

컴퓨터·소프트웨어

서울과학기술대학교 컴퓨터공학과
이길흥

정보기술

- Information Technology (IT)
 - 정보를 수집하여 가공하고 저장하고,
 - 보관된 정보를 검색하여 활용
- 정보통신기술
 - + 정보를 주고받는데 사용되는 기술
- 컴퓨터는 정보기술의 핵심

컴퓨터·소프트웨어 전공

- 컴퓨터에 관련된 모든 학문적 지식을 전문적으로 공부하는 것
- 다양한 분야에서 컴퓨터를 이용하여 좀 더 쉽고 편하게, 더 빠르고 정확하게, 좀 더 비용을 줄이거나 이익을 늘리기 위해 새로운 방법을 찾고 새로운 기술을 만들어 내는 것
- “인간이 할 일을 스스로 대신할 수 있는 기계” 개발

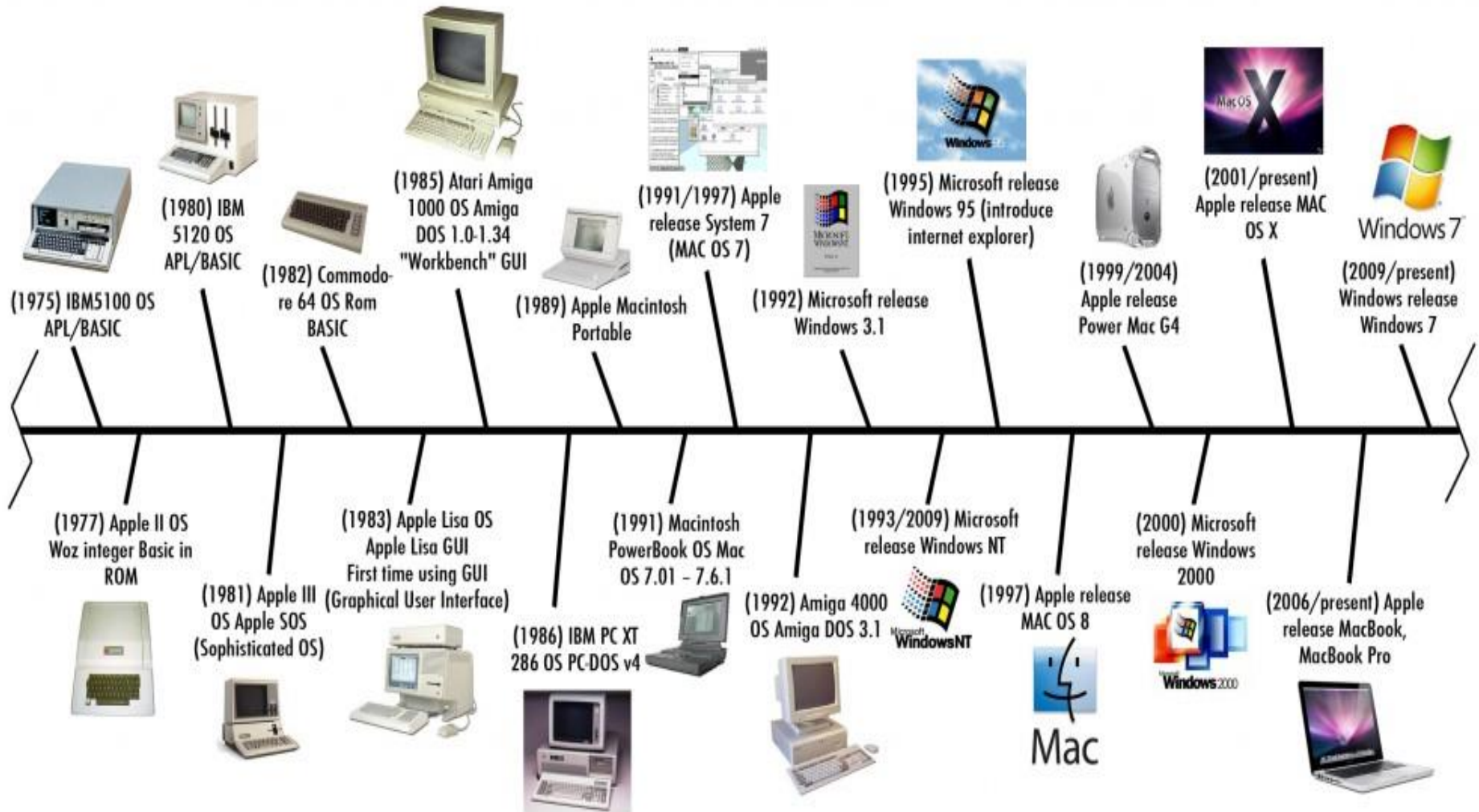
컴퓨터/소프트웨어 과학/공학

- 컴퓨터과학 (전산학, 전자계산학)
 - 이과대학 소속
 - Computation의 이론과 응용에 초점
 - 이론적 측면 (자료구조, 알고리즘, ...)
- 컴퓨터공학
 - 공과대학 소속
 - 컴퓨터 장치의 설계, 제작, 구현, 관리에 초점
 - 실제적 측면 (슈퍼컴퓨터, 멀티미디어, 통신, ...)
- 소프트웨어공학
 - 공과대학 소속
 - 프로그래밍 개발 분야 초점
 - 응용 측면 (인공지능, 보안, ...)

컴퓨터 · 소프트웨어분야

- 컴퓨터과학이론 분야
 - 이산수학, 알고리즘, 자료구조, 계산이론, 인공지능
- 컴퓨터시스템 분야
 - 하드웨어 : CPU, 메모리, 입출력장치, 디지털논리
 - 소프트웨어 : 운영체제, 시스템소프트웨어
 - 제품 : 슈퍼컴, PC, 임베디드시스템(휴대폰, 로봇, 게임기, 냉장고, 전자사전, ...)
- 소프트웨어응용 분야
 - 프로그래밍, 데이터베이스, 정보보호, 소프트웨어공학, 컴퓨터비전, 그래픽스, 정보검색, 인공지능
- 컴퓨터통신 분야
 - 아날로그/디지털통신, 유선/무선/광통신, 인터넷, 초고속통신, 멀티미디어통신

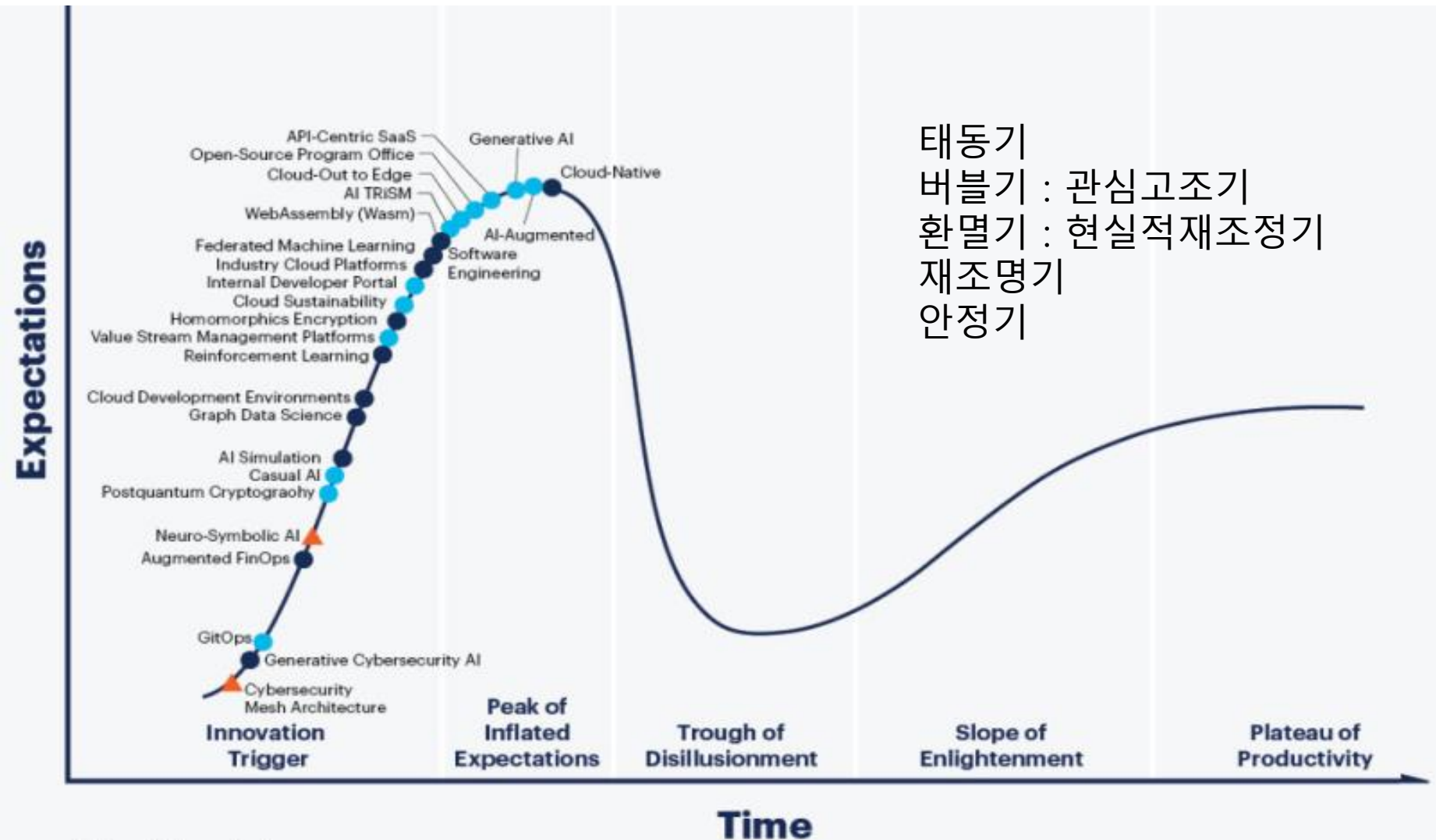
Computer History



신기술 분야

- 클라우드 컴퓨팅과 웹
 - 인터넷 기반(Cloud)의 컴퓨터기술(Computing)
- 빅데이터
 - 기존 데이터베이스 관리도구로 데이터를 수집,저장,관리,분석할 수 있는 역량을 넘어서는 대량의 정형 또는 비정형 데이터
- IoT (Intenet of Things, 사물인터넷)
 - 모든 사물이 인터넷으로 연결, 유비쿼터스 컴퓨팅
- 모바일로봇, 스마트카, 자율주행
- 차세대 통신, 정보보안
- 인공지능 챗GPT

Gartner's Hype Cycle for Emerging Technology, 2024



2023-2024 가트너 선정 10대 전략기술

No.	2023	2024
1	Digital Immune System	AI Trust, Risk and Security Management (AI TRISM)
2	Applied Observability	Continuous Threat Exposure Management (CTEM)
3	AI Trust, Risk and Security Management (AI TRISM)	Sustainable Technology
4	Industry Cloud Platforms	Platform Engineering
5	Platform Engineering	AI-Augmented Development
6	Wireless-Value Realization	Industry Cloud Platforms
7	Superapps	Intelligent Applications
8	Adaptive AI	Democratized Generative AI
9	Metaverse	Augmented Connected Workforce
10	Sustainable Technology	Machine Customers
11		
12		

컴퓨터 · 소프트웨어 업무

- 컴퓨터 하드웨어 개발
- 컴퓨터시스템소프트웨어개발
- 응용프로그램개발
- 웹서비스개발
- 융합기술개발
- 인공지능응용기술개발
- 컴퓨터시스템응용
- 정보보안시스템개발
- 멀티미디어응용 및 서비스개발
- 정보통신기술개발
- 정보가전기술개발

졸업 후 진로

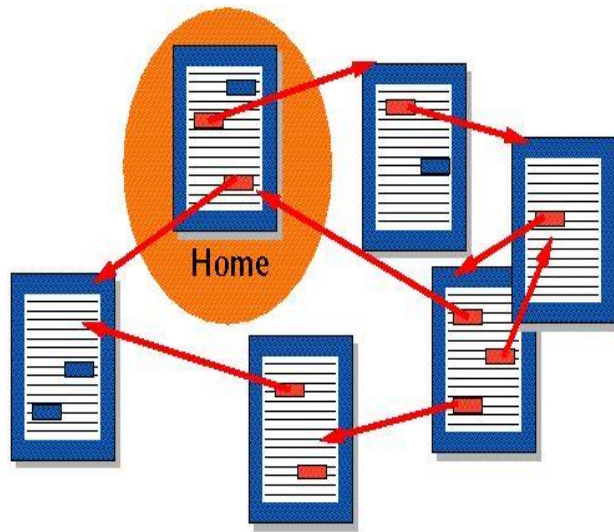
- KT, SKT, LG 통신회사
- 삼성, LG 등 컴퓨터 및 응용제품 제조회사
- 대우, 농협, 롯데, KCC 등 정보통신회사
- 대기업, 은행, 관공서, 학교 전산실
- 벤처기업
- 창업
- 연구소
- 대학원(국내, 국외)

웹(Web) 개요

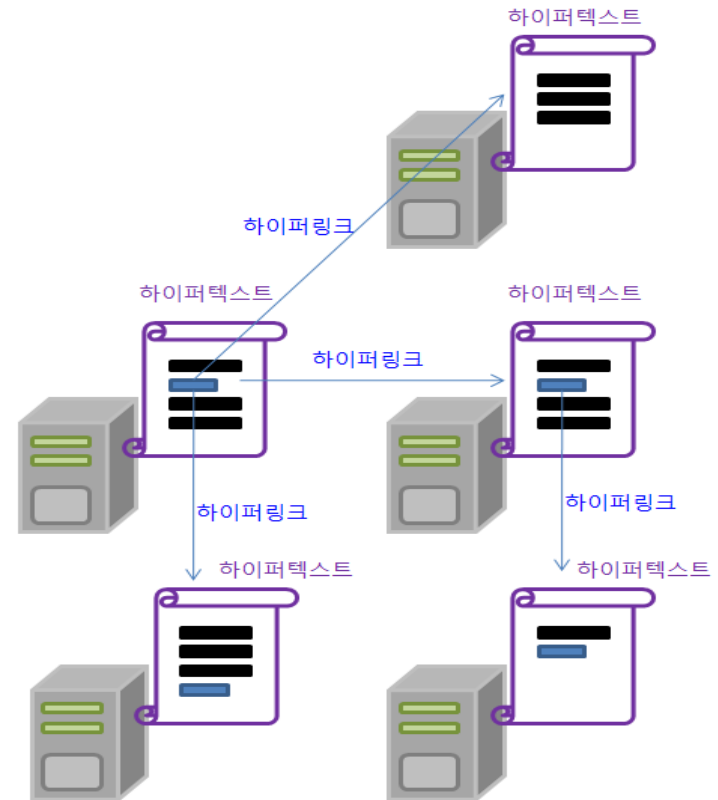
웹 개요

- 웹(Web, World Wide Web, WWW)
 - 정보를 하이퍼텍스트 형태로 제공하는 서비스
 - 문서안에 다른 문서에 대한 참조(하이퍼링크)를 포함시켜, 많은 문서가 유기적으로 결합됨
 - 인터넷에 연결된 컴퓨터들이 서로 정보를 공유하고 찾아볼 수 있는 공간
- 하이퍼텍스트
 - 정보들을 비순차적이고 비 연속적이며 비선형적인 체제로 유기적으로 결합시켜, 제목의 제시 순서에 관계없이, 이용자가 원하는 제목과 관련된 정보를 검색할 수 있도록 하는 정보 제공 방법

