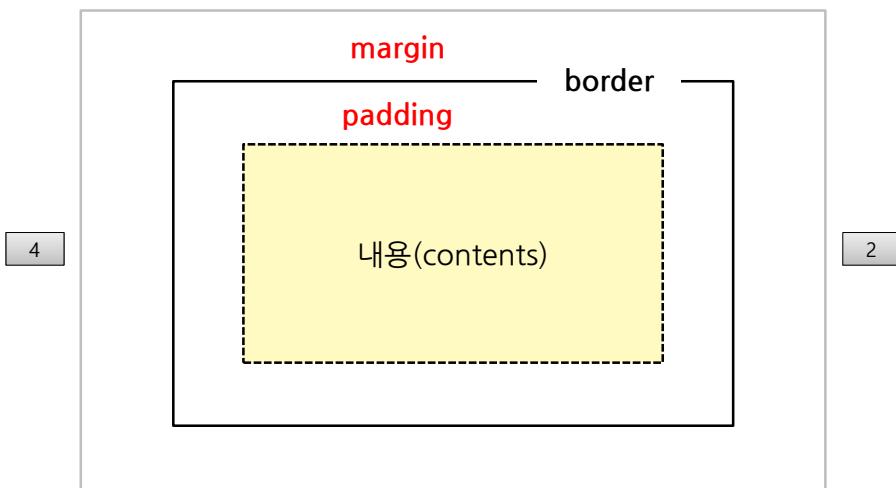
```
margin: 10px 20px 30px 40px;
```



207

마진과 패딩

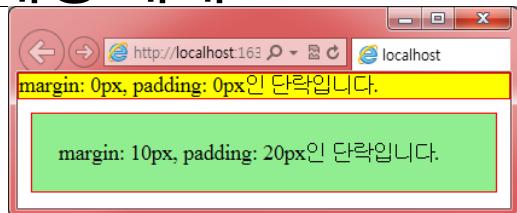
1



208

마진과 패딩 예제

```
...  
<style>  
    body {  
        margin: 0px;  
        padding: 0px;  
    }  
    p {  
        margin: 0px;  
        padding: 0px;  
        background-color: yellow; border: 1px solid red;  
    }  
    #target {  
        margin: 10px;  
        padding: 20px;  
        background-color: lightgreen; border: 1px solid red;  
    }  
</style>  
</head>  
<body>  
    <p>margin: 0px, padding: 0px인 단락입니다.</p>  
    <p id="target">margin: 10px, padding: 20px인 단락입니다.</p>  
</body>  
</html>
```



209

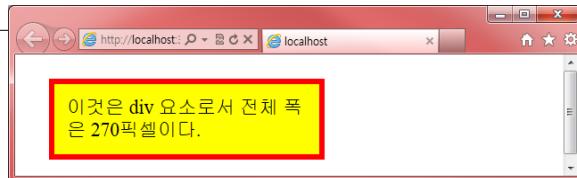
박스의 크기 계산



210

예제

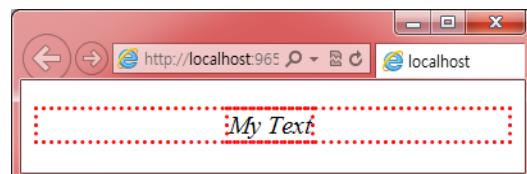
```
<!DOCTYPE html>
<html>
<head>
<style>
div.test { background-color: yellow;
            width: 200px; padding: 10px;
            border: 5px solid red; margin: 20px; }
</style>
</head>
<body>
<div class="test">이것은 div 요소로서 전체 폭은 270픽셀이다.</div>
</body>
</html>
```



211

수평정렬(인라인 요소)

```
<!DOCTYPE html>
<html>
<head>
<style>
p, em, strong, div
{
    border: dotted 3px red;
}
</style>
</head>
<body>
<p style="text-align: center">
<em>My Text </em>
</p>
</body>
</html>
```



212

수평정렬(블록 요소)

```
<!DOCTYPE html>
<html>
<head>
<style>
p, h1 {
    border: dotted 3px red;
}
</style>
</head>
<body>
<p style="margin-left: auto; margin-right: auto; width: 50%">
    My Text
</p>
</body>
</html>
```



213

배경 설정하기

속성	설명
background	한줄에서 모든 배경 속성을 정의한다.
background-attachment	배경 이미지가 고정되어 있는지 스크롤되는지를 지정한다.
background-color	배경색을 정의한다.
background-image	배경 이미지를 정의한다.
background-position	배경 이미지의 시작위치를 지정한다.
background-repeat	배경 이미지의 반복 여부를 지정한다.

```
body { background-color: red; } /* 배경을 빨간색으로 설정한다 */
body { background-color: rgb(255,0,0); }/* 배경을 빨간색으로 설정한다 */
body { background-color: #ff0000; } /* 배경을 빨간색으로 설정한다 */
```

214

배경 이미지 설정

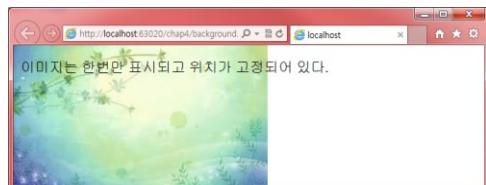
```
<!DOCTYPE html>
<html>
<head>
    <style>
        body {
            background-image: url('back1.jpg')
        }
    </style>
</head>
<body>
    <h1>삶이 그대를 속일지라도</h1>
    <p> 삶이 그대를 속일지라도
        슬퍼하거나 노하지 말아라.
        ...
        지나가 버린 것 그리움이 되리니.
    </p>
</body>
</html>
```



215

고정된 배경 이미지

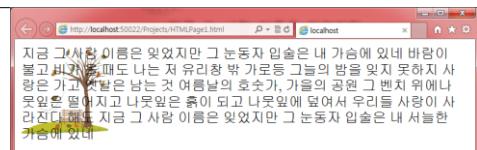
```
<!DOCTYPE html>
<html>
<head>
    <style>
        body {
            background-image: url('back1.jpg');
            background-repeat: no-repeat;
            background-attachment: fixed;
        }
    </style>
</head>
<body>
    <p>이미지는 한번만 표시되고 위치가 고정되어 있다.</p>
</body>
</html>
```



216

배경 이미지 크기

```
<!DOCTYPE html>
<html>
<head>
<style>
    div {width: 500px; height: 100px;
        background: url(back.jpg);
        background-repeat: no-repeat;
        background-size: 100px 100px; }
</style>
</head>
<body>
<div>
    지금 그 사람의 이름은 잊었지만 그의 눈동자 입술은 내 가슴에 있네... 내
    가슴에 있네
</div>
</body>
</html>
```



217

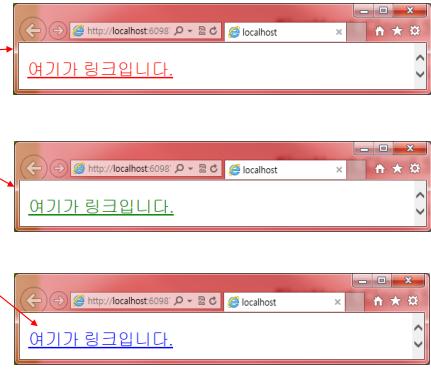
링크 스타일

- a:link - 방문되지 않은 링크의 스타일
- a:visited - 방문된 링크의 스타일
- a:hover - 마우스가 위에 있을 때의 스타일
- a:active - 마우스로 클릭되는 때의 스타일

218

링크 예제

```
<!DOCTYPE html>
<html>
<head>
<style>
    a:link { color: red; }
    a:visited { color:green; }
    a:hover { color:blue; }
    a:active { color:yellow; }
</style>
</head>
<body>
<p><a href="" target="_blank">
    여기가 링크입니다.</a>
</p>
</body>
</html>
```



219

링크 예제

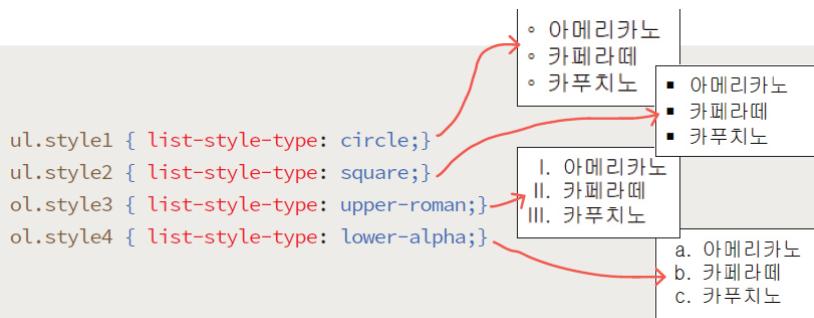
```
<!DOCTYPE html>
<html>
<head>
<style>
    a.style1:link {color: #ff0000; }
    a.style1:visited {color: #0000ff; }
    a.style1:hover {font-size: 150%; }
    a.style2:link {color: #ff0000; }
    a.style2:visited {color: #0000ff; }
    a.style2:hover {background: #66ff66; }
</style>
</head>
<body>
<p>마우스를 올려놓으면 스타일이 변경됩니다.</p>
<p><a class="style1" href="index.html" target="_blank">
    폰트크기를 변경하는 링크</a></p>
<p><a class="style2" href="index.html" target="_blank">
    배경색을 변경하는 링크</a></p>
</body>
</html>
```



220

리스트 스타일

속성	설명
list-style	리스트에 대한 속성을 한줄로 설정한다.
list-style-image	리스트 항목 마커를 이미지로 지정한다.
list-style-position	리스트 마커의 위치를 안쪽인지 바깥쪽인지를 지정한다.
list-style-type	리스트 마커의 타입을 지정한다.



221

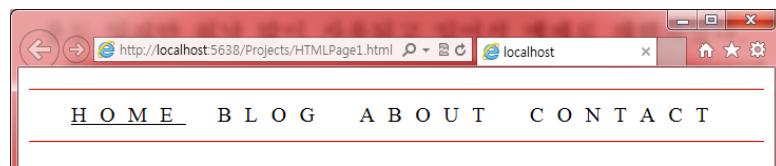
수평 리스트 예제

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
    list-style:none; text-align:center; border-top:1px solid red;
    border-bottom:1px solid red; padding:10px 0;
}
ul li {
    display:inline; text-transform:uppercase;
    padding:0 10px; letter-spacing:10px;
}
ul li a { text-decoration:none; color:black; }
ul li a:hover { text-decoration:underline; }
</style>
</head>
```

222

수평 리스트 예제

```
<body>
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">Blog</a></li>
    <li><a href="#">About</a></li>
    <li><a href="#">Contact</a></li>
  </ul>
</body>
</html>
```



223

테이블 스타일

속성	설명
border	테이블의 경계선
border-collapse	이웃한 셀의 경계선을 합칠 것인지 여부
width	테이블의 가로 길이
height	테이블의 세로 길이
border-spacing	테이블 셀 사이의 거리
empty-cells	공백 셀을 그릴 것인지 여부
table-align	테이블 셀의 정렬 설정

224

테이블의 경계

```
<!DOCTYPE html>
<html>
<head>
<style>
    table, td, th { border: 1px solid blue; }
</style>
</head>
<body>
<table>
<tr><th>이름</th><th>이메일</th></tr>
<tr><td>김철수</td><td>chul@google.com</td></tr>
<tr><td>김영희</td><td>young@google.com</td></tr>
</table>
</body>
</html>
```

이름	이메일
김철수	chul@google.com
김영희	young@google.com

225

경계 통합

- collapse : 이웃하는 셀의 경계선을 합쳐서 단일선으로 표시한다.
- separate : 이웃하는 셀의 경계선을 합치지 않고 분리하여 표시한다.

```
<!DOCTYPE html>
<html>
<head>
<style>
    table {border-collapse: collapse;}
    table, th, td {border: 1px solid blue; }
</style>
</head>
<...>
```

이름	이메일
김철수	chul@google.com
김영희	young@google.com

226

테이블 배경색

```
<!DOCTYPE html>
<html>
<head>
<style>
td, th { color: white; background-color: green; }
</style>
</head>
<body>
<table>
<tr><th>이름</th><th>이메일</th></tr>
<tr><td>김철수</td><td>chul@google.com</td></tr>
<tr><td>김영희</td><td>young@google.com</td></tr>
</table>
</body>
</html>
```

이름	이메일
김철수	chul@google.com
김영희	young@google.com

227

헤더와 데이터의 분리

```
<!DOCTYPE html>
<html>
<head>
<style>
table, td, th { border: 1px solid green; }
th {background-color: green; color: white; }
</style>
</head>
...
```

이름	이메일
김철수	chul@google.com
김영희	young@google.com

228

테이블 텍스트 정렬

```
<!DOCTYPE html>
<html>
<head>
<style>
table, td, th { border: 1px solid blue; }
table { width: 100%; }
td { text-align: center; }
</style>
</head>
<body>
<table>
<tr><th>이름</th><th>이메일</th></tr>
<tr><td>김철수</td><td>chul@google.com</td></tr>
<tr><td>김영희</td><td>young@google.com</td></tr>
</table>
</body>
</html>
```

이름	이메일
김철수	chul@google.com
김영희	young@google.com

229

테이블 캡션

```
<!DOCTYPE html>
<html>
<head>
<style>
caption { caption-side:bottom; }
</style>
</head>
<body>
<table border="1">
<caption>VIP 고객 리스트</caption>
<tr><th>이름</th><th>이메일</th></tr>
<tr><td>김철수</td><td>chul@google.com</td></tr>
<tr><td>김영희</td><td>young@google.com</td></tr>
</table>
</body>
</html>
```

이름	이메일
김철수	chul@google.com
김영희	young@google.com

230

짝수행과 홀수행 다르게 하기

```
<!DOCTYPE html>
<html>
<head>
    <style>
        #list {font-family: "Trebuchet MS", sans-serif; width: 100%; }
        #list td, #list th {
            border: 1px dotted gray;
            text-align: center;
        }
        #list th {color: white; background-color: blue;}
        #list tr.alt td {background-color: yellow;}
    </style>
</head>
```

231

짝수행과 홀수행 다르게 하기

```
<body>
    <table id="list">
        <tr><th>이름</th><th>이메일</th></tr>
        <tr><td>김철수</td><td>chul@google.com</td><tr>
        <tr class="alt"><td>김영희</td><td>young@google.com</td><tr>
        <tr><td>홍길동</td><td>hong@google.com</td><tr>
        <tr class="alt"><td>김수진</td><td>sujin@google.com</td><tr>
    </table>
</body>
</html>
```

A screenshot of a Microsoft Internet Explorer browser window displaying a table. The table has two columns: '이름' (Name) and '이메일' (Email). There are four rows of data. The first and third rows have a light blue background color, while the second and fourth rows have a yellow background color. The browser's address bar shows 'http://localhost:601'. The table data is as follows:

이름	이메일
김철수	chul@google.com
김영희	young@google.com
홍길동	hong@google.com
김수진	sujin@google.com

232

06 CSS 레이아웃

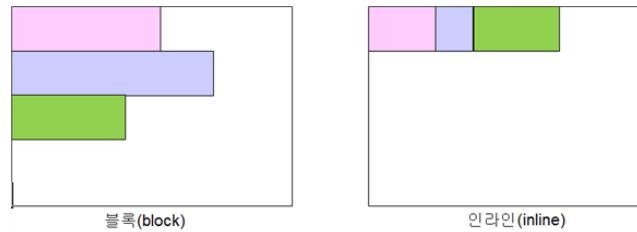
레이아웃이란?

- 웹페이지에서 HTML 요소의 위치, 크기 등을 결정하는 것
 - 집안에서의 가구 배치와 비슷하다.



블록요소와 인라인 요소

- 블록(block) 요소 - 화면의 한 줄을 전부 차지한다.
- 인라인(inline) 요소 - 한 줄에 차례대로 배치된다. 현재 줄에서 필요한 만큼의 너비만을 차지한다.

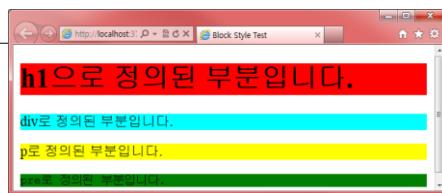


235

블록요소

- 한 줄을 전부 차지
- <h1>, <p>, , , <table>, <blockquote>, <pre>, <div> <form>, <header>, <nav> 요소
- 예제 실행과 소스보기

```
<body>
  <h1 style="background-color: red">h1으로 정의된 부분입니다.</h1>
  <div style="background-color: aqua">div로 정의된 부분입니다.</div>
  <p style="background-color: yellow">p로 정의된 부분입니다.</p>
  <pre style="background-color: green">pre로 정의된 부분입니다.</pre>
</body>
```

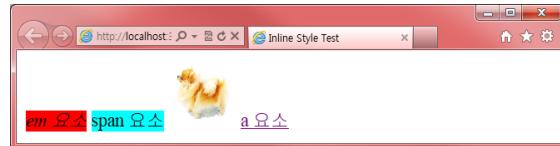


236

인라인요소

- 인라인 요소들은 한 줄 안에 차례대로 배치
- <a>, , , ,
, <input>, 요소

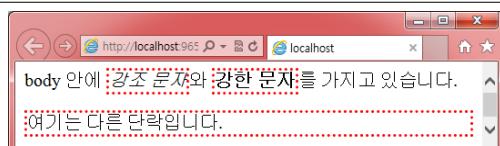
```
<body>
    <em style="background-color: red">em 요소</em>
    <span style="background-color: aqua">span 요소</span>
    
    <a href="http://www.w3c.org">a 요소</a>
</body>
```



237

블록 요소와 인라인 요소의 혼합

```
<!DOCTYPE html>
<html>
<head>
    <style>
        p, em, strong {
            border: dotted 3px red;
        }
    </style>
</head>
<body>
    body 안에
    <em>강조 문자</em>와 <strong>강한 문자</strong>를 가지고 있습니다.
    <p>여기는 다른 단락입니다. </p>
</body>
</html>
```



238

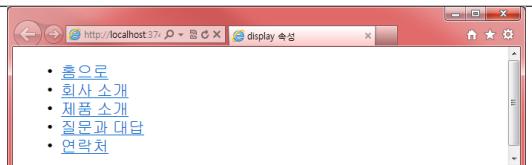
CSS의 display 속성

- 속성 display를 block으로 설정하면 -> 블록 요소처럼 배치
- display를 inline으로 설정-> 인라인 요소처럼 배치
 - display:block : 블록(block)
 - display:inline : 인라인.inline)
 - display:none : 없는 것으로 간주됨
 - visibility:hidden : 화면에서 감춰짐

239

예제

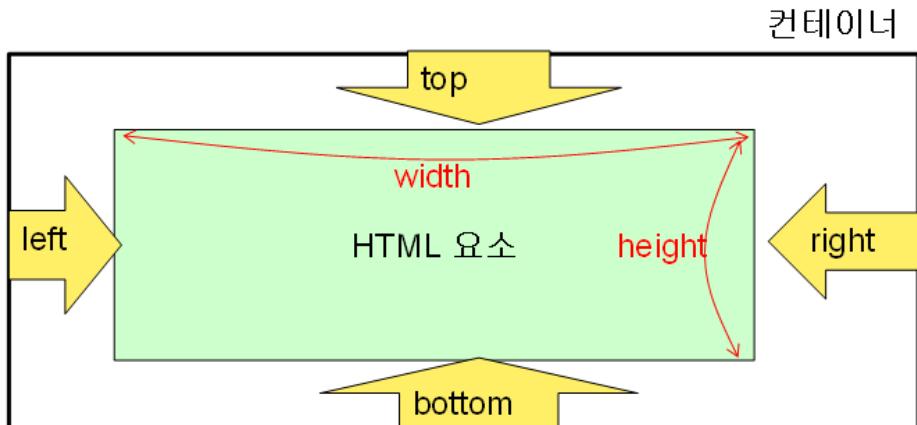
```
<!DOCTYPE html>
<html>
<head>
<title>display 속성</title>
<style>
.menubar li {display: inline; background-color: yellow; margin: 0;
              border: 1px solid; border-color: red; padding: .5em;}
</style>
</head>
<body>
  <ul class="menubar">
    <li><a href="#">홈으로</a></li>
    <li><a href="#">회사 소개</a></li>
    <li><a href="#">제품 소개</a></li>
    <li><a href="#">질문과 대답</a></li>
    <li><a href="#">연락처</a></li>
  </ul>
</body>
</html>
```



240

요소의 위치

- top, bottom, left, right 속성으로 결정



241

위치 설정 방법

- 정적 위치 설정(static positioning) - 정상적인 흐름에 따른 배치
- 상대 위치 설정(relative positioning) - 정상적인 위치가 기준점이 된다.
- 절대 위치 설정(absolute positioning) - 컨테이너의 원점이 기준점이 된다.
- 고정 위치 설정(fixed positioning) - 윈도우의 원점이 기준점이 된다.

242

정적 위치 설정

- 정적 위치 설정(static positioning)

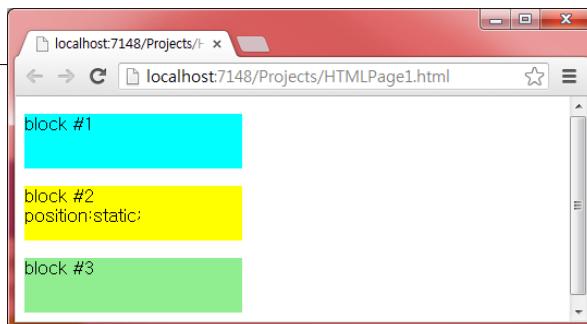
- 블록 요소들은 박스처럼 상하로 쌓이게 되고 인라인 요소들은 한 줄에 차례대로 배치

