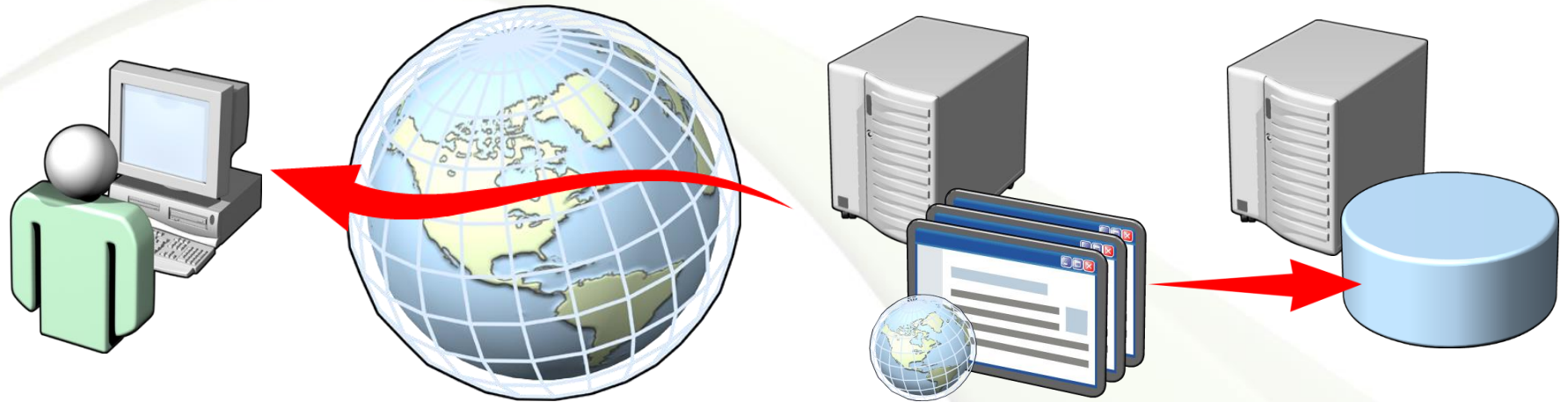


The background features a large, abstract, wavy shape in shades of green and white, resembling a stylized wave or a flowing ribbon. The shape is composed of several overlapping, curved segments that create a sense of movement and depth. The colors range from a bright, vibrant green to a soft, pale green, with white highlights and shadows that give it a three-dimensional appearance. The overall effect is clean, modern, and visually appealing.

# **Introduction to Web Client**

# 웹 애플리케이션 구조



Browser

Internet

Web Application

Data Server

HTML +  
CSS +  
Javascript

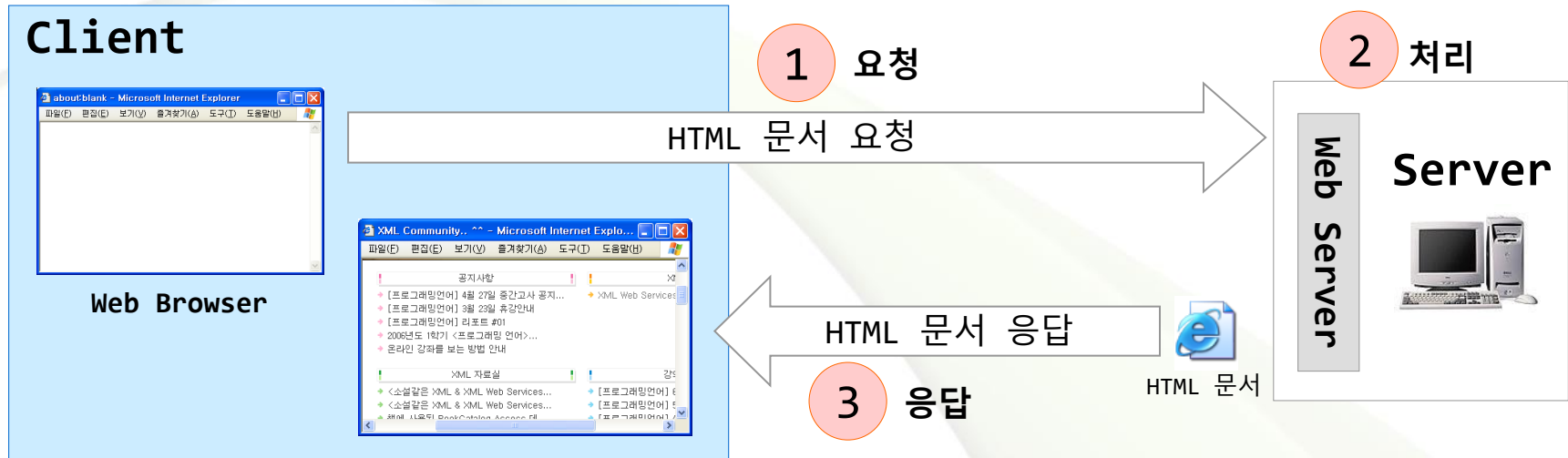
HTTP 기반  
Request &  
Response

Static Resources  
(html, image...)  
  
Dynamic Resource  
(aspx, asmx...)

DBMS

# 웹 구성 요소

## ■ 웹 애플리케이션 작동 구조



## ■ 웹 서버(Web Server)

- » 클라이언트의 요청을 받아서 처리한 후 결과를 클라이언트에 전송해 주는 주체

## ■ 웹 클라이언트(Web Client)

- » 필요한 데이터를 웹 서버에 요청하는 주체
- » 일반적으로 웹 브라우저(Web Browser)

# 정적 웹응용프로그램 vs 동적 웹 응용프로그램

## ■ 정적 웹 응용프로그램

- » Html, image 등 미리 만들어진 자원으로 사용자의 요청에 대해 응답
- » 미리 생성된 자원이 없으면 응답할 수 없으며
- » 작은 차이라도 각각 다른 페이지로 구성해야 합니다.
- » 웹 서버 수준에서 응답이 처리됩니다.

## ■ 동적 웹 응용프로그램

- » 요청이 들어오면 요청에 따라 동적으로 응답할 결과를 생성
- » Web Server는 요청을 받고 적절한 웹 응용프로그램에 요청을 전달
- » CGI 를 이용한 프로세스 기반 웹 응용프로그램 서비스 또는
- » php, jsp, asp ( → asp.net) 등의 쓰레드 기반 동적 웹 응용프로그램 기술 사용

# 표준 웹 클라이언트 기술



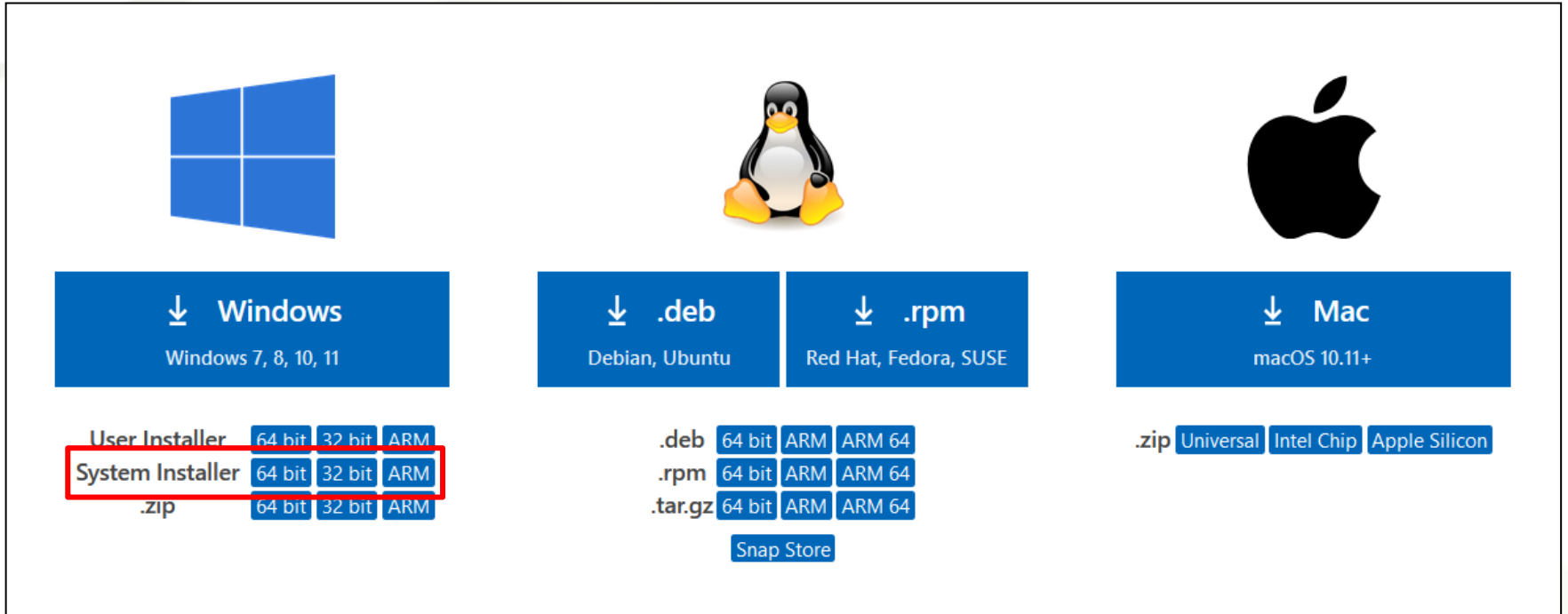
- HTML5
  - » 작게는 웹 문서 작성을 위한 문법 시스템, 크게는 웹 표준 기술을 총칭하는 의미
- CSS ( Cascading Style Sheet )
  - » 웹 페이지의 스타일을 지정하는 문법 시스템
- 자바스크립트
  - » HTML 문서 내에서 동적인 기능을 수행하는 프로그램 구현을 위한 문법 시스템

The background features a large, abstract, wavy shape in shades of green and white, resembling a stylized wave or a flowing ribbon. The shape is composed of several overlapping, curved segments that create a sense of movement and depth. The colors range from a bright, vibrant green to a soft, pale green, with white highlights and shadows that give it a three-dimensional appearance. The overall effect is clean, modern, and organic.

# 테스트 환경 구축

# Visual Studio Code

- 다운로드 → <https://code.visualstudio.com/#alt-downloads>



The screenshot displays the download options for Visual Studio Code. It is organized into three main columns: Windows, Linux, and Mac.

- Windows:** Features the Windows logo and a blue button labeled "↓ Windows" with "Windows 7, 8, 10, 11" below it. Below the button is a table of download options:

	64 bit	32 bit	ARM
User Installer	64 bit	32 bit	ARM
System Installer	64 bit	32 bit	ARM
.zip	64 bit	32 bit	ARM

**Linux:** Features the Tux penguin logo and two blue buttons: "↓ .deb" (Debian, Ubuntu) and "↓ .rpm" (Red Hat, Fedora, SUSE). Below these is a table of download options:

	64 bit	ARM	ARM 64
.deb	64 bit	ARM	ARM 64
.rpm	64 bit	ARM	ARM 64
.tar.gz	64 bit	ARM	ARM 64

Below the table is a "Snap Store" button.