Web



서버 내

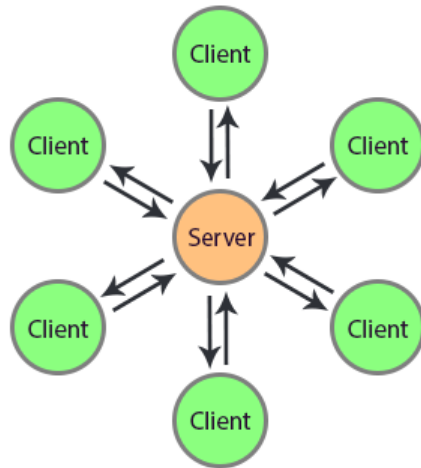


서버 간

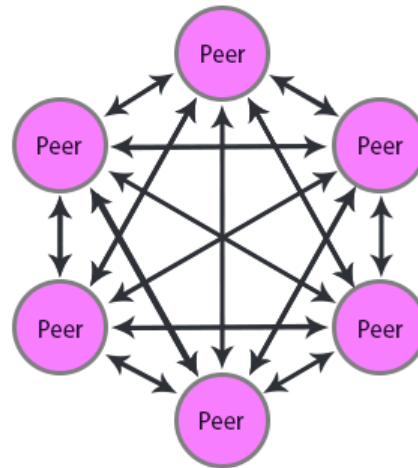
웹 서비스 구조

- 클라이언트/서버 구조 \leftrightarrow 피어투피어(Peer-to-Peer, P2P) 구조
 - 서버 : 서비스 제공
 - 클라이언트 : 서비스 이용
- 웹 서버
 - 웹 페이지 보유
 - 클라이언트가 서비스 요청을 하면 정보를 HTML 문서형태로 내려 보냄
- 웹 클라이언트
 - 웹 서버에게 서비스 요청을 보냄
 - 서버에서 내려온 HTML 형태의 정보를 화면으로 보여줌

Client/Server and P2P



Client / Server



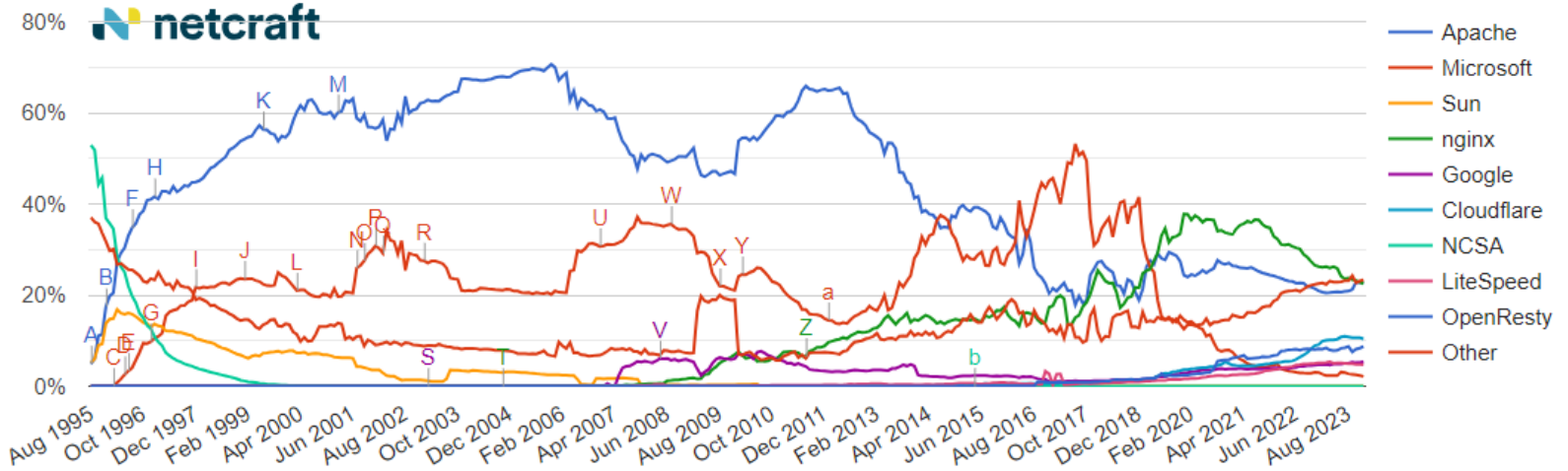
Peer To Peer

웹 프로그램 (소프트웨어)

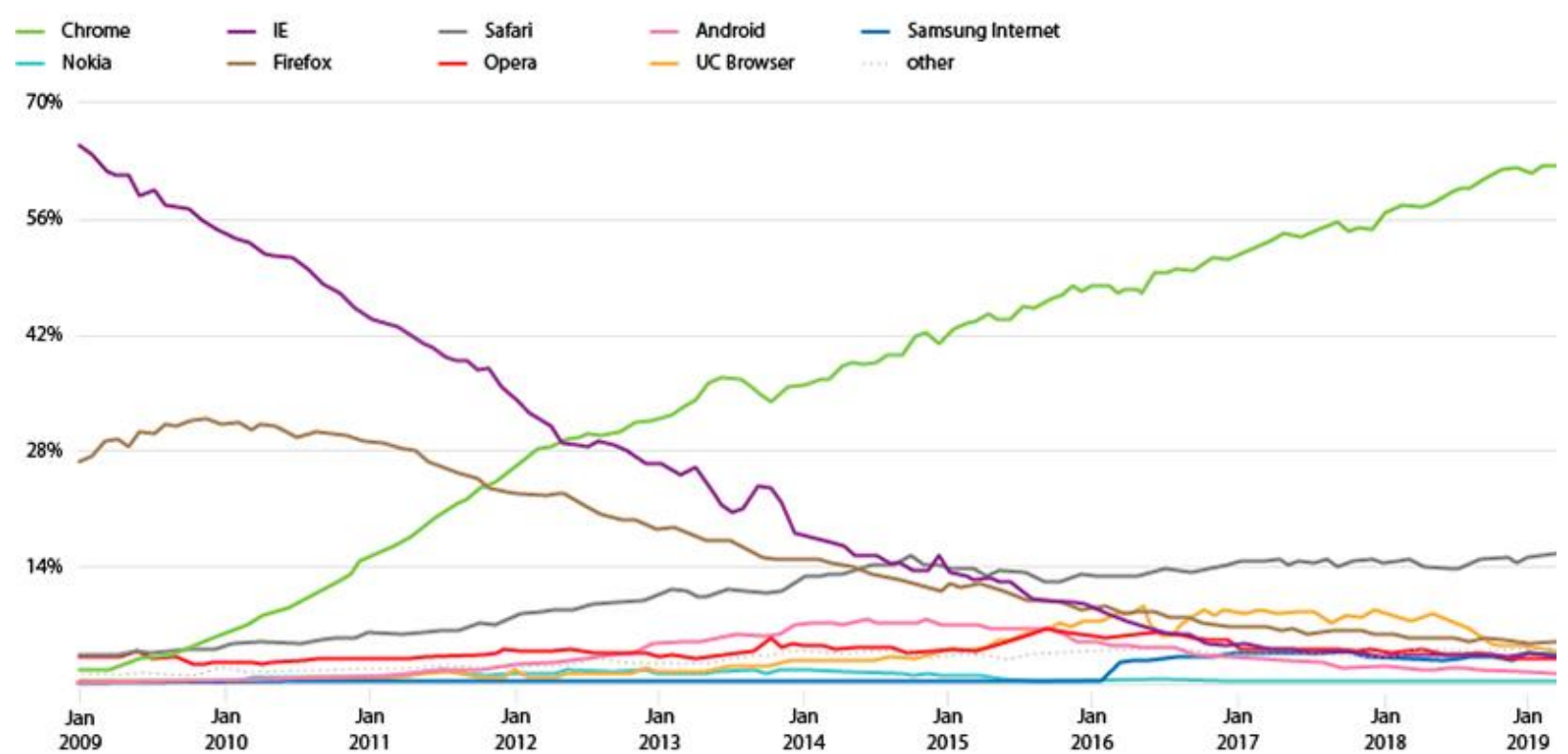


- 웹 서버
 - 브라우저의 요청을 처리
 - HTML 형태의 정보 제공 (Text, 멀티미디어 제공 가능)
 - Unix/Linux Httpd, Windows IIS, Apache Tomcat, NginX, Node.js
- 브라우저 (웹 브라우저) : 웹 클라이언트 프로그램
 - 웹 서버에 요청을 보내고, 서버로부터 오는 정보를 해석하여 화면에 표시해주는 프로그램
 - GUI (Graphic User Interface)를 통해 이용자의 요청을 처리
 - 크롬, 에지, Firefox, Safari

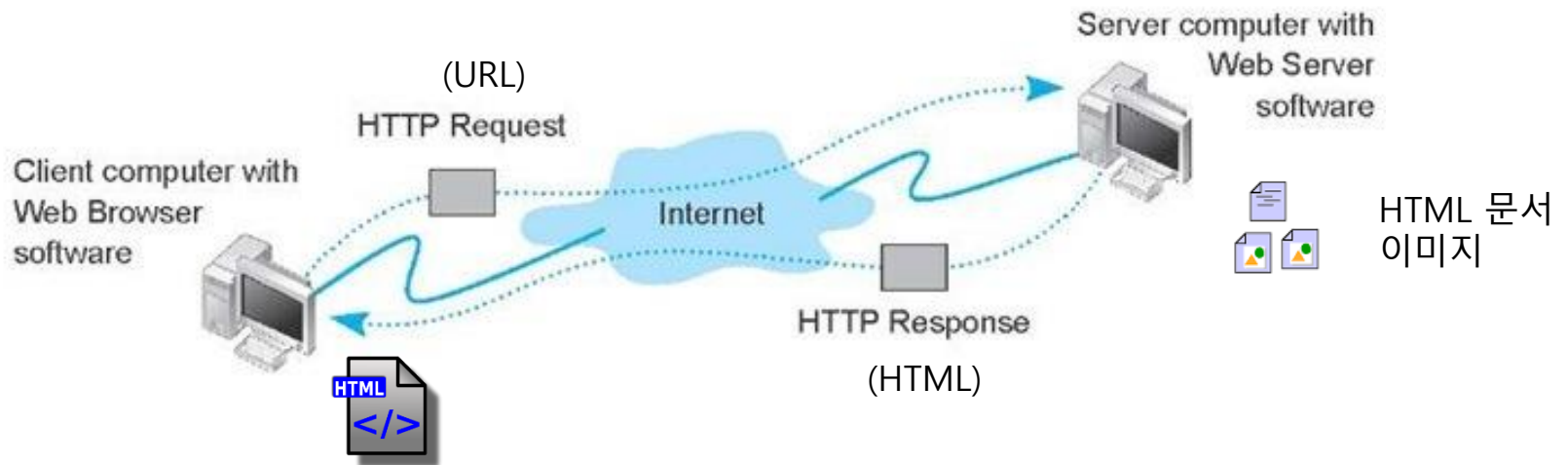
Web server developers: Market share of all sites



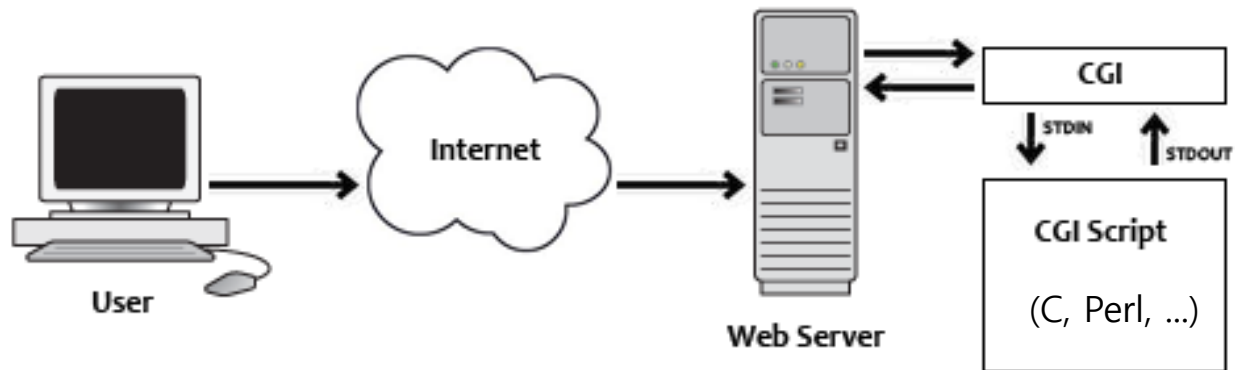
Developer	November 2023	Percent	December 2023	Percent	Change
Apache	248,343,154	22.74%	248,118,087	22.80%	0.06
nginx	249,368,944	22.83%	244,903,710	22.51%	-0.32
Cloudflare	115,937,937	10.62%	113,499,479	10.43%	-0.18
OpenResty	91,405,835	8.37%	94,737,403	8.71%	0.34

[illegible]