```
<!DOCTYPE html>
<html>
<head>
<style>
#one {background-color: cyan; width: 200px; height: 50px; }
#two {position: static; background-color: yellow; width: 200px;
height: 50px; }
#three {background-color: lightgreen; width: 200px; height: 50px; }
</style>
```

243

예제

```
<body>
<p id="one">block #1</p>
<div id="two">
    block #2<br />
    position:static;<br />
</div>
<p id="three">block #3</p>
</body>
</html>
```



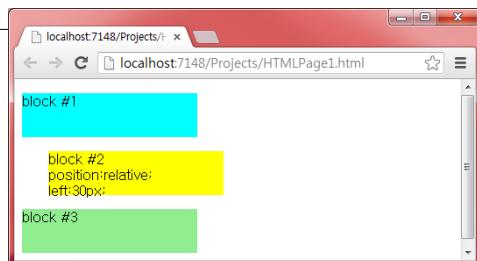
244

상대 위치 설정

- 상대 위치 설정(relative positioning)

- 정상적인 위치에서 상대적으로 요소가 배치

```
<style>
#one {background-color: cyan; width: 200px; height: 50px; }
#two {position: relative; left: 30px; background-color: yellow;
       width: 200px; height: 50px; }
#three {background-color: lightgreen; width: 200px; height: 50px; }
</style>
```



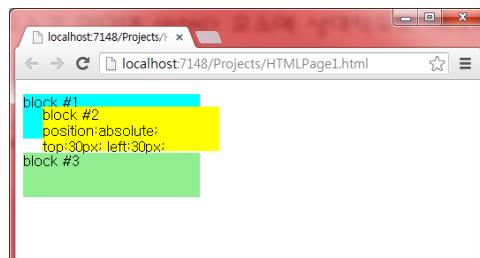
245

절대 위치 설정

- 절대 위치(absolute positioning)

- 전체 페이지를 기준으로 시작 위치에서 top, left, bottom, right 만큼 떨어진 위치에 배치

```
...
#two {
    position: absolute;
    top: 30px;
    left: 30px;
    background-color: yellow;
    width: 200px;
    height: 50px;
}
...
```



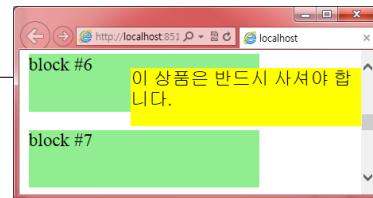
246

고정 위치 설정

• 고정 위치 설정(fixed positioning)

- 브라우저 윈도우에 상대적으로 요소의 위치를 잡는 것

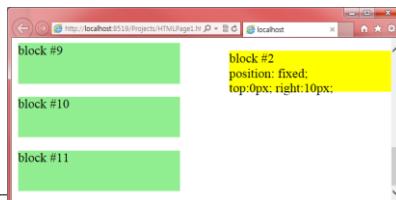
```
<!DOCTYPE html>
<html>
<head>
<style>
p {background-color: lightgreen; width: 200px; height: 50px; }
#two {background-color: yellow; position:fixed; top:0px; right:0px; }
</style>
</head>
```



247

고정 위치 설정

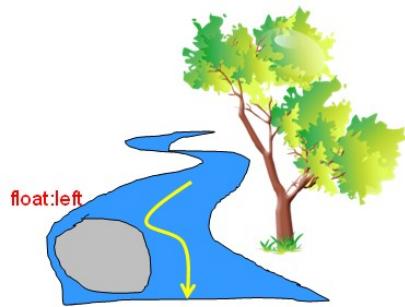
```
<body>
<p>block #1</p>
<p id="two">block #2<br>position: fixed;<br>top:0px; right:10px;<p>
<p>block #3</p>
<p>block #4</p>
<p>block #5</p>
<p>block #6</p>
<p>block #7</p>
<p>block #8</p>
<p>block #9</p>
<p>block #10</p>
<p>block #11</p>
</body>
</html>
```



248

float 속성

- 하나의 콘텐츠 주위로 다른 콘텐츠들이 물처럼 흘러가는 스타일 지정



249

예제

```
<!DOCTYPE html>
<html>
<head>
    <style>
        img.a {float: left;}
    </style>
</head>
<body>
    
    <p>
        생활이 그대를 속일지라도  

        슬퍼하거나 노여워 말라.  

        ...
    </p>
</body>
</html>
```



250

예제

```
<!DOCTYPE html>
<html>
<head>
<style>
img {float: left; width: 110px; height: 90px; margin: 5px; }
</style>
</head>
<body>
<h3>이미지 갤러리</h3>



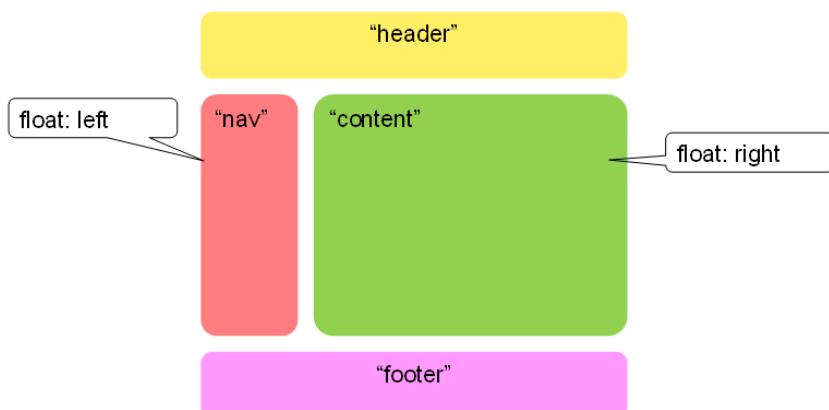



</body>
</html>
```

251

float의 용도

- 레이아웃에 많이 사용된다.



252

clear 속성

- float 속성을 중단할 때 사용된다.

```
#footer {  
    clear: both;  
}
```

float: left

“content”

“nav”

float: right

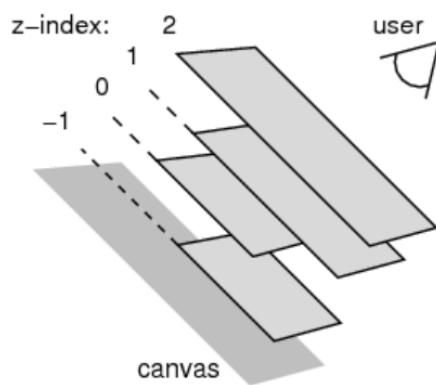
“footer”

clear: both로
지정하면 빈칸을
채우지 않고
정상적으로
배치된다.

253

z-index

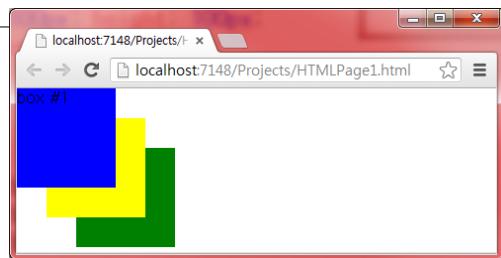
- 요소의 스택 순서를 지정



254

예제

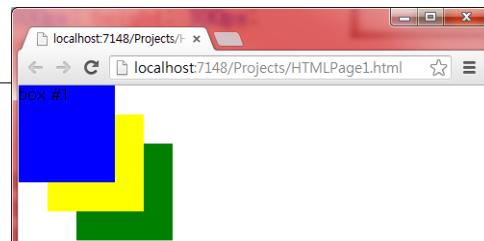
```
...
<style>
#box1 {position: absolute; top: 0px; left: 0px; width: 100px;
       height: 100px; background: blue; z-index: 200; }
#box2 {position: absolute; top: 30px; left: 30px;
       width: 100px; height: 100px; background: yellow; z-index: 100; }
#box3 {position: absolute; top: 60px; left: 60px;
       width: 100px; height: 100px; background: green; z-index: 0; }
</style>
```



255

예제

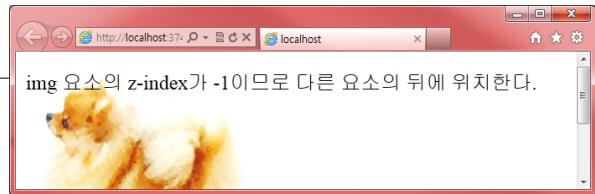
```
</head>
<body>
  <div id="box1">box #1 </div>
  <div id="box2">box #2 </div>
  <div id="box3">box #3 </div>
</body>
</html>
```



256

예제

```
<!DOCTYPE html>
<html>
<head>
<style>
img {position: absolute; left: 0px; top: 0px; z-index: -1; }
</style>
</head>
<body>
img 요소의 z-index가 -1이므로 다른 요소의 뒤에 위치한다.
</body>
</html>
```



257

요소의 크기 지정

- width, height - 요소의 크기
- min-width, min-height: 요소의 최소 크기
- max-width, max-height: 요소의 최대 크기

258

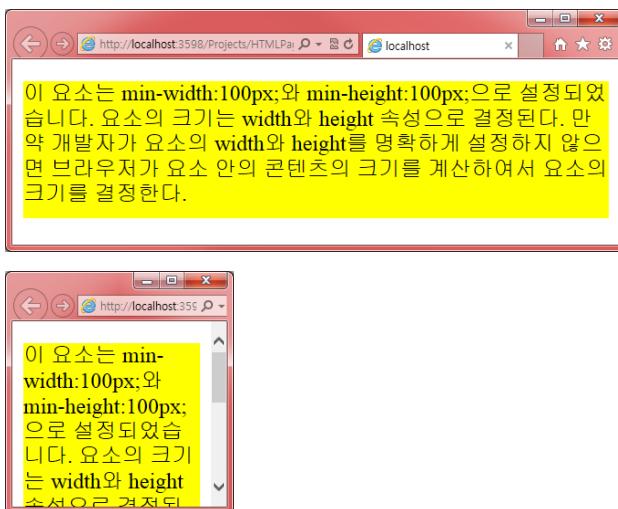
예제

```
<!DOCTYPE html>
<html>
<head>
<style>
p {min-width: 100px; min-height: 100px; background-color: yellow; }
</style>
</head>
<body>
<p>이 요소는 min-width:100px;와 min-height:100px;으로 설정되었습니다.  
요소의 크기는 width와 height 속성으로 결정된다.  
만약 개발자가 요소의 width와 height를 명확하게 설정하지 않으면 브라우저  
가 요소 안의 콘텐츠의 크기를 계산하여서 요소의 크기를 결정한다.</p>
</body>
</html>
```

259

실행 결과

• 실행 결과(클릭)



260

overflow 속성

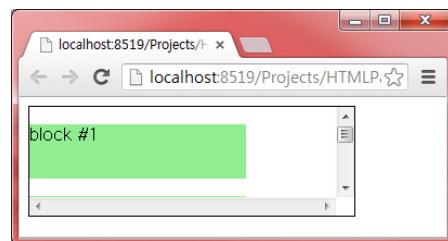
- overflow 속성: 자식 요소가 부모 요소의 범위를 벗어났을 때, 어떻게 처리할 것인지를 지정

- hidden - 부모 영역을 벗어나는 부분을 보이지 않게 한다.
- scroll - 부모 영역을 벗어나는 부분을 스크롤 할 수 있도록 한다.
- auto - 자동으로 스크롤 바가 나타난다.

261

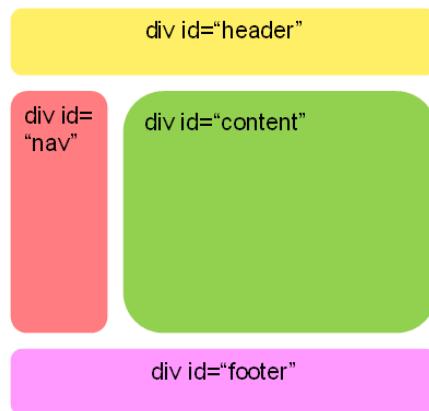
예제

```
<!DOCTYPE html>
<html>
<head>
    <style>
        p { background-color: lightgreen; width: 200px; height: 50px; }
        #target { border: 1px solid black; width: 300px;
                  height: 100px; overflow: scroll; }
    </style>
</head>
<body>
    <div id=target>
        <p>block #1</p>
        <p>block #2</p>
        <p>block #3</p>
        <p>block #4</p>
        <p>block #5</p>
    </div>
</body>
</html>
```



262

<div>를 이용한 레이아웃



263

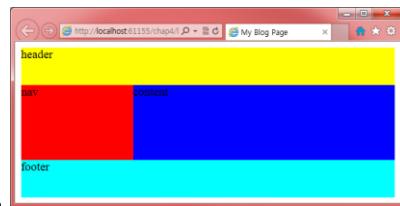
예제

```
<!DOCTYPE html>
<html>
<head>
    <title>My Blog Page</title>
    <style>
#header {
    background-color: yellow;
    width: 100%;
    height: 50px;
}
#nav {
    width: 30%;
    background-color: red;
    height: 100px;
    float: left;
}
```

264

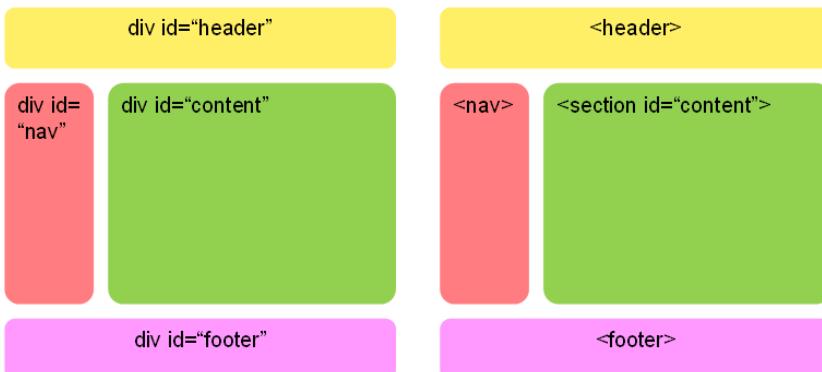
예제

```
#content { width: 70%; background-color: blue;
            float: right; height: 100px;}
#footer { background-color: aqua; width: 100%;
            height: 50px; clear: both;}
</style>
</head>
<body>
    <div id="wrapper">
        <div id="header"> header </div>
        <div id="nav"> nav </div>
        <div id="content"> content </div>
        <div id="footer"> footer </div>
    </div>
</body>
</html>
```



265

시맨틱 요소 레이아웃



266

시맨틱 요소

태그	설명
<header>	문서의 머리말(header)
<hgroup>	<h1>에서 <h6>요소들의 그룹
<nav>	내비게이션 링크
<article>	문서의 내용이나 블로그의 포스트
<section>	문서의 섹션을 의미한다.
<aside>	사이드바와 같이 옆에 위치하는 내용
<footer>	문서의 꼬리말(footer)
<figure>	그림이나 도표
<time>	날짜와 시간을 표시

267

연습

My Blog Page

Links

- [W3C](#)
- [MOZILLA](#)
- [HTML Dogs](#)

Semantic Tags

시멘틱 요소(Semantic elements)들은 브라우저에게 요소의 의미나 목적을 명확하게 알려주는 요소이다.

div와 span

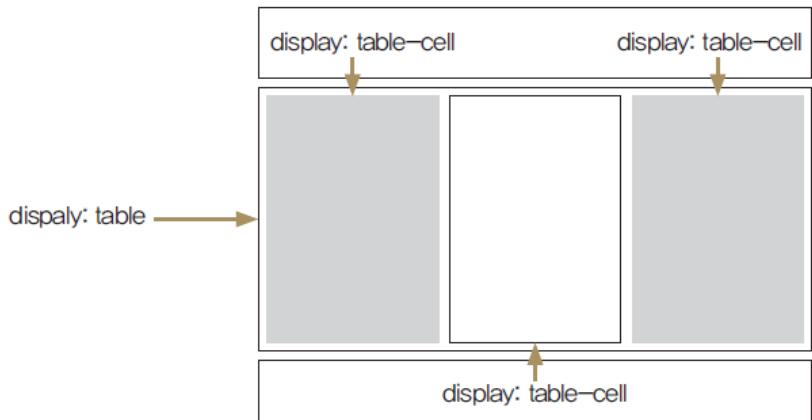
div은 "divide"의 약자로서 페이지를 논리적인 섹션으로 분리하는데 사용되는 태그이다. span 요소는 인라인 요소로서 텍스트를 위한 컨테이너로 사용할 수 있다.

Copyright (c) 2013 Hong

268

table-cell 속성

- display 속성에 table-cell을 하면 자식 요소들을 테이블의 셀처럼 배치하라는 의미가 된다.



269

CSS3 효과: 투명도

```
<!DOCTYPE html>
<html>
<head>
  <style>
    img { opacity: 0.4; }
    img:hover { opacity: 1.0; }
  </style>
</head>
<body>
  <h1>Opacity 속성</h1>
  
  
</body>
</html>
```



270

CSS3 효과: 가시성

```
<!DOCTYPE html>
<html>
<head>
  <style>
    #a {
      visibility: hidden;
      border: 1px dotted red;
    }
    #b {
      visibility: visible;
      border: 1px dotted red;
    }
  </style>
</head>
<body>
  <h1>Visibility 속성</h1>
  