**Mac:** Features the Apple logo and a blue button labeled "↓ Mac" with "macOS 10.11+" below it. Below the button is a table of download options:

	Universal	Intel Chip	Apple Silicon
.zip	Universal	Intel Chip	Apple Silicon

- 다운로드 완료 후 설치

# Node.js

- 다운로드 → <https://nodejs.org/ko/>

Node.js®는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.

## 다운로드 - Windows (x64)

**16.13.1 LTS**

안정적, 신뢰도 높음

**17.3.0 현재 버전**

최신 기능

[다른 운영 체제](#) | [변경사항](#) | [API 문서](#)

[다른 운영 체제](#) | [변경사항](#) | [API 문서](#)

LTS 일정은 [여기서 확인하세요](#)

- 다운로드 완료 후 설치

명령 프롬프트

```
C:\#>node --version  
v16.13.1
```

```
C:\#>npm --version  
8.1.2
```

```
C:\#>
```



# http-server ( node based )

- 설치 → 터미널에서 명령 실행

```
npm install -g http-server
```

```
added 34 packages, and audited 35 packages in 3s
```

```
7 packages are looking for funding  
run `npm fund` for details
```

```
found 0 vulnerabilities
```

```
npm notice  
npm notice New minor version of npm available! 8.1.2 -> 8.3.0  
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.3.0  
npm notice Run npm install -g npm@8.3.0 to update!  
npm notice
```

- 실행 → 터미널에서 명령 실행

```
C:\>http-server . -p 8081  
Starting up http-server, serving .  
  
http-server version: 14.0.0  
  
http-server settings:  
CORS: disabled  
Cache: 3600 seconds  
Connection Timeout: 120 seconds  
Directory Listings: visible  
AutoIndex: visible  
Serve GZIP Files: false  
Serve Brotli Files: false  
Default File Extension: none  
  
Available on:  
  http://192.168.123.108:8081  
  http://127.0.0.1:8081  
Hit CTRL-C to stop the server
```

The background features a large, abstract, wavy shape in shades of green and white, resembling a stylized wave or a flowing ribbon. The shape is composed of several overlapping, curved segments that create a sense of movement and depth. The colors range from a bright, vibrant green to a soft, pale green, with white highlights and shadows that give it a three-dimensional appearance. The overall effect is clean, modern, and visually appealing.

# **HTML Basic**

# HTML

- 하이퍼 텍스트 마크업 언어 (Hyper Text Markup Language, HTML)
- 웹 페이지를 위한 지배적인 마크업 언어
  - » 제목, 단락, 목록 등과 같은 본문을 위한 구조적 의미 표시
  - » 링크, 인용과 그 밖의 항목으로 구조적 문서를 만들 수 있는 방법 제공
  - » 이미지와 객체를 내장하고 대화형 양식을 생성하는 데 사용
- 웹 브라우저와 같은 HTML 처리 장치의 행동에 영향을 주는 자바스크립트 및 본문과 그 밖의 항목의 외관과 배치를 정의하는 CSS 같은 스크립트를 포함하거나 불러올 수 있다

버전	발표 연도
HTML 1	1991년 10월
HTML 2	1995년 11월
HTML 3	1997년 1월
HTML 4	1997년 12월
HTML 5	2014년 11월

# HTML5

- 월드 와이드 웹 (World Wide Web)의 핵심 마크업 언어
  - » HTML 4.01, XHTML 1.0, DOM 레벨 2 HTML에 대한 차기 표준 제안
- 2014년 10월 28일 표준이 확정된 HTML의 완전한 5번째 버전
  - » 2016년 11월 1일 버전인 HTML5.1 표준안 확정
  - » 2017년 12월 14일 HTML5.2 표준안 확정
  - » HTML5.3 표준안은 현재 작업 초안 단계로 진행 중
- 비디오, 오디오 등 다양한 부가기능과 최신 멀티미디어 콘텐츠를 액티브X 없이 브라우저에서 쉽게 볼 수 있게 하는 목적

# HTML5 주요 기능

- 웹 브라우저에서 음악, 동영상 재생, 온라인 게임까지 가능
- HTML5에서 추가된 주요 기능

버전	발표 연도
멀티미디어	플러그인 도움 없이 음악과 동영상 재생
그래픽	하드웨어 가속 기반 2차원 및 3차원 그래픽 구현
통신	소켓 통신 기반 실시간 쌍방향 통신 수행
디바이스 접근	디바이스의 정보와 기능 사용
오프라인 및 저장소	인터넷이 연결되지 않은 상태에서 응용 프로그램 동작
시맨틱	시맨틱 웹을 구현하는 시맨틱 태그 추가
CSS3 스타일시트	CSS3 스타일시트 완벽하게 지원
웹의 성능 극대화 및 통합	웹 페이지가 하나의 독립적인 응용 프로그램 하이브리드 응용 프로그램 크롬 북, MS 윈도우 운영체제에서 사용

# 태그와 요소

- 요소
  - » 제목, 본문, 이미지 등 HTML 페이지에 포함된 것
- 태그
  - » 요소를 만드는 문법 표현

`<h1>Hello HTML5</h1>`

시작 태그

끝 태그

`<br>`

HTML5 표기법

`<br />`

XHTML5 표기법

# H태그와 요소

- 계층 구조

- » 태그 내부에 다른 태그를 포함하는 것 가능
- » 가능한 태그와 불가능한 태그 구분

```
<article>
```

```
  <h1>Article Header</h1>
```

```
  <p>Lorem ipsum dolor sit amet.</p>
```

```
</article>
```

# HTML

- 속성

- » 태그에 추가 정보를 설정하는 방법

<sup>속성 블록</sup>  
<h1 <sup>속성 이름</sup>title="header" <sup>속성 값</sup>>Hello HTML5</h1> <sup>내용 문자</sup>

<sup>속성 블록</sup>  
<img <sup>속성 이름</sup>src="image.png" <sup>속성 값</sup>/ >



# 주석

- 브라우저가 해석하지 않는 영역 표시 → 코드에 대한 설명을 위해 사용

```
<!DOCTYPE html>
<html>
<head>
  <!-- title 태그 -->
  <title>TITLE</title>
</head>
<body>
  <!-- h1 태그 -->
  <h1>Hello HTML5</h1>
</body>
</html>
```

실행



# Hello HTML5

# HTML 페이지 기본 구조

- 전체 문서는 다음 요소로 구성
  - » Document type declaration
  - » html
  - » head
  - » body

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML5 Basic Page</title>
</head>
<body>

</body>
</html>
```

# HTML 페이지 기본 구조

- <!DOCTYPE html>

- » HTML5 표준을 따르는 문서임을 표시
- » 문서의 첫 행에 표시

- <html> ... </html>

- » HTML 문서의 최상위 요소
- » lang 속성 사용

<html lang="ko">

언어	속성 값	언어	속성 값
한국어	ko	영어	en
일본어	ja	러시아어	ru
중국어	zh	독일어	de

# HTML 페이지 기본 구조

## ▪ <head> ... </head>

- » 화면에 표시되지 않는 영역
- » 브라우저 등 문서를 해석하는 도구에게 문서에 대한 정보 제공
- » head에 포함되는 하위 요소

태그 이름	설명
meta	웹 페이지에 추가 정보 전달
title	웹 페이지의 제목 지정
script	웹 페이지에 스크립트 추가
link	웹 페이지에 다른 파일 추가
style	웹 페이지에 스타일시트 추가
base	웹 페이지의 기본 경로 지정

## » <title> 하위 태그

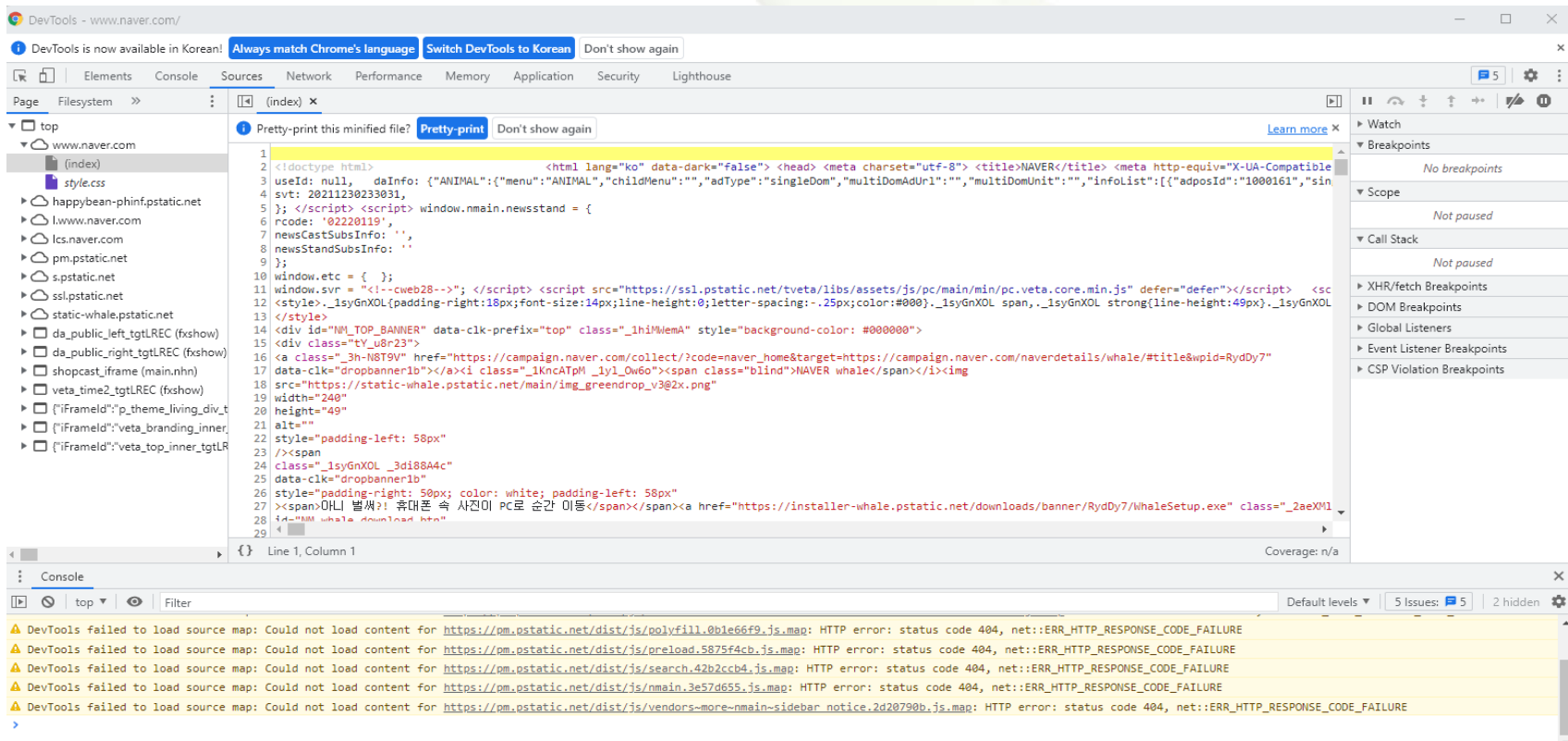
- › <head> 요소의 하위 요소
- › 브라우저에 표시하는 문서 제목 지정



# 브라우저에서 문서 상세 보기 (inspection)

## ■ 문서 검사 실행

- » 마우스 우측 버튼 → 요소 검사
- » F12
- » ctrl + shift + i



# HTML 기본 태그

The background features a large, abstract, wavy shape in shades of green and white, resembling a stylized wave or a flowing ribbon. The shape starts from the left, curves upwards and then downwards, creating a sense of movement. The colors transition from a light green to a darker green, with white highlights. The overall composition is clean and modern.

# 글자 태그

## ■ 제목 표시 태그

태그 이름	설명	태그 이름	설명
h1	첫 번째로 큰 제목 글자	h4	네 번째로 큰 제목 글자
h2	두 번째로 큰 제목 글자	h5	다섯 번째로 큰 제목 글자
h3	세 번째로 큰 제목 글자	h6	여섯 번째로 큰 제목 글자

## ■ 공백 표시

- » html은 연속된 여러 개의 공백, tab, enter를 하나의 공백으로 처리
- » &nbsp; 특수 문자 사용해서 공백 표시

특수문자	출력 문자
&nbsp;	공백
&lt;	<
&gt;	>
&amp;	&

# 글자 태그

태그	태그 설명
<p> ... </p>	문단 표시 태그
 	줄 바꿈 태그
<hr>	횡방향 선 태그

## ■ 문서 이동 태그

» <a href="target-path"> ... </a>

» 지정된 페이지로 이동하는 기능

<a href="http://hanb.co.kr"> 한빛리더어 </a>

이동할 페이지                      출력 글자

» 경로 → 상대 경로 vs 절대 경로

» 페이지 내부 이동 → #id



# 목록 태그

- 순서 있는 목록 태그

- » `ol (ordered list)` → 순서 번호 목록 태그
- » `li (list item)` → 목록 요소

- 순서 없는 목록 태그

- » `ul (unordered list)` → 글머리 기호 목록 태그
- » `li (list item)` → 목록 요소

# 테이블 태그

- 행과 열로 구성된 표를 표현하는 태그
- 테이블 태그 구성 요소

태그 이름	설명
table	표(table)
thead	표의 머리(table head)
tbody	표의 본문(table body)
tfoot	표의 꼬리(table foot)
tr	표의 행(table row)
th	표의 제목 요소(table header)
td	표의 일반 요소(table data)

# 테이블 태그

## ■ 테이블 태그 사례