Web 서비스 구조

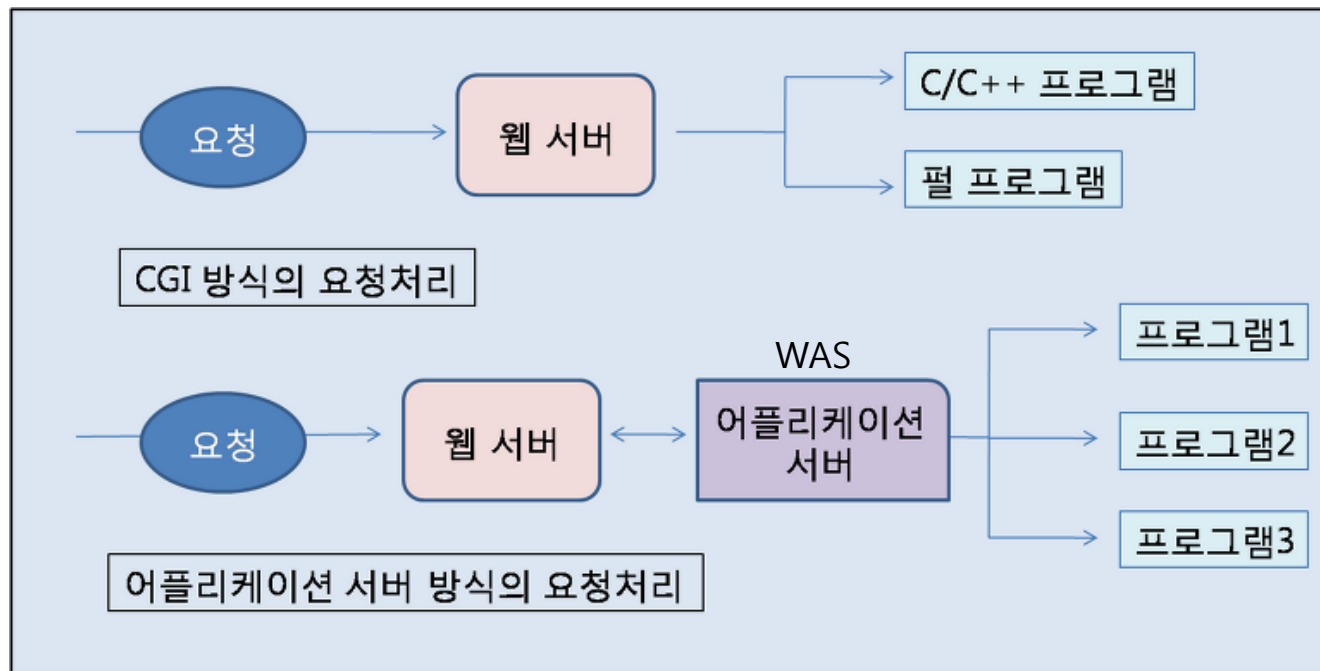


Web Server 기능 확장



이용자의 요청 처리

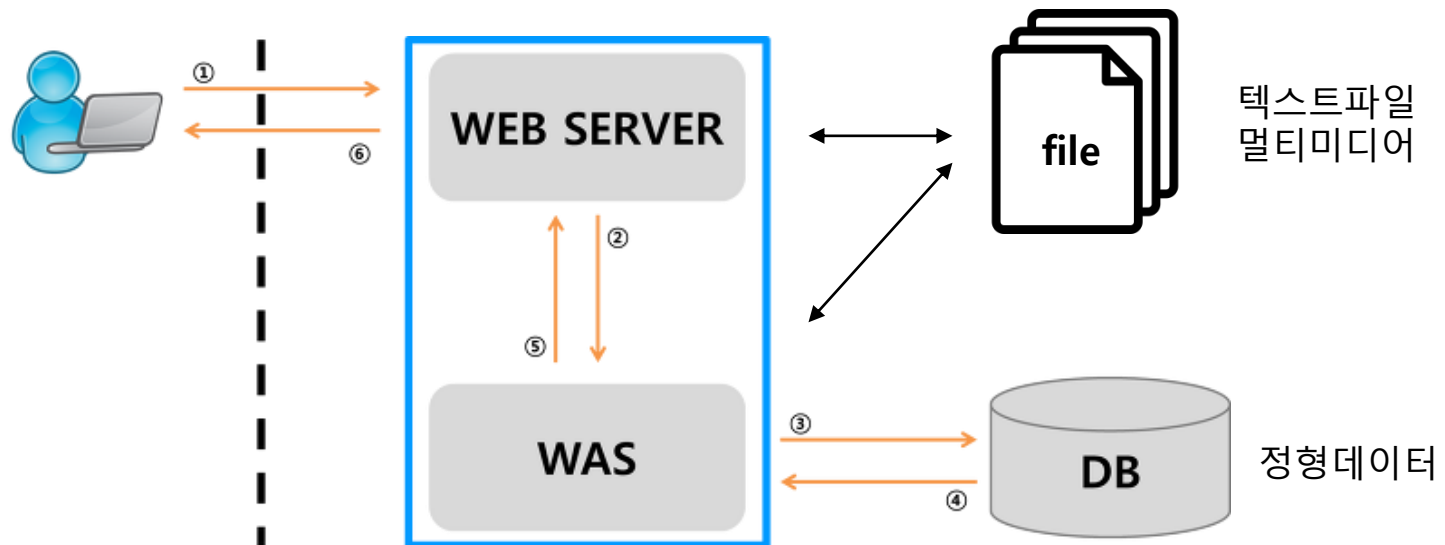
WAS (Web Application Server)



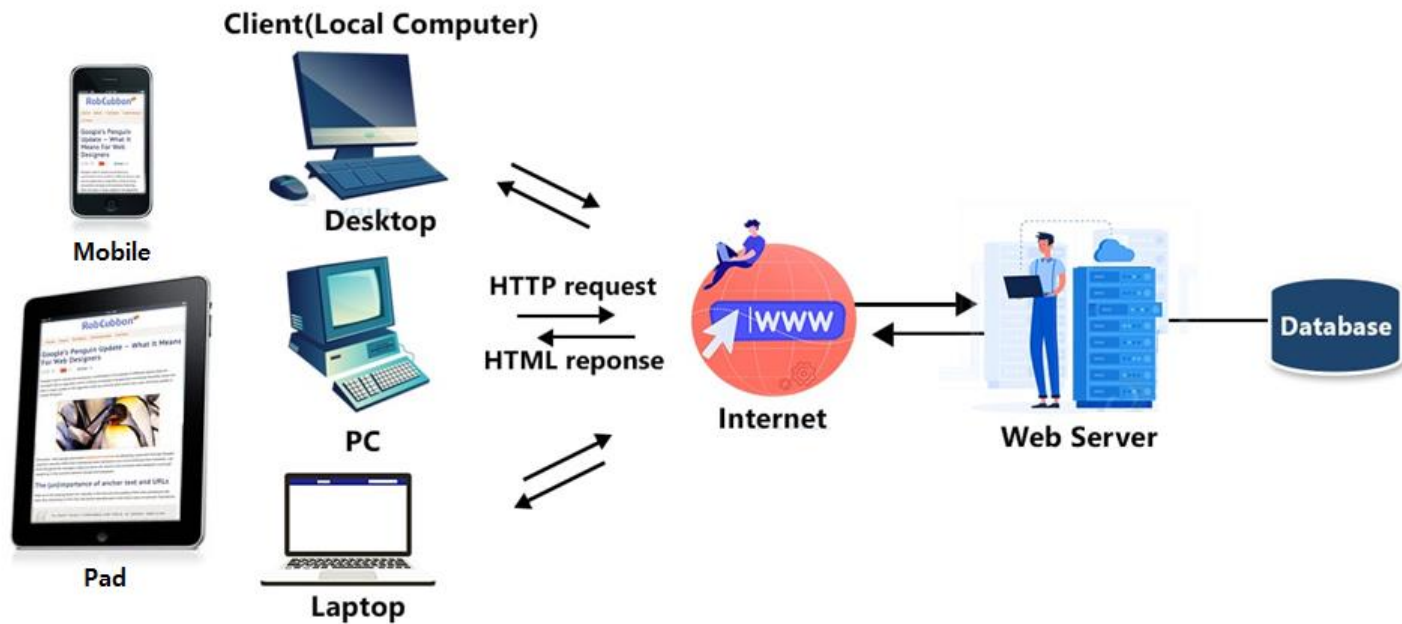
데이터베이스 (Database)

- 구조화된 정보 또는 데이터의 조직화된 모음
- 데이터의 처리 및 데이터 쿼리를 효율적으로 수행
- 데이터에 쉽게 액세스하고 관리, 수정, 업데이트, 제어 및 구성
- 데이터 작성 및 쿼리에 SQL(Structured Query Language) 사용
- 데이터베이스 관리 시스템(DBMS)에 의해 제어됨
- DBMS (Database Management System)
 - 데이터베이스 소프트웨어
 - Oracle, MSSQL, MySQL, MongoDB, Redis, H2DB