  
</body>
</html>
```



271

07 HTML와 CSS로 웹사이트 만들기

웹사이트 구축 과정



273

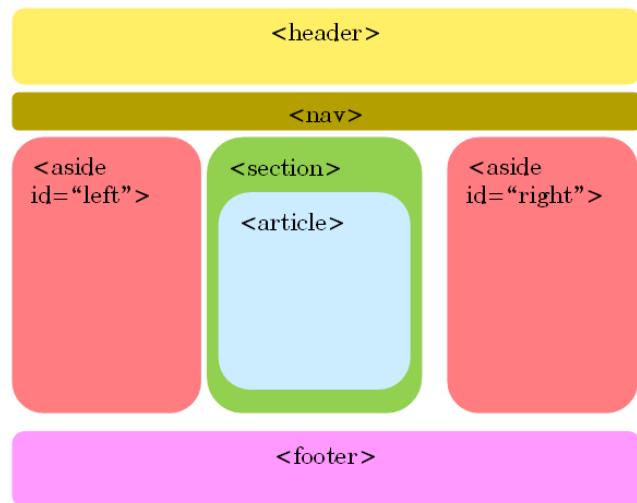
내비게이션 구조도



문서 이름	설명
index.html	WebShop의 홈 화면, 상품 카테고리, 로고, 로그인 등이 포함되어 있다.
register.html	회원 가입 화면. 다양한 HTML5 입력 양식을 이용해 본다.
shopcart.html	현재 쇼핑 카트에 들어 있는 상품을 보여준다.
computer.html	컴퓨터 상품을 보여주는 화면이다. index.html 화면 안에 포함된다.
clothing.html	의류 상품을 보여주는 화면이다. index.html 화면 안에 포함된다.
music.html	음악 파일을 판매하는 화면이다. index.html 화면 안에 포함된다.
movie.html	영화 파일을 판매하는 화면이다. index.html 화면 안에 포함된다.
mystyle.css	index.html에 대한 스타일을 가지고 있는 파일이다.

274

홈페이지 레이아웃



275

main.html

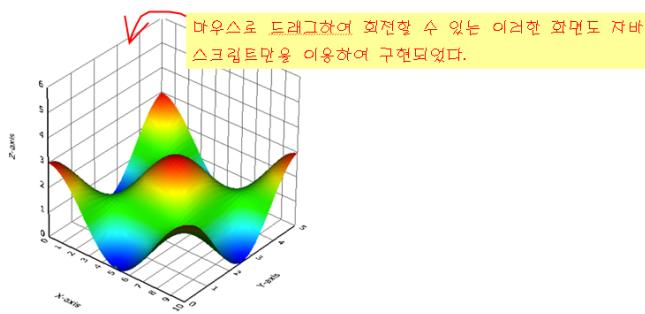


276

08 자바스크립트 기초

자바스크립트 소개

- 자바스크립트(javascript): 동적인 웹 페이지를 작성하기 위하여 사용되는 언어
- 웹의 표준 프로그래밍 언어
- 모든 웹브라우저들은 자바스크립트를 지원



HTML5 기술의 핵심



279

자바 vs 자바 스크립트

특징	자바 언어	자바스크립트
언어 종류	소스 파일을 컴파일하여 실행하는 컴파일 언어이다.	브라우저가 소스 코드를 직접 해석하여 실행하는 인터프리트 언어이다.
실행 방식	자바 가상 기계 위에서 실행한다.	브라우저 위에서 실행된다.
작성 위치	별도의 소스 파일에 작성	HTML 파일 안에 삽입 가능
변수 선언	변수의 타입을 반드시 선언해야 함	변수의 타입을 선언하지 않아도 사용 가능

280

자바스크립트 역사

- 넷스케이프의 브렌던 아이크(Brendan Eich)가 개발
- 처음에는 라이브스크립트(LiveScript)
- 최신 버전은 자바스크립트 1.8.5
- ECMA(European Computer Manufacturer's Association)이 ECMAScript라는 이름으로 표준을 제정-> ECMA-262



281

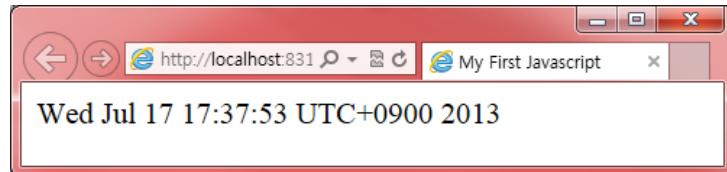
자바스크립트 특징

- 인터프리트 언어- 컴파일 과정을 거치지 않고 바로 실행시킬 수 있는 언어
- 동적 타이핑(dynamic typing) - 변수의 자료형을 선언하지 않고도 변수를 사용할 수 있는 특징
- 구조적 프로그래밍 지원 - C언어의 구조적 프로그래밍을 지원한다. 즉 if else, while, for등의 제어 구조를 완벽 지원
- 객체 기반 - 전적으로 객체지향언어이다. 자바스크립트의 객체는 연관배열(associative arrays)
- 함수형 프로그래밍 지원 - 자바스크립트에서 함수는 일급 객체(first-class object)이다. 즉 함수는 그 자체로 객체이다. 함수는 속성과 .call()과 같은 메서드를 가진다.
- 프로토타입-기반(prototype-based) - 상속을 위해 클래스 개념 대신에 프로토타입을 사용

282

첫번째 예제

```
<!DOCTYPE HTML>
<html>
<head>
    <title> 첫번째 Javascript </title>
</head>
<body>
    <script>
        var now = new Date();
        document.write(now);
    </script>
</body>
</html>
```



283

자바스크립트의 용도

- 이벤트에 반응하는 동작을 구현할 수 있다.
- AJAX를 통하여 전체 페이지를 다시 로드하지 않고서도 서버로 부터 새로운 페이지 콘텐츠를 받거나 데이터를 제출할 때, 사용 한다.
- HTML 요소들의 크기나 색상을 동적으로 변경할 수 있다.
- 게임이나 애니메이션과 같은 상호 대화적인 콘텐츠를 구현할 수 있다.
- 사용자가 입력한 값들을 검증하는 작업도 자바스크립트를 이용 한다.

284

자바스크립트의 미래

- 자바스크립트는 본래 클라이언트 웹페이지를 위한 프로그래밍 언어였지만 그 용도는 점점 더 확장되고 있다.
- Node.js : 웹서버와 같은 애플리케이션을 작성하기 위해 서버-사이드(Server-Side) 소프트웨어 시스템
- jQuery : 자바스크립트 라이브러리
- JSON : 자바스크립트의 객체 표기법(Javascript Object Notation)은 개발 언어 독립적인 데이터 형식으로서 데이터 전송용 XML을 대체하고 있다. 심지어 문서 데이터베이스의 표준 저장 형식으로도 사용된다.



285

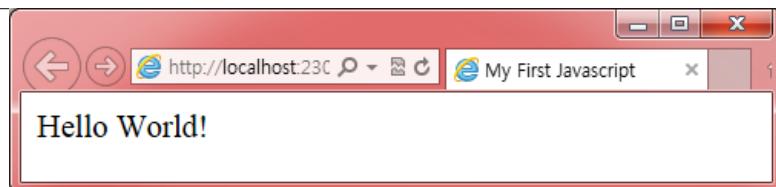
자바스크립트의 위치

- 내부 자바스크립트
- 외부 자바스크립트
- 인라인 자바스크립트

286

내부 자바 스크립트

```
<!DOCTYPE HTML>
<html>
<head>
    <title>My First Javascript </title>
    <script>
        document.write("Hello World!");
    </script>
</head>
<body></body>
</html>
```



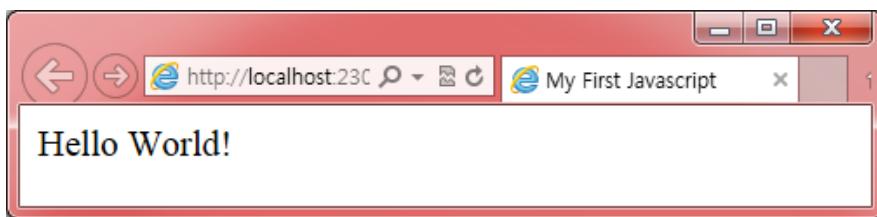
287

외부 자바 스크립트

```
<!DOCTYPE html>
<html>
<head>
    <script src="myscript.js"></script>
</head>
<body>
</body>
</html>
```

myscript.js

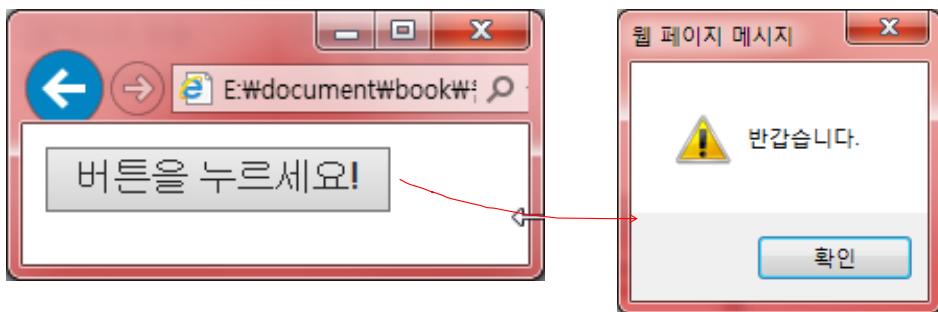
```
document.write("Hello World!");
```



288

인라인 자바 스크립트

```
<!DOCTYPE html>
<html>
<body>
  <button type="button" onclick="alert('반갑습니다.')">버튼을  
누르세요!</button>
</body>
</html>
```



289

문장

- 자바스크립트 문장(statement)들은 웹 브라우저에게 내리는 명령

- 문서에 "Hello World!"를 추가하시오.
- 화면에 경고창을 띄우시오.
- 변수를 하나 만드시요.
- ...



```
document.write("Hello World!");
alert("warning!");
var count;
...
```

순차적
으로
실행된
다.

290

주석문

- // - 단일문장 주석

```
// id가 heading1인 헤딩요소를 찾아서 내용을 바꾼다.  
document.getElementById("heading1").innerHTML = "My HomePage";
```

- /* */ - 다중 문장 주석

```
/*  
이 코드는 웹 페이지의 헤딩의 내용을 변경한다.  
*/  
document.getElementById("heading1").innerHTML = "My HomePage";
```

291

변수

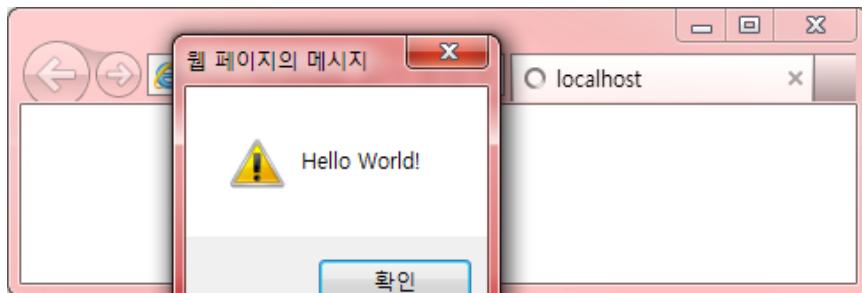
- 변수(variable)는 데이터를 저장하는 상자
- var 키워드를 사용하여서 선언(declare)한다.



292

예제

```
<script>
    var x;
    x = "Hello World!";
    alert(x);
</script>
```



293

변수 명명 규칙

- 변수 이름은 문자로 시작해야 한다.(숫자로 시작하면 안된다.)
- 변수 이름은 \$나 _로 시작할 수 있다.
- 변수 이름은 대소문자를 구별한다.(count와 Count는 서로 다른 변수이다.)
- 예약어는 변수명으로 사용할 수 없다.

294

자료형

- 수치형(number) - 정수나 실수
- 문자열(string) - 문자가 연결된 것, ""나 "로 표현
- 부울형(Boolean) - true 또는 false
- 객체형(object) - 객체를 나타내는 타입
- Undefined - 값이 정해지지 않은 상태

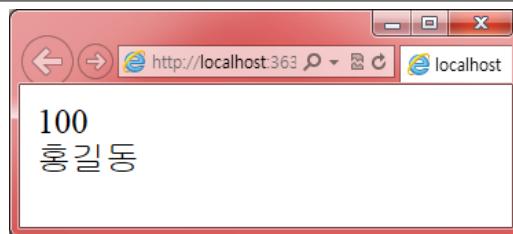
295

예제

```
<script>
  var s;

  s = 100;
  document.write(s + "<br>");

  s = "홍길동";
  document.write(s + "<br>");
</script>
```

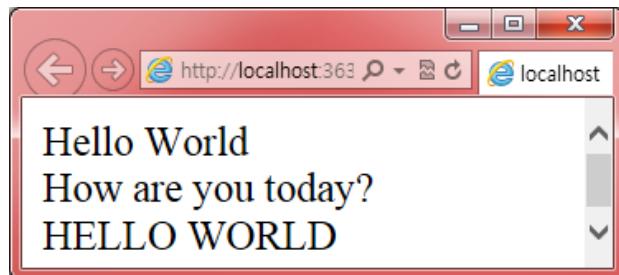


296

예제

```
<script>
    var s = "Hello World";
    var t = "How are you" + " today?";

    document.write(s + "<br>");
    document.write(t + "<br>");
    document.write(s.toUpperCase() + "<br>");
</script>
```



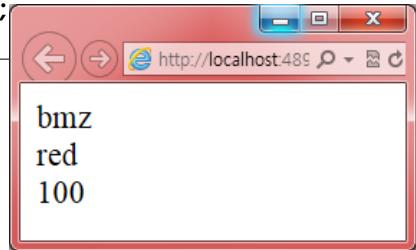
297

객체형

- 객체(object)는 사물의 속성과 동작을 묶어서 표현하는 기법
- (예) 자동차는 메이커, 모델, 색상, 마력과 같은 속성도 있고 출발하기, 정지하기 등의 동작도 가지고 있다.

```
<script>
    var myCar = {model:"bmw", color:"red", hp:100};

    document.write(myCar.model + "<br>");
    document.write(myCar.color + "<br>");
    document.write(myCar.hp + "<br>");
</script>
```



298

산술 연산자

연산자	설명	예	수식의 값
+	덧셈	$x = 3 + 2$	
-	뺄셈	$x = 3 - 2$	
*	곱셈	$x = 3 * 2$	
/	나눗셈	$x = 3 / 2$	
%	나머지	$x = 3 \% 2$	
++	증가	$++x$	
--	감소	$--x$	

299

대입 연산자

- 변수에 값을 할당한다.
- 수식 ' $z = x + y$ '는 x 값과 y 값을 더한 값을 z 에 대입한다는 의미이다.
- 대입연산자 "="는 산수에서의 같다라는 의미가 아니라 오른쪽에 있는 값을 왼쪽에 있는 변수에 저장하겠다라는 의미를 갖는다.
- "같다"를 표현할 때는 "=="을 사용한다.

300

복합 대입 연산자

- 다음 표는 $x = 10$, $y = 5$ 라고 가정하고 대입연산이 어떻게 수행되는지를 설명한다.

연산자	예	동일한 수식	결과
$+=$	$x += y$	$x = x + y$	
$-=$	$x -= y$	$x = x - y$	
$*=$	$x *= y$	$x = x * y$	
$/=$	$x /= y$	$x = x / y$	
$%=$	$x %= y$	$x = x \% y$	

301

문자열에서의 '+' 연산자

- + 연산자는 문자열을 결합하는 용도로도 사용된다.
- 즉 + 연산자가 문자열에서 사용되면 문자열 결합의 의미가 된다.

```
s1 = "Welcom to ";
s2 = "Javascript";
s3 = s1 + s2;
```

- 숫자와 문자열을 + 연산자로 합하면 숫자를 문자열로 변환하여, 결합된 문자열을 반환한다.

```
x = 1 + 1;
y = "Car" + 1;
document.write(x + "<br>");
document.write(y + "<br>");
```

302

비교 연산자

- 논리문장에서 값들을 비교하는 용도로 사용
- 다음 표에서 x의 값은 1이라고 가정한다.

연산자	설명	예	결과
==	값이 같으면 true	x == 1	
		x == 2	
!=	값이 다르면 true	x != 2	
>	크면 true	x > 2	
<	작으면 true	x < 2	
>=	크거나 같으면 true	x >= 2	
<=	작거나 같으면 true	x <= 2	

303

비교 연산자

- 비교연산자는 다음과 같이 조건문에서 많이 사용된다. 아직 학습하지 않았지만 다음 문장의 의미를 추리하여 보자.

```
if (age > 18) {  
    msg = "입장하실 수 있습니다.>";  
}
```

- 다음의 결과를 확인해보자.

```
<script>  
    var x = 10;  
    var y = 20;  
    document.write((x > y) + "<br>");  
    document.write((x < y) + "<br>");  
    document.write((x == y) + "<br>");  
    document.write((x != y) + "<br>");  
</script>
```

304

비교 연산자

- ==== 연산자와 !== 연산자

연산자	설명	예	결과
====	값과 타입이 모두 같으면 참	x === 1	
		x === "1"	
!==	값과 타입이 다르면 참	x !== 1	
		x !== "1"	

305

논리 연산자

- 여러 개의 조건을 조합하여 참인지 거짓인지를 따질 때 사용
- 예를 들어 "비가 오지 않고 휴일이면 테니스를 친다."라는 문장에는 "비가 오지 않는다"라는 조건과 "휴일이다"라는 조건이 동시에 만족이 되면 테니스를 친다는 의미가 포함되어 있다.

연산자	사용 예	의미
&&	x && y	AND 연산, x와 y가 모두 참이면 참, 그렇지 않으면 거짓
	x y	OR 연산, x나 y중에서 하나만 참이면 참, 모두 거짓이면 거짓
!	!x	NOT 연산, x가 참이면 거짓, x가 거짓이면 참

306

조건 연산자(삼항 연산자)

- $x > y$ 가 참이면 x 가 수식의 값이 된다.
- $x > y$ 가 거짓이면 y 가 수식의 값이 된다.