```
<body>
  <table border="1">
    <thead>
      <tr>
        <td>thead 태그 내부</td>
        <td>thead 태그 내부</td>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>tbody 태그 내부</td>
        <td>tbody 태그 내부</td>
      </tr>
      <tr>
        <td>tbody 태그 내부</td>
        <td>tbody 태그 내부</td>
      </tr>
    </tbody>
    <tfoot>
      <tr>
        <td>tfoot 태그 내부</td>
      </tr>
    </tfoot>
  </table>
</body>
```

thead 태그 내부	thead 태그 내부
tbody 태그 내부	tbody 태그 내부
tbody 태그 내부	tbody 태그 내부
tfoot 태그 내부	tfoot 태그 내부

# 테이블 태그

- 셀 병합 속성

태그 이름	설명
colspan	셀의 너비 지정
rowspan	셀의 높이 지정

# 이미지 태그

- 이미지 표시 태그

- » `<img />`

- » 주요 속성

태그 이름	설명
src	이미지의 경로 지정
alt	이미지가 없을 때 나오는 글자 지정
width	이미지의 너비 지정
height	이미지의 높이 지정

# 공간 분할 태그

## ■ 공간 분할 태그

» `<div> ... </div>`



```
</div>  
</div>  
<!--EMPTY-->  
▶ <div id="gnb">...</div>  
</div>  
▶ <div id="container" role="main">...</div>  
▶ <div id="footer" role="contentinfo">...</div>  
</div>
```

## ■ span

» 스타일을 지정하거나 속성을 공유하는 요소를 묶을 때 사용

» 인라인 컨테이너

# 인라인 형식 vs 블록 형식

- 블록 형식 요소
  - » 한 행에 하나의 요소만 표시하는 요소
  - » 사례 → `<a> ... </a>`, `<span> ... </span>`
- 인라인 형식 요소
  - » 한 행에 여러 개의 요소만 표시하는 요소
  - » 사례 → `<h1> ... </h1>`, `<div> ... </div>`

블록 형식	인라인 형식
div	span
h1 - h6	a
p	input
ul, ol, li	글자 태그
table	
input	

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

## h1 태그

p 태그

a 태그 i 태그

# 오디오 태그

## ■ 오디오 태그

» `<audio> ... </audio>`

» HTML5에서 추가되어 플러그인 없이 브라우저에서 오디오 재생 가능

## ■ 주요 속성

속성 이름	설명
src	음악 파일의 경로 지정
preload	음악을 재생하기 전에 데이터를 모두 불러올지 여부 지정
autoplay	음악의 자동 재생 여부 지정
loop	음악의 반복 여부 지정
controls	음악 재생 도구 출력 여부 지정

» source 속성

› 각 브라우저 별 지원 파일 형식을 지정하기 위한 호환성 태그



# 비디오 태그

## ■ 비디오 태그

» `<video> ... </video>`

» HTML5에서 추가되어 플러그인 없이 브라우저에서 비디오 재생 가능

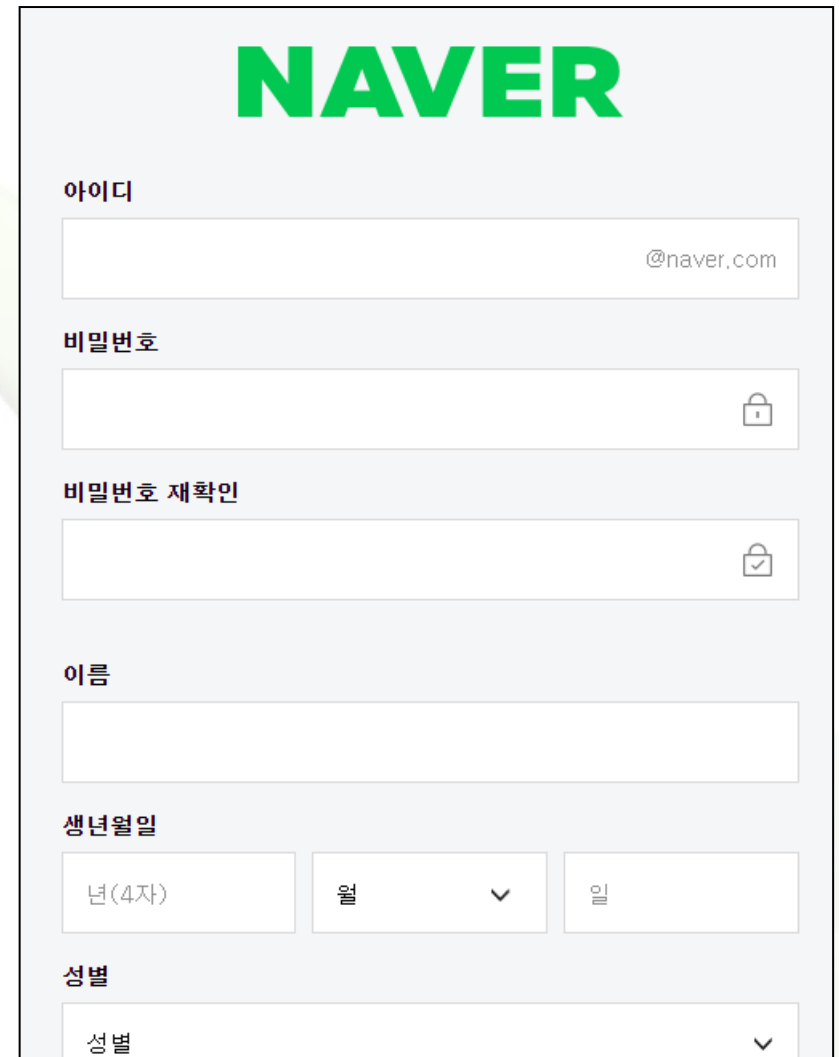
## ■ 주요 속성

속성 이름	설명
src	비디오 파일의 경로 지정
preload	비디오를 재생하기 전에 데이터를 모두 불러올지 여부 지정
autoplay	비디오의 자동 재생 여부 지정
loop	비디오의 반복 여부 지정
Controls	비디오 재생 도구 출력 여부 지정
poster	비디오 준비 중일 때 표시할 이미지의 파일 경로 지정
width	비디오의 너비 지정
height	비디오의 높이 지정

# 입력 양식

## ■ 입력 양식

- » 사용자 입력을 받을 수 있는 html 요소
- » form 단위로 입력 요소를 묶어서 서버로 데이터 전송



The image shows a Naver login form. At the top is the Naver logo in green. Below it are several input fields: '아이디' (ID) with a placeholder '@naver.com', '비밀번호' (Password) with a lock icon, '비밀번호 재확인' (Confirm Password) with a checkmark icon, '이름' (Name), '생년월일' (Date of Birth) with separate boxes for year (4 digits), month (dropdown), and day, and '성별' (Gender) with a dropdown menu.

**NAVER**

**아이디**

**비밀번호**

**비밀번호 재확인**

**이름**

**생년월일**

년(4자) 월 일

**성별**

성별

# 입력 양식

## ▪ form 태그

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

 쿼리 전송

## » 주요 속성

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

# 입력 양식

## ■ 데이터 전송 방식

- » 브라우저의 데이터를 서버로 전송하는 방식 지정
- » form 태그의 method 속성으로 지정

전송 방식	설명
GET	주소에 데이터를 직접 입력해 전달
POST	별도의 방법을 사용해 데이터를 해당 주소로 전달

# 입력 양식

## ■ 입력 양식

» 사용자 입력을 처리하는 태그

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

# 입력 양식

## ■ 입력 양식 (계속)

» 사용자 입력을 처리하는 태그

태그 이름	속성	설명
textarea	cols	여러 행의 글자 입력 양식 생성
	rows	cols는 너비, rows는 높이 지정
select		선택 양식 생성
optgroup		옵션 그룹화
option		옵션 생성
fieldset		입력 양식의 그룹 지정
legend		입력 양식 그룹의 이름 지정



# **CSS Basic**

# CSS (Cascading Style Sheet)

## ■ CSS

- » 웹 페이지를 꾸미기 위한 표준 문법 시스템
- » HTML 문서의 요소들에 선택적으로 스타일 적용
- » 형식