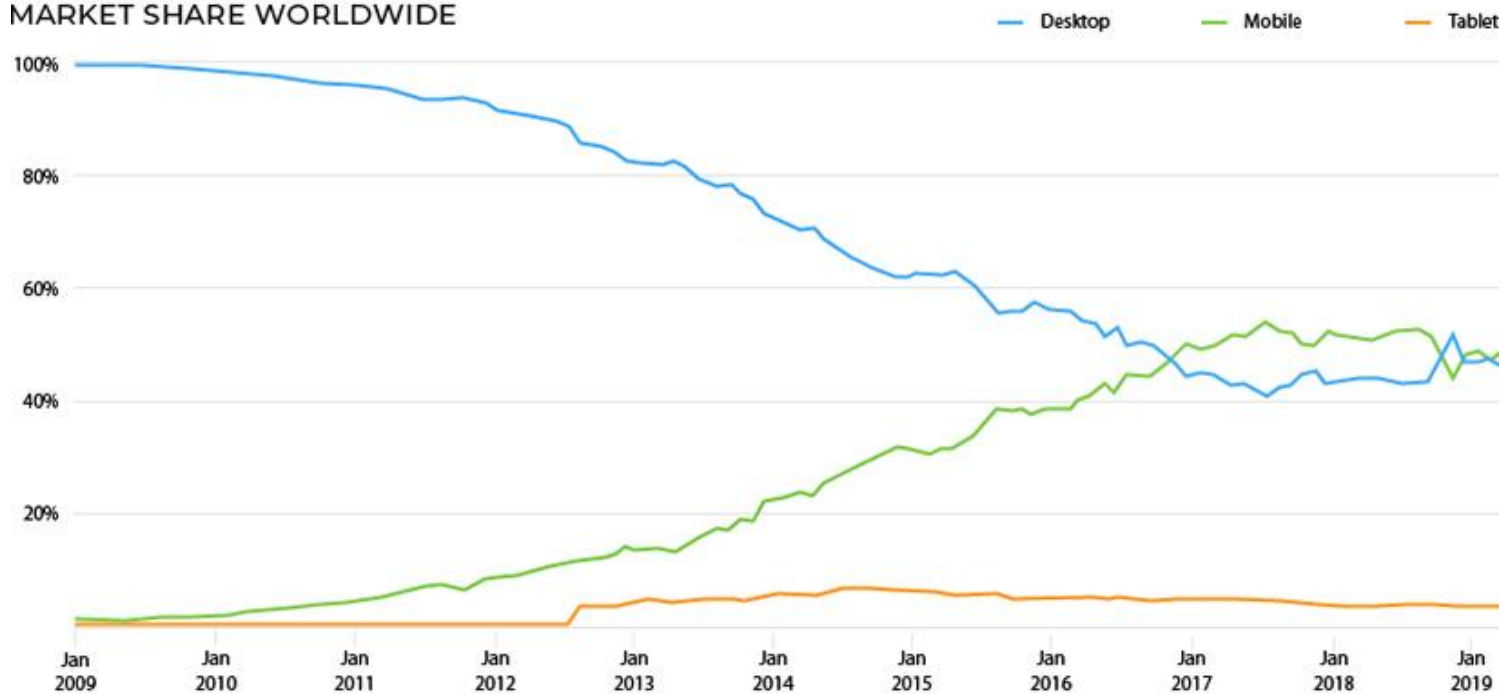
Web and Database



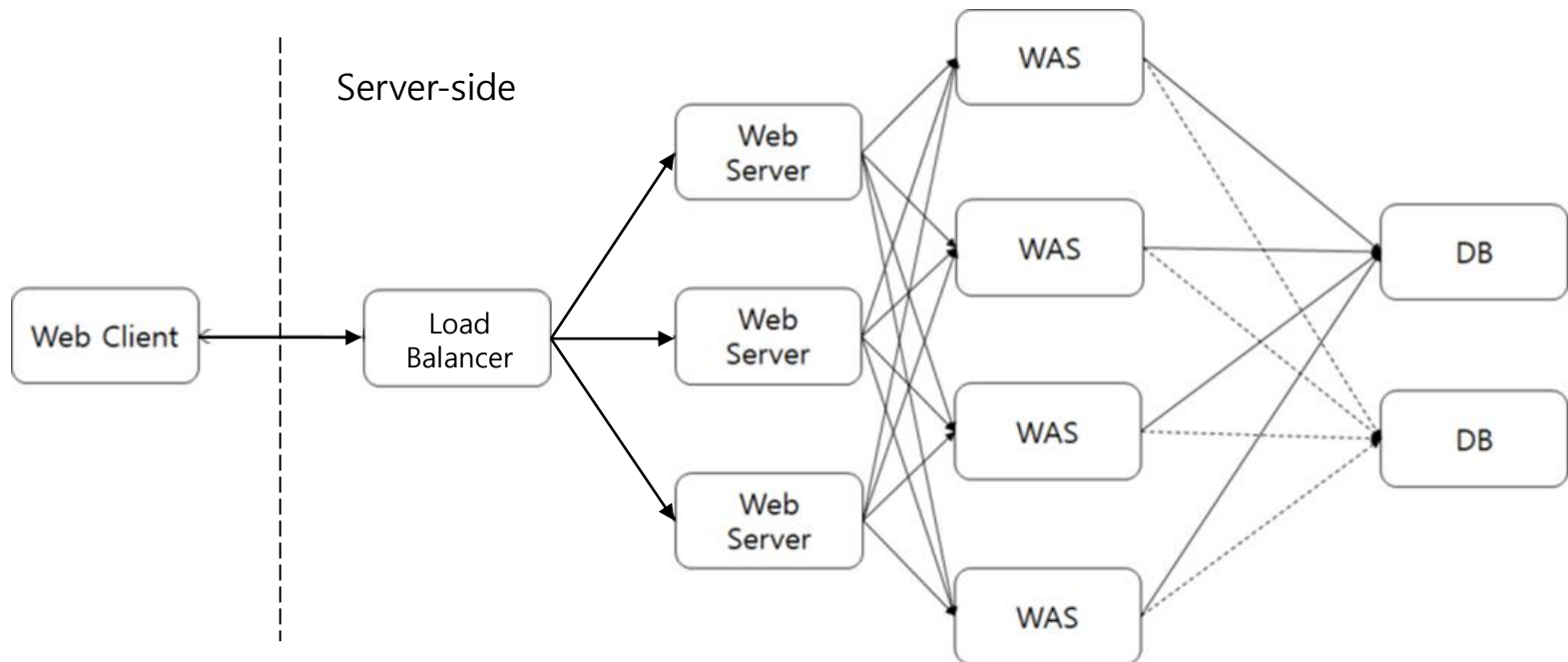
Web Client and Server



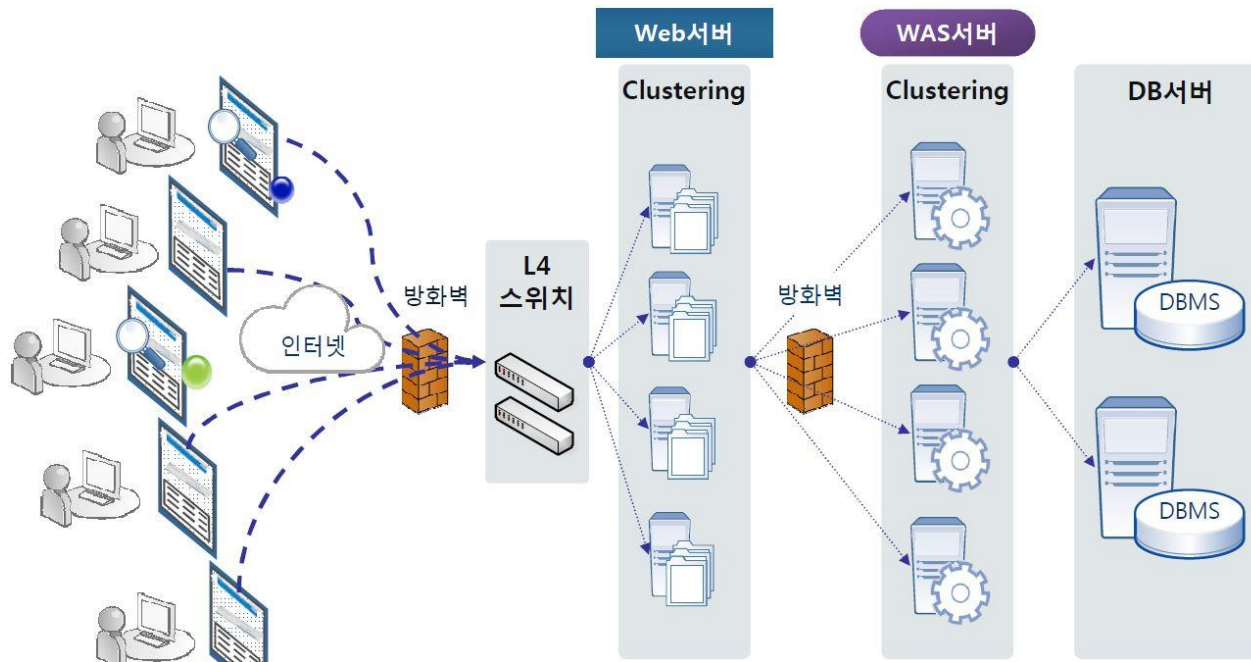
MARKET SHARE WORLDWIDE



Web Server Architecture



Secure Web System



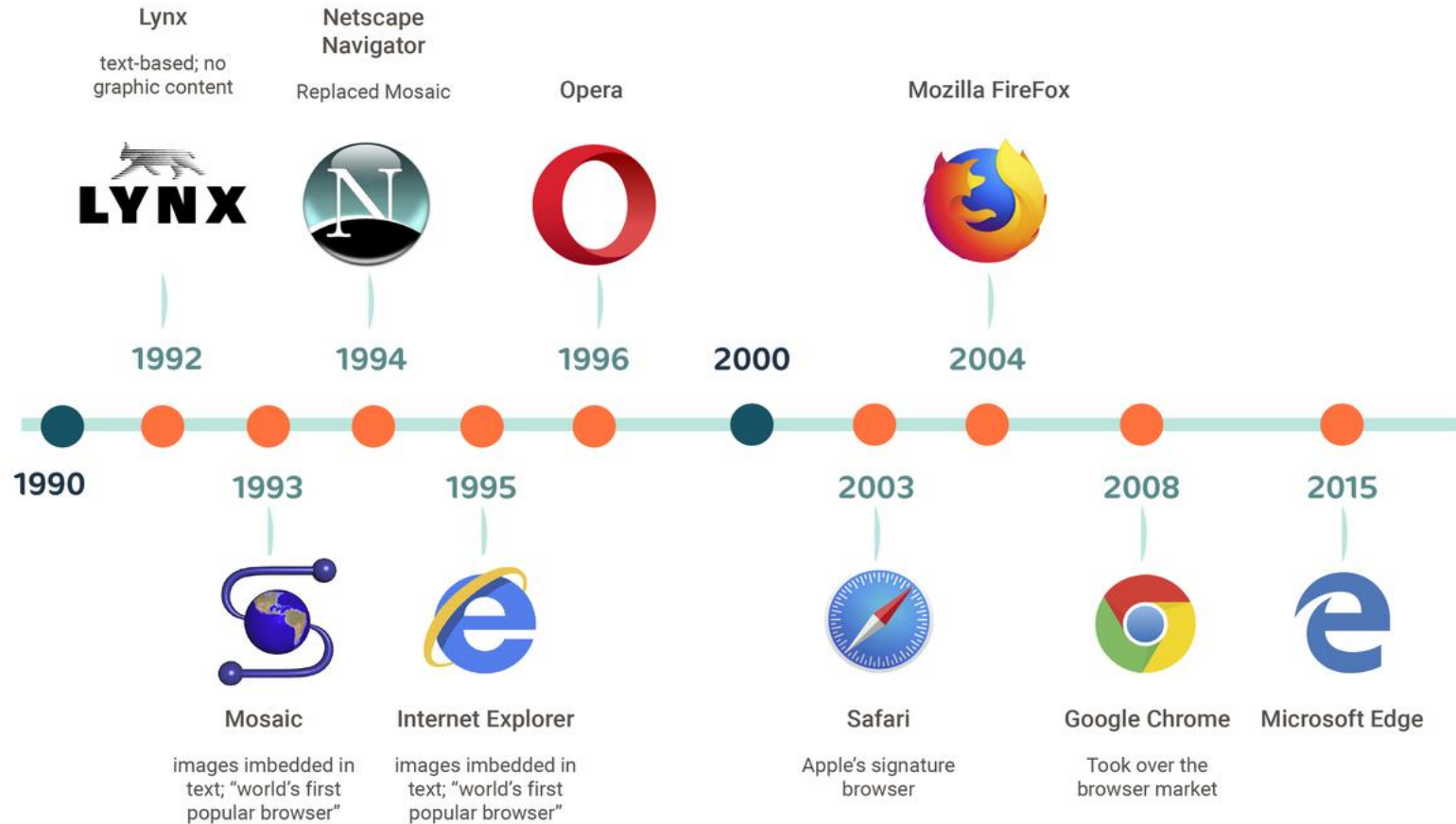
웹 구성 요소

- 클라이언트-서버 구조
- 브라우저 : 클라이언트 프로그램
- 웹 서버 : 정보를 제공
- HTTP 프로토콜 : 웹 서비스 요청과 응답
- URL : 요청 서비스 주소
 - HTTP 요청 메시지에 포함됨
- HTML 문서 : 브라우저 화면에 표시되도록 구성된 페이지
 - HTTP 응답 메시지 내에 포함됨

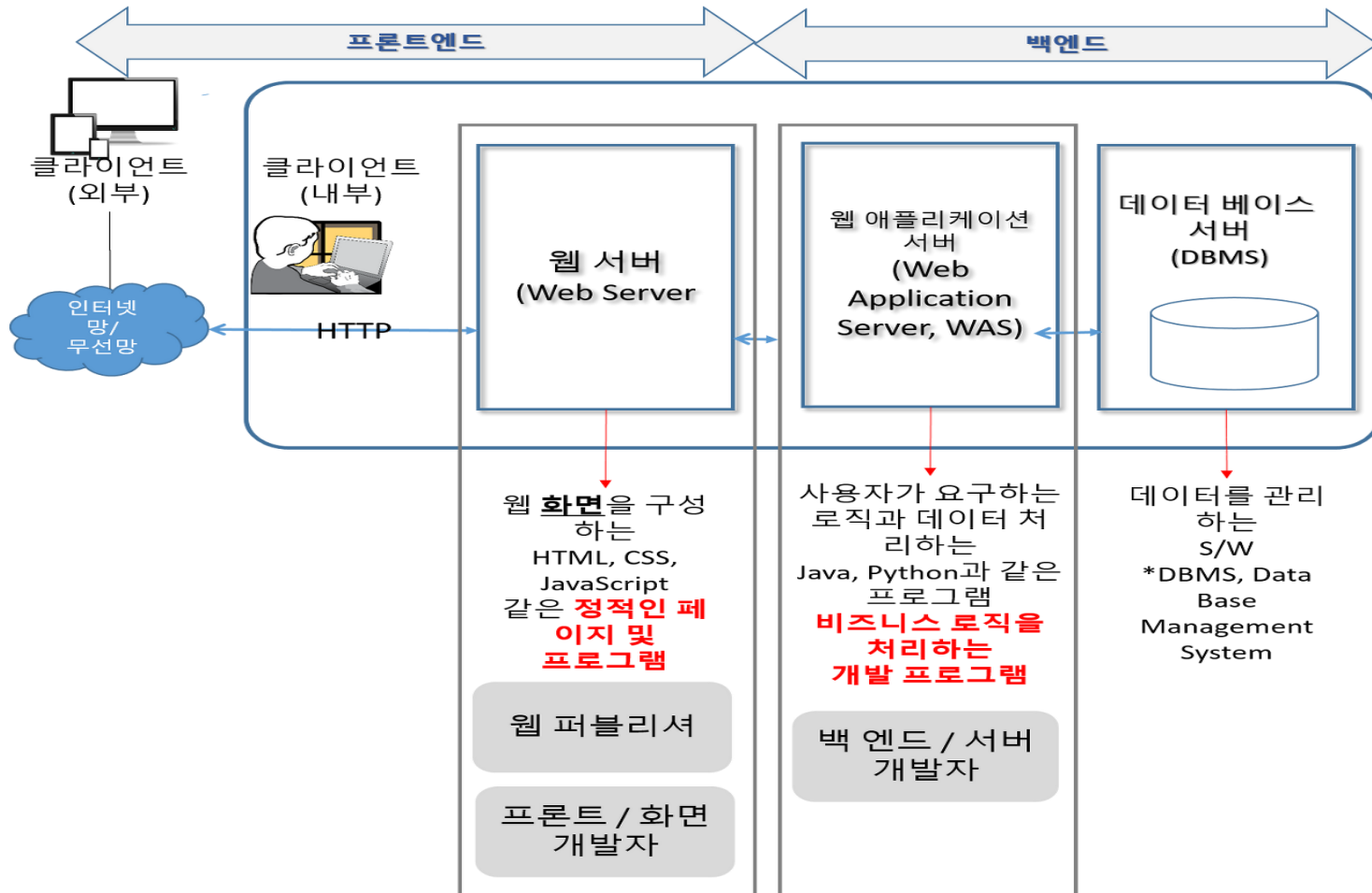
Web 클라이언트 (프로그램)

- 브라우저 (Browser) : web client program
- 모자이크 (Mosaic)
- Internet navigator
- Internet Communicator
- Internet Explorer
- 크롬 (Chrome)
- 사파리(Safari) (apple)
- 에지(Edge) Browser

브라우저의 발전



프론트엔드와 백엔드



웹의 3대 요소

- HTML (HyperText Markup Language)
 - 정보의 표현
 - 화면에 표시 방법 : 크게, 강조해서, 흘림체로, 새줄로
 - 문서의 구조 표현 : 머리말, 내용, 하단부
 - 정보의 의미 표현 : 글, 그림, 오디오, 동영상
 - 태그(tag) 이용하여 표현
- HTTP (HyperText Transfer Protocol)
 - HTML 형식의 정보를 교환하는 통신 규약 (protocol)
 - 인터넷의 프로토콜중의 하나 : TCP, IP, FTP, Telnet, ...
- URL (Uniform Resource Locator)
 - 인터넷의 자원을 표시하는 통일된 방식

HTML 문서 구성 요소 및 역할

- 구성요소
 - 정보 + 태그, CSS, JavaScript
 - `<button style="font:..." onclick="alert('hi')">선택</button>`
- 역할 분리
 - 태그 : 정보의 구조, 의미 표현
 - CSS : 스타일 정의
 - 초기에는 Javascript 없이 정적인 문서 (Static)
 - 나중에 Javascript가 추가됨 → Dynamic HTML
 - JavaScript : 다이나믹 & 인터랙티브 제어
 - 그래픽 제어, 이벤트 제어, 애니메이션, 서버와의 통신
 - JavaScript는 웹 문서의 기능 확장

URL (Uniform Resource Locator)

	host		path		fragment
	http://	www.google.com	:5883	/search/food.html	?topic=pizza #top
scheme			Port		query

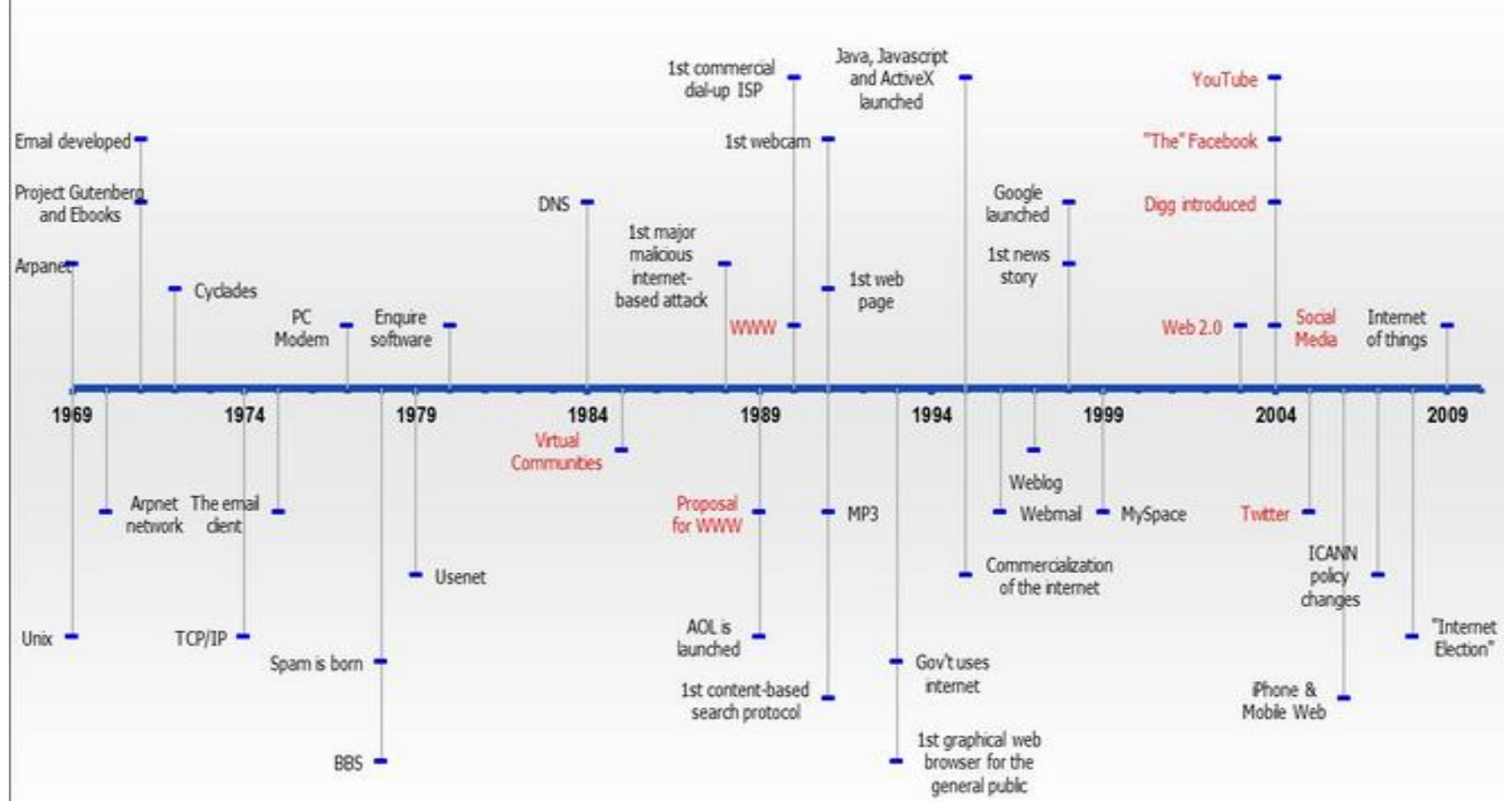
- Scheme : 접근방법. HTTP/FTP/e-mail/file
- Host : 서버 도메인 이름 혹은 IP 주소
- Port : 서비스 포트 번호. 웹 서버의 디폴트 포트는 80이고 생략 가능
- Path : 서버 내의 정보 위치 정보