```
maxValue = (x > y) ? x : y;
```

307

연산자 우선순위

우선순위	연산자	우선순위	연산자
1	. [] new	10	&
2	()	11	^
3	++ --	12	
4	! ~ + -(부호) typeof void delete	13	&&
5	* / %	14	
6	+ -(사칙연산자)	15	? : (삼항연산자)
7	<< >> >>>	16	yield
8	<<= >= in instanceof	17	= += -= *= /= %= <<= >= >>= &= ^= =
9	== != === !==	18	,

308

prompt() 함수

```
<script>
  var age = prompt("나이를 입력하세요", "만나이로 입력합니다.");
</script>
```



309

덧셈 예제1

```
<script>
  var x, y;
  var input;
  input = prompt("정수를 입력하시오", "정수로");
  x = parseInt(input);
  input = prompt("정수를 입력하시오", "정수로");
  y = parseInt(input);
  document.write(x + y + "<br>");
</script>
```

310

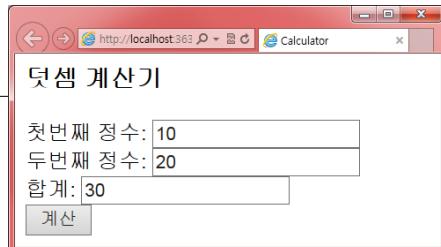
덧셈 예제2

```
<html>
<head>
    <title>Calculator</title>
    <script>
        function calc() {
            var x = document.getElementById("x").value;
            var y = document.getElementById("y").value;
            var sum;
            sum = parseInt(x) + parseInt(y);
            document.getElementById("sum").value = sum;
        }
    </script>
</head>
```

311

덧셈 예제2

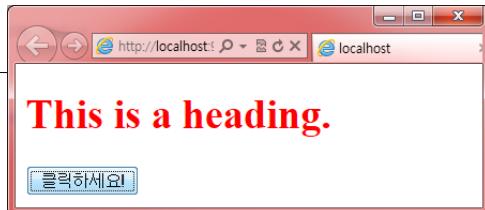
```
<body>
    <h3>덧셈 계산기</h3>
    <form name="myform" action="..." method="POST">
        첫번째 정수:
        <input id="x"><br>
        두번째 정수:
        <input id="y"><br>
        합계:
        <input id="sum"><br>
        <input type="button" value="계산" onclick="calc();">
    </form>
</body>
</html>
```



312

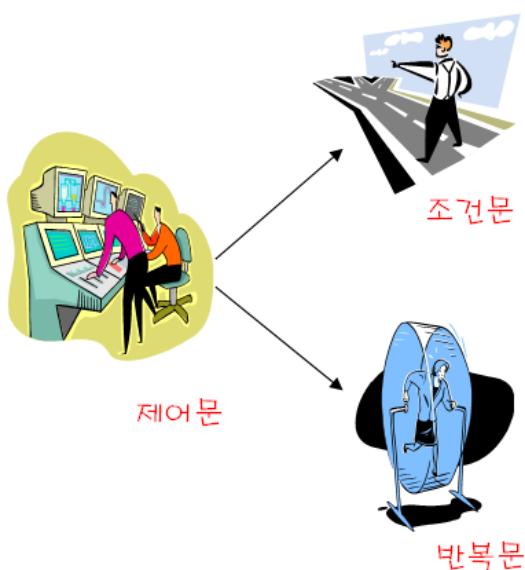
HTML 요소에 접근하기

```
<!DOCTYPE html>
<html>
<body>
  <h1 id="test">This is a heading.</h1>
  <script>
    function func() {
      e = document.getElementById("test");
      e.style.color = "red";
    }
  </script>
  <button type="button" onclick="func()">클릭하세요!</button>
</body>
</html>
```



313

제어문



```
if( 연봉 > 2500 )
  쥐업;
else
  고시 준비;
```

```
while( 토플성적 < 800 )
  영어공부;
```

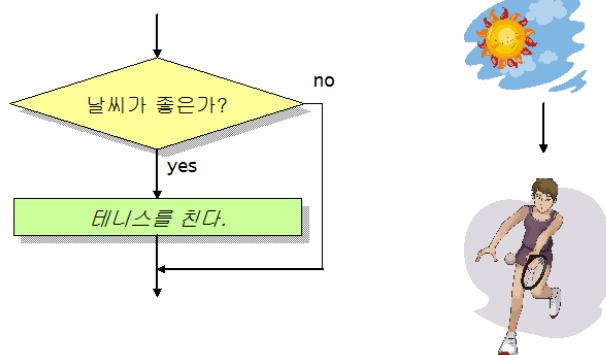
314

조건문의 종류

- if 문
- if else 문
- switch 문

315

if 문



```
if (time < 12) {  
    greeting = "Good Morning!";  
}
```

316

if-else 문

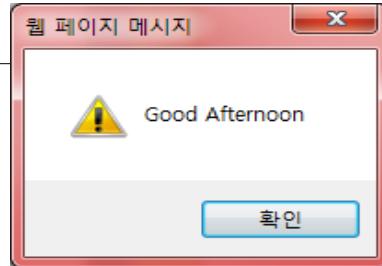
형식	<pre>if(조건식) { 문장 1; } else { 문장 2; }</pre>
설명	만약 조건식이 참이면 문장1이 실행된다. 그렇지 않으면 문장2가 실행된다.

```
if (time < 12) {
    msg = "Good Morning!";
} else {
    msg = "Good Afternoon!";
}
```

317

연속적인 if 문

```
<script>
    var msg = "";
    var time = new Date().getHours();
    if (time < 12) {           // 12시 이전이면
        msg = "Good Morning";
    } else if (time < 18) {     // 오후 6시 이전이면
        msg = "Good Afternoon";
    } else {                   // 그렇지 않으면(오후 6시 이후이면)
        msg = "Good evening";
    }
    alert(msg);
</script>
```



318

if 문제

- 숫자 2개와 연산자 1개를 입력 받아 연산자에 맞는 계산결과를 출력하는 프로그램을 작성하시오.



319

switch 문

- if문과 비슷하게 조건에 따라 프로그램의 흐름을 분기시키기 위해 사용된다.
- if문의 경우 조건식이 참이냐 거짓이냐에 따라서 실행할 문장이 둘 중의 하나로 결정되기 때문에 연속적인 if문을 쓸 경우에는 switch문을 사용하는 것이 좋다.
- switch문은 제어식의 값에 따라 다음에 실행할 문장을 결정하게 된다.

형식	<pre>switch(제어식) { case c1: 문장1; break; case c2: 문장2; break; default: 문장d; break; }</pre>
----	---

320

switch 문

```
<script>
    var grade = prompt("성적을 입력하시오:", "A-F사이의 문자로");
    switch (grade) {
        case 'A': alert("잘했어요!");
        break;
        case 'B': alert("좋은 점수군요");
        break;
        case 'C': alert("괜찮은 점수군요");
        break;
        case 'D': alert("좀더 노력하세요");
        break;
        case 'F': alert("다음학기 수강하세요");
        break;
        default: alert("알수없는 학점입니다.")
    }
</script>
```



321

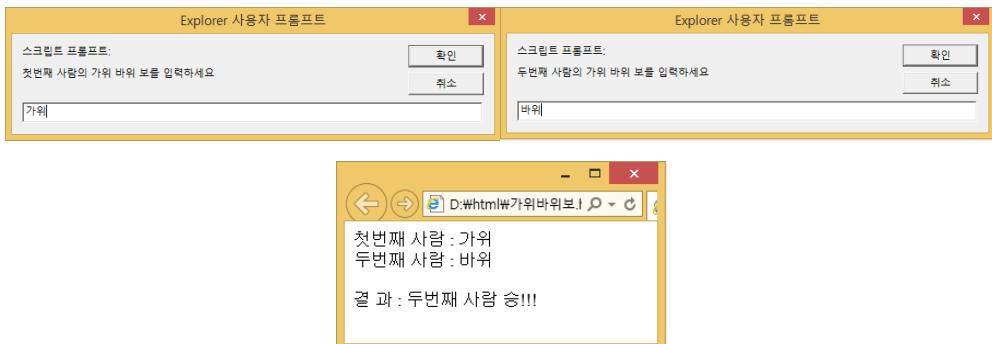
switch 문제

- 점수를 입력받아 학점을 출력하시오.(switch문을 이용)
 - 점수가 90 ~ 100이면 ‘A’
 - 점수가 80 ~ 89이면 ‘B’
 - 점수가 70 ~ 79이면 ‘C’
 - 점수가 60 ~ 69이면 ‘D’
 - 점수가 0 ~ 59이면 ‘F’
 - 출력은 document.write()를 이용

322

문제 1

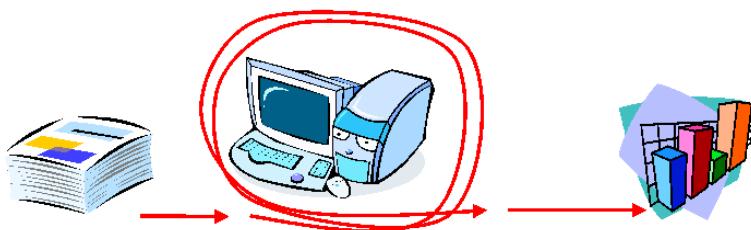
- 두 사람의 가위, 바위, 보를 입력 받아 승자를 출력하는 프로그램을 작성하시오.



323

반복문

- 같은 처리 과정을 여러 번 되풀이하는 것



324

반복문의 종류

- while - 지정된 조건이 참이면 반복 실행한다.
- for - 주로 정해진 횟수 동안 코드를 반복 실행한다.

325

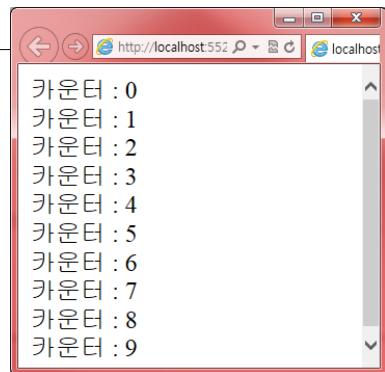
while 문

```
var i = 0;  
  
while( i < 10 ) ← 반복 조건, 변수 i가 10보다 작으면 반복 계속  
{  
    document.write(i + " <br>" ); ← 반복되는 문장, i를 화면에 출력한다.  
    i++; ← 한 번의 반복마다 i를 하나 증가시킨다.  
}
```

326

while 문

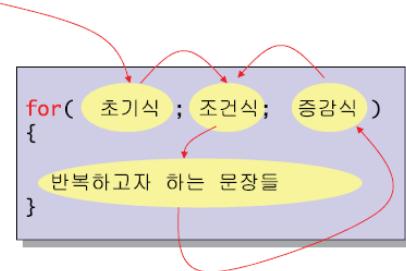
```
<script>
    var i = 0;
    while (i < 10) {
        document.write("카운터 : " + i + "<br />");
        i++;
    }
</script>
```



327

for 문

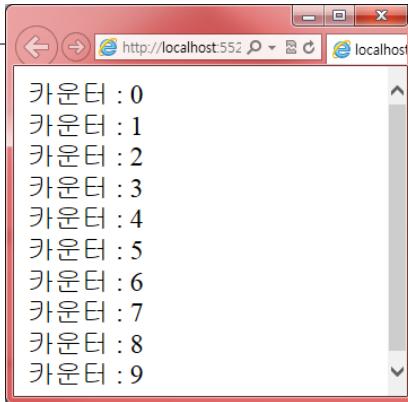
초기식 조건식 증감식
↓ ↓ ↓
for(i=0 ; i<10 ; i++)
{
 document.write(i+"
");
}



328

for 문

```
<script>
    for (var i = 0; i < 10; i++) {
        document.write("카운터 : " + i + "<br />");
    }
</script>
```



329

for문 예제

```
<html>
<head>
    <title>온도 변환기</title>
</head>
<body>
    <table border="3">
        <tr>
            <td>섭씨온도</td>
            <td>화씨온도</td>
        </tr>
        <script>
            for (celsius = 0; celsius <= 10; celsius = celsius + 1) {
                document.write("<tr><td>" + celsius + "</td><td>" +
                    ((celsius * 9.0 / 5) + 32) + "</td></tr>");
            }
        </script>
    </table>
</body>
</html>
```

섭씨온도	화씨온도
0	32
1	33.8
2	35.6
3	37.4
4	39.2
5	41
6	42.8
7	44.6
8	46.4
9	48.2
10	50

330

중첩 반복문

- 하나의 for문 안에 다른 for문이 내장될 수 있다.
- 반복문이 중첩될 때는 반복문 제어 변수로 서로 다른 변수를 사용해야 한다.

331

중첩 반복문 예제

```
<style>
    table, td {border:1px solid black;}
</style>
<script>
    document.write("<h1>구구단표</h1>");
    document.write("<table>");
    for (var i = 1; i <= 9; i++) {
        document.write("<tr>");
        document.write("<td>" + i + "</td>");
        for (var j = 2; j <= 9; j++) {
            document.write("<td>" + i * j + "</td>");
        }
        document.write("</tr>");
    }
    document.write("</table>");
</script>
```

The screenshot shows a web browser window with the URL http://localhost:552. The title of the page is '구구단표'. The content of the page is a 9x9 multiplication table where each cell contains the product of its row and column indices. The table is defined by a single outer loop (i) from 1 to 9, and each row is defined by an inner loop (j) from 2 to 9. The table starts with 1x1=1 at the top-left and ends with 9x9=81 at the bottom-right.

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

332

do/while문

- while문과 비슷하나 반복 조건을 처음이 아니라 끝에서 검사한다는 점이 다르다.
- do/while문은 일단 문장을 한 번 실행하고 나서 조건을 검사하고 싶을 때 사용한다.

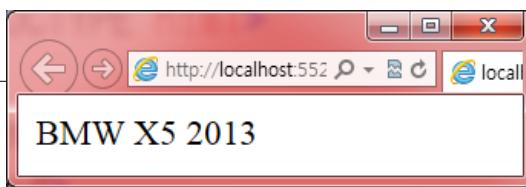
```
<script>
    var i = 0;
    do {
        document.write("카운터 : " + i + "<br>");
        i++;
    } while (i < 10);
</script>
```

333

for/in 반복문

- 객체 안의 속성들에 대하여 어떤 처리를 반복할 수 있는 구조
- for/in 반복문을 이용하면 객체 안의 모든 속성에 대하여 어떤 연산을 실행할 수 있다.

```
<script>
    var myCar = { make: "BMW", model: "X5", year: 2013 };
    var txt = "";
    for (var x in myCar) {
        txt += myCar[x] + " ";
    }
    document.write(txt);
</script>
```



334

break 문장

- 반복문을 벗어나기 위해 사용
- 반복문 안에서 break 문이 실행되면 반복문을 빠져나오게 된다.

```
<script>
  var msg = "";
  for (var i = 0; i < 10; i++) {
    if (i == 3) {
      break;
    }
    msg += i + "<br>";
  }
  document.write(msg);
</script>
```

335

continue 문장

- 현재 실행하고 있는 반복 과정의 나머지를 생략하고 다음 반복문을 시작하게 만든다.
- 예를 들어 0부터 10까지의 정수 중에서 3만 제외하고 출력하는 예제를 보면 0부터 10까지 정수를 하나씩 조사하다가 현재 정수가 3이면 continue를 실행해서 현재 반복을 중지하고 다음 반복을 시작한다.

```
<script>
  var msg = "";
  for (var i = 0; i < 10; i++) {
    if (i == 3) {
      continue;
    }
    msg += i + "<br>";
  }
  document.write(msg);
</script>
```

336

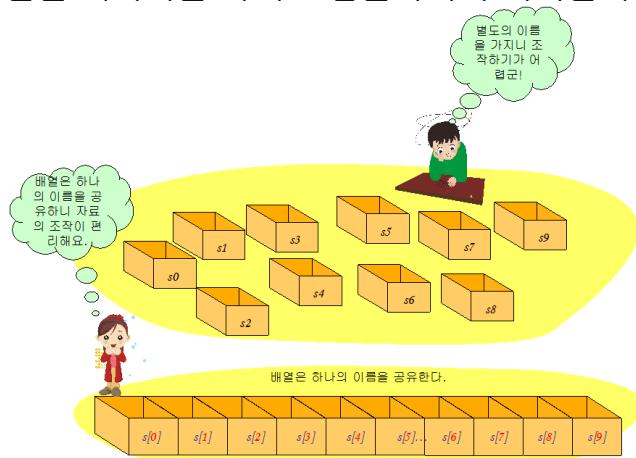
문제

1. 1부터 10까지의 합을 구하는 프로그램을 작성하시오.
2. 1부터 200까지의 짝수의 합을 구하는 프로그램을 작성하시오.(break나 continue를 이용)
3. 사용자가 입력한 값을 계속 더하고, 사용자가 0을 입력하면 그 때까지 누적된 값을 출력하는 프로그램을 작성하시오.

337

배열

- 많은 값을 저장할 수 있는 공간이 필요할 때 배열을 사용한다.
- 서로 관련된 데이터를 차례로 접근하여서 처리할 수 있다.



338

배열을 생성하는 2가지 방법

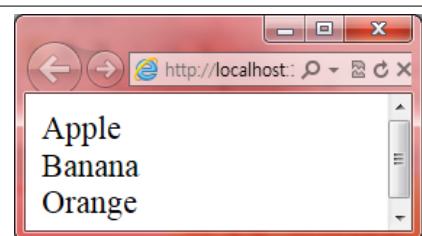
- 리터럴로 배열 생성
 - var fruits = ["apple", "banana", "peach"];
- Array 객체로 배열 생성
 - var fruits = new Array("apple", "banana", "orange");

339

예제

```
<!DOCTYPE html>
<html>
<body>
<script>
    var i;
    var fruits = new Array();
    fruits[0] = "Apple";
    fruits[1] = "Banana";
    fruits[2] = "Orange";

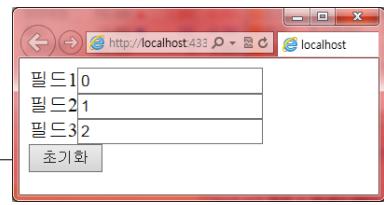
    for (i = 0; i < fruits.length; i++) {
        document.write(fruits[i] + "<br>");
    }
</script>
</body>
</html>
```



340

연관 배열

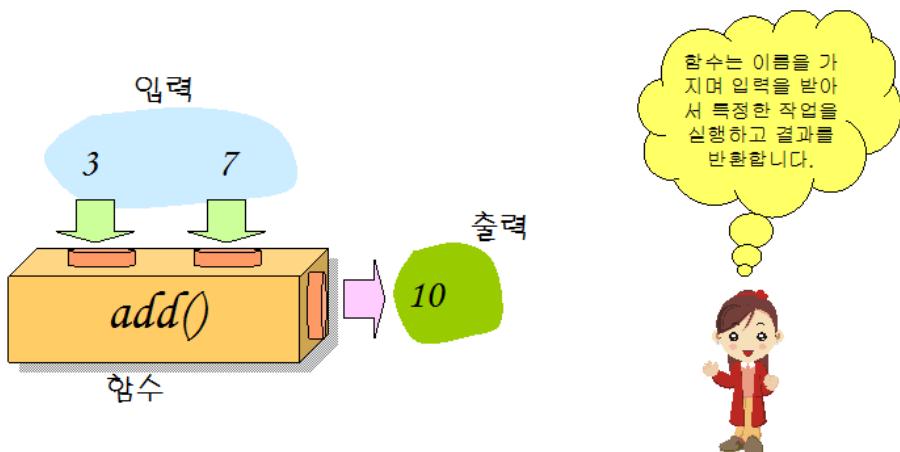
```
<!DOCTYPE html>
<html>
<body>
  <form name="myForm">
    필드1<input type="text" name="a0"><br>
    필드2<input type="text" name="a1"><br>
    필드3<input type="text" name="a2"><br>
    <input type="button" value="초기화" onclick="init();">
  </form>
  <script>
    function init() {
      for (var i = 0; i < 3; i++) {
        document.myForm["a" + i].value = i;
      }
    }
  </script>
</body>
</html>
```



341

함수

- 함수는 입력을 받아서 특정한 작업을 수행하여서 결과를 반환하는 블랙 박스



342

함수 만들기

1. 파라미터도 있고 반환 값도 있는 함수

```
function 함수명(파라미터1, 파라미터2, ...) {  
    return 반환값;  
}
```

2. 파라미터는 있고 반환 값은 없는 함수

```
function 함수명(파라미터1, 파라미터2, ...) {  
    명령문;  
}
```

3. 파라미터는 없고 반환 값은 있는 함수

```
function 함수명() {  
    return 반환값;  
}
```

4. 파라미터도 없고 반환 값도 없는 함수

```
function 함수명() {  
    명령문;  
}
```

343

함수의 호출

- 함수는 호출에 의해서 실행

```
function showDialog(para1, para2) {  
    명령문1;  
    명령문2;  
}  
  
showDialog(arg1, arg2);
```

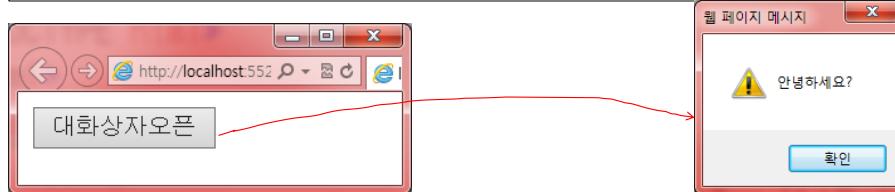
매개변수
인수

- 인수(argument) : 함수를 호출할 때는 어떤 값을 함수로 전달하는 값
- 인수는 데이터 타입이 없을 뿐만 아니라 개수에도 제약이 없다.
- 실 인수가 남으면 무시되고, 모자라는 인수는 undefined가 된다.
- 매개변수 (parameter) : 함수를 만들 때 인수로 받을 변수를 선언하는 것

344

예제

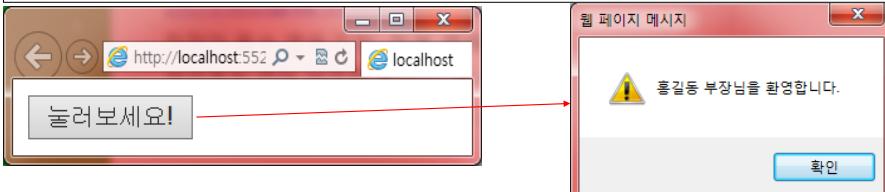
```
<!DOCTYPE html>
<html>
<head>
    <script>
        function showDialog() {
            alert("안녕하세요?");
        }
    </script>
</head>
<body>
    <button onclick="showDialog()">대화상자오픈</button>
</body>
</html>
```



345

인수와 매개 변수

```
<!DOCTYPE html>
<html>
<head>
    <script>
        function greeting(name, position) {
            alert(name + " " + position + "님을 환영합니다.");
        }
    </script>
</head>
<body>
    <button onclick="greeting('홍길동', '부장')">눌러보세요!</button>
</body>
</html>
```



346

무명 함수

- 함수를 만들어서 한번만 사용할 때 이름을 주지 않고 한번만 사용하는 경우 무명함수(anonymous function)라고 한다.