```
h1 { color: red; }
```

선택자

스타일 속성

스타일 값

## ■ 선택자

- » 특정한 HTML 태그를 선택할 때 사용
- » 선택한 태그에 원하는 스타일이나 스크립트 적용 가능



# 선택자

## ■ 종류

선택자 종류	선택자 형태	사용 예
전체 선택자	*	*
태그 선택자	태그	h1
아이디 선택자	#아이디	#header
클래스 선택자	.클래스	.item
속성 선택자	선택자[속성 = 값]	input[type = text]
	선택자[속성 ~= 값]	div[data-role ~= row]
	선택자[속성  = 값]	div[data-role  = row]
	선택자[속성 ^= 값]	div[data-role ^= row]
	선택자[속성 \$= 값]	div[data-role \$= 9]
	선택자[속성 *= 값]	div[data-role *= row]
후손 선택자	선택자 선택자	header h1
자손 선택자	선택자 > 선택자	header > h1
반응 선택자	선택자:active	div:active
	선택자:hover	div:hover

# 선택자

## ■ 종류 (계속)

선택자 종류	선택자 형태	사용 예
전체 선택자	*	HTML 문서 안에 있는 태그를 모두 선택
태그 선택자	태그	HTML 페이지 내부의 특정 태그를 모두 선택
아이디 선택자	#아이디	특정 id 속성이 있는 태그를 선택. 특정 태그 하나를 선택할 때 사용
클래스 선택자	.클래스	특정 클래스가 있는 태그를 선택

# 선택자

## ■ 다중 선택자

- » 한 번에 여러 개의 선택자 지정
- » 쉼표 사용

```
<style>  
    body, p, h1, h2, h3, h4, h5, h6 { margin: 0; padding: 0 }  
</style>
```

## ■ 속성 선택자

- » 특정 속성과 값을 가지는 태그 선택
- » []를 사용해서 속성 지정

선택자 형태	설명
선택자[속성]	특정한 속성이 있는 태그를 선택
선택자[속성 = 값]	특정한 속성 안의 값이 특정 값과 같은 태그를 선택

# 선택자

- 후손 선택자

- » 계층 구조상의 하위 요소를 계층의 깊이와 관계 없이 선택

선택자 형태	설명
선택자A 선택자B	선택자A의 후손인 선택자B 선택

- 자식 선택자

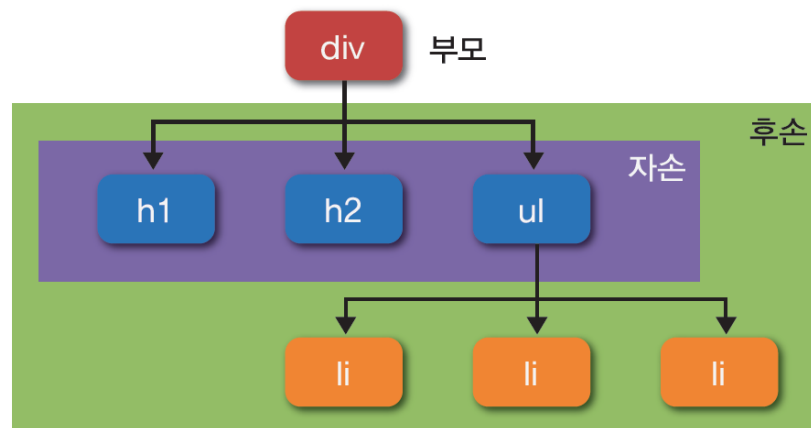
- » 계층 구조상의 한 단계 하위 요소 중에서 선택

선택자 형태	설명
선택자A > 선택자B	선택자A의 자식인 선택자B 선택

# 선택자

## ■ 자식 / 후손 선택자 사례

```
<body>
  <div>
    <h1>CSS3 선택자 기본</h1>
    <h2>기본 선택자</h2>
    <ul>
      <li>전체 선택자</li>
      <li>태그 선택자</li>
      <li>아이디 선택자와 클래스 선택자</li>
    </ul>
  </div>
</body>
```



# 선택자

## ■ 반응 선택자

» 사용자의 반응에 의해 만들어지는 특정한 상태 선택

선택자 형태	설명
:active	사용자가 마우스로 클릭한 태그 선택
:hover	사용자가 마우스 커서를 올린 태그 선택

## ■ 상태 선택자

» 동일한 요소를 상태에 따라 다르게 선택

선택자 형태	설명
:checked	체크 상태의 input 태그 선택
:focus	초점이 맞추어진 input 태그 선택
:enabled	사용 가능한 input 태그 선택
:disabled	사용 불가능한 input 태그 선택

# 선택자

- 구조 선택자

- » 목록에서 특정한 위치에 있는 태그 선택

선택자 형태	설명
:first-child	형제 관계에서 첫 번째로 등장하는 태그 선택
:last-child	형제 관계에서 마지막으로 등장하는 태그 선택
:nth-child(수열)	형제 관계에서 앞에서 수열 번째로 등장하는 태그 선택
:nht-last-child(수열)	형제 관계에서 뒤에서 수열 번째로 등장하는 태그 선택

The background features a series of overlapping, wavy, ribbon-like shapes in various shades of green and white, creating a dynamic, flowing effect. The text is centered over this background.

# **Javascript Language Specification**



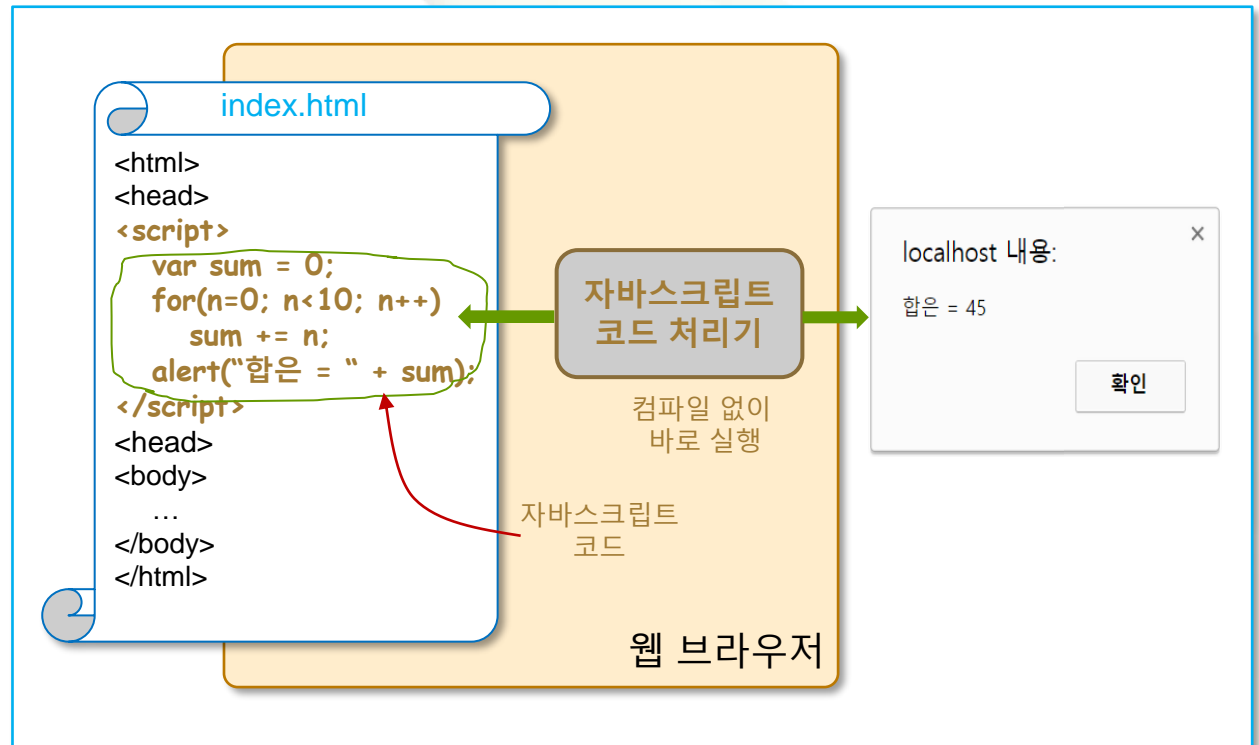
# 자바스크립트

## ■ 개발

- » 1995년 Netscape에서 개발
- » Netscape Navigator 2.0 브라우저에 최초 탑재
- » 웹 프로그래밍 개념 시초

## ■ 특징

- » HTML 문서에 내장
- » 인터프리터 언어
- » 단순한 구조
- » 초기 학습곡선 완만



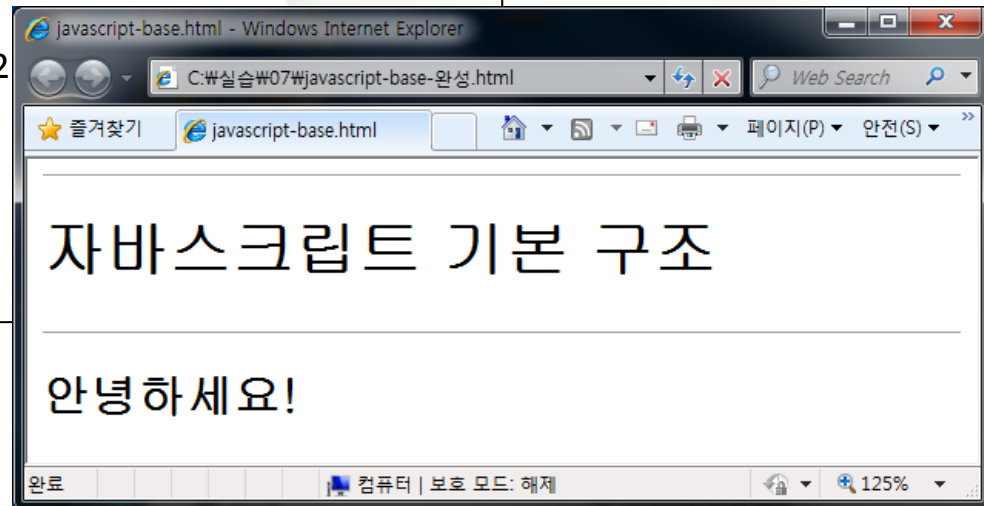
# 자바스크립트 기능

- 브라우저 제어
  - » 새 브라우저 윈도우 열기 및 닫기
  - » History 제어 ( 앞으로 / 뒤로 이동 )
  - » 주소 입력 제어 ( 다른 웹 페이지로 이동 )
- 웹 페이지 내용 및 모양 동적 제어
  - » HTML 태그의 속성, 내용 제어
  - » DOM 트리 제어
- 사용자와 동적 상호작용 처리
  - » 사용자 입력 조작에 대한 이벤트 처리
- 웹 서버와 통신
  - » 비동기 데이터 전송 및 수신

# 자바스크립트의 기본 구조

- <script> 태그를 사용
- type 속성값으로 "text/javascript"를 입력

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset=euc-kr" />
    <title> javascript-base.html </title>
  </head>
  <body>
    <hr /><h1> 자바스크립트 기본 구조 </h1><hr />
    <script type="text/javascript">
      <![CDATA[
        document.write("<h2>안녕하세요!</h2>
      //]]>
    </script>
  </body>
</html>
```



# 자바스크립트 실행 위치

- 자바스크립트가 <head> 태그 영역에 배치되는 경우

```
01  <!DOCTYPE html>
03  <html>
04    <head>
05      <title> javascript-head.html </title>
07      <script type="text/javascript">
08        function message( ){
09          alert("이 메시지 상자는 onload 이벤트가 발생할 때 보인다.");
10        }
11        document.write("<h2>자바스크립트 세계에 오신 걸 환영합니다.</h2>");
12      </script>
13    </head>
14    <body onload="message( );">
15      <hr /><h1> 자바스크립트가 head 요소 영역에 위치했을 때 </h1><hr />
16    </body>
17  </html>
```