Web 역사

- 1969 unix
- TCP/IP 프로토콜 탑재
- ARPANET : 4개 unix 연결
- 정부기관, 군 computer 네트워크
- 1983 Military Network 분리 -> Internet 탄생
- 1989 CERN Web (HTTP 0.9)
- 1999 HTML4
- 2003 Web 2.0
- 2010 HTML5
- HTTP 2.0
- 2019 Web 3.0

인터넷 개발 타임라인

Timeline of the development of the Internet



웹 : Internet 서비스

- 초기 인터넷 서비스
 - Telnet
 - Ftp
 - E-mail
 - Usenet : 뉴스그룹
- Web
 - 1889.3 팀 버너스 리
 - CERN 연구원으로. 정보 접근. 공유 서비스 개발 프로젝트 수행

제 1 장

자바스크립트 프로그래밍

자바스크립트

- 소스코드를 한 줄씩 읽어서 실행하는 스크립트 언어의 한 종류
 - 컴파일 언어 : 소스코드를 컴파일 과정을 거쳐 기계어로 바꾼 다음 실행하는 언어. C/C++
 - 스크립트 언어 : JavaScript, Python(파이썬)
- 자바스크립트는 웹 브라우저에서 기본적으로 사용하는 프로그래밍 언어임

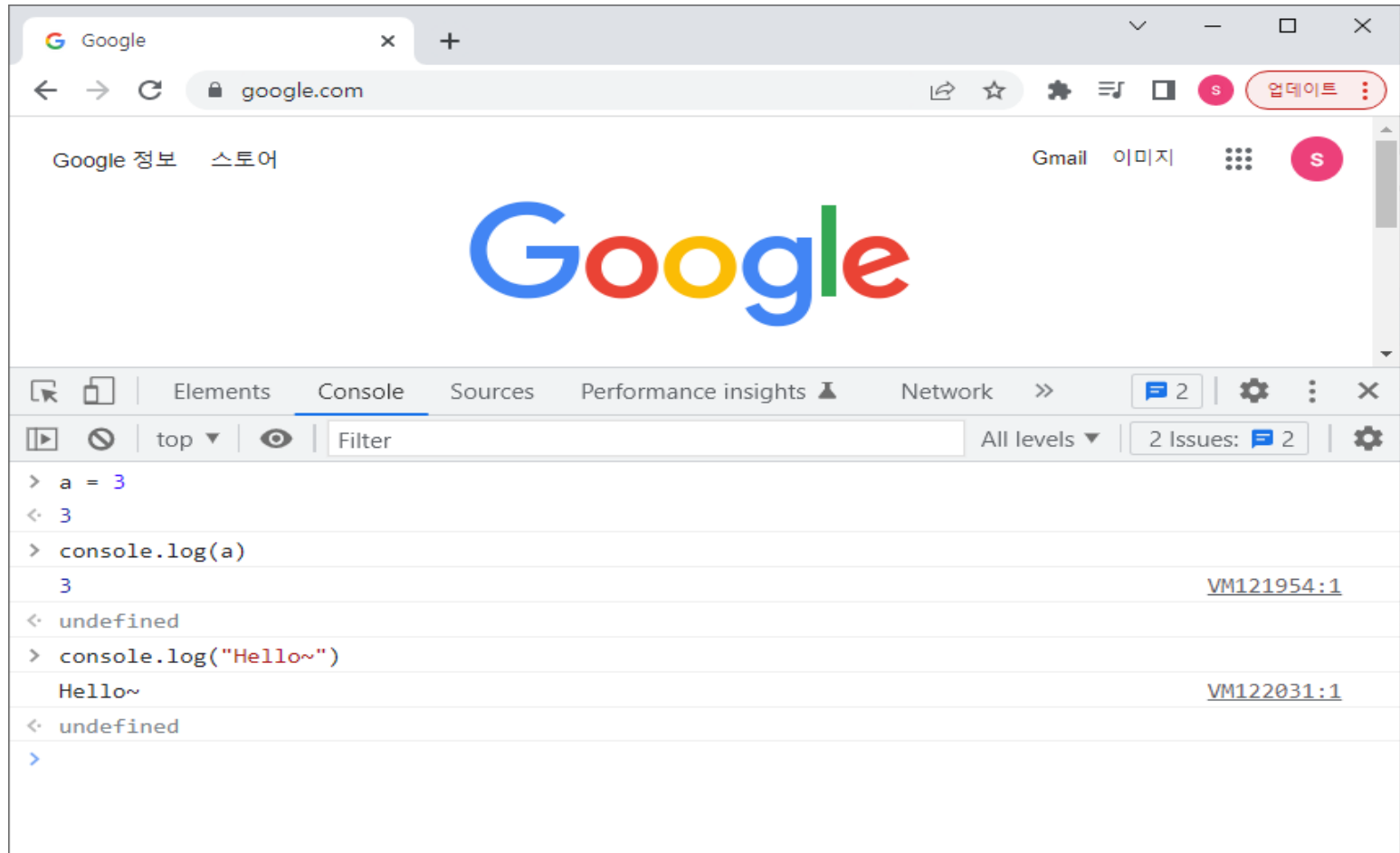
프로그래밍 언어 사용

- 자바스크립트 : 웹 서버/브라우저
- 파이썬 : 인공지능, 과학용
- C : 범용 개발 (운영체제/임베디드)
- C++/C# : 윈도우/웹 응용 프로그램
- 자바/코틀린 : 안드로이드 앱
- 스위프트 : 애플 앱
- PHP/JSP/ASP : 웹서버 애플리케이션

브라우저 콘솔

- 크롬 (chrome)
 - 우측상단의 "Chrome 맞춤설정및제어 → 도구 더보기 → 개발자도구" 선택 (혹은 F12키 누름)
 - "콘솔" 탭 선택
 - `console.log("Hello~")`
 - Control키와 +/-키로 글씨 크게/작게
 - Control키 와 L키로 화면 지우기(화면초기화)
 - Up키와 Down키로 전/후 명령 재사용
 - Shift+Enter : 줄바꿈 (여러 줄 입력 시)

크롬 도구 콘솔 화면



프로그래밍 요소

- 자료형 : 숫자, 문자, 참/거짓, 객체
- 변수 : 자료를 담는 것
- 연산자 : +, -, *, /, AND, OR, NOT
- 제어문, 반복문 : if, for, while
- 함수 : 정해진 작업을 하는 코드 집합
- 주석 : 설명문
- 키워드 : 미리 정해진 특수 이름

변수 (variable)

- 변수란
 - 값을 저장하는 저장소
 - 값이 변할 수 있음 \leftrightarrow 상수 (변하지 않는 값)
- 변수 이름
 - 영문자로 시작 (대소문자는 다른 문자 임)
 - 숫자, _(언더바) 가능
 - a, A, abc, myname, myName, my_name
 - 잘못 사용하는 경우 : 123, ab cd, 33abc, ab+cd, if, for

변수 선언 및 할당

- 선언
 - `var abc;`
 - `name;`
- 할당
 - `abc = 1;`
 - `name = "khlee"`
- 선언 및 할당
 - `const PI = 3.14;`
- 선언 없이도 바로 사용(할당) 가능
 - `myName = "nobody";`
 - `age = 18`

변수 값 할당

- $abc = 1$
 \leftarrow abc 라는 변수에 값 1을 할당 (넣음)
 (= 왼쪽의 값을 = 오른쪽으로 할당)
- $abc = 3$
 \leftarrow 변수 abc 의 값이 3으로 바뀜
- $def = abc$
 \leftarrow def 변수에 abc 변수의 값인 3이 들어감
- $abc = 5$
 \rightarrow abc 의 값이 바뀜
 \rightarrow def 의 값은 바뀌는 것이 아님 (전에 넣은 3으로 유지됨)

데이터의 종류

- 123 : 숫자
- 123.45 : 숫자
- "abc" : 문자열
- '123' : 문자열
- "abc123def" : 문자열
- true
- false
- abc는 무엇일까
→ 이름 (변수 또는 함수 이름)

숫자 값

- `a = 1;` // 변수
- `a = 1` // ;생략가능
- `a = 3; b = 5` // 한 줄에 여러 명령
- `c = a + b;`
- `console.log(c);` // 8
- `console.log("c");` // c
- `d = 3 + 5;`
- `console.log(d);` // 8
- `console.log(3 + 5);` // 8

변수의 값 교환

- $a = 1$
- $b = 2$
- 잘못된 교환

$a = b$

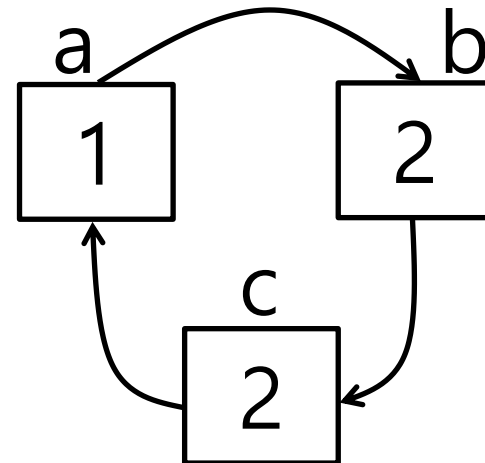
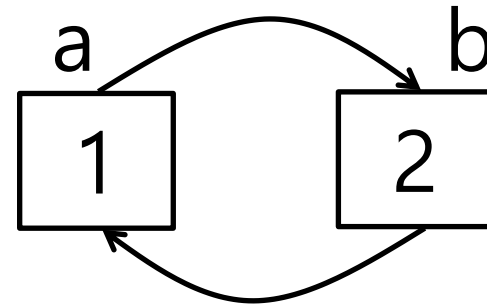
$b = a$

- 제대로 된 교환

$c = a$

$a = b$

$b = c$

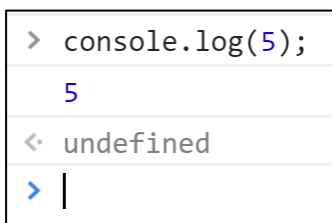


문자열

- `a = "1";` // 문자열 "1"
- `b = "5"` // 문자열 "5"
- `c = a + b;` // 문자열 더하기
- `console.log(c);` // "15"
- `d = "abc" + "de";` // 문자열 더하기
- `console.log(d);` // "abcde"

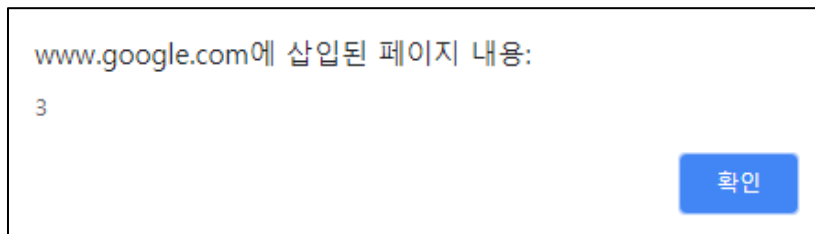
출력함수

- `console.log(5);`
 - 콘솔의 명령 바로 밑줄에 출력



```
> console.log(5);  
5  
< undefined  
> |
```

- `alert(3);`
 - 브라우저 경고창으로 출력



www.google.com에 삽입된 페이지 내용:

3

확인

alert() 함수

- `alert("Hello~")`
- `windows.alert('Hello~')` // same
- `alert("Hello~\nHow are you?")`
- `alert(navigator.appName)`



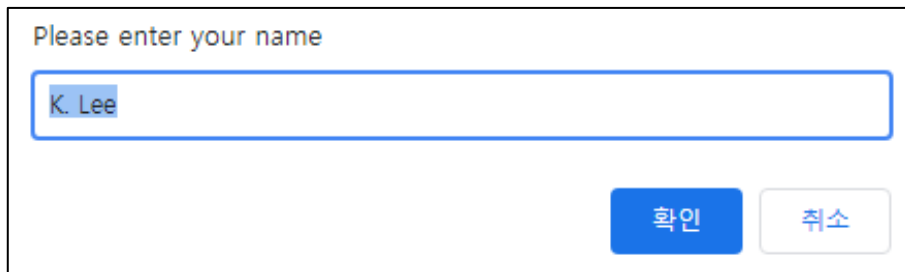
출력함수

- document.write() 메소드
 - 웹 페이지로 출력
- document.write("Hello JavaScript")
- document.write(1234567890)
- document.write("
" + "abcdefghijklmn")

```
Hello JavaScript1234567890  
abcdefghijklmn
```

prompt()

- `var person = prompt("Please enter your name", "K. Lee");`
- `alert("Hello~ " + person);`



A screenshot of a JavaScript prompt dialog box. The title bar is not visible. The main area contains the text "Please enter your name" in a light blue font. Below this is a text input field with a blue border and a light blue background, containing the text "K. Lee". At the bottom right of the dialog are two buttons: a blue button with the text "확인" (Confirm) and a light blue button with the text "취소" (Cancel).



A screenshot of a JavaScript alert dialog box. The main area contains the text "Hello~ K. Lee" in a light blue font. At the bottom right of the dialog is a single blue button with the text "확인" (Confirm).