```
function showDialog(str) {  
    alert(str);  
}  
showDialog("안녕하세요.");
```

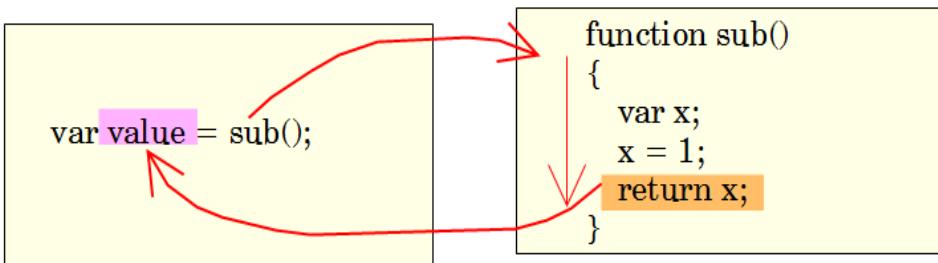


```
// 무명함수의 실행  
(function (str) {  
    alert(str);  
})( "안녕하세요" );
```

347

함수의 반환값

- return 문장을 사용하여 외부로 값을 반환



- 반환된 값을 어디에 저장하기 않고 바로 수식에 사용해도 된다.

```
document.getElementById("para1").innerHTML = sub();
```

348

함수의 반환값

- 단순히 함수를 종료하고 싶은 경우에도 사용할 수 있다.

```
function divide(a, b) {  
    if (b == 0) {  
        return;  
    }  
    return a / b;  
}
```

만약 분모가 0이면 나눗셈을 할 수 없으므로 함수를 종료한다.

349

지역변수

- 함수 안에서 선언된 변수
- 함수 안에서만 사용 가능
- 다른 함수에서도 똑같은 이름으로 선언이 가능함
- 지역변수는 함수가 종료되면 자동적으로 소멸된다.

```
function add(a, b) {  
    var sum = 0;  
  
    sum = a + b;  
    return sum;  
}
```

350

전역변수

- 함수 외부에서 선언된 변수
- 웹 페이지 상의 모든 스크립트와 모든 함수는 전역변수를 사용할 수 있다.
- 전역변수는 사용자가 웹페이지를 닫으면 소멸된다.

```
var sum = 0;

function add(a, b) {
    sum = a + b;
    return sum;
}
```

351

전역변수

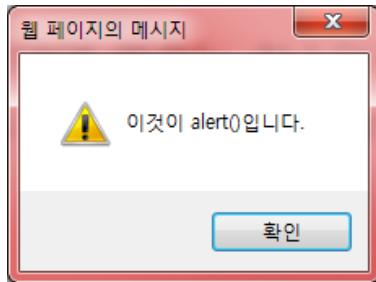
- 선언되지 않은 변수에 값을 대입하면 그 변수는 자동적으로 전역 변수가 된다.
- 예를 들면 다음과 같은 문장은 함수 안에서 실행되더라도 변수 userName을 전역변수로 선언하는 것이나 마찬가지이다.

```
function add(a, b) {
    userName = "초파";
    //var userName = "나초";
    document.write(userName);
}
```

352

alert() 함수

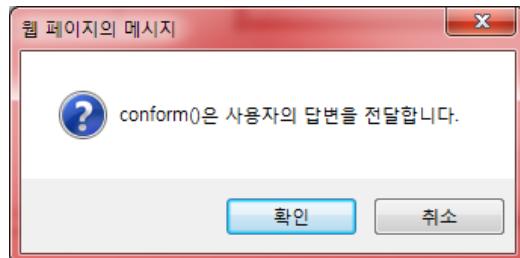
```
<script>
    alert("이것이 alert()입니다.");
</script>
```



353

confirm() 함수

```
<script>
    var user = confirm("confirm()은 사용자의 답변을 전달합니다.");
</script>
```



354

prompt() 함수

```
<script>
  var age = prompt("나이를 입력하세요", "만나이로 입력합니다.");
</script>
```

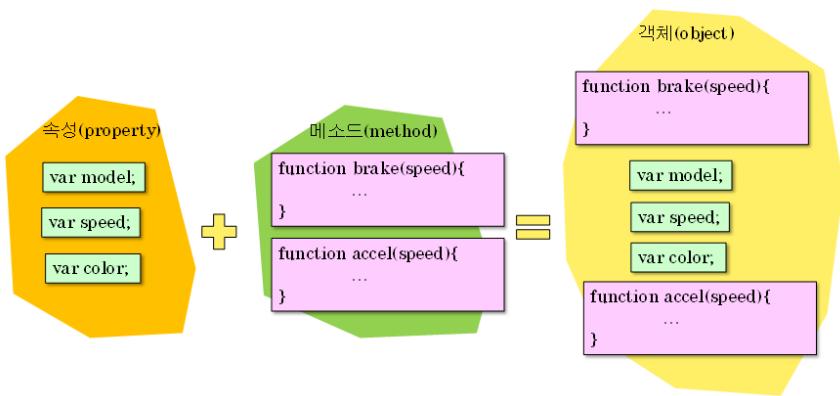


355

09 자바 스크립트 객체

객체

- **객체(object)**는 사물의 속성과 동작을 묶어서 표현하는 기법
- (예) 자동차는 메이커, 모델, 색상, 마력과 같은 속성도 있고 출발하기, 정지하기 등의 동작도 가지고 있다.



357

객체의 종류

- 객체의 2가지 종류
 - **내장 객체(bult-in object)**: 생성자가 미리 작성되어 있다.
 - **사용자 정의 객체(custom object)**: 사용자가 생성자를 정의한다.
- 내장 객체들은 생성자를 정의하지 않고도 사용이 가능하다. `Date`, `String`, `Array`와 같은 객체들이 내장 객체이다.

358

객체 생성 방법

- 객체를 생성하는 2가지 방법
 - 객체를 객체 리터럴로부터 직접 생성한다.
 - 생성자 함수를 이용하여 객체를 정의하고 new 연산자를 통하여 객체의 인스턴스를 생성한다.

359

객체 상수로부터 객체 생성

```
var myCar = {  
    model: "520d",  
    speed: 60,  
    color: "red",  
    brake: function () { this.speed -= 10; },  
    accel: function () { this.speed += 10; }  
};
```

객체의 속성

객체의 메소드

```
myCar.color = "yellow";  
myCar.brake();
```

360

생성자를 이용한 객체 생성

```
생성자도 함수  
이다.  
  
this 키워드로  
일반 변수와  
객체 속성을  
구별한다.  
  
function Car(model, speed, color) {  
    this.model = model;  
    this.speed = speed;  
    this.color = color;  
    this.brake = function () {  
        this.speed -= 10;  
    }  
    this.accel = function () {  
        this.speed += 10;  
    }  
}
```

생성자 이름은 항상 대문자로 한다.

객체의 속성

객체의 메소드

361

생성자를 이용한 객체 생성

이 변수에
객체가 저
정된다.

new 연산자는 새로운 객체를
만든다.

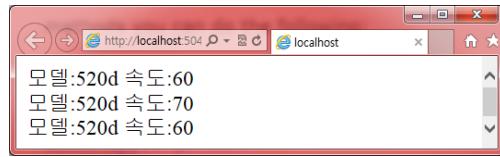
이 인수들
로 객체가
초기화된다.

```
var myCar = new Car("520d", 60, "white");  
myCar.color = "yellow";  
myCar.brake();
```

362

객체 생성 예제

```
<!DOCTYPE html>
<html>
<body>
<script>
    function Car(model, speed, color) {
        this.model=model;
        this.speed=speed;
        this.color = color;
        this.brake = function () {
            this.speed -= 10;
        }
        this.accel = function () {
            this.speed += 10;
        }
    }
    myCar = new Car("520d", 60, "red");
    document.write("모델:" + myCar.model + " 속도:" + myCar.speed + "<br>");
    myCar.accel();
    document.write("모델:" + myCar.model + " 속도:" + myCar.speed + "<br>");
    myCar.brake();
    document.write("모델:" + myCar.model + " 속도:" + myCar.speed + "<br>");
</script>
</body>
</html>
```



363

객체에 속성과 메소드 추가

- 기존에 존재하고 있던 객체에도 속성을 추가할 수 있다.
- 생성자 함수는 변경할 필요가 없다.

```
myCar.turbo = true;
myCar.showModel = function () {
    alert("모델은 " + this.model + "입니다.")
}
```

364

자바 스크립트 내장 객체

- Array 객체
- Date 객체
- Number 객체
- String 객체
- Math 객체
- ...

365

Array 객체

- 배열을 나타내는 객체

```
var myArray = new Array();
myArray[0] = "apple";
myArray[1] = "banana";
myArray[2] = "orange";
```

- 배열의 크기가 자동으로 조절된다. 다른 언어에서는 배열의 크기가 고정되어 있다. 하지만 자바스크립트에서 배열의 크기는 현재 배열의 크기보다 큰 인덱스를 사용하면 자동으로 증가한다.
- 자바스크립트에서는 하나의 배열에 여러 가지 자료형을 혼합해서 저장할 수 있다. 즉 데이터의 종류를 다르게 하면서 배열에 저장할 수 있는 것이다. 하나의 배열에 정수와 문자열을 동시에 저장하는 것이 가능하다.

366

예제

```
<html>
<head>
<script>
function printArray(a) {
    document.write("[ ");
    for (var i = 0; i < a.length; i++) {
        document.write(a[i] + " ");
    }
    document.write(" ] <br>");
}
var myArray1 = new Array();
myArray1[0] = "apple";
myArray1[1] = "banana";
myArray1[2] = "orange";
var myArray2 = new Array("apple", "banana", "orange");
var myArray3 = ["apple", "banana", "orange"];
printArray(myArray1);
printArray(myArray2);
printArray(myArray3);
</script>
</head>
<body>
</body>
</html>
```



367

Array 객체의 메소드

- 속성
 - length
- 메소드

메서드	설명
indexOf(item, start)	배열에서 요소를 찾아 위치를 리턴한다.
lastIndexOf(item, start)	역순으로 요소를 찾아 위치를 리턴한다.
push(a,b,c,⋯)	배열 끝에 요소를 추가한다.
pop()	마지막 요소를 제거하고 리턴한다.
shift()	배열 처음의 원소를 제거하고 리턴한다.
unshift(a,b,c,⋯)	배열 처음에 요소를 추가한다.

368

Array 객체의 메소드

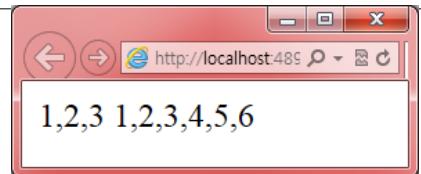
메소드	설명
sort(sortfunction)	배열을 정렬한다. 인수로 값을 비교하는 함수를 지정할 수 있으며 생략시 사전순으로 정렬된다.
splice(index, n, a, b, c, ...)	배열 일부를 수정한다. 일정 범위를 삭제하고 새로운 요소를 삽입한다.
join(deli)	배열 요소를 하나의 문자열로 합친다. 구분자를 지정할 수 있으며 생략시 콤마로 구분한다.

369

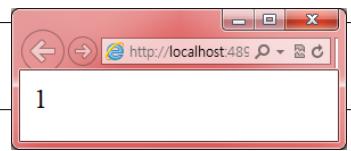
예제

```
<script>
  var x = [1, 2, 3];
  var y = [4, 5, 6];
  var joined = x.concat(y);

  document.writeln(x);    // 출력: 1,2,3
  document.writeln(joined); // 출력: 1,2,3,4,5,6
</script>
```



```
<script>
  var fruits = ["apple", "banana", "grape"];
  document.writeln(fruits.indexOf("banana")); // 출력: 1
</script>
```



370

예제

```
<script>
    var numbers = [1, 2, 3, 4, 5];

    numbers.push(6);
    document.writeln(numbers + '<BR>'); // 출력: 1,2,3,4,5,6
    item = numbers.pop();
    document.writeln(numbers + '<BR>'); // 출력: 1,2,3,4,5,
</script>
```

```
<script>
    var numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

    var item = numbers.shift();
    document.writeln(item + '<BR>'); // 출력: 1
    document.writeln(numbers + '<BR>'); // 출력: 2,3,4,5,6,7,8,9,10
</script>
```

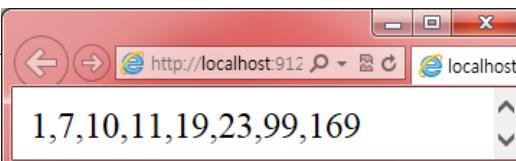
371

예제

```
<script>
    var myArray = [10, 7, 23, 99, 169, 19, 11, 1];
    myArray.sort()
    document.writeln(myArray);
</script>
```



```
<script>
    var myArray = [10, 7, 23, 99, 169, 19, 11, 1];
    myArray.sort(function (a, b) { return a - b });
    document.writeln(myArray);
</script>
```



372

Array 객체 문제

1. 사람 이름을 계속 입력 받아 배열에 저장하고 그 저장된 이름을 출력하는 프로그램을 작성하시오.(단, 입력은 prompt 명령을 이용하고, 입력의 마지막은 공백문자를 입력하거나 "취소" 버튼을 눌렀을 때로 한다. "취소" 버튼은 null 또는 "null"이 입력될 때이다.)
2. 서로 중복되지 않은 정수 5개를 입력 받아 출력하는 프로그램을 작성하시오.

373

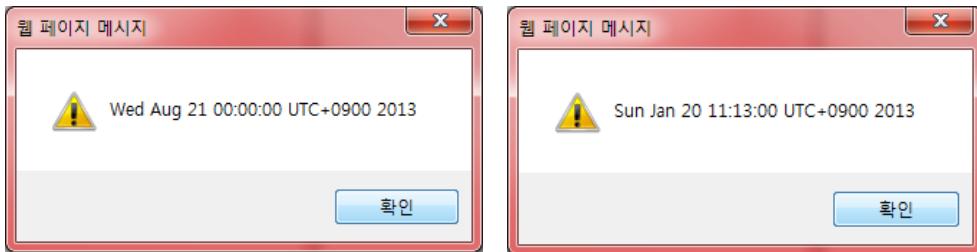
Date 객체

- Date 객체는 날짜와 시간 작업을 하는데 사용되는 가장 기본적인 객체
 - new Date() // 현재 날짜와 시간
 - new Date(milliseconds) //1970/01/01 이후의 밀리초
 - new Date(dateString)// 다양한 문자열
 - new Date(year, month, date[, hours[, minutes[, seconds[, ms]]]])

374

예제

```
<script>
var d1 = new Date(2013, 7, 21, 0, 0, 0);
var d2 = new Date("January 20, 2013 11:13:00");
alert(d1);
alert(d2);
</script>
```



375

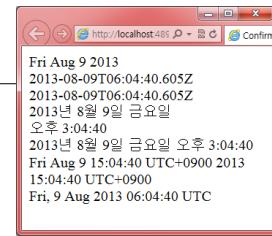
Date 객체의 메소드

get 함수명	반환값	set 함수명
getDay()	0(일요일) ~ 6(토요일)	setDay(day)
getDate()	1 ~ 31	setDate(date)
getMonth()	0 ~ 11	setMonth(month-1)
getFullYear()	4개의 숫자로 된 연도	setYear(year)
getHours()	0 ~ 23	setHours(hours)
getMinutes()	0 ~ 59	setMinutes(minutes)
getSeconds()	0 ~ 59	setSeconds(seconds)
getMilliseconds()	0 ~ 999	setMilliseconds(millisec)
getTime()	경과시간(milliseconds 단위)	setTime(millisec)

376

예제

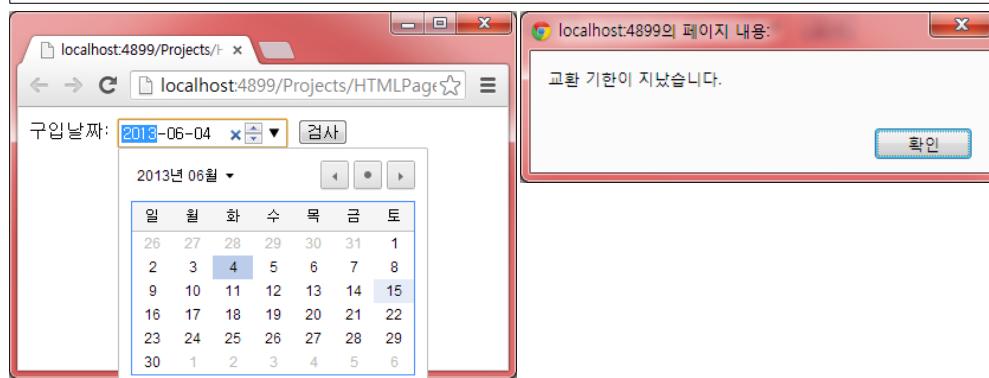
```
<script>
    var today = new Date();
    document.write(today.toDateString() + "<br>");
    document.write(today.toISOString() + "<br>");
    document.write(today.toJSON() + "<br>");
    document.write(today.toLocaleDateString() + "<br>");
    document.write(today.toLocaleTimeString() + "<br>");
    document.write(today.toLocaleString() + "<br>");
    document.write(today.toString() + "<br>");
    document.write(today.toTimeString() + "<br>");
    document.write(today.toUTCString() + "<br>");
</script>
```



377

예제

```
<body>
    구입날짜:
    <input type="date" id="pdate">
    <button onclick="checkDate()">검사</button>
</body>
</html>
```



378

Date 객체 문제

1. 2010년 1월 1일부터 오늘까지의 날짜수를 계산하는 프로그램을 작성하시오.(소수점은 parseInt()을 이용하여 버린다.)
2. 오늘부터 500일 후의 날짜를 출력하는 프로그램을 작성하시오.

379

Number 객체

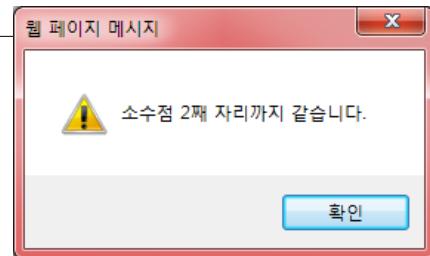
- Number 객체는 수치형 값을 감싸서 객체로 만들어 주는 랩퍼(wrapper) 객체

```
var num = new Number(7);
```
- 속성
 - MAXVALUE : 표현할 수 있는 가장 큰 값
 - MINVALUE : 표현할 수 있는 가장 작은 값
 - NaN : "Not a Number"의 약자
- 메소드
 - toExponential([digits]) : 지수형으로 반환, 인수는 소수점 이하 개수
 - toFixed([digits]) : 고정소수점 방식으로 반환, 인수는 소수점 이하 개수
 - toPrecision([precision]) : 유효숫자 수를 지정
 - toString([radix]) : 주어진 진법으로 숫자를 반환

380

예제