# 자바스크립트 실행 위치

- 자바스크립트가 <body> 태그 영역에 배치되는 경우

```
01  <!DOCTYPE html>
03  <html>
04    <head>
05      <title> javascript-body.html </title>
07    </head>
08    <body>
09      <hr /><h2> 자바스크립트가 body 요소 영역 내에 위치했을 때 </h2><hr />
10      <script type="text/javascript">
11        document.write("자바스크립트의 문장");
12      </script>
13      <p> xhtml의 문장 </p>
14    </body>
15  </html>
```

## 2. 자바스크립트 실행 위치

- 자바스크립트를 외부 파일에서 가져오는 경우
  - » 기능을 구현할 곳에 삽입
  - » 외부 파일 자바스크립트의 확장자는 ".js"
  - » script element의 src 속성값에 호출할 자바스크립트의 파일명 지정
  - » 사용 방법

```
<script type="text/javascript"  
      src="javascript-file-name.js"></script>
```

# 기본 구문 규칙 - 표현식과 문장

## ■ 표현식

» 값을 만들어내는 간단한 코드

- 273
- 10 + 20 + 30 \* 2
- 'RintIanTta'

## ■ 문장

» 하나 이상의 표현식이 모인 것  
» 문장이 모여 프로그램 구성  
» 문장의 끝은 세미콜론(;)으로 표시

- 10 + 20 + 30 \* 2;
- var rintiantta = 'Rint' + 'Ian' + 'Tta';
- alert('Hello JavaScript..!');
- 273;

# 기본 구문 규칙 - 키워드

- 자바 스크립트에 이미 정의된 특별한 의미가 있는 단어

break	else	instanceof	true
case	false	new	try
catch	finally	null	typeof
continue	for	return	var
default	function	switch	void
delete	if	this	while
do	in	throw	with

abstract	enum	int	short
boolean	export	interface	static
byte	extends	long	super
char	final	native	synchronized
class	float	package	throws
const	goto	private	transient
debugger	implements	protected	volatile
double	import	public	



# 기본 구문 규칙 - 식별자

- 식별자는 변수, 함수 등에 붙여진 이름

- 필수 식별자 규칙

- » 키워드를 사용할 수 없음
- » 숫자로 시작할 수 없음
- » 특수 문자는 \_와 \$만 허용
- » 공백 문자 포함할 수 없음

```
alpha  
alpha10  
_alpha  
$alpha  
AlPha  
ALPHA
```

- 권장 식별자 규칙

- » 모든 언어의 문자 사용 가능하나 영문자 사용이 관례
- » 대상의 의미를 정확하게 표현할 수 있는 단어 사용
- » 생성자 함수의 이름은 항상 대문자로 시작
- » 변수, 인스턴스, 함수, 메서드의 이름은 항상 소문자로 시작
- » 여러 단어로 이뤄진 식별자는 각 단어의 첫 글자를 대문자로

# 기본 구문 규칙 - 주석

- 엔진에 의해 처리되지 않는 영역으로 코드에 대한 설명을 위해 작성
- HTML 태그 주석
  - » <!-- 주석 내용 -->로 문자열을 감싸 생성
- 자바스크립트 주석
  - » 한 줄 주석 : //뒤의 문장은 실행되지 않음
  - » 영역 주석 : /\* 주석 내용 \*/을 사용해 주석 영역 표현

```
<script>  
    // 주석은 코드의 실행에 아무 영향을 미치지 않습니다.  
    /*  
        alert('Hello JavaScript');  
        alert('Hello JavaScript');  
        alert('Hello JavaScript');  
    */  
</script>
```

# 변수

- 데이터를 담을 수 있는 메모리 할당 영역
- 변수 선언

```
var A=8;  
var people_name="kim";
```

- 잘못 된 변수 이름 사용 예

변수명	올바르지 않은 이유	올바른 변수명의 예시
Str*	특수문자 포함	Str
1234Num	숫자로 시작	Num1234
if	예약어 사용	ifn
마우스	한글 사용	mouse

# 자료형

- 크기, 저장 형식 등 데이터의 정보를 의미
- 명시적인 자료형 표현을 사용하지 않습니다.
- 데이터에 따라 자료형 구분
  - » 문자열
  - » 숫자 (정수, 부동소수점)
  - » Boolean
  - » null

# 숫자 자료형

## ■ 정수

```
var num1 = 64; // 10진수
var num2 = 0100; // 8진수, 10진수로 64
var num3 = 0x40; // 16진수, 10진수로 64
document.write("num1의 값은 " + num1 + " 입니다 <br/>");
document.write("num2의 값은 " + num2 + " 입니다 <br/>");
document.write("num3의 값은 " + num3 + " 입니다 <br/>");
```

## ■ 실수

```
var num1 = 3.2541, num2 = 2e3; // 소수점 4자리수와 2,000의 지수표현
document.write("num1은 " + num1 + "이고 num2는 " + num2 + "입니다 <br/>");
```

# 문자형

- 큰따옴표(" ") 또는 작은따옴표(' ') 사이에 들어가는 0개 이상의 문자들

```
var str = "야구"; // 나중에 바뀔 수 있는 문자열을 변수에 입력
document.write("사람들이 좋아하는 \"스포츠\"는 " +
               str + "이고 " +
               str + "(은)는 \'재미\' 있다.");
```

- 이스케이프 문자

» 특수한 문자 표시 도구

특수문자	내용
\n	커서를 다음 줄로 이동
\t	커서를 탭(Tab) 이동
\b	커서를 앞 문자를 지우며 이동
\f	커서를 다음 페이지 처음으로 이동
\r	커서를 그 줄의 처음으로 이동
\"	큰따옴표
\'	작은따옴표
\\	역슬래시

# 논리형

- 크기가 1비트
- 참(true)과 거짓(false)의 두 값만 가짐
- 비교(<, >, =) 연산, 논리 연산, 함수 반환 값으로 사용

```
var t = (3>2);  
var f = (2>3);  
document.write("참일 때에는 "+t+"로표현되 고거짓일때에는 "+f+"로 표현된다.");
```

# 연산자

- 데이터의 처리 명령을 표현하는 기호
- 종류
  - » 산술 연산자
  - » 대입 연산자
  - » 관계 연산자
  - » 논리 연산자
  - » 증감 연산자
  - » ...



# 산술 연산자

- 정수형 변수, 실수형 변수, 정수값, 실수값을 이용하여 산술 연산

연산자	사용 예	설 명
-	$C = -A$	A 값이 양수이면 음수로, 음수이면 양수로 변환
-	$C = A - B$	A에서 B를 뺀 차를 C에 저장, 뺄셈 연산
+	$C = A + B$	A와 B의 합을 C에 저장, 덧셈 연산
*	$C = A * B$	A와 B의 곱을 C에 저장, 곱셈 연산
/	$C = A / B$	A를 B로 나눈 몫을 C에 저장, 나눗셈 연산
%	$C = A \% B$	A를 B로 나누었을 때 나머지를 C에 저장, 나머지 연산
++	$C = ++A$	A 값에 1을 더한 값을 C에 저장, 증가 연산
--	$C = --A$	A 값에서 1을 뺀 값을 C에 저장, 감소 연산

# 대입 연산자

- 연산에 사용되는 변수와 연산 결과를 저장하는 변수가 동일할 경우에 사용
- 수식을 축약하여 표현할때 사용

연산자	사용 예	동일 연산	설명
<code>+=</code>	<code>C += A</code>	<code>C = C + A</code>	C와 A의 합을 C에 저장
<code>-=</code>	<code>C -= A</code>	<code>C = C - A</code>	C에서 A를 뺀 차를 C에 저장
<code>*=</code>	<code>C *= A</code>	<code>C = C * A</code>	C와 A의 곱을 C에 저장
<code>/=</code>	<code>C /= A</code>	<code>C = C / A</code>	C를 A로 나눈 몫을 C에 저장
<code>%=</code>	<code>C %= A</code>	<code>C = C % A</code>	C를 A로 나누었을 때 나머지를 C에 저장

# 관계 연산자

- 변수 사이의 일치성 판단하는 조건문에 사용
- 두 개의 값을 비교하여 참(true)과 거짓(false) 값으로 반환.

연산자	사용 예	설명
==	A == B	A와 B의 값이 같은지 비교
===	A === B	A와 B의 값뿐만 아니라 자료형도 같은지 비교
!=	A != B	A와 B의 값이 다른지 비교
>	A > B	A가 B보다 큰지 비교
>=	A >= B	A가 B보다 크거나 같은지 비교
<	A < B	A가 B보다 작은지 비교
<=	A <= B	A가 B보다 작거나 같은지 비교

# 논리 연산자

- 변수 또는 값의 조합이 논리적으로 참(true)인지 거짓(false)인지를 판별
- True : "1", false : "0" 반환

연산자	사용 예	설명
&&	A && B	and 연산, A와 B 둘 다 참일 때만 참이다.
	A    B	or 연산, A와 B 둘 중에 하나만 참이면 참이다.
!	!A	not 연산, A가 참이면 거짓이 되고, 거짓이면 참이 된다.

A	B	A && B	A    B	!A
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

# 삼항 연산자

## ■ 형식

```
값 = 조건식 ? 처리내용 1 : 처리내용 2;
```

» 조건 결과값이

- › 참(true)이면 "처리내용 1",
- › 거짓(false)이면 "처리내용 2" 반환

## ■ 사례

```
val = ("a" == "a") ? "같습니다." : "같지 않습니다."  
document.write("a와 a가 같나요? " + val + "<br />");
```

```
val = ("a" == "b") ? "같습니다." : "같지 않습니다."  
document.write("a와 b가 같나요? " + val + "<br />");
```

# 증감 연산자

## ■ 증감 연산자

연산자	설 명	사용 예	연산자 구분
++	1씩 증가	++A : 먼저 1 증가한 값 출력 A++ : 처음은 증가하지 않고 다음 값부터 1씩 증가	전위 증감 후위 증감
--	1씩 감소	--A : 먼저 1 감소한 값 출력 A-- : 처음은 감소하지 않고 다음 값부터 1씩 감소	전위 증감 후위 증감

# 제어문

- 실행 코드의 실행 흐름을 논리적으로 제어하는 구문
- 기본 구조
  - » 조건 검사 ➔ 선택 또는 반복
- 종류
  - » 선택문
    - › if - else
    - › switch - case
  - » 반복문
    - › while
    - › for
    - › do - while

# if - else 구문

## ▪ if 문

- » if 문의 조건식에 만족(true)하면 { } 안의 문장을 수행

```
if(조건식) {  
    문장... 문장...  
}
```

## ▪ if~else 문

- » 조건식에 만족하면 문장 1 수행,  
만족하지 않으면 else 다음문장 수행

```
if(조건식) { 문장 1... }  
else { 문장 2... }
```