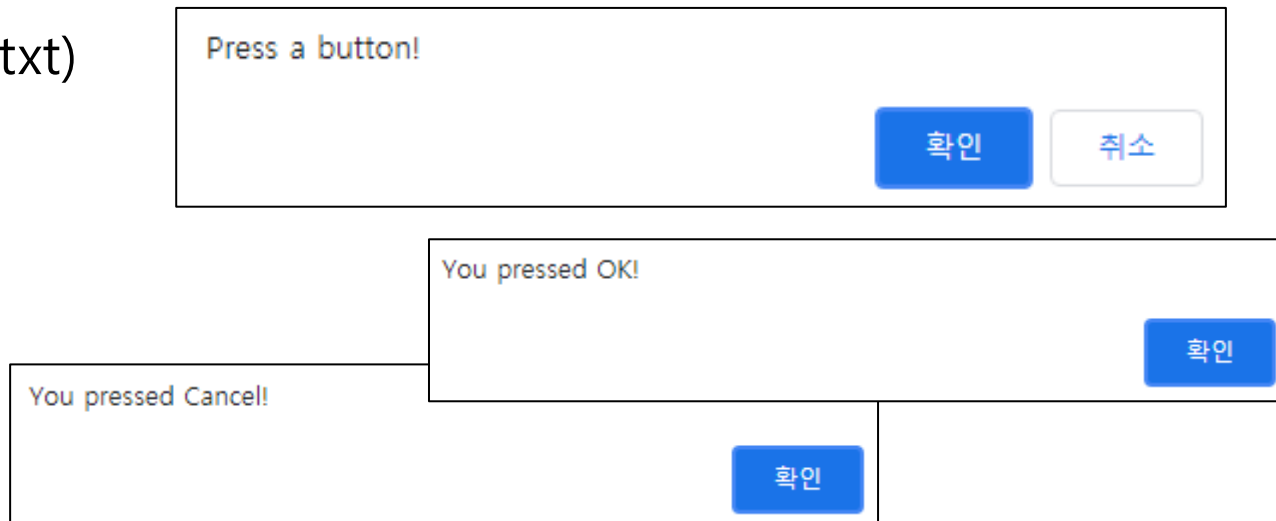
prompt() 함수 활용

- data = prompt()
 ← 프롬프트창에서 1 5 7 입력
- alert(data) // 1 5 7
- data.split(' ') // ["1","5","7"]
- data.map(Number) // [1,5,7]
- alert(data[0]) // 1
- alert(data[1]) // 5
- alert(data.length) // 3
- alert(data[data.length-1]) // 7

confirm() 함수

- ```
var txt;
var r = confirm("Press a button!");
if (r == true) {
 txt = "You pressed OK!";
} else {
 txt = "You pressed Cancel!";
}

alert(txt)
```



# 제 2 장

## 웹사이트 만들기

# HTML 문서 구조

```
<HTML>
<HEAD>
.....
</HEAD>
<BODY>
.....
.....
</BODY>
</HTML>
```

## Header

(브라우저에게 필요 정보 제공,  
스타일, 자바스크립트,  
화면에 표시되지 않음)

## Body

(정보, 태그,  
브라우저 화면에 표시되는 부분)

# 태그 (tag)

- 정보를 감싸고 있음 : `<H1>오늘의 날씨</H1>`
- 정보를 설명해 줌 : H1 → 큰 제목
- 보통 쌍으로 구성
  - Opening tag : 시작태그
  - Closing tag : 종료태그
  - 단독으로 존재하는 경우도 있음 : `<br>`, `<hr>`
    - `<br>` → `<br />`로 표시하는게 표준적 방법
- 대소문자 구분 없음
  - 한가지로 통일을 하는 것이 좋음 (소문자가 보기 좋음)

# HTML 문서 구성 요소 및 역할

- 구성요소
  - 정보 + 태그, CSS, JavaScript
  - `<button style="font:..." onclick="alert()">선택</button>`
- 역할 분리
  - 태그 : 정보의 구조, 의미 표현
  - CSS : 스타일 정의
  - 초기에는 Javascript 없이 정적인 문서 (Static)
  - 나중에 Javascript가 추가됨 → Dynamic HTML
  - JavaScript : 다이나믹 & 인터랙티브 제어
    - 그래픽 제어, 이벤트 제어, 애니메이션, 서버와의 통신
  - JavaScript는 웹 문서의 기능 보완 및 확장

# GitHub 홈페이지 구축

- Github.com에 가입 : id
- 새 Repository 만들기
  - Repository → New → home
  - 주소가 <https://id.github.io/home> 이 됨
- Repository 설정
  - Settings → Pages
  - Source : Branch:main
  - / (root) → save
- 홈페이지 주소
  - <https://id.github.io/home/>

# GitHub Pages 제한 사항

- GitHub Pages 소스 저장소의 권장 제한은 1GB
- 게시 된 GitHub 페이지 사이트는 1GB를 초과 할 수 없음
- GitHub 페이지 사이트의 대역폭 제한은 한 달에 100GB
- GitHub 페이지 사이트의 builds 제한은 시간당 10회
- 단점
  - 내용 수정 후 반영하는데 시간이 걸림
  - 서버에서 데이터 처리 못함 (로그인, 게시판, 데이터 저장, ...)

# 페이지 추가

- Code → main → Add File
  - Create New File : 웹 화면에서 추가
  - Upload File : 파일 업로드
- Create New File
  - abc.html : 루트 디렉토리에 만듦
  - abc/123.html : 특정 디렉토리 하부에 만듦
  - 입력/수정 후에 [commit New File] 버튼 클릭
- Upload File
  - 드래그 혹은 파일 선택
  - [Commit Changes] 버튼 클릭



# 페이지 수정/삭제

- 수정
  - 페이지 웹 화면에서 수정
    - 페이지 보기 → 연필 모양의 수정 아이콘
  - 같은 이름의 파일을 업로드하여 덮어 씌움
    - 복수개의 파일 선택 가능
  - Commit Changes하여 수정 완료
- 삭제
  - 페이지 보기 → 쓰레기통 모양의 아이콘
  - Commit Changes하여 삭제 완료

# 제 3 장

## 자바스크립트 기초문법

# 산술연산

- $a = 10;$
- $b = 3;$
- $c = a + b;$
- $d = a - b;$
- $e = a * b;$       // 곱하기
- $f = a / b;$       // 나누기, 3.33...
- $g = a \% b;$       // 몫, 1

# 사칙연산

- $a = 7;$
- $b = 2;$
- $cc = a \% b;$  // 1, 나머지
- $dd = ((2 + 3) * 5) / 2$  // 괄호먼저
- $a = a + 1; a += 1;$
- $a = a - 1; a -= 1;$
- $a++; ++a;$  // 1 증가
- $a--; --a;$  // 1 감소

# 데이터 입력 받기

- `a = prompt("당신의 이름은 무엇입니까? ")`
- `alert(a + "씨 반갑습니다.")`
  
- `b = prompt("당신의 나이는? ")`
- `alert(a + "씨의 나이는 " + b + " 입니다.")`

# 숫자데이터 입력 받기

- `a = prompt("숫자1 입력: ")` `// 3`
- `num1 = parseInt(a)`
- `b = prompt("숫자2 입력: ")` `// 5`
- `num2 = parseInt(b)`
- `// 문자열과 숫자 차이`
- `alert(a + b)` `// 35`
- `alert(num1 + num2)` `// 8`
- `// 한 줄로 다 처리`
- `num1 = parseInt(prompt("숫자1 입력: "))`
- `num2 = parseInt(prompt("숫자2 입력: "))`

# if 조건문

- Shift + Enter : 여러 줄 입력 가능

```
if (불 표현식) {
 참이면 실행하는 문장
}
```

```
if (5 > 3) {
 console.log("5가 3보다 크다");
}
```

# Boolean 데이터

- `a = true`
- `b = false`
- `c = 5 > 3`                      `// c는 true`
- `d = 3 > 5`                      `// d는 false`

```
if (c) {
 console.log("c는 참입니다."); → 실행됨
}
else {
 console.log("c는 거짓입니다."); → 실행되지 않음
}
```



# 비교 연산

- `c = 5 > 3`                      `// c는 true`
- `d = 3 > 5`                      `// d는 false`

```
if (d) {
 console.log("d는 참입니다."); → 실행되지 않음
}
else {
 console.log("d는 거짓입니다."); → 실행됨
}
```

# 비교 연산자

- == : 같다
- != : 같지 않다
- > : 크다
- < : 작다
- >= : 크거나 같다
- <= : 작거나 같다
- === : 데이터 형과 값 모두 일치한다

# 논리 연산자

- && : and
- || : or
- ! : not
- 연산자 우선순위  
괄호() > !(not) > &&(and) > ||(or)

# 논리연산 예제

- `const a = 3;`
- `const b = -2;`
- `console.log(a > 0 && b > 0);`  
`// expected output: false`
- `console.log(a > 0 || b > 0);`  
`// expected output: true`
- `console.log(!(a > 0 || b > 0));`  
`// expected output: false`

# if-else

```
if (불 표현식) {
 참이면 실행하는
 문장
}
else {
 거짓이면 실행
 하는 문장
}
```

```
if (a > b) {
 alert("a is larger than
b")
}
else {
 alert("b is larger than
a")
}
```

# if-else if-else

```
if (불 표현식1) {
 표현식1이 참이면 실행
}
else if (불 표현식2) {
 표현식1이 거짓, 표현식 2가 참이면 실행
}
else {
 표현식1이 거짓, 표현식 2가 거짓이면 실행
}
```

# 중복 if 문

```
if (표현식1) {
 if (표현식2) { 표현식1 참, 표현식2 참이면 실행 }
 else { 표현식1 참, 표현식2 참이면 실행 }
}
else {
 if (표현식3) { 표현식1이 거짓, 표현식3 참이면 실행 }
 else { 표현식1 거짓, 표현식3 거짓이면 실행 }
}
```

# 값 비교 및 판별

- if 조건문 형식
- 판별문 : 참과 거짓 판별
- if 문장 활용
- 큰 값 찾기
- 작은 값 찾기
- 홀짝수 판별하기



# 큰 수 찾기

```
var a = parseInt(prompt("정수", 0));
var b = parseInt(prompt("정수", 0));
if (a > b)
 document.write("큰 수: ", a);
else
 document.write("큰 수: ", b);
```

# 제 4 장

## 프로그램 제어

# for 반복문

- 정해진 횟수만큼 루프 실행

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

```
for (i=1 ; i<=10 ; i++) {
 console.log(i);
}
```

# 합 구하기

```
sum = 0;
for (i=1 ; i<=10 ; i++) {
 console.log(i);
 sum = sum + i;
}
console.log(sum);
```

# for 반목문 출력

```
<script>
 // 변수 선언
 output = "";

 // 반복문
 for (i=0 ; i<10 ; i++) {
 for (j=0 ; j<i ; j++) {
 output += '*';
 }
 output += '\n';
 }

 // 출력
 alert(output);
</script>
```



# 구구단 출력

```
var n=0, sum=0;
for(var i=2; i<=9; i++){
 for(var j=1; j<=9; j++)
 document.write(i,"*",j,"=",i*j," ");
 document.write("
");
}
```

# while 문

- 판단식이 참일 동안 계속 반복

```
while (판단 식 혹은 값) {
 판단 식이나 값이 참일 경우 실행

}
```

괄호 안 문장을 실행하고, 다시 판단 식이나 값을 따져, 계속 여부를 판단

# while 문

```
var n = 0;
var x = 0;
while (n < 3) {
 n++;
 x += n;
}
console.log(x)
// 6 출력됨
```

0 < 3

- 첫번째 반복; n=1 과 x=1이 됨

1 < 3

- 두번째 반복; n=2 과 x=3 이 됨

2 < 3

- 세번째 반복; n=3 과 x=6 이 됨

3 < 3

- 거짓이므로 종료



# 제 5 장

## 배열과 함수

# 배열

- 많은 수의 데이터 다루기
- $a = [1, 3, 5, 7, 9];$
- $a$  : 배열의 이름
- 배열의 요소 접근
  - 인덱스 : 0 부터 시작
  - 첫번째 요소는  $a[0]$
  - 두번째 요소는  $a[1]$
  - 마지막 요소는  $a[4]$
  - $a[5]$ 는 없음 (중요 !!!)

# 배열 데이터 접근

- `a = [1, 3, 5, 7, 9];`
- 배열 데이터 입력
  - `a[1] = 2;`
  - $\rightarrow$  `a` 는 `[2, 3, 5, 7, 9];`
- 배열 데이터 출력
  - `console.log(a[2])`    `// 3`
- 배열 데이터 접근
  - `c = a[0] + a[1];`    `// c는 5`
  - `d = a[3] - 1`        `// d는 6`

# 배열 길이

- 배열이름.length → 배열의 길이 (크기)
- a = [1, 3, 5, 7, 9, 11];
- console.log(a.length); // 6
- console.log(a[length - 1]); // 11
- console.log(a[length]); // undefined

# 함수

- 함수는 코드 블록

```
function printhello() {
 document.write("Hello");
}
printhello();
printhello();
printhello();
```

# 함수 파라미터 전달

```
function printValue(a) {
 document.write(a);
}
```

```
a = 3
```

```
printValue(a);
```

```
a = 5
```

```
printhello(a);
```

# 함수 리턴값

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

```
a = add(2, 3) // 5
```

```
b = 3;
```

```
c = add(a, c) // 8
```

# 1부터 합 구하기

```
function sum(n) {
 var sum = 0;
 for (i=1 ; i<=n ; i++) {
 sum += i;
 }
 return sum;
}
result = sum(10); // 55
```



# 최대값 찾기

```
function max(arr) {
 len = arr.length;
 max = 0;
 for (i=0 ; i<len-1 ; i++) {
 if (max < arr[i])
 max = arr[i]
 }
 alert(max + ":" + max);
}
max([6,1,2,3,4,5]);
```

# 최소값 찾기

```
function min(arr) {
 len = arr.length;
 min = 10000;
 for (i=0 ; i<=len ; i++) {
 if (min > arr[i])
 min = arr[i];
 }
 alert(min + ":" + min);
}
min([6,1,2,3,4,5]);
```

# 구구단 출력

```
var n=0, sum=0;
for(var i=2; i<=9; i++){
 for(var j=1; j<=9; j++)
 document.write(i,"*",j,"=",i*j," ");
 document.write("
");
}
```

# 제 5 장

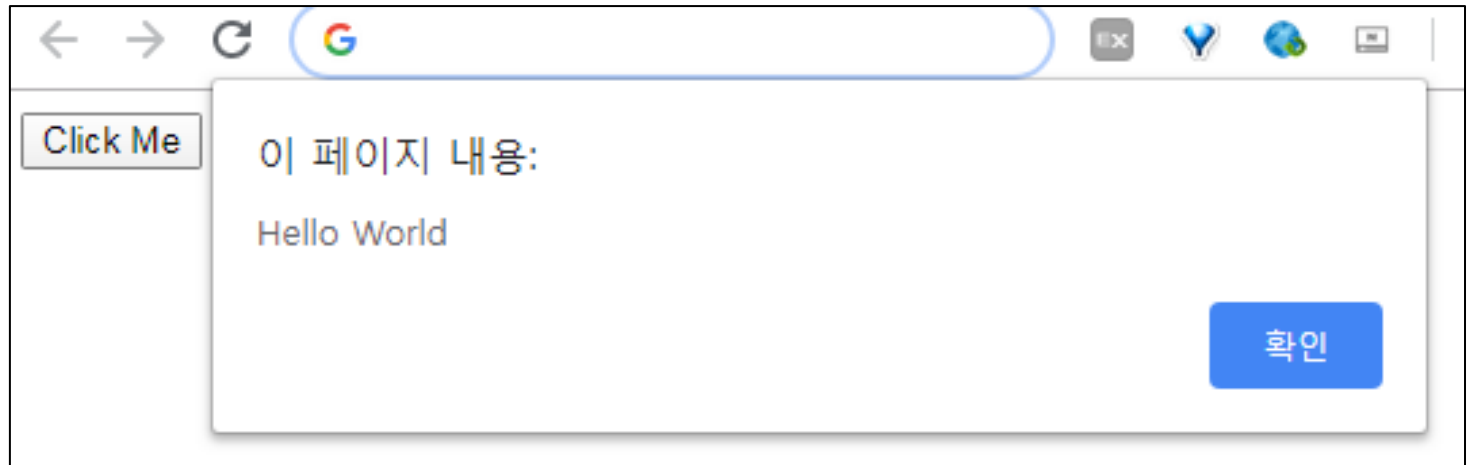
## 게임 프로그래밍

# 이벤트

- 동적 프로그래밍의 기초
- 이벤트 종류
  - 마우스 입력
  - 키보드 입력
  - 터치
  - 시간 경과
- 이벤트 핸들러 : 이벤트 처리 함수