```
<script>
    var count1, count2;
    count1 = new Number(1.237);
    count2 = 1.238;
    if (count1.toFixed(2) === count2.toFixed(2)) {
        alert("소수점 2째 자리까지 같습니다.");
    }
</script>
```



381

String 객체

- 속성
 - length : 문자열의 길이.
- 메소드

메서드	설명
charAt(index)	index 위치의 문자를 구한다. index가 문자열의 범위를 벗어나면 빈 문자열이 리턴된다.
charCodeAt(index)	index 위치의 문자에 대한 유니코드를 구한다.
indexOf(searchvalue, start)	부분 문자열의 위치를 검색한다. start는 검색 시작 위치이며 생략 시 0이 적용되어 처음부터 검색한다. 없을 경우 -1을 리턴한다.
lastIndexOf(searchvalue, start)	부분 문자열의 위치를 역방향에서 검색한다. start는 검색 시작 위치이며 생략 시 문자열의 제일 끝이 적용된다. 없을 경우 -1을 리턴한다.
concat(s1, s2, ...)	여러 개의 문자열을 연결한다. + 연산자와 동일하다.
trim()	앞 뒤의 공백을 제거한다.

382

String 메소드

메소드	설명
toLowerCase()	소문자로 변환한다
toUpperCase()	대문자로 변환한다
replace(search,value)	문자열을 대체한다. 정규식도 사용 가능
search(searchvalue)	부분 문자열 또는 정규식을 검색하여 그 위치를 리턴한다.
match(regexp)	정규식으로 검색하여 일치하는 결과를 배열로 리턴한다. 발견되지 않으면 null을 리턴한다.
substring(from, to)	두 위치 사이의 부분 문자열을 추출한다. to를 생략하면 뒤쪽 모든 문자열을 추출한다.
slice(start, end)	start 위치에서 end 위치까지 부분 문자열을 추출한다. 음수로 끝에서부터 위치를 지정할 수 있다.
substr(start, length)	start에서 시작하여 length 길이만큼 부분 문자열을 추출한다. 길이를 생략하면 뒤쪽 모든 문자열을 추출한다.
split(separator, limit)	구분자로 구분된 문자열을 분리하여 배열로 리턴한다. limit는 최대 몇 개까지 리턴할 것인가를 지정한다.

383

글자 위치 찾기

앞에서 찾기

0	1	2	3	4	5	6	7	8	9	10	11	12	13
우	리	나	라		대	한	민	국		좋	은	나	라

↑
찾음

↑
찾음

뒤에서 찾기

```
var s = "우리나라 대한민국 좋은나라";
s.charAt(2);
s.charCodeAt(2);
s.indexOf("나라");
s.lastIndexOf("나라");
```

384

글자 추출

찾음													
0	1	2	3	4	5	6	7	8	9	10	11	12	13
우	리	나	라		대	한	민	국		좋	은	나	라

```
var s = "우리나라 대한민국 좋은나라";
s.substring(3, 6);
s.substr(3, 6);
s.slice(3, 6);
s.slice(3, -2);
```

385

예제

```
<script>
var s = '독도는 일본땅';
document.write(s + "이라는 말도 안되는 소리를 바꿔줍니다.<br>");
document.write("검색 위치 : " + s.search("일본") + "<br>");
document.write("진실된 말 : " + s.replace("일본", "한국") + "<br>");
</script>
```

386

예제

```
<script>
  var s = 'aBcDeF';
  var result1 = s.toLowerCase();
  var result2 = s.toUpperCase();
  document.write(result1); // 출력: abcdef
  document.write(result2); // 출력: ABCDEF
</script>
```

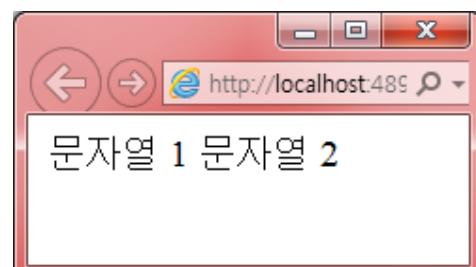


387

예제

```
<script>
  var s1 = " 문자열 1 ";
  var s2 = " 문자열 2 ";

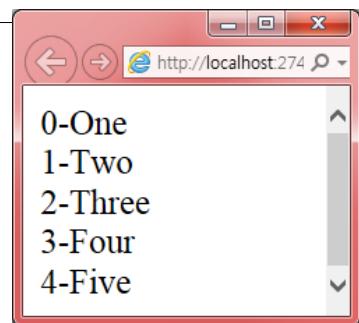
  s3 = s1.concat(s2);
  document.writeln(s3 + '<br>'); // "문자열 1 문자열 2"
</script>
```



388

예제

```
<script>
    s = "One,Two,Three,Four,Five";
    array = s.split(',');
    for (i = 0; i < array.length; i++) {
        document.writeln(i + '-' + array[i] + '<BR>');
    }
</script>
```



389

예제

```
<script>
    var s = "This is a test.";
    document.write("Big: " + s.big() + "<br>");
    document.write("Small: " + s.small() + "<br>");
    document.write("Bold: " + s.bold() + "<br>");
    document.write("Italic: " + s.italics() + "<br>");
    document.write("Fixed: " + s.fixed() + "<br>");
    document.write("Strike: " + s.strike() + "<br>");
    document.write("Fontcolor: " + s.fontcolor("green") + "<br>");
    document.write("FontSize: " + s.fontSize(6) + "<br>");
    document.write("Subscript: " + s.sub() + "<br>");
    document.write("Superscript: " + s.sup() + "<br>");
    document.write("Link: " + s.link("http://www.google.com") +
    "<br>");
```

390

String 객체 문제

- 주민등록번호를 입력 받아 생년월일과 성별을 출력하는 프로그램을 작성하시오.(입력은 prompt로 입력받는다.)
 - 예) 주민등록번호를 110326-4432618로 입력 받은 경우
 - 생일 : 2011년 3월 26일
 - 성별 : 여자
- 주민등록번호를 입력 받아 주민등록번호의 유효성을 검사하는 프로그램을 작성하시오.(ABCDEF-GHIJKLM)
 - $A*2 + B*3 + \dots + H*9 + I*2 + \dots + L*5$ 의 총합을 구한다.
 - 1번의 합을 11로 나눈 나머지를 구한다.
 - 11에서 2번의 결과를 뺀다.
 - 3번의 결과가 0~9이면 값 그대로, 10이면 0, 11이면 1로 변환
 - 4번의 결과와 M자리의 값이 같으면 맞는 번호이다.

391

Math 객체 속성

속성	설명
E	오일러의 상수(약 2.718)
LN2	자연 로그(밑수:2)(약 0.693)
LN10	자연 로그(밑수:10)(약 2.302)
PI	파이 상수(약 3.14)
SQRT1_2	$\frac{1}{2}$ 의 제곱근(약 0.707)
SQRT2	2의 제곱근(약 1.414)

392

Math 객체 메소드

메소드	설명
abs(x)	절대값
acos(x), asin(x), atan(x)	아크 삼각함수
ceil(x), floor(x)	실수를 정수로 올림, 내림 함수
cos(x), sin(x), tan(x)	삼각함수
exp(x)	지수함수
log(x)	로그함수
max(x,y,z,⋯,n)	최대값
min(x,y,z,⋯,n)	최소값
pow(x,y)	지수함수 X^y
random()	0과 1사이의 난수값 반환
round(x)	반올림
sqrt(x)	제곱근

393

원하는 범위의 랜덤값 만들기

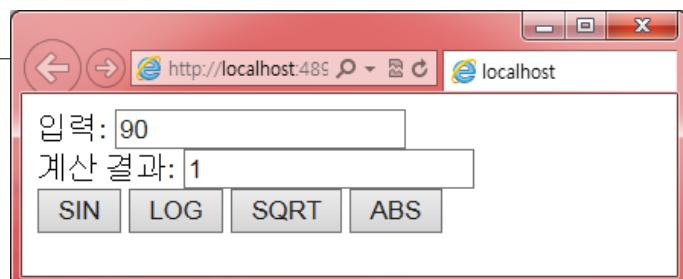
- `Math.floor(Math.random() * (최대값 - 최소값+1) +최소값);`
- `Math.round(Math.random() * (최대값 - 최소값) + 최소값);`
- 예) 1부터 10까지의 랜덤수 만들기

```
var ranNum = Math.floor(Math.random() * (10 - 1 + 1) + 1);
document.write(ranNum);
```

394

예제

```
<body>
  <form name="calculator">
    입력: <input type="text" name="number1"><br>
    계산 결과: <input type="text" name="total"><br>
    <input type="button" value="SIN" onclick="calc(1);">
    <input type="button" value="LOG" onclick="calc(2);">
    <input type="button" value="SQRT" onclick="calc(3);">
    <input type="button" value="ABS" onclick="calc(4);">
  </form>
</body>
</html>
```



395

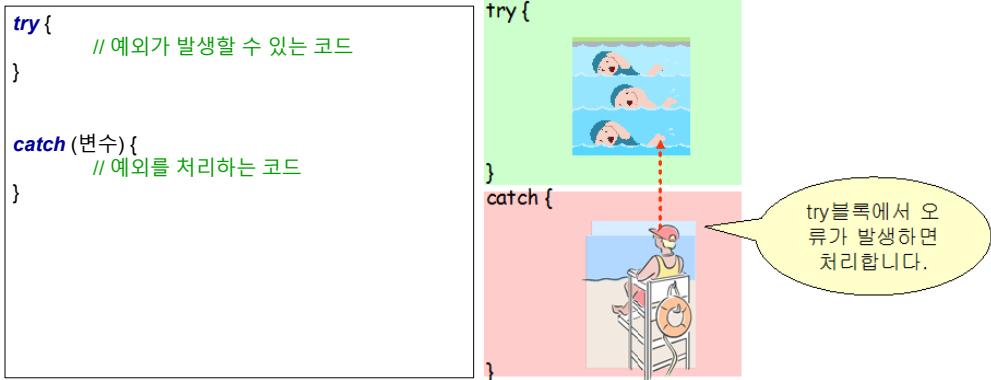
Math 객체 문제

- 1 ~ 10 사이의 난수를 발생 후 사용자가 이 값을 맞추는 프로그램을 작성하시오.
- 가위 바위 보 게임을 할 수 있는 프로그램을 작성하시오. (단, 컴퓨터는 랜덤을 이용하고, 사용자는 prompt로 입력 받아서 처리)
- 로또 번호를 생성하는 프로그램을 작성하시오.(1번 ~ 45번 중 6개의 번호를 추첨)

396

오류 처리

- 자바스크립트에서의 예외 처리기는 try 블록과 catch 블록으로 이루어진다.



397

예제

The screenshot shows a browser window with the URL `http://localhost:386`. A button labeled "try-catch 시험" is clicked. An alert box appears with the message: "다음과 같은 오류가 발생하였음: " + error.message;. The browser's status bar shows the message: "다음과 같은 오류가 발생하였음: 'alert'이(가) 정의되지 않았습니다." (An error occurred: 'alert' is not defined).

```
<!DOCTYPE html>
<html>
<head>
<script>
var msg = "";
function test() {
    try {
        alert("Hello World!");
    } catch (error) {
        msg = "다음과 같은 오류가 발생하였음: " + error.message;
        alert(msg);
    }
}
</script>
</head>
<body>
<input type="button" value="try-catch 시험" onclick="test()" />
</body>
</html>
```

398

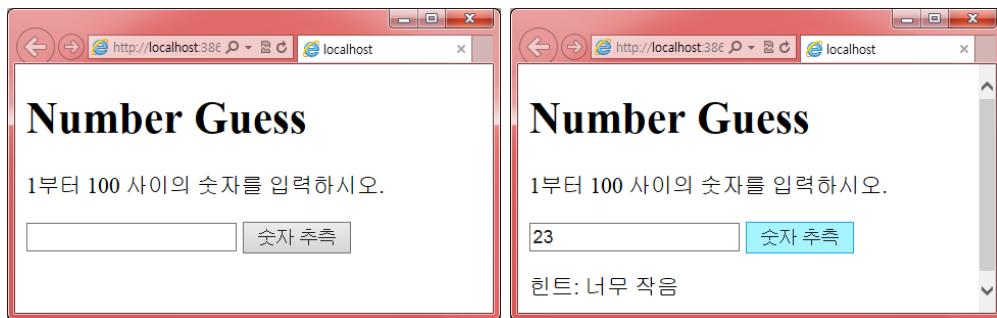
throw 문장

- throw 문장은 개발자가 오류를 생성할 수 있도록 한다.
- throw 문장을 사용하여서 오류 처리를 이용할 수도 있다.
 - (예) 숫자 맞추기 게임

399

예제

```
<h1>Number Guess</h1>
<p>1부터 100 사이의 숫자를 입력하시오.</p>
<input id="number" type="text">
<button type="button" onclick="test()">숫자 추측</button>
<p id="message"></p>
</body>
</html>
```

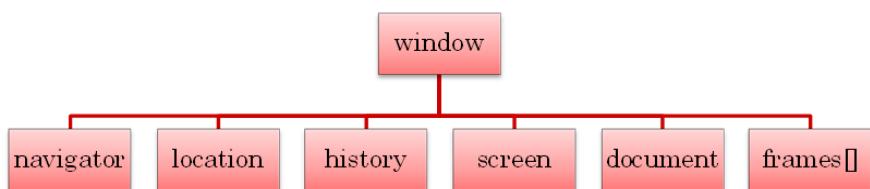


400

10 BOM과 DOM

브라우저 객체 모델

- 브라우저 객체 모델(BOM: Browser Object Model): 웹 브라우저가 가지고 있는 모든 객체를 의미
- 최상위 객체는 window이고 그 아래로 navigator, location, history, screen, document, frames 객체가 있다.



window 객체의 하위 객체

- window - 웹 브라우저 최상위 객체
- navigator - 브라우저에 대한 정보(버전 번호와 같은 정보들)
- screen - 사용자 화면
- history - 사용자가 방문한 URL 기록
- location - 현재 URL에 대한 정보
- frames - 브라우저 윈도우를 차지하고 있는 프레임들(현재는 잘 사용하지 않음)
- document - 메인 브라우저에 표시된 HTML 문서

403



window 객체

- 브라우저 모델 객체(BOM)에서 최상위 객체로서 웹 브라우저 윈도우를 나타낸다.
- 모든 전역 자바스크립트 객체, 함수, 변수는 자동적으로 window 객체의 멤버(속성)가 된다.
- 모든 전역 변수는 윈도우 객체의 속성이고 모든 함수는 윈도우 객체의 메서드이다.
- window 객체는 최상위 객체로서 생략이 가능하다.
- window 객체 속성
 - opener : open()을 통해 새로운 창을 열었을 때 그 창을 자식창이라 한다면 자식창에서 부모창을 가리킬 때 opener라 한다.

405

window 객체 함수

- open() : 새로운 창을 연다.
- close() : 열려진 창을 닫는다.
- alert(내용) : 내용을 나타내는 경고 창이 뜬다.(확인버튼)
- confirm(내용) : 내용을 나타내는 확인 창이 뜬다(확인,취소버튼)
- prompt(메시지, 초기값) : 메시지와 초기값을 나타내고 새로운 값을 입력할 수 있는 창이 뜬다.
 - 확인 버튼은 입력 값, 취소버튼은 null 또는 "null"을 반환한다.
 - 입력 값은 문자열(String)로 반환된다.
- setTimeout(function, millisecond) : 주어진 시간이 경과하면 지정된 함수가 호출되어 실행된다.
- setInterval(function, millisecond) : 주어진 시간이 경과할 때마다 지정된 함수가 호출되어 실행된다.

406

window 객체 함수

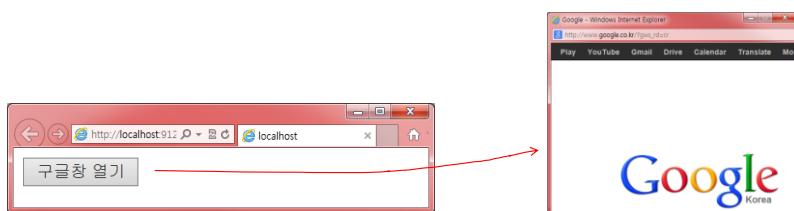
- clearInterval(시간변수) : setInterval() 메소드를 종료시킨다.
- moveTo(x, y) : 절대적인 위치 x와 y로 이동
- moveBy(dx, dy) : 상대적인 위치로 x와 y값 만큼 이동
- resizeTo(x, y) : x와 y값으로 창의 크기를 재조정
- resizeBy(dx, dy) : 현재 크기에서 x와 y값 만큼 크기를 재조정
- scrollTo(x, y) : 스크롤을 x와 y로 이동
- scrollBy(dx, dy) : 스크롤을 x와 y값 만큼 이동
- focus() : 특정 윈도우로 키보드 포커스를 이동

407

새로운 윈도우 오픈 예제

```
window.open(URL, name, specs, replace);
```

- URL : 오픈할 페이지의 URL
- name : 타겟(target)을 지정하거나 윈도우의 이름
- specs : 여러가지 속성
- replace : 히스토리 리스트에서 새로운 엔트리인지 아니면 현재 엔트리를 대체하는지 여부



408

새로운 윈도우 오픈 예제

```
<!DOCTYPE html>
<html>
<head>
<script>
function openPopup() {
    open("http://www.google.co.kr", "_blank", "width=200,height=200", true);
}
</script>
</head>
<body>
<input type="button" value="구글창 열기" onclick="openPopup();">
</body>
</html>
```



409

setTimeout()

setTimeout(code, millisec)

호출되는 함수의 이름. 호출되는 함수를 여기서 직접 정의할 수도 있다.

함수를 호출하기 전에 풀어야 하는 시간

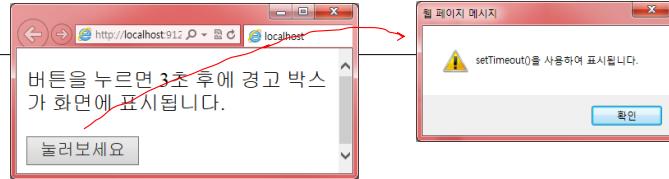
setTimeout()은 지정된 시간이 지난 후에 한번만 호출합니다.



410

예제

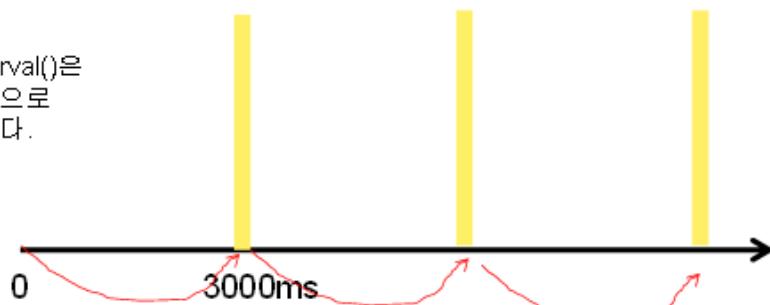
```
<!DOCTYPE html>
<html>
<head>
    <script>
        function showAlert() {
            setTimeout(function () { alert("setTimeout()을 사용하여
표시됩니다."), 3000);
        }
    </script>
</head>
<body>
    <p>버튼을 누르면 3초 후에 경고 박스가 화면에 표시됩니다.</p>
    <button onclick="showAlert()">눌러보세요</button>
</body>
</html>
```



411

setInterval()

setInterval()은
주기적으로
호출한다.



412

예제