## ▪ 다중 if~else 문

- » 여러 문장 중에서 조건에 만족하는 문장을 실행

```
if(조건식) { 문장1... }  
else if(조건식) { 문장2... }  
else if(조건식) { 문장3... }  
else if(조건식) { 문장4... }  
else { 문장5... }
```



# switch - case 구문

- 다중 if 문을 대신하여 switch 문을 사용하면 간단하게 구현이 가능하다.
- switch 문에 제공된 식의 결과 값과 일치하는 case 문 실행

```
switch(변수) {  
    case 상수1 : 문장1; break;  
    case 상수2 : 문장2; break;  
    case 상수3 : 문장3; break;  
    default : 문장4;  
}
```

# 반복문

## ▪ while 문

- » 주어진 조건을 만족하는 동안 반복해서 지정된 기능 수행
- » 조건이 참인 동안 계속해서 문장이 실행되고, 거짓이 되는 순간 루프를 벗어남
- » 초기값이 while 문 밖에 존재

```
while(조건식) {  
    문장...  
}
```

```
do {  
    문장...  
} while(조건식)
```

## ▪ do~while 문

- » do~while 문은 최소한 한 번은 루프를 실행 - while 문과의 차이점
- » 조건이 아래에 있기 때문에 루프를 실행한 후에 조건을 체크

# 반복문

## ▪ for 문

- » 초기값과 조건식, 증가분을 한꺼번에 지정하는 반복문
- » 일반적으로 반복 횟수가 정해져 있거나 확인할 수 있을 때 사용
- » 변수에 초기값을 주고 이 값이 조건을 만족시키는 동안에만 작동

```
for (초기값; 조건식; 증감식) {  
    실행문...  
}
```

## ▪ 다중 for 문

- » 단일 for 문을 동시에 여러 번 사용하는 반복문
- » 조건이 아래에 있기 때문에 루프를 실행한 후에 조건을 체크

```
for (초기식; 조건식; 증감식) {  
    실행문...  
    for (초기식; 조건식; 증감식) {  
        실행문...  
    }  
}
```

# 함수

## ■ 함수

- » 프로그램 내에서 특정 작업 수행 위해 독립적으로 만들어진 하나의 단위
- » 예약어 `function` 으로 시작
- » 함수의 시작과 끝을 구분하는 기호 → 중괄호(`{ }`)
- » 호출 영역에서 함수 영역으로 데이터를 전달하기 위해 전달인자 사용

## ■ 구문 형식

### 함수 정의

```
function name(arguments)
{
    statements;
}
```

예약어      함수명  
형식매개변수  
함수몸체

### 함수 호출

```
name(arguments_values);
```

실매개변수  
함수명

# 이벤트와 이벤트 핸들러

- 이벤트(event)
  - » 프로그램 실행 중 발행하는 특정한 상황을 표시
  - » 주로 사용자의 특정한 액션에 의해 발생
- 이벤트 핸들러(Event Handler)
  - » 특정 이벤트에 대응하는 실행 코드
  - » 함수로 정의

# 이벤트와 이벤트 핸들러

## ■ 종류

	이벤트 핸들러	설명
마우스	onclick	버튼 등을 마우스로 클릭할 때
	ondblclick	버튼 등을 마우스로 더블 클릭할 때
	onmouseover	링크나 그림, 버튼 위에 마우스 포인터를 올릴 때
	onmouseout	링크나 그림, 버튼에서 마우스 포인터가 빠져나갈 때
	onmousedown	마우스 버튼을 클릭하는 순간
	onmouseup	마우스 버튼을 클릭했다가 떼는 순간
	onmousemove	마우스를 움직이는 순간

# 이벤트와 이벤트 핸들러

## ■ 종류

	이벤트 핸들러	설명
마우스	ondragdrop	마우스를 클릭한 상태에서 움직일 때
포커스	onfocus	커서가 위치할 때
	onblur	커서가 다른 곳으로 옮겨갈 때
키보드	onkeydown	키보드를 누를 때
	onkeyup	키보드를 눌렀다가 떼는 순간
폼	onsubmit	전송(확인) 버튼을 클릭할 때
	onreset	취소 버튼을 클릭할 때
	onselect	문자열을 선택하거나 체크 박스, 라디오 버튼 등을 선택할 때
	onchange	리스트 박스의 값을 변경할 때
윈도우	onload	브라우저에서 문서를 읽을 때
	onunload	브라우저에서 문서를 닫을 때
	onmove	브라우저를 이동했을 때
	onresize	브라우저의 크기를 변경했을 때

# 이벤트 핸들러 활용

## ■ 이벤트 핸들러 실행 방법

```
<element event-handler="javascript-code">
```

- » element : HTML element
- » event-handler : 이벤트 핸들러 이름
- » javascript-code : Javascript 문장 또는 함수 호출

## ■ 사례

```
<a href="http://www.naver.com"  
  onclick='alert("naver로 이동");'>NAVER</a>
```



# 객체 개요

- 객체 정의
  - » 프로그램으로 제어 하는 모든 대상
- 속성(Attribute)
  - » 객체가 가지는 형태나 성격
  - » 버튼의 크기 및 색상, 이름 등
- 메서드(Method) or 함수(Function)
  - » 객체가 행하는 동작

# 객체 사용

- 속성과 메소드를 사용하는 방법

객체명.속성

예 location.href

객체명.속성=속성값

예 location.href = "http://www.hanb.co.kr"

객체명.메소드(인수)

예 document.write("...")

- 객체명과 속성 또는 메소드 사이에 점(.)을 사용하여 계층적 관계 표현

상위객체이름.하위객체이름.속성

예 document.links.length

상위객체이름.하위객체이름.메소드

예 document.images[0].name

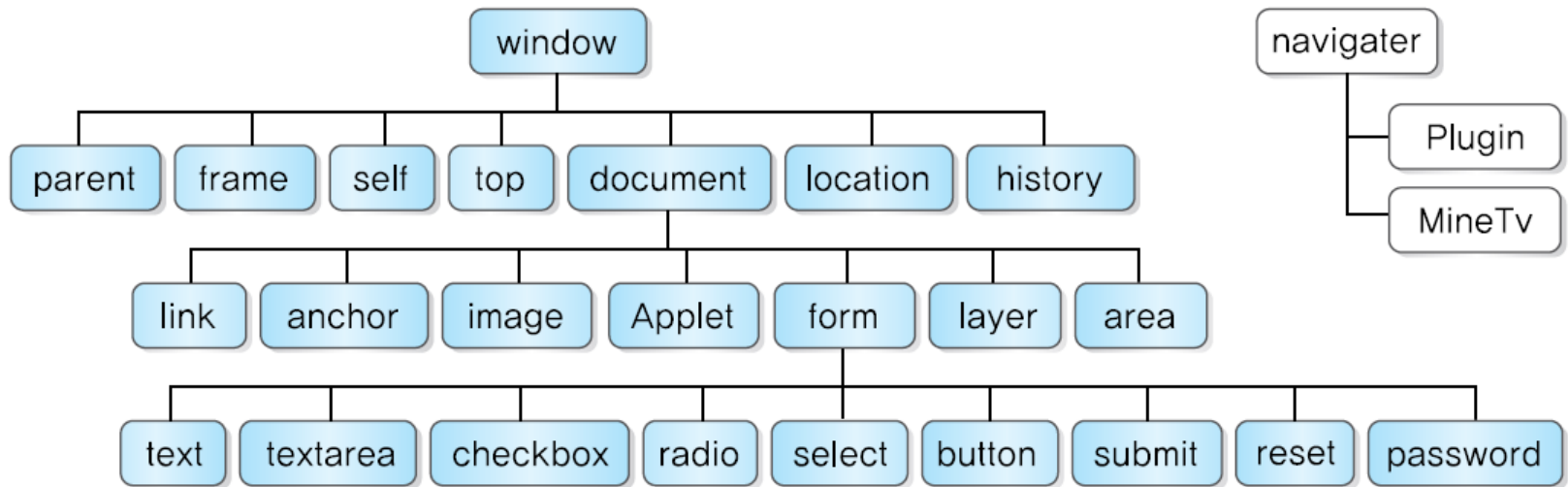
# 자바스크립트 객체 종류

- 내장 객체

- » Date 객체, String 객체, Array 객체, Screen 객체

- 클라이언트 객체

- » 계층구조



# 클라이언트 객체

## ■ 종류

객체	특징
window	최상위 객체로 자바스크립트에서 사용되는 모든 객체를 처리
navigator	웹 브라우저에 관련된 정보를 제공
link	하이퍼링크와 관련된 작업 처리
anchor	특정 위치로의 이동 객체
image	문서 내의 그림에 관련된 처리
location	현재 문서의 URL과 관련된 정보를 처리
history	방문기록에 관련한 처리
document	문서에 대한 정보를 처리
form	폼에 있는 양식 객체를 처리

# window 객체

- 창 열기, 창 닫기, 창 크기 조절 등의 작업을 수행
- 속성과 메소드

속성	설명
defaultStatus	상태표시줄에 나타낼 초기 문자열 설정
status	상태표시줄에 나타낼 문자열 설정
frames	창에 포함된 프레임들을 배열로 지정
opener	새 창을 열도록 한 문서
parent	현재 프레임의 부모 문서 창(프레임)
self	현재 프레임 자신
top	프레임을 무시한 하나의 전체 창
screenX	창의 x 좌표값 반환
screenY	창의 y 좌표값 반환
closed	창이 닫힌 상태에 대한 true, false 반환
name	창의 이름 반환
length	창 안의 프레임수 반환

메소드	설명
alert("내용")	경고창 표시
prompt("창 제목", "입력창에 미리 들어가 있는 문장")	입력창 표시
confirm("내용")	확인창 표시
open("문서명", "창이름", "속성")	새 창 열기
close( )	창 닫기
moveBy(x,y)	브라우저를 x, y만큼 이동
MoveTo(x,y)	브라우저를 x, y로 이동
resizeBy(x,y)	브라우저의 크기를 상대적 x, y로 설정
resizeTo(x,y)	브라우저의 크기를 x, y 크기로 설정
scroll(x,y)	스크롤함
scrollBy(x,y)	현재 위치에서 스크롤을 x, y만큼 이동
scrollTo(x,y)	스크롤을 x, y로 이동
setTimeout("명령", "시간")	지정된 시간 간격으로 명령을 반복 수행, 시간(msec)
clearTimeout( )	타이머 해제
print( )	문서 출력

# window 객체

▪ open() 메소드 → 다른 윈도우를 표시할 때 사용하는 메서드

▪ 메소드의 속성

속성	속성값	설명
menubar	yes/no	메뉴바 표시 여부
toolbar	yes/no	툴바 표시 여부
location	yes/no	주소표시줄 표시 여부
resizable	yes/no	창 크기 조절 가능 여부
status	yes/no	상태표시줄 표시 여부
scrollbars	yes/no	스크롤바 표시 여부
fullscreen	yes/no	전체 화면으로 표시
width	pixel	창의 너비
height	pixel	창의 높이
left	pixel	화면에서 창이 왼쪽에서 떨어진 간격
top	pixel	화면에서 창이 위쪽에서 떨어진 간격