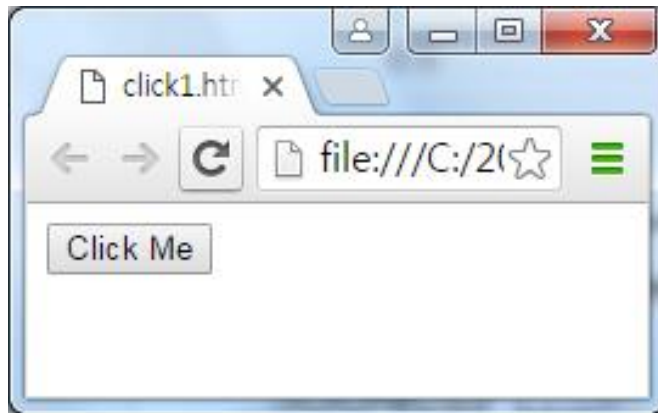
# Onclick 이벤트

```
<html> <head> <script>
function callme() {
 alert("Hello World");
}
</script> </head> <body>
<button onclick="callme()">Click Me</button>
</body>
</html>
```



# click 이벤트

```
<html> <head> <script>
function myFunction() {
 document.getElementById("demo").innerHTML = "Hello World"; }
</script> </head> <body>
<button onclick="myFunction()">Click Me</button>
<p id="demo"> </p>
</body>
</html>
```



# 타이머 이벤트 함수

- `setTimeout(func, milliseconds)`
  - 일정 시간 지난 후 한번 수행
  - `func` : 수행할 함수
- `setInterval(func, milliseconds)`
  - 일정시간마다 주기적으로 계속 수행
  - `clearInterval` 함수로 중지 가능



# setTimeout 예제

```
<body>
```

```
<p>Click "Try it". Wait 3 seconds. The page will alert "Hello".</p>
```

```
<p>Click "Stop" to prevent the first function to execute.</p>
```

```
<p>(You must click "Stop" before the 3 seconds are up.)</p>
```

```
<button onclick=
```

```
 "myVar = setTimeout(myFunction, 3000)">Try it</button>
```

```
<button onclick="clearTimeout(myVar)">Stop it</button>
```

```
<script>
```

```
function myFunction() {
```

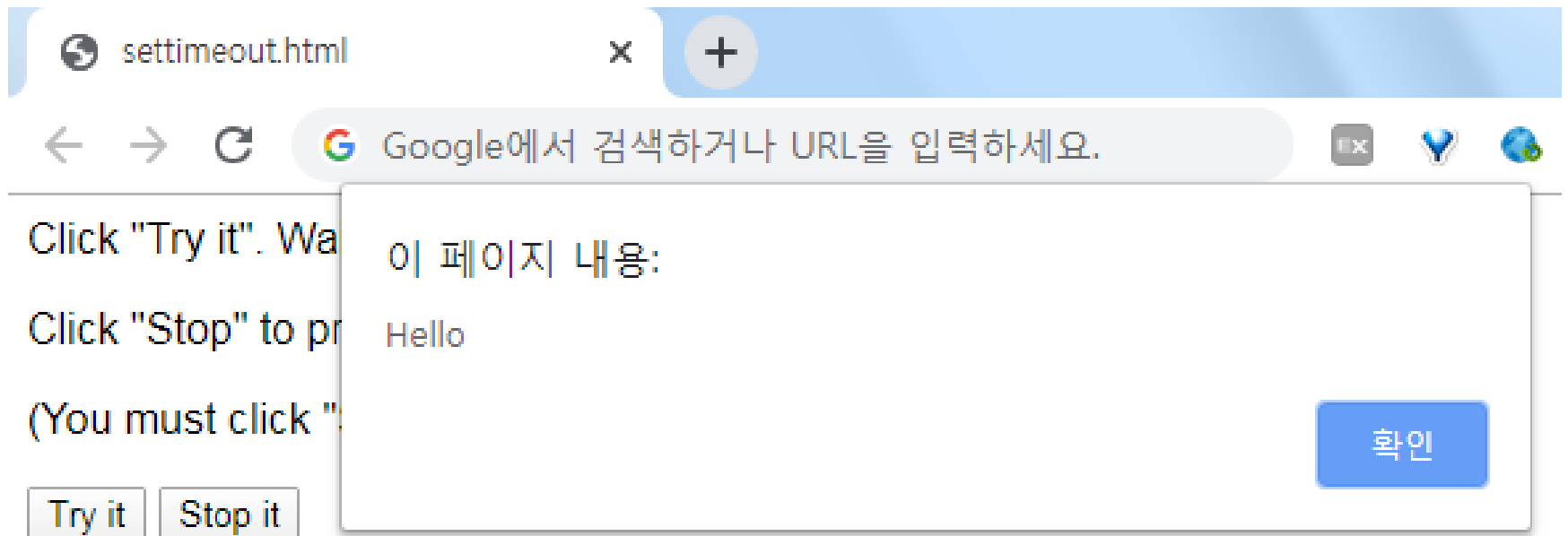
```
 alert("Hello");
```

```
}
```

```
</script>
```

```
</body>
```

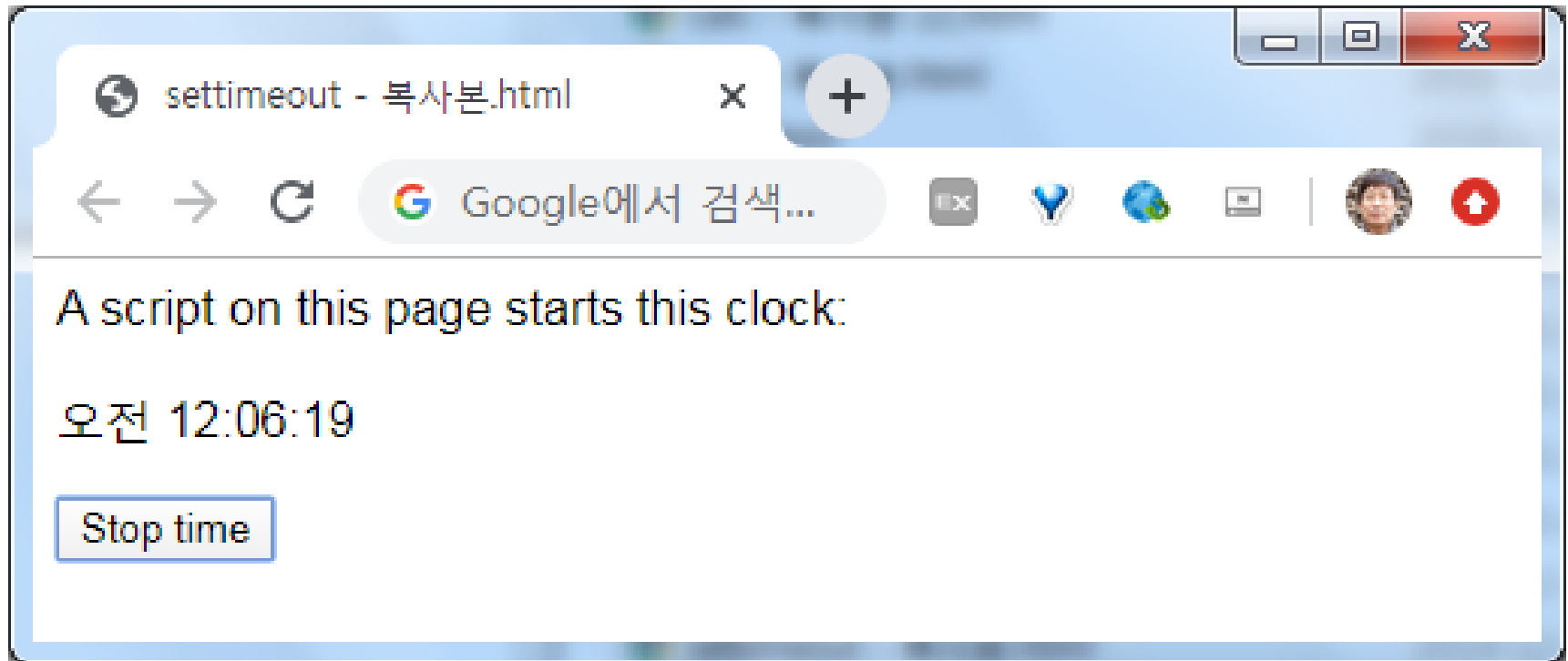
# setTimeout 예제결과



# setInterval 예제

```
<body>
<p>A script on this page starts this clock:</p>
<p id="demo"></p>
<button onclick="clearInterval(myVar)">Stop time</button>
<script>
var myVar = setInterval(myTimer ,1000);
function myTimer() {
 var d = new Date();
 document.getElementById("demo").
 innerHTML = d.toLocaleTimeString();
}
</script>
</body>
```

# setInterval 예제 결과

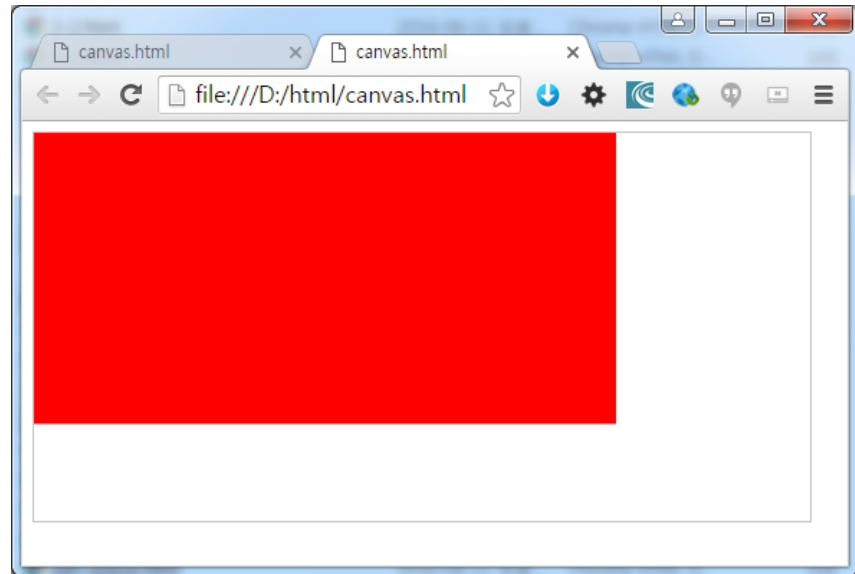


# 캔버스

- <canvas> 요소는 스크립트를 통해 그래픽을 그리는데 사용
- <canvas> 요소는 그래픽을 위한 컨테이너
  - 실제로 그래픽을 그리기 위해서는 스크립트를 사용
  - 경로, 상자, 원, 텍스트 그리기, 이미지 추가

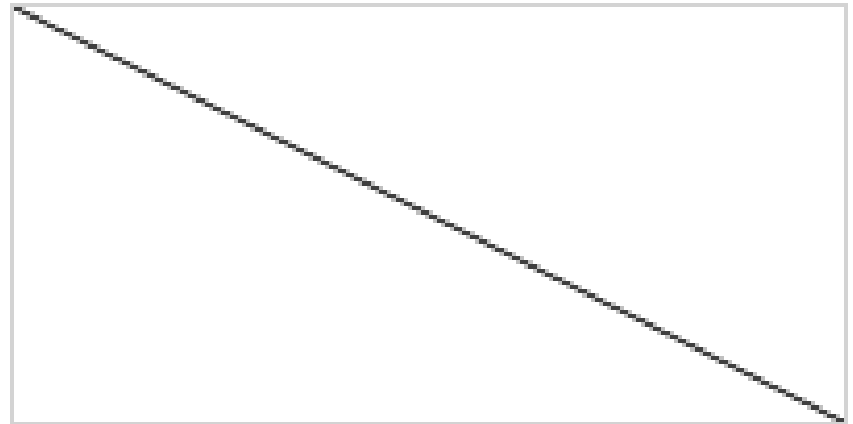
# <canvas> 예제

```
<script type="text/javascript">
 var c=document.getElementById("myCanvas");
 var cxt=c.getContext("2d");
 cxt.fillStyle="#FF0000";
 cxt.fillRect(0,0,150,75);
< /script>
```



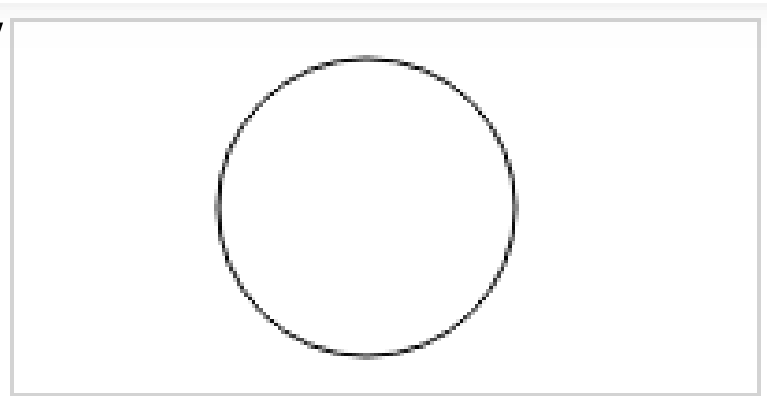
# 라인 그리기

```
var c =
document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.moveTo(0, 0);
ctx.lineTo(200, 100);
ctx.stroke();
```



# 원 그리기

- ```
var c =  
document.getElementById("myCanvas");  
var ctx = c.getContext("2d");  
ctx.moveTo(0, 0);  
ctx.lineTo(200, 100);  
ctx.stroke();
```



텍스트 쓰기

```
var c =  
document.getElementById("myCanvas");  
var ctx = c.getContext("2d");  
ctx.font = "30px Arial";  
ctx.fillText("Hello World", 10, 50);
```



Hello World

Stroke Text

```
var c =  
document.getElementById("myCanvas");  
var ctx = c.getContext("2d");  
ctx.font = "30px Arial";  
ctx.strokeText("Hello World", 10, 50);
```



Linear Gradient

```
var c = document.getElementById("myCanvas");  
var ctx = c.getContext("2d");  
// Create gradient  
var grd = ctx.createLinearGradient(0, 0, 200, 0);  
grd.addColorStop(0, "red");  
grd.addColorStop(1, "white");  
// Fill with gradient  
ctx.fillStyle = grd;  
ctx.fillRect(10, 10, 150, 80)
```