```
<!DOCTYPE html>
<html>
<head>
<script>
    function changeColor() {
        id = setInterval(flashText, 500);
    }
    function flashText() {
        var elem = document.getElementById("target");
        elem.style.color = (elem.style.color == "red") ? "blue" : "red";
        elem.style.backgroundColor =
            (elem.style.backgroundColor == "green") ? "yellow" : "green";
    }
    function stopTextColor() {
        clearInterval(id);
    }
</script>
</head>
```

413

예제

```
<body onload="changeColor();">
    <div id="target">
        <p>This is a Text.</p>
    </div>
    <button onclick="stopTextColor();">중지</button>
</body>
</html>
```



414

문제

- 1초마다 브라우저 배경색이 랜덤으로 바뀌는 프로그램을 작성하시오.
- 참고 : HTML에서 색 지정 방법
 - 색 이름을 영문으로 지정(black, pink, orange, yellow, blue, red···)
 - #RRGGBB : 16진수로 표현
 - 10진수를 16진수로 표현하는 방법
 - var a = 230;
 - a.toString(16);
 - 만약에 파랑을 만들고 싶으면 var red = 0, green = 0, blue = 255;
 - "#" + red.toString(16) + green.toString(16) + blue.toString(16);

415

screen 객체

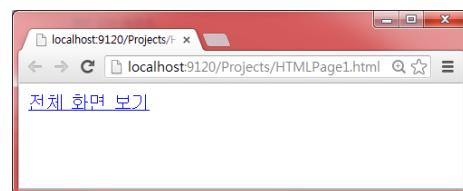
속성	설명
availHeight	화면의 높이를 반환(윈도우에서 태스크바를 제외한 영역)
availWidth	화면의 너비를 반환(윈도우에서 태스크바를 제외한 영역)
colorDepth	컬러 팔레트의 비트 깊이를 반환
height	화면의 전체 높이를 반환
pixelDepth	화면의 컬러 해상도(bits per pixel)를 반환
width	화면의 전체 너비를 반환

416

예제

```
<script>
    function maxopen(url, winattributes) {
        var maxwindow = window.open(url, "", winattributes)
        maxwindow.moveTo(0, 0);
        maxwindow.resizeTo(screen.availWidth, screen.availHeight)
    }

</script>
<a href="#" onClick="maxopen('http://www.google.com', 'resize=1,
scrollbars=1, status=1'); return false;">전체 화면 보기</a>
```



417

location 객체

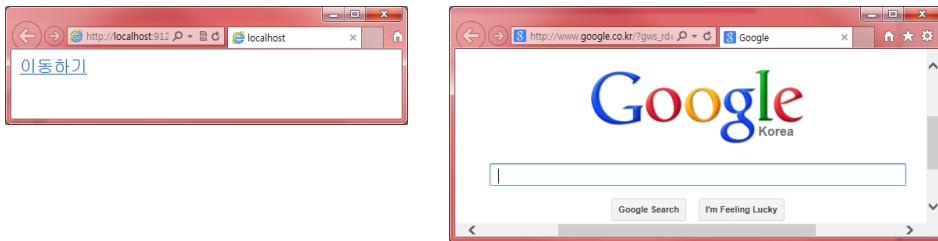
속성	설명
hash	URL 중에서 앵커 부분을 반환한다.
host	URL 중에서 hostname과 port를 반환
hostname	URL 중에서 hostname을 반환
href	전체 URL을 반환
pathname	URL 중에서 경로(path)를 반환
port	URL 중에서 port를 반환
protocol	URL 중에서 protocol 부분을 반환
search	URL 중에서 쿼리(query) 부분을 반환

메서드	설명
assign()	새로운 문서를 로드한다.
reload()	현재 문서를 다시 로드한다.
replace()	현재 문서를 새로운 문서로 대체한다.

418

예제

```
<script>
    function replace() {
        location.replace("http://www.google.com");
    }
</script>
<a href="#" onClick="replace()">이동하기</a>
```



419

navigator 객체

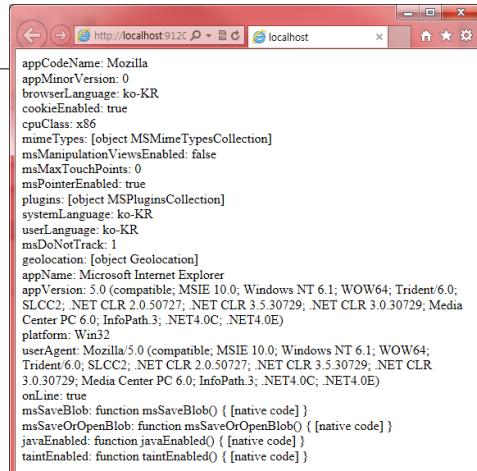
속성	설명
appCodeName	브라우저의 코드 네임
appName	브라우저의 이름
appVersion	브라우저의 버전 정보
cookieEnabled	브라우저에서 쿠키가 활성화되어 있는지 여부
online	브라우저가 인터넷에 연결되어 있으면 true
platform	브라우저가 컴파일 된 플랫폼
userAgent	브라우저에서 서버로 가는 user-agent 헤더

메서드	설명
JavaEnabled()	자바가 사용 가능한지 여부
taintEnabled()	브라우저에서 data tainting이 가능한지 여부

420

예제

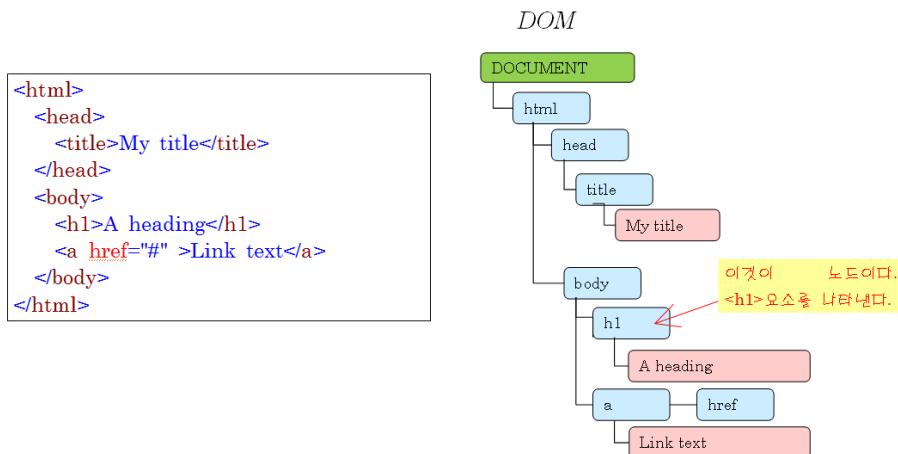
```
<script>
  for (var key in navigator) {
    value = navigator[key];
    document.write(key + ":" + value + "<br>");
  }
</script>
```



421

문서 객체 모델(DOM)

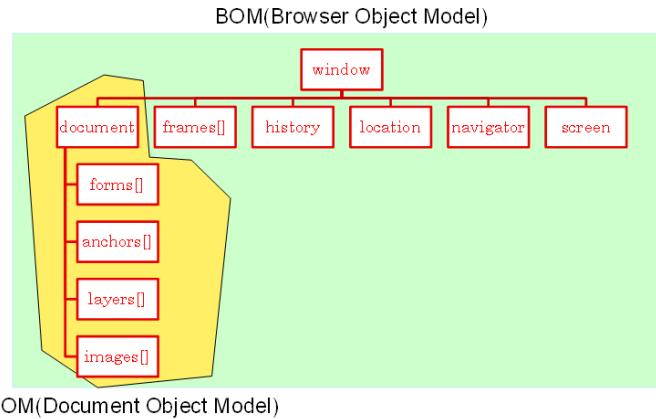
- DOM은 HTML 문서의 계층적인 구조를 트리(tree)로 표현



422

DOM과 BOM

- HTML 문서를 객체로 표현한 것을 DOM
- 웹브라우저를 객체로 표현한 것을 BOM(Browser Object Model)



423

HTML 요소 찾기

- 동적인 웹페이지를 작성하려면 원하는 요소를 찾아야 한다.
 - 요소 속성 중 id로 찾기

```
<script>
    document.getElementById("");
</script>
```

- 요소 속성 중 name으로 찾기

```
<script>
    document.getElementsByName("");
</script>
```

- 태그 이름으로 찾기

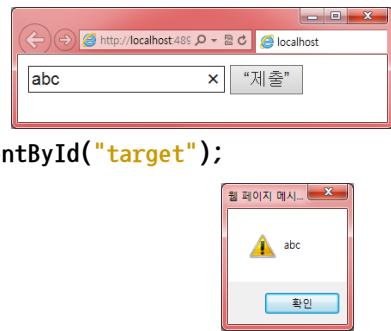
```
<script>
    document.getElementsByTagName("");
</script>
```

424

id로 HTML 요소 찾기

```
<!DOCTYPE html>
<html>
<head>
<script>
function process() {
    var obj = document.getElementById("target");
    alert(obj.value);
}

</script>
</head>
<body>
<form name="myform">
    <input type="text" id="target" name="text1">
    <input type="submit" value="제출" onclick="process()">
</form>
</body>
</html>
```



425

DOM 트리 순회

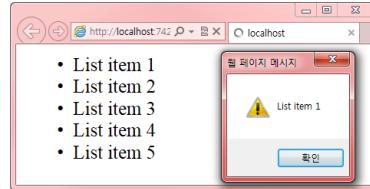
속성	내용
childNodes	한 요소의 모든 자식 요소에 접근할 수 있다. 배열이 반환
firstChild	"childNodes" 배열의 첫번째 자식 노드가 반환된다. "childNodes[0]"와 같다.
lastChild	"childNodes" 배열의 마지막 자식 노드가 반환된다. "childNodes[childNodes.length - 1]"와 같다.
parentNode	현재 노드의 부모 노드를 반환한다.
nextSibling	현재 노드의 다음 형제 노드를 반환한다.
previousSibling	현재 노드의 이전 형제 노드를 반환한다.

426

태그 이름으로 HTML 요소 찾기

```
<!DOCTYPE html>
<html>
<body>
<ul>
    <li>List item 1</li>
    <li>List item 2</li>
    <li>List item 3</li>
    <li>List item 4</li>
    <li>List item 5</li>
</ul>
<script>
var list = document.getElementsByTagName('ul')[0];
var allItems = list.getElementsByTagName('li');

for (var i = 0, length = allItems.length; i < length; i++) {
    alert(allItems[i].firstChild.data);
}
</script>
</body>
</html>
```



427

img와 form 태그 찾기

- 이름을 지정한 경우

```
 ← document.myImg  

<form name="myform"> ← document.myform
    <input type="text" name="text1"> ← document.myform.text1
    <input type="text" name="text2"> ← document.myform.text2
    <input type="submit">
</form>
```

- 이름이 없는 경우 배열 형태로 찾을 수 있음

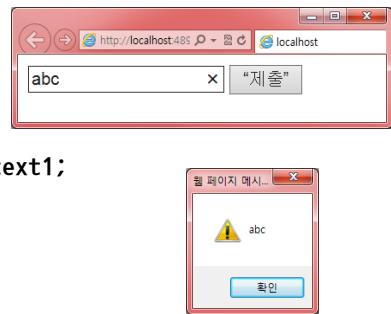
```
 ← document.images[0]  

<form> ← document.forms[0]
    <input type="text"> ← document.forms[0].elements[0]
    <input type="text"> ← document.forms[0].elements[1]
    <input type="submit">
</form>
```

428

입력 양식 찾기

```
<!DOCTYPE html>
<html>
<head>
<script>
    function process() {
        var obj = document.myform.text1;
        alert(obj.value);
    }
</script>
</head>
<body>
<form name="myform">
    <input type="text" id="target" name="text1">
    <input type="submit" value="제출" onclick="process()">
</form>
</body>
</html>
```



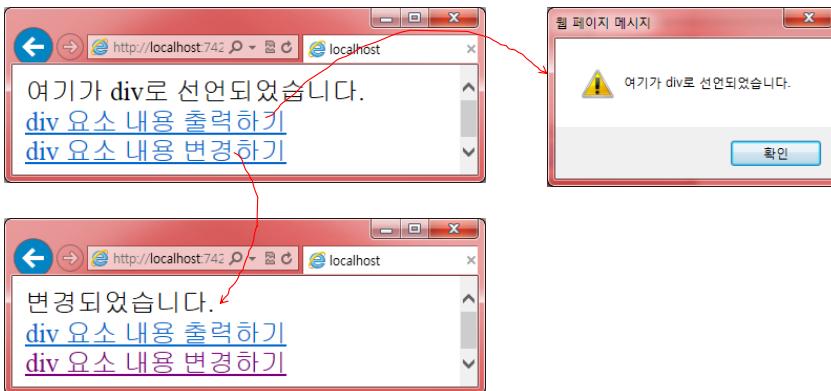
429

HTML 요소의 내용 변경

```
<!DOCTYPE html>
<html>
<head>
<title></title>
<script>
    function get() {
        var val = document.getElementById("ex").innerHTML;
        alert(val);
    }
    function set(v) {
        document.getElementById("ex").innerHTML = v;
    }
</script>
</head>
<body>
    <div id="ex">여기가 div로 선언되었습니다.</div>
    <a href="#" onclick="get()">div 요소 내용 출력하기</a><br>
    <a href="#" onclick="set('변경되었습니다.')">div 요소 내용
    변경하기</a>
</body>
</html>
```

430

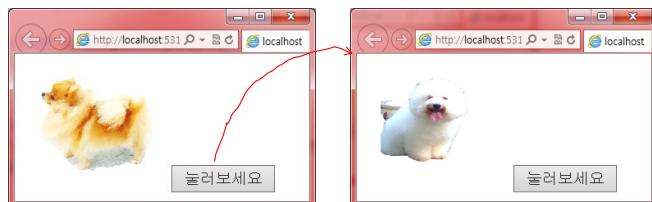
실행결과



431

요소의 속성 변경하기

```
<!DOCTYPE html>
<html>
<body>
    
    <script>
        function changeImage() {
            document.getElementById("image").src = "poodle.png";
        }
    </script>
    <input type="button" onclick="changeImage()" value="눌러보세요"/>
</body>
</html>
```



432

이미지 문제 1

- 다음 화면과 같이 이미지를 클릭하면 이미지가 스위치가 켜지고 다시 클릭하면 스위치가 꺼지는 프로그램을 작성하시오.



433

이미지 문제 2

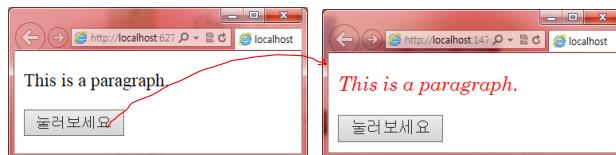
- 이미지를 이용하여 가위 바위 보 게임을 다음과 같이 완성하시오.
(게임 시작 버튼을 누르면 프롬프트 창이 열리고 가위바위보 중 하나를 입력하면 그 결과를 출력하는 프로그램)



434

요소의 스타일 변경하기

```
<!DOCTYPE html>
<html>
<body>
<p id="p1">This is a paragraph.</p>
<script>
    function changeStyle() {
        document.getElementById("p1").style.color = "red";
        document.getElementById("p1").style.fontFamily = "Century
Schoolbook";
        document.getElementById("p1").style.fontStyle = "italic";
    }
</script>
<input type="button" onclick="changeStyle()" value="눌러보세요" />
</body>
</html>
```



435

새로운 HTML 요소 생성

```
<script>
function addtext(t) {
    if (document.createTextNode) {

        var node = document.createTextNode(t);
        document.getElementById("target").appendChild(node);
    }
}
</script>
<div id="target" onclick="addtext('동적으로 텍스트가 추가됩니다.')" style="font: 20px bold;">여기클릭하세요.</div>
```



436

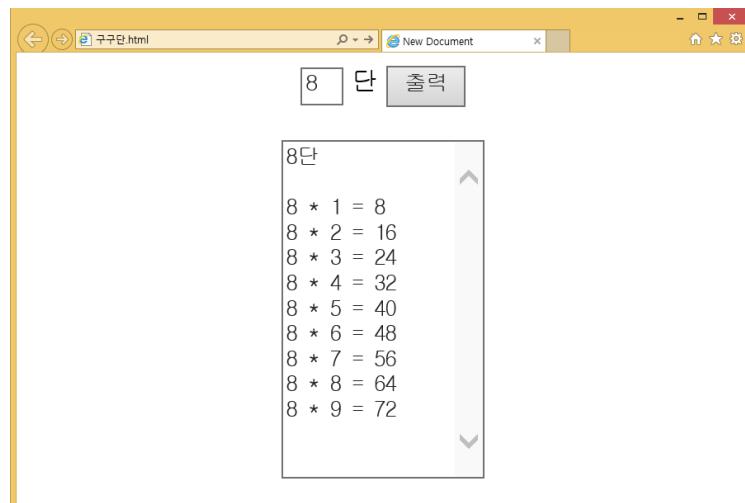
HTML 요소 삭제

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function removeNode() {
        var parent = document.getElementById("target");
        var child = document.getElementById("p1");
        parent.removeChild(child);
      }
    </script>
  </head>
  <body>
    <div id="target">
      <p id="p1">첫번째 단락</p>
      <p id="p2">두번째 단락</p>
    </div>
    <button onclick="removeNode()">누르세요!</button>
  </body>
</html>
```

437

form 문제1

- 다음과 같은 프로그램을 작성하시오.



438

form 문제2

- 다음과 같은 입력 양식을 만들고 결과를 출력하는 프로그램을 작성하시오.(취미가 없을 경우는 "취미가 없군요"를 출력)

이 름 : 샘

성 별 : ♂남 ○여

취 미 : 여행 독서 영화감상 바둑 장기 등산 낚시

결과출력

샘씨!
당신은 남자이고,
취미는 여행, 영화감상, 등산이군요!

439

form 문제 3

- 다음과 같이 데이터를 교환하는 프로그램을 만드시오.

select 객체 예제

4
5
2
3

>

>>

<

<<

1
6
7

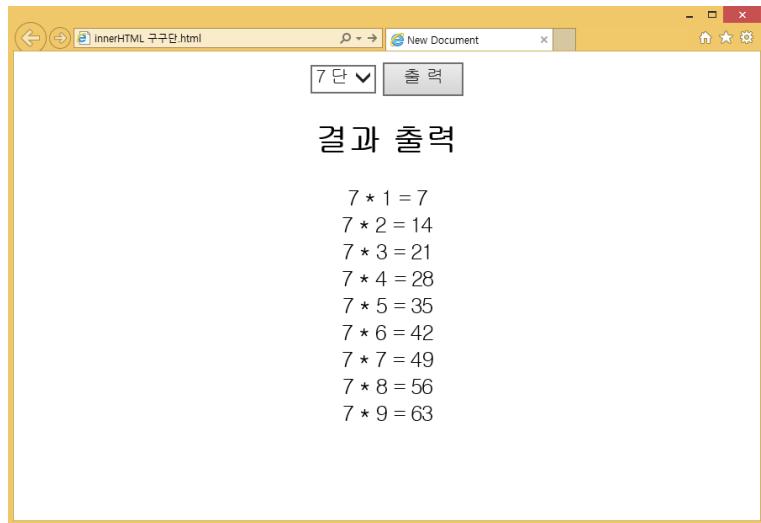
8

추가하기

440

innerHTML 문제

- <select>와 innerHTML을 이용하여 다음과 같이 만드시오.