# document 객체

- HTML 문서 정보와 포함된 객체에 대한 정보를 포함하며 관련된 동작을 수행

- 속성과 메서드

메소드	설명
open( )	문서의 내용을 나타냄
close( )	open( )으로 나타낸 문서를 닫음
clear( )	문서의 모든 내용을 삭제
write( )	태그를 포함하여 문자열을 출력
writeln( )	행 마지막에 new line 처리

속성	설명
bgColor	문서의 배경색을 지정
fgColor	문서의 글자색을 지정
linkColor	하이퍼링크의 문자 색상 지정
vlinkColor	방문한 링크의 문자 색상을 지정
lastModified	문서가 마지막으로 수정된 날짜를 저장
location	문서의 URL 주소를 저장
URL	문서의 URL 주소값을 반환
domain	서버의 도메인명을 지정 또는 반환
title	문서의 제목을 지정하거나 반환
cookie	쿠키 파일의 정보를 지정하거나 반환
images	문서에 삽입된 그림을 배열로 제공
links	문서의 링크를 배열로 제공
forms	문서의 form 태그로 이루어진 폼을 배열로 제공
embeds	문서에 포함된 플러그인을 배열로 제공

# history 객체

- 웹 브라우저 실행 후 사용자가 접속한 페이지의 목록을 순서대로 관리하는 객체
- 페이지의 앞/뒤 이동 기능 지원
- 속성과 메소드

속성	설명
length	히스토리에 보관되어 있는 URL 주소 수

메소드	설명
back( )	이전 페이지로 이동
forward( )	다음 페이지로 이동
go(n)	n=양수 : 히스토리 목록에서 n만큼 다음 페이지로 이동 n=음수 : 히스토리 목록에서 n만큼 이전 페이지로 이동



# location 객체

- 웹 브라우저의 주소 입력줄의 주소를 관리하는 객체
- 속성과 메소드

속성	설명
host	호스트명과 포트 번호 정보를 표시
hostname	호스트명을 표시
href	전체 URL을 표시
pathname	웹 문서의 경로를 표시
port	포트 번호를 표시
protocol	프로토콜명을 표시

메소드	설명
reload( )	문서 새로고침
replace("URL")	현재 문서를 지정한 URL로 대체

# frame 객체

- 프레임에 관련한 정보를 제공하고 관리할 수 있는 객체
- 인덱스 번호는 위에서 아래로, 왼쪽에서 오른쪽 순서로 부여

frames[0]		
frames[1]	frames[2]	frames[3]
frames[4]		

- 속성과 메소드 : window 객체에서 사용하는 속성과 메소드 사용

속성	설명
length	프레임의 개수를 반환
name	해당 프레임의 이름을 반환
parent	해당 프레임의 상위 프레임을 지정
self	해당 프레임 자신을 지정

# image 객체

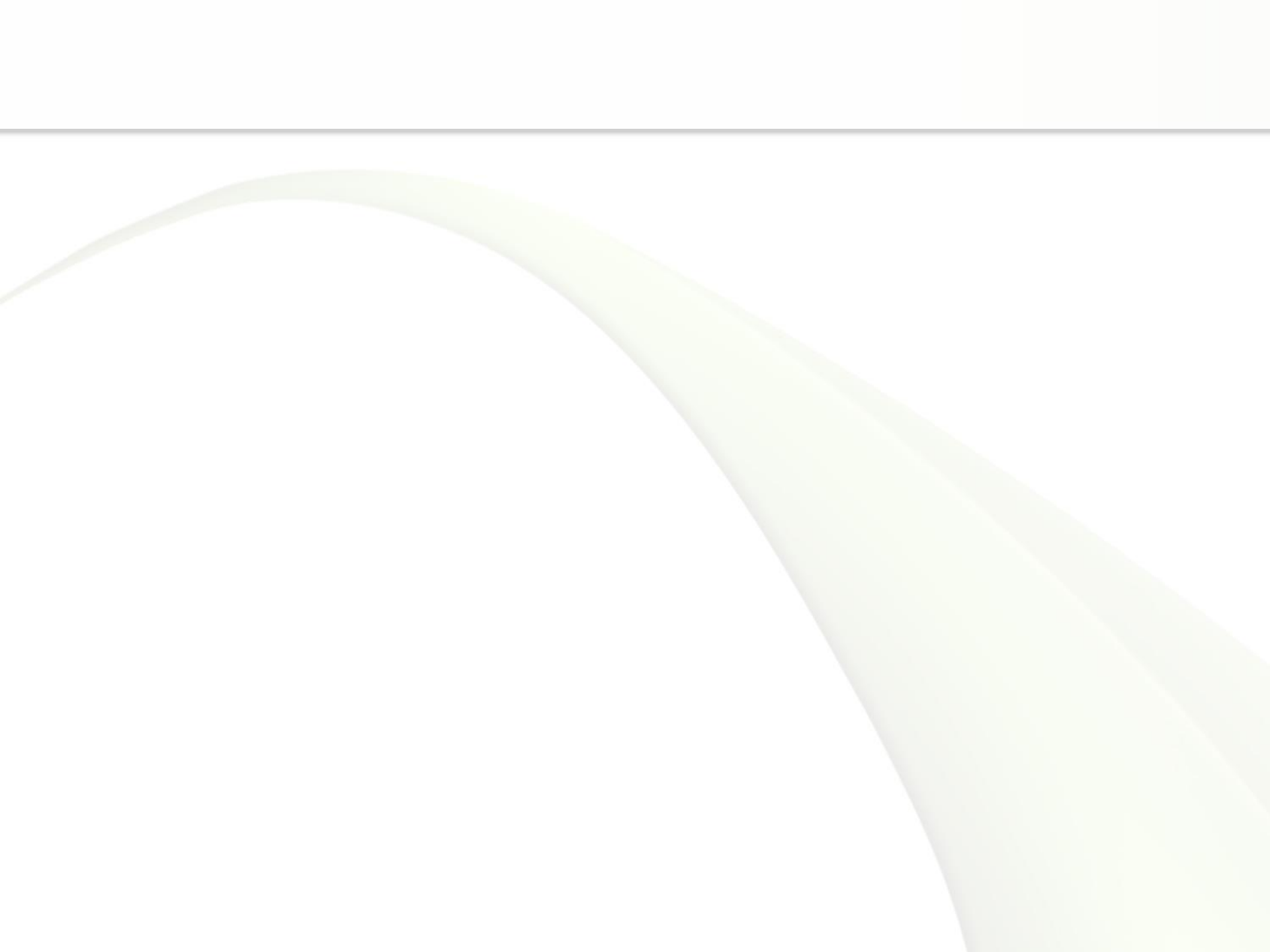
- 삽입 그림에 대한 정보를 이용하여 다양한 형태로 제어
- 속성

속성	설명
name	그림에 설정된 name 속성값을 반환
length	문서에 삽입된 그림의 개수를 반환
src	그림의 src 속성값을 반환
hspace	설정된 그림의 좌/우 여백값을 반환
vspace	설정된 그림의 상/하 여백값을 반환
width	설정된 그림의 폭을 반환
height	설정된 그림의 높이를 반환
border	설정된 그림의 테두리 값을 반환

# event 객체

- 마우스를 클릭하거나 키보드를 누르는 등의 이벤트를 제어하고 관리
- 속성

속성	설명
altKey	 키를 누르면 true 값 발생
button	마우스 버튼을 누른 경우 이벤트 발생
clientX	윈도우 영역에서 마우스의 x 좌표값
clientY	윈도우 영역에서 마우스의 y 좌표값
ctrlKey	 키를 누르면 true 값 발생
keyCode	키보드의 키 값을 전달받음
screenX	전체 화면에서 마우스의 x 좌표값 설정
screenY	전체 화면에서 마우스의 y 좌표값 설정
shiftKey	 키를 누르면 true 값을 발생



# String 객체

- 문자열에 관련하여 글자의 속성이나 하이퍼링크 등을 설정

- 메서드

메소드	설명
fontSize(n)	문자열에 크기를 지정
fontcolor("color")	문자열에 색상을 지정
bold( )	문자열을 진하게 표현
italics( )	문자열을 기울임꼴로 표현
sup( )	문자열을 윗첨자로 표현
sub( )	문자열을 아래첨자로 표현
link("URL")	문자열에 하이퍼링크를 설정
indexOf("문자",n)	문자열에서 해당 문자의 위치를 왼쪽 n번째부터 찾아서 찾은 위치를 반환 n이 생략되면 왼쪽에서부터 찾고 없으면 -1을 반환 인덱스 번호는 0부터 시작
lastIndexOf("문자",n)	문자열에서 해당 문자의 위치를 오른쪽 n번째부터 찾아서 찾은 위치를 반환 n이 생략되면 오른쪽에서부터 찾고 없으면 -1을 반환 인덱스 번호는 0부터 시작
charAt(n)	문자열에서 n번째 위치한 문자를 반환 인덱스 번호는 0부터 시작

메소드	설명
substring(n,m)	문자열의 n번째 문자부터 m번째 문자까지를 반환
substr(n,m)	문자열의 n번째 문자부터 m개의 문자를 반환
split("구분문자")	구분 문자로 문자열을 분리
toUpperCase( )	모두 대문자로 변환
toLowerCase( )	모두 소문자로 변환

# Date 객체

## ■ 날짜와 시간에 관련된 처리 - new 연산자

```
var today = new Date();  
document.write(today);
```

## ■ 메소드

메소드	설명
getFullYear( )	연도 정보를 반환
getMonth( )	월 정보를 반환(0~11)
getDate( )	일 정보를 반환(1~31)
getDay( )	요일 정보를 숫자로 반환(일요일 0 ~ 토요일 6)
getHours( )	시간 정보를 반환
getMinutes( )	분 정보를 반환
getSeconds( )	초 정보를 반환
getTime( )	1970년 1월 1일 0시부터 지정한 시간까지를 1/1000초로 반환

# Array 객체

- 배열 객체를 생성하고 생성된 변수에 인덱스 번호로 접근할 수 있는 객체

```
변수명 = new Array(배열개수);
```

```
변수명 = new Array(값1, 값2, 값3, ... );
```



# Math 객체

- 수학 연산 기능을 제공
- 속성과 메소드

속성	설명
PI	원주율(3.14159..)
E	오일러 상수 자연대수의 밑수(2.71828..)
LN10	밑수가 10인 자연로그(2.302585..)
LN2	밑수가 2인 자연로그(0.6931471,,)
SQRT2	2의 제곱근(1.41421..)
LOG10E	밑수가 10인 E로그(0.43429448..)
LOG2E	밑수가 2인 E로그(1.442695)

메소드	설명
floor(x)	x의 소수점 이하 값을 절사하는 함수
ceil(x)	x의 소수점 이하 값을 무조건 반올림하는 함수
max(x, y)	x와 y 중 큰 값을 반환
min(x, y)	x와 y 중 작은 값을 반환

메소드	설명
sin(x)	sine 함수
cos(x)	cosine 함수
tan(x)	tangent 함수
asin(x)	arc sine 함수
acos(x)	arc cosine 함수
atan(x)	arc tangent 함수
abs(x)	x의 절대값
exp(x)	E에 대한 지수 함수
log(x)	x 로그값
pow(x, y)	지수 함수, x 의 y 제곱
sqrt(x)	제곱근 함수 x의 루트값
random( )	난수 발생 함수
round(x)	x값에 대한 반올림 함수