이미지 그리기

```
<body> <p>Image to use:</p>

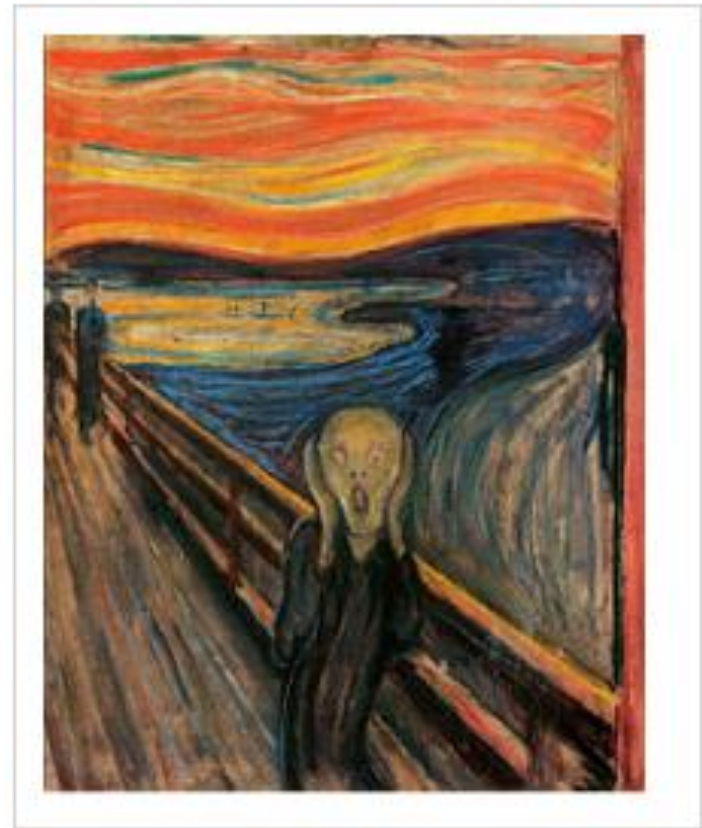
<p>Canvas to fill:</p>
<canvas id="myCanvas" width="250" height="300"
style="border:1px solid #d3d3d3;"/>
<p> <button onclick="myCanvas()">Try it</button> </p>
<script>
function myCanvas() {
    var c = document.getElementById("myCanvas");
    var ctx = c.getContext("2d");
    var img = document.getElementById("scream");
    ctx.drawImage(img,10,10);
} </script> </body>
```

이미지 그리기 결과

Image to use:



Canvas to fill:



Try it

제 6 장

게임 제작

계산기

Simple Calculator			
<input type="text"/>			
1	2	3	+
4	5	6	-
7	8	9	/
.	0	*	%
C		=	

계산기 Code (1)

```
<html> <head>
<title>Simple Calculator</title>
<style>
table { border:2px solid black; background-color:darkgray; }
td { padding:5px; height:40px; width:50px;
      text-align:center; color:white; border:2px solid black;
}
input { width:100%; height:100%; background-color: white; }
button { width: 100%; height:100% }
</style>
</head>
<body>
```


계산기 Code (2)

```
<form name="form1" onsubmit="return false">
<table align="center">
<tr><td colspan="4">Simpe Calculator</td></tr>
<tr><td colspan="4">
<input type="text" name="result"
        style="text-align:right"></input></td></tr>
<tr><td>1</td><td>2</td><td>3</td><td>+</td></tr>
<tr><td>4</td><td>5</td><td>6</td><td>-</td></tr>
<tr><td>7</td><td>8</td><td>9</td><td>/</td></tr>
<tr><td>.</td><td>0</td><td>*</td><td>%</td></tr>
<tr><td colspan="2" class="btn">
<button onclick="clr()">C</button></td>
<td colspan="2" class="btn">
<button onclick="calc()">=</button></td></tr>
```

계산기 Code (3)

```
</table> </form> <script>
tdl = document.getElementsByTagName("td");
for (i=0 ; i<tdl.length ; i++) {
    if (!tdl[i].classList.contains("btn")) {
        tdl[i].addEventListener("click", function() {
            document.form1.result.value += this.childNodes[0].nodeValue; });
    }
}
function clr() { document.form1.result.value = "0"; }
function calc() { document.form1.result.value =
                    eval(document.form1.result.value); }
</script> </body> </html>
```

숫자 맞추기 게임

숫자 맞추기 게임

1부터 100 사이의 숫자 맞추기

시작

이 페이지 내용:

Tell me your guess. 1 - 100

Your guess:

확인

취소

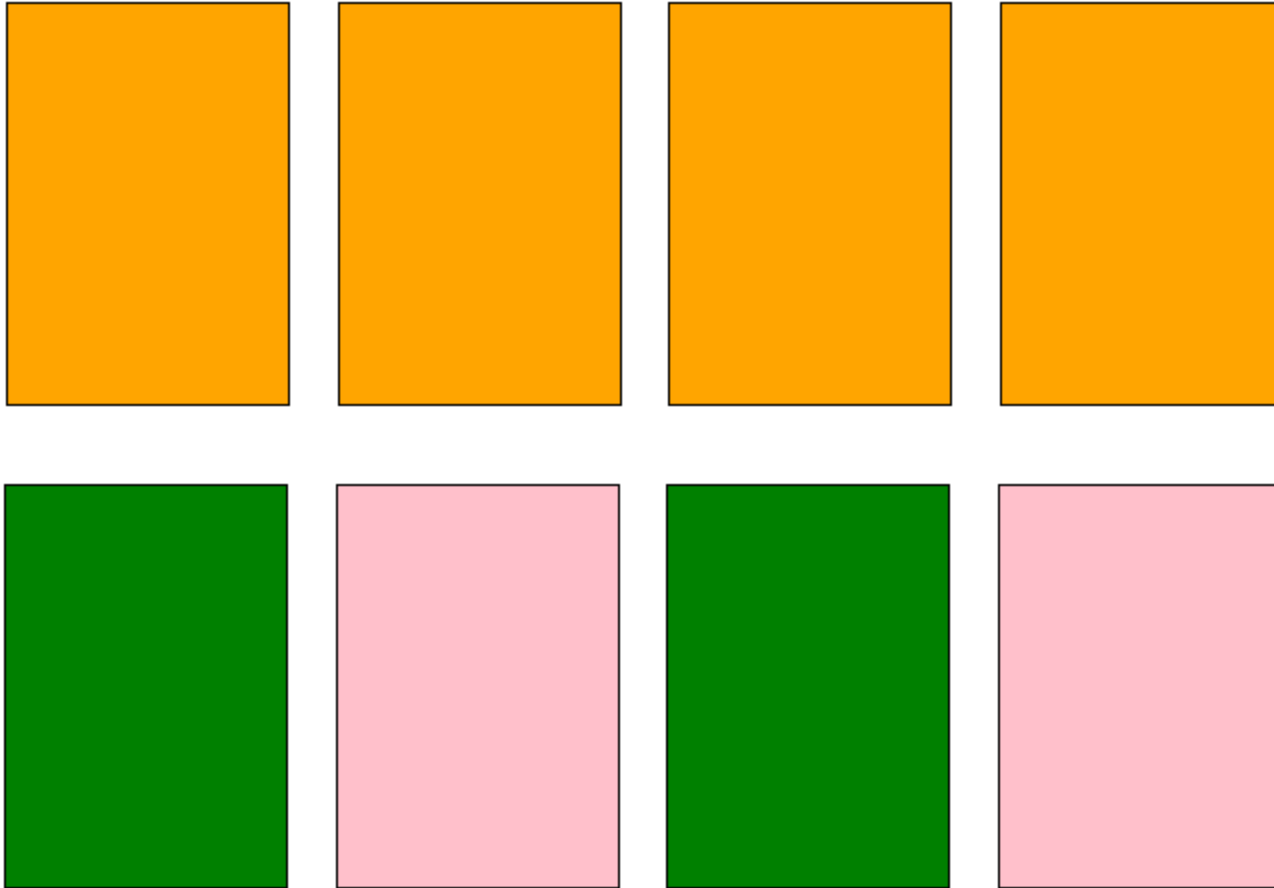
숫자 맞추기 HTML

```
<html> <head>
<script type="text/javascript">
var num;
function checkNumber() {
}
</script> </head> <body>
<h1>숫자 맞추기 게임</h1>
<p>1부터 100 사이의 숫자 맞추기</p>
<form id="Input" name="Input">
  <input onclick="checkNumber()" type="button" value="시|
작">
</form>
</body> </html>
```

숫자 맞추기 Script

```
<script>
var num;  var turns;
function checkNumber() {
  var num = Math.floor(Math.random() * 100) + 1;  var turns = 10;
  while (turns > 0) {
    guess=prompt("Tell me your guess. 1 - 100 ", "Your guess: ");
    if ( guess == null ) { break; }
    else if (guess==num) { alert("You won! The number was "+num+"."); break;}
    else if (guess<num) { turns--; alert("Too low. Turns remaining: " + turns); }
    else if (guess>num) { turns--; alert("Too high. Turns remaining: " + turns); }
  }
  if (turns==0) alert ("You failed to guess.");
} </script>
```

카드 짝 맞추기



카드 짝 맞추기 head

```
<!DOCTYPE html> <html> <head>
<meta charset="utf-8" /> <title>짝맞추기</title>
<style>
.card { display: inline-block; margin-top: 30px; margin-right: 20px;
margin-bottom: 10px; width: 140px; height: 200px; perspective: 280px; }
.card-inner { position: relative; width: 100%; height: 100%; text-align:
center; transition: transform 0.8s; transform-style: preserve-3d; }
.card.flipped .card-inner { transform: rotateY(180deg); }
.card-front { background: orange; }
.card-back { transform: rotateY(180deg); }
.card-front, .card-back { position: absolute; width: 100%; height: 100%;
border: 1px solid black; backface-visibility: hidden; }
</style> </head>
```

카드 짝 맞추기 body

```
<body>
```

```
<div id="wrapper">
```

```
<div class="card"><div class="card-inner"><div class="card-  
front"></div><div class="card-back"></div></div></div>
```

```
<div class="card"><div class="card-inner"><div class="card-  
front"></div><div class="card-back"></div></div></div>
```

```
<div class="card"><div class="card-inner"><div class="card-  
front"></div><div class="card-back"></div></div></div>
```

```
<div class="card"><div class="card-inner"><div class="card-  
front"></div><div class="card-back"></div></div></div>
```

```
</div>
```


카드 짝 맞추기 script

```
<script>
const $wrapper = document.querySelector('#wrapper');
const colors = ['pink', 'green'];
const total = colors.length * 2;
let colorCopy = colors.concat(colors);
let shuffled = []; let clicked = [];
let completed = []; let clickable = false;
function shuffle() {}
function startGame() {}
function resetGame() {}
startGame();
</script>
```

shuffle()

```
function shuffle() {  
  for (let i = 0; colorCopy.length > 0; i += 1) {  
    const randomIndex = Math.floor(Math.random() * total);  
    shuffled = shuffled.concat(colorCopy.splice(randomIndex, 1));  
  }  
}
```

onClickCard()

```
function onClickCard() {  
  if (!clickable || completed.includes(this) || clicked[0] === this) {  
    return;  
  }  
  this.classList.toggle('flipped');  
  clicked.push(this);  
  if (clicked.length !== 2) {    return;    }  
  const firstBackColor = clicked[0].querySelector('.card-back').style.backgroundColor;  
  const secondBackColor = clicked[1].querySelector('.card-back').style.backgroundColor;
```

onClickCard()

```
if (firstBackColor === secondBackColor) {  
    // 두 카드가 같은 카드면  
    completed.push(clicked[0]);  
    completed.push(clicked[1]);  
    clicked = [];  
    if (completed.length !== total) {    return;    }  
    setTimeout(function() { alert('축하합니다!'); resetGame(); }, 1000);  
    return;    }  
    clickable = false;  
    setTimeout(function() { clicked[0].classList.remove('flipped');  
        clicked[1].classList.remove('flipped'); clicked = [];  
        clickable = true; }, 500);  
}
```

startGame()

```
function startGame() {  
    clickable = false;  
    shuffle();  
    var card = document.querySelectorAll('.card');  
    for (let i=0; i<total; i++) {  
        card[i].addEventListener('click', onClickCard);  
    }  
    var backCard = document.querySelectorAll('.card-back');  
    for (let i=0; i<total; i++) {  
        let card = backCard[i];  
        card.style.backgroundColor = shuffled[i];  
        card.addEventListener('click', onClickCard);  
    }  
}
```

startGame()

```
document.querySelectorAll('.card').forEach((card, index) => {  
  // 초반 카드 공개  
  setTimeout(function() {  
    card.classList.add('flipped');  
  }, 1000 + 100 * index);  
  setTimeout(() => { // 카드 감추기  
    document.querySelectorAll('.card').forEach((card) => {  
      card.classList.remove('flipped');  
    });  
    clickable = true;  
  }, 5000);  
}
```

resetGame()

```
function resetGame() {  
  colorCopy = colors.concat(colors);  
  shuffled = [];  
  completed = [];  
  startGame();  
}
```

알파벳 정렬 게임

3x3 Shift Game		
A	B	C
D	E	G
H	F	

33

알파벳 정렬 HTML

```
<html> <head>
<meta name="viewport" content="width=device-width">
<style> input button {
  border: none; color: white; padding: 15px 32px; text-align: center;
  text-decoration: none; display: inline-block; font-size: 16px;
} </style> </head> <body> <center>
<form name=pad>
<table border=2 cellpadding=2 cellspacing=2>
  <tr> <td colspan=3 align=center>3x3 Shift Game</td> </tr>
  <tr> <td colspan=3><input type=text size=40
name=msg> </td> </tr>
</table> </form> </center>
<script> </script> </body> </html>
```

알파벳 정렬 Script (1)

```
<SCRIPT>
bx = 2; by = 2;
for ( y=0 ; y<3 ; y++ ) { document.write('<tr>');
  for ( x=0 ; x<3 ; x++ ) {
    document.write('<td><input type=button
      style="font-size:48px; width:100px; height:100px;" value=" "')
    document.write('onclick="move('+x+', '+y+');" > </td>');  }
  document.write('</tr>');
}
function move(x, y) {
  ax = Math.abs(bx-x);  ay = Math.abs(by-y);
  if (((ax*ay)==0) && ((ax+ay)==1)) {
    f=document.pad;
    f.elements[3*by+bx].value = f.elements[3*y+x].value;
    f.elements[3*y+x].value="  ";  bx = x;  by = y;  f.msg.value++;
  }
}
```

알파벳 정렬 Script (2)

```
function rndize() {  
    alpha = "ABCDEFGH ";  
    for ( i=0 ; i<9 ; i++ ) {  
        x=0;  
        y=0;  
        while (document.pad.elements[3*y+x].value!=" ") {  
            x=Math.floor(Math.random()*3);  
            y=Math.floor(Math.random()*3);  
        }  
        document.pad.elements[3*y+x].value=alpha.substring(i,i+1);  
    }  
    bx=x;  
    by=y;  
}  
rndize();  
</script>
```