441

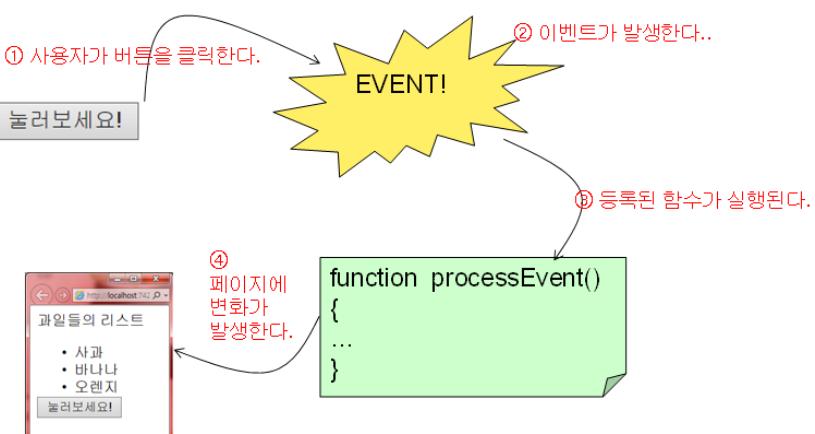
11 이벤트 처리

이벤트

- 마우스 클릭
- 웹 페이지 로딩
- 호버링(hovering)으로 마우스를 어떤 요소 위에서 움직이는 것
- HTML 입력 양식에서 입력 박스를 선택하는 것
- 키보드의 키를 누르는 것

443

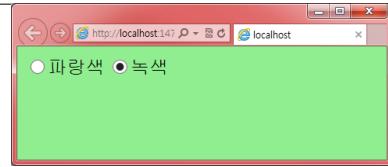
이벤트 처리



444

onclick 이벤트

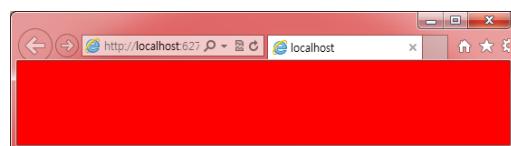
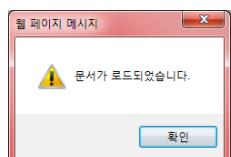
```
<!DOCTYPE html>
<html>
<head>
<script>
    function changeColor(c) {
        document.getElementById("target").style.backgroundColor = c;
    }
</script>
</head>
<body id="target">
    <form method="POST">
        <input type="radio" name="c1" value="v1"
            onclick="changeColor('lightblue')">파랑색
        <input type="radio" name="c1" value="v2"
            onclick="changeColor('lightgreen')">녹색
    </form>
</body>
</html>
```



445

onload와 onunload 이벤트

```
<html>
<head>
<script>
    function onLoadDoc() {
        alert("문서가 로드되었습니다.");
        document.body.style.backgroundColor = "red";
    }
</script>
</head>
<body onload="onLoadDoc();">
</body>
</html>
```



446

onchange 이벤트

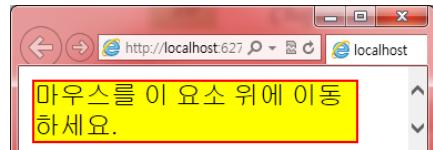
```
<!DOCTYPE html>
<html>
<head>
<script>
    function sub() {
        var x = document.getElementById("name");
        x.value = x.value.toLowerCase();
    }
</script>
</head>
<body>
영어단어: <input type="text" id="name" onchange="sub()">
<p>입력필드를 벗어나면 소문자로 변경됩니다.</p>
</body>
</html>
```



447

onmouseover 이벤트

```
<html>
<head>
<script>
    function OnMouseIn(elem) {
        elem.style.border = "2px solid red";
    }
    function OnMouseOut(elem) {
        elem.style.border = "";
    }
</script>
</head>
<body>
    <div style="background-color: yellow; width: 200px"
        onmouseover="OnMouseIn(this)" onmouseout="OnMouseOut(this)">
        마우스를 이 요소 위에 이동하세요.
    </div>
</body>
</html>
```



448

onmousedown 이벤트

```
<html>
<head>
<script>
    function OnButtonDown(button) {
        button.style.color = "#ff0000";
    }
    function OnButtonUp(button) {
        button.style.color = "#000000";
    }
</script>
</head>
<body>
<button onmousedown="OnButtonDown(this)"
        onmouseup="OnButtonUp(this)">눌러보세요!</button>
</body>
```



449

폼의 유효성 검증

- 입력 필드에서의 잘못을 검증하는 작업

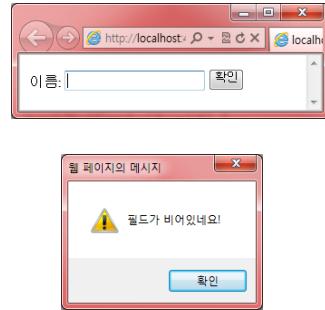
```
<h3>회원가입</h3>
<form>
이름: <input type='text' id='name' ><br>
주소: <input type='text' id='addr' ><br>
생일: <input type='date' id='birthday' ><br>
아이디(6-8 문자):
<input type='text' id='username' ><br>
이메일: <input type='email' id='email' ><br>
휴대폰: <input type='tel' id='phone' ><br>
<input type='submit' value='확인' ><br>
</form>
```



450

공백 검증

```
<script>
    function checkNotEmpty(field) {
        if (field.value.length == 0) {
            alert("필드가 비어있네요!");
            field.focus();
            return false;
        }
        return true;
    }
</script>
<form>
    이름: <input type='text' id='user' />
    <input type='button'
        onclick="checkNotEmpty(document.getElementById('user'))"
        value='확인' />
</form>
```



451

데이터 길이 검증

```
<script>
function checkLength(elem, min, max) {
    var s = elem.value;
    if (s.length >= min && s.length <= max) {
        return true;
    } else {
        alert(min + " 문자와 " + max + " 문자 사이로 입력해주세요!");
        elem.focus();
        return false;
    }
}
</script>
<form>
    이름(6-8 문자): <input type='text' id='name' />
    <input type='button'
        onclick="checkLength(document.getElementById('name'), 6, 8)"
        value='확인' />
</form>
```

452

정규식

- 정규식(regular expression): 특정한 규칙을 가지고 있는 문자열들을 표현하는 수식이다
- 정규 표현식은 /과 / 내부에 위치 (예) `^/[0-9]+abc$`

식	기능	설명
<code>^</code>	시작	문자열의 시작을 표시
<code>\$</code>	끝	문자열의 끝을 표시
<code>.</code>	문자	한 개의 문자와 일치
<code>\d</code>	숫자	한 개의 숫자와 일치
<code>\w</code>	문자와 숫자	한 개의 문자나 숫자와 일치
<code>\s</code>	공백문자	공백, 탭, 줄바꿈, 캐리지리턴 문자와 일치
<code>[]</code>	문자 종류, 문자 범위	[abc]는 a 또는 b 또는 c를 나타낸다. [a-z]는 a부터 z 까지 중의 하나, [1-9]는 1부터 9까지 중의 하나를 나타낸다.

453

수량한정자

	기능	설명
<code>()</code>	문자를 그룹핑한다.	"abcladc"와 "a(b d)c"는 같은 의미를 가진다.
<code>*</code>	0회 이상 반복	"a*b"는 "b", "ab", "aab", "aaab"를 나타낸다.
<code>+</code>	1회 이상	"a+b"는 "ab", "aab", "aaab"를 나타내지만 "b"는 포함하지 않는다.
<code>?</code>	0 또는 1회	"a?b"는 "b", "ab"를 나타낸다.
<code>{m}</code>	m회	"a{3}b"는 "aaab"와 매칭된다.

- 이메일을 검사하는 정규식

- `/^[a-zA-Z0-9.!#$%&'*+/=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$/`

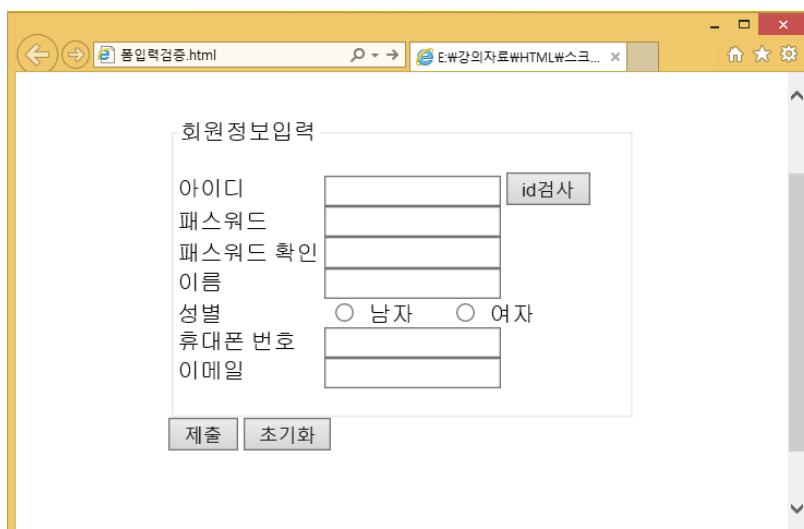
454

숫자 검증 예제

```
<script>
    function checkNumeric(elem, msg) {
        var exp = /^[0-9]+$/;
        if (elem.value.match(exp)) {
            return true;
        } else {
            alert(msg);
            elem.focus();
            return false;
        }
    }
</script>
<form>
    전화번호(-없이 입력): <input type='text' id='phone' />
    <input type='button'
        onclick="checkNumeric(document.getElementById('phone'), '숫자만
        입력하세요!')" value='확인' />
</form>
```

455

정규식 문제 화면



456

정규식문제

- 아이디는 영문자 소문자로 시작하고 8 ~ 20 글자로 입력 받는다.
- 비밀번호는 영문 소문자, 대문자, 숫자, 특수문자가 반드시 1글자 포함되도록 8 ~ 20 글자 사이로 입력 받는다.
- 이름은 한글로 2자에서 5자로 입력받는다.
- 생일, 전화번호는 숫자만 입력 받을 수 있도록 한다.
- 이메일 형식에 맞게 입력받는다.

457

12 HTML5 위치정보

HTML5 위치정보

- 위치정보(Geolocation)은 자신의 위치를 웹 사이트와 공유
- 현재 지역의 날씨, 유명한 맛집 등의 정보를 제공받을 수 있다.



459

geolocation 객체

- var geolocation = navigator.geolocation;

메소드	설명
getCurrentPosition()	사용자의 현재 위치 정보를 반환한다.
watchPosition()	장치의 현재 위치에 대한 정보를 주기적으로 반환한다.
clearWatch()	현재 진행 중인 watchPosition() 실행을 중지한다.

460

예제

```
<!DOCTYPE html>
<html>
<body>
<button onclick="getGeolocation()">위치 정보 얻기</button>
<div id="target"></div>
<script>
    var myDiv = document.getElementById("target");
    function getGeolocation() {
        if (navigator.geolocation) {
            navigator.geolocation.getCurrentPosition(showLocation);
        }
    }
    function showLocation(location) {
        myDiv.innerHTML = "(위도: " + location.coords.latitude +
            ", 경도: " + location.coords.longitude + ")";
    }
</script>
</body>
</html>
```

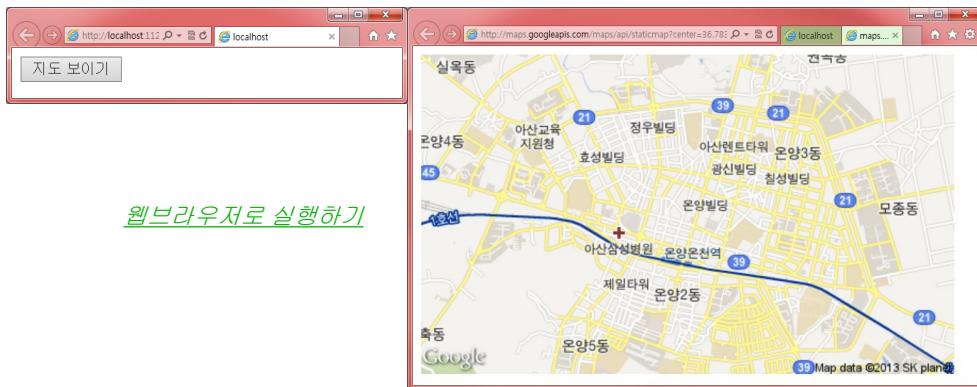
461

지도에 위치 표시하기

```
<!DOCTYPE html>
<html>
<body>
<button onclick="getGeolocation()">지도 보기</button>
<script>
    var myDiv = document.getElementById("target");
    function getGeolocation() {
        if (navigator.geolocation) {
            navigator.geolocation.getCurrentPosition(showGeolocation);
        }
    }
    function showGeolocation(position) {
        var pos = position.coords.latitude + "," +
            position.coords.longitude;
        var url = "http://maps.googleapis.com/maps/api/staticmap?center=" +
            pos + "&zoom=14&size=500x300&sensor=false";
        window.open(url);
    }
</script>
</body>
</html>
```

462

실행결과



[웹브라우저로 실행하기](#)

463

13 HTML5 웹스토리지와 파일API

웹 스토리지

- 웹스토리지(web storage)는 클라이언트 컴퓨터에 데이터를 저장하는 메카니즘
- 웹스토리지는 쿠키보다 안전하고 속도도 빠르다.
- 약 5MB 정도까지 저장이 가능하다.
- 데이터는 키/값(key/value)의 쌍으로 저장



465

localStorage와 sessionStorage

- localStorage 객체
 - 만료 날짜가 없는 데이터를 저장한다.
 - 도메인이 다르면 서로의 로컬 스토리지에 접근할 수 없음.
- sessionStorage 객체
 - 각 세션(하나의 윈도우)마다 데이터가 별도로 저장
 - 해당 세션이 종료되면 데이터가 사라진다.

466

localStorage 예제

```
<!DOCTYPE html>
<html>
<head></head>
<body>
<p>
    페이지 방문 횟수: <span id="count"> </span>번
</p>
<script>
    if (!localStorage.pageLoadCount) {
        localStorage.pageLoadCount = 0;
    }
    localStorage.pageLoadCount = parseInt(localStorage.pageLoadCount)+1;
    document.getElementById('count').textContent =
                                localStorage.pageLoadCount;
</script>
</body>
</html>
```

467

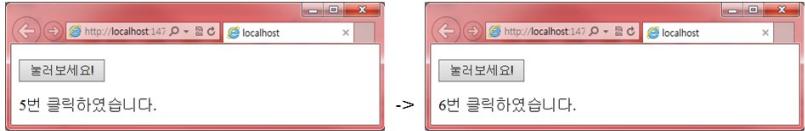
버튼을 클릭한 횟수 저장

```
<!DOCTYPE html>
<html>
<head></head>
<body>
<p>
    <button onclick="incrementCounter()" type="button">눌러보세요!</button>
</p>
<div id="target"></div>
```

468

버튼을 클릭한 횟수 저장

```
<script>
function incrementCounter() {
    if (('localStorage' in window) && window['localStorage'] !== null) {
        if (localStorage.count) {
            localStorage.count++;
        } else {
            localStorage.count = 1;
        }
        document.getElementById("target").innerHTML = localStorage.count
        + "번 클릭 하였습니다.";
    } else {
        document.getElementById("target").innerHTML = "브라우저가
웹스토리지를 지원하지 않습니다.";
    }
}
</script>
</body>
</html>
```



469

sessionStorage 예제

```
<!DOCTYPE html>
<html>
<head></head>
<body>
<p>
<button onclick="incrementCounter()" type="button">눌러보세요!</button>
</p>
<div id="target"></div>
```

470

sessionStorage 예제

```
<script>
function incrementCounter() {
    if ('sessionStorage' in window) && window['sessionStorage'] !== null)
    {
        if (sessionStorage.count) {
            sessionStorage.count++;
        } else {
            sessionStorage.count = 1;
        }
        document.getElementById("target").innerHTML = sessionStorage.count
        + "번 클릭 하였습니다.";
    } else {
        document.getElementById("target").innerHTML = "브라우저가
웹스토리지를 지원하지 않습니다.";
    }
}
</script>
</body>
</html>
```

471

파일 API

- 파일 API: 웹브라우저가 사용자 컴퓨터에 있는 로컬 파일들을 읽어올 수 있도록 해주는 API
- PC에서 실행되는 일반적인 프로그램처럼 동작(웹 애플리케이션)
- 파일 API의 가장 전형적인 응용 분야는 아무래도 사용자가 파일을 선택하여서 원격 서버로 전송하는 작업
- 파일 API에서 사용되는 객체는 File, FileReader
 - File 객체는 로컬 파일 시스템에서 얻어지는 파일 데이터를 나타낸다.
 - FileReader 객체는 이벤트 처리를 통하여 파일의 데이터에 접근하는 메소드들을 제공하는 객체이다.

472

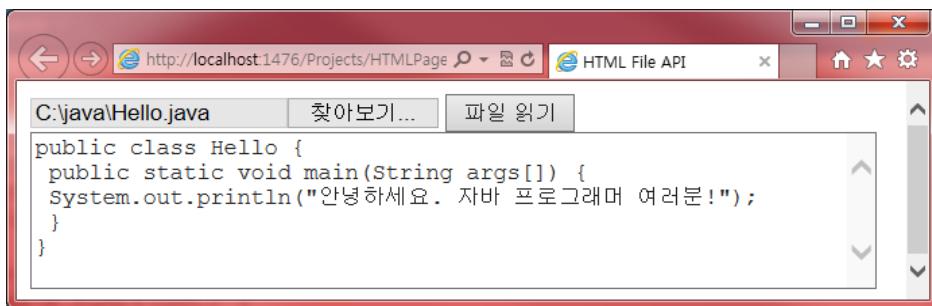
파일 API 예제

```
<!DOCTYPE html>
<html>
<head>
<title>HTML File API</title>
<script>
    function readFile() {
        if (!window.File || !window.FileReader) {
            alert('File API가 지원되지 않습니다.');
            return;
        }
        var files = document.getElementById('input').files;
        if (!files.length) {
            alert('파일을 선택하시오!');
            return;
        }
        var file = files[0];
        var reader = new FileReader();
        reader.onload = function () {
            document.getElementById('result').value = reader.result;
        };
        reader.readAsText(file, "euc-kr");
    }
</script>
```

473

파일 API 예제

```
</head>
<body>
    <input type="file" id="input" name="input">
    <button id="readfile" onclick="readFile()">파일 읽기</button><br>
    <textarea id="result" rows="6" cols="60"></textarea>
</body>
</html>
```



474

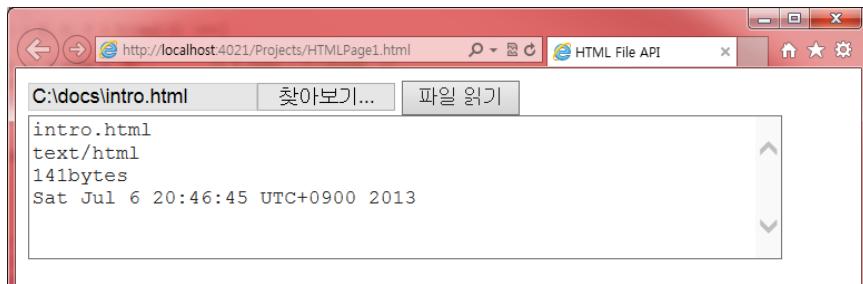
파일 정보 표시 예제

```
<!DOCTYPE html>
<html>
<head>
<title>HTML File API </title>
<script>
function readFile() {
    var files = document.getElementById('input').files;
    output = "";
    for (var i = 0, f; f = files[i]; i++) {
        output += f.name + "\n"; /* f.name - Filename */
        output += f.type + "\n"; /* f.type - File Type */
        output += f.size + "bytes\n"; /* f.size - File Size */
        output += f.lastModifiedDate + "\n"; /* f.lastModifiedDate */
    }
    document.getElementById('result').value = output;
}
</script>
</head>
```

475

실행 결과

```
<body>
<input type="file" id="input" name="input">
<button id="readfile" onclick="readFile()">파일 읽기</button><br />
<textarea id="result" rows="6" cols="60"></textarea>
</body>
</html>
```



476