# Screen 객체

- 해상도, 화면 크기, 색상 등에 관련한 정보를 제공
- 사용자 해상도에 따라 적절한 처리 필요시 유용
- 메소드

메소드	설명
height	화면 전체 높이 정보를 반환
width	화면 전체 너비 정보를 반환
availHeight	작업표시줄을 제외한 사용 가능한 화면의 높이 정보를 반환
availWidth	작업표시줄을 제외한 사용 가능한 화면의 너비 정보를 반환
colorDepth	화면의 색상 수를 반환

# form 객체

- <form> 태그 내의 폼 양식을 제어하고 관리할 수 있는 객체
- 속성과 메소드

속성	설명
action	<form> 태그의 action 속성에 기록된 정보를 보관
elements	텍스트 입력상자, radio 버튼 등 폼 양식을 배열로 저장
encoding	encoding 속성에 기록된 정보를 보관
method	method 속성에 기록된 정보를 보관
target	target 속성에 기록된 정보를 보관
length	폼 양식의 개수
name	name 속성에 기록된 정보를 보관

메소드	설명
reset( )	form 양식에 입력된 값을 초기화
submit( )	form 양식에 입력된 값을 전달

# 텍스트 입력 상자

- 키보드를 이용한 자유 형식의 텍스트 입력 상자
- type 속성에 따라 일반 텍스트, 패스워드 등의 입력 방식을 조정
- 속성과 메소드

속성	설명
type	텍스트 입력상자의 종류를 지정(text, password)
name	텍스트 입력상자의 이름
value	텍스트 입력상자에 입력되는 내용
defaultValue	텍스트 입력상자에 입력된 초기값
form	텍스트 입력상자를 포함하는 폼 객체

메소드	설명
focus( )	텍스트 입력상자로 커서 옮기기
blur( )	텍스트 입력상자에서 커서 제거하기
select( )	텍스트 입력상자에 있는 값을 블록으로 선택하기

# 리스트 박스

- 유사 또는 동일한 객체의 목록을 표시하고 사용자의 선택을 지원하는 입력 요소
- `select` 객체 이용, `option` 속성 이용
- `select` 객체의 속성

속성	설명
<code>type</code>	<code>multiple</code> 정보를 반환
<code>length</code>	목록의 개수를 반환
<code>options</code>	<code>&lt;option&gt;</code> 태그를 배열로 구성
<code>selectedIndex</code>	목록을 배열 번호로 표시하거나 배열 번호를 반환

# button 객체

- 폼 양식에서 입력에 사용되지는 않고 폼의 동작을 제어하는 객체
- 속성과 메소드

속성	설명
form	버튼의 입력 양식을 포함하는 객체
name	버튼의 name 속성을 반환
value	버튼의 value 속성을 반환
type	버튼의 type 속성을 반환

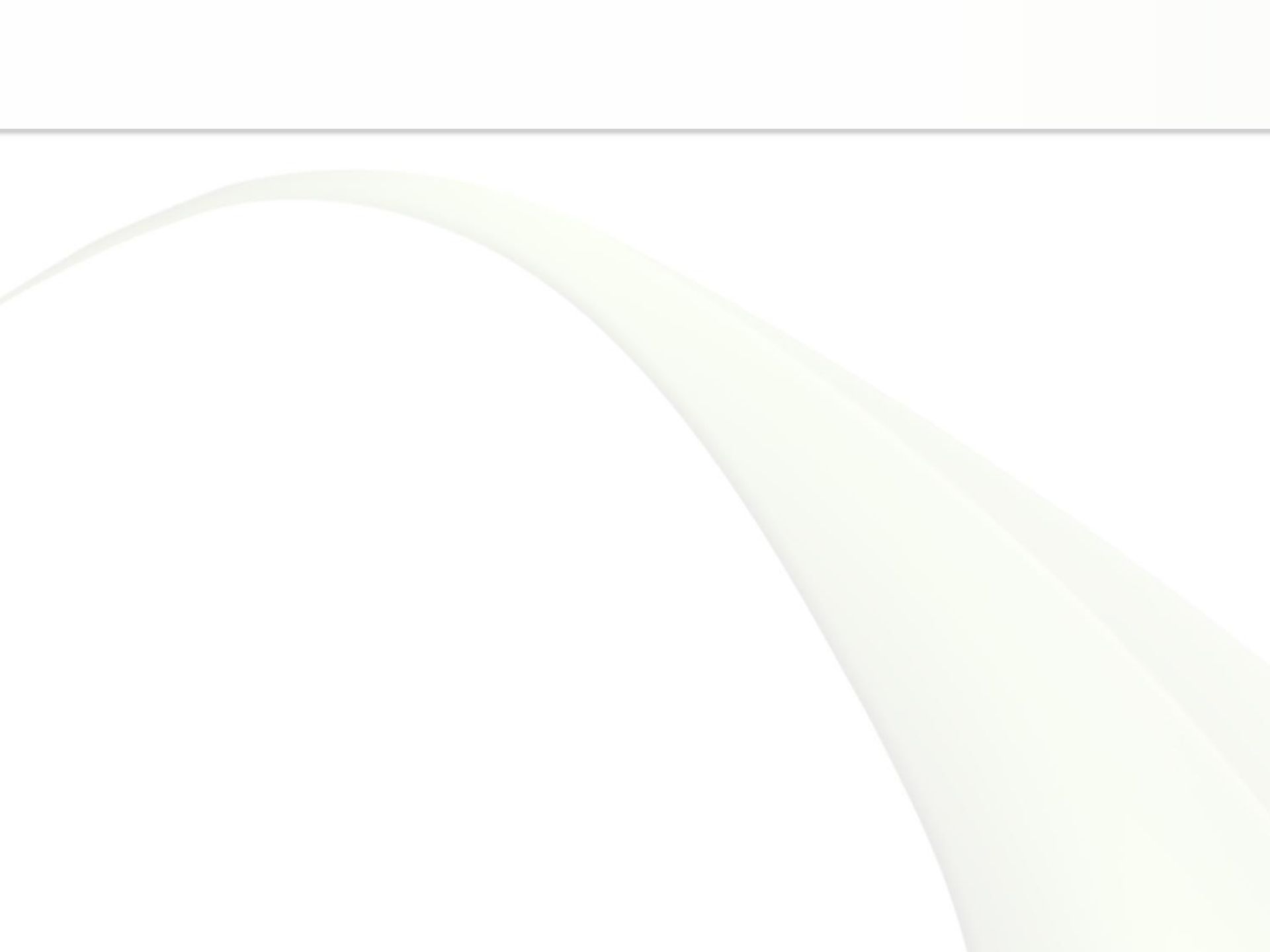
메소드	설명
click( )	버튼을 클릭하는 메소드

# radio, checkbox 객체

- radio 객체 : 여러 선택 옵션 중 하나를 선택
- checkbox 객체 : 다중 선택
- radio, checkbox 객체 속성과 메소드

속성	설명
form	입력 양식을 포함하는 폼 객체
name	입력 양식의 이름
value	입력 양식에 할당되는 값
length	입력 양식 수
checked	입력 양식 선택 여부
defaultChecked	기본적으로 선택된 상태로 표시될 입력 양식
type	type 속성값을 반환

메소드	설명
click( )	입력 양식을 클릭하는 메소드





# JSON

- Javascript Object Notation
- 사람에게 가독성이 높고 코드를 통해 간단하게 생성 및 처리할 수 있는 형식
- 일부 목적을 공유하는 XML에 비해 빠르고 효율적이고 사용 편의성이 높음
- 프로그램 언어로부터 완전하게 독립적인 텍스트 형식 → 높은 호환성
- 대부분의 주요 프로그램 언어에서 공식 데이터 교환 양식으로 사용
  - » 이기종 시스템 사이의 데이터 교환에서 실질적인 표준

# JSON 표기법

## ■ 기본 형식

» key : value → (사례) "name" : "John Doe"

## ■ 사용 사례

데이터 타입	표현 방법	예
number	integer 또는 float	"number" : 1
string	큰 따옴표로 묶음	"name" : "장동건"
boolean	true 또는 false	"isResult" : true
object	여러 개의 key/value 를 입력하여 중괄호로 묶음	{ "name" : "장동건", "gender" : "남자" }
array	여러 개의 object를 대괄호로 묶음	{ "employees" : [ {"name" : "장동건", "gender" : "남자"}, {"name" : "싸이", "gender" : "남자"}, {"name" : "김태희", "gender" : "여자"} ] }

# JSON 사용 사례

```
<html>
<body>
  이름 : <span id="name"></span><br/>
  나이 : <span id="age"></span><br/>
  주소 : <span id="address"></span><br/>
  전화번호 : <span id="phone"></span><br/>

<script type="text/javascript">
  var person= {
    "name" : "장동건",
    "address" : "서울시 강남구 역삼동 1",
    "age" : 18,
    "phone" : "070-1234-5678"
  };
  document.querySelector("#name").innerHTML = person.name
  document.querySelector("#age").innerHTML = person.age
  document.querySelector("#address").innerHTML = person.address
  document.querySelector("#phone").innerHTML = person.phone
</script>

</body>
</html>
```

# JSON / XML 비교

```
1: {  
2: "book": {  
3:   "type": "textbook",  
4:   "pages": "256",  
5:   "title": "Programming Pearls 2nd Edition",  
6:   "description": "The first edition of Programming",  
7:   "rating": "4.5",  
8:   "coverType": "paperback",  
9:   "genre": "Computer Science",  
10:  "author": "Jon Bentley",  
11:  "publisher": "Addison-Wesley Professional",  
12:  "copyright": "1999"  
13: }  
14: }
```

JSON 포맷

XML 포맷

```
1: <book>  
2:   <type>textbook</type>  
3:   <pages>256</pages>  
4:   <title>Programming Pearls 2nd Edition</title>  
5:   <description>The first edition of Programming</description>  
6:   <rating>4.5</rating>  
7:   <coverType>paperback</coverType>  
8:   <genre>Computer Science</genre>  
9:   <author>Jon Bentley</author>  
10:  <publisher>Addison-Wesley Professional</publisher>  
11:  <copyright>1999</copyright>  
12: </book>
```

# JSON / XML 비교

구분	장점
XML	<ul style="list-style-type: none"><li>• 작성하기가 간편하다(tag구조)</li><li>• 사람이 읽기가 쉽다 (즉 각 정보들이 의미하는 바를 한눈에 보기가 좋다)</li><li>• DTD 등 XML자체의 기능을 확장할 여지가 많이 있다</li><li>• 파싱 : DOM 또는 SAX</li></ul>
JSON	<ul style="list-style-type: none"><li>• 내용이 함축적으로 최소한의 정보만을 가지고있다</li><li>• XML대비 용량이 획기적으로 줄어들고 속도는 그만큼 빨라지게 된다</li><li>• 객체구조와 {} 배열구조의 [] 적절한 만남으로 아주 효율적인 데이터 구성이 가능하다</li><li>• 파싱이 매우 간편하고, 때로는 일반적인 변수처럼 사용도 가능하다</li></ul>