

# JSP 프로그래밍

## 1차시

- 개발환경 구축
- 웹 프로그래밍의 이해
- JSP 맛보기
- Servlet 맛보기



중요한 것은 큰 뜻을 품고 그것을 완수하기 위한  
기능과 인내를 지니는 일이다. 그 밖의 것은  
하나도 중요하지 않다.

— 괴테 —



- JSP를 활용하여 웹 프로그램을 제작한다.
- 웹을 이해하고, 웹으로 서비스를 제공한다.
- HTML의 개념을 이해하고, 웹 서비스를 위한 환경을 구성한다.
- JSP / Servlet의 이해
- 웹 서버와 WAS(Web Application Server)의 이해



- JDK 설치
- Path 설정
- 이클립스 다운로드
- 톰캣 설치 및 환경 설정



## □ JDK 설치

- <http://www.oracle.com/technetwork/java/index.htm>
- Java SE 8 버전 다운로드
- 자신의 PC에서 사용하는 운영체제에 맞는 Java를 설치



## □ Path 설정

○ 시스템 > 설정 변경 > 고급 탭 > 환경 변수

○ 변수 이름: JAVA\_HOME

○ 변수값: 설치한 Java 디렉토리

– ex) c:/Dev/Java/jdk1.8.0\_x

○ Path 변수 다음의 값 추가하기

– ;%JAVA\_HOME%/bin;

○ CMD 창에서 java -version으로 설치 확인하기



## □ 이클립스 다운로드

- <http://www.eclipse.org/>

- download 클릭 > 자신의 OS 버전에 맞는 이클립스 선택

- Eclipse IDE for Java EE 를 선택하여 설치



## □ 톰캣 설치 및 환경 설정

- <http://tomcat.apache.org/>
- Tomcat 8 > 8.5.29 Core > zip
- 다운로드 받은 파일 압축풀기





## □ 이클립스와 Tomcat 연동

- 웹 서비스를 제공하는 것은 Server에 Web Server와 WAS를 설치하여 서비스하는 것이지만 개발환경에서는 보통 이클립스에 Tomcat을 연동하여 로컬 개발 환경을 구성한다.
- 강사는 Tomcat을 WAS라고 생각합니다.

# 본격적인 수업을 시작하기전에...



## □ 자기진단

- 서버와 클라이언트의 차이를 알고 있다.
- 요청(Request)와 응답(Response)의 차이를 알고 있다.
- 프로토콜과 status code를 알고 있다.
- 웹 서버(Web server)와 WAS(Web Application Server)의 차이를 알고 있다.
- HTML, Javascript, java, css를 설명할 수 있다.
- GET과 POST의 차이를 알고 있다.

# 웹 프로그래밍의 이해



□ 웹 프로그래밍이란 웹 어플리케이션을 구현하는 것

□ 웹 어플리케이션이란 웹을 기반으로 작동되는 프로그램

□ 웹이란 1개 이상의 사이트가 연결되어 있는 인터넷 서비스의 형태

□ 인터넷이란 1개 이상의 네트워크가 연결되어 있는 형태

뭐 소리인치 당체  
(원지함 궁서체)  
무르겠는대요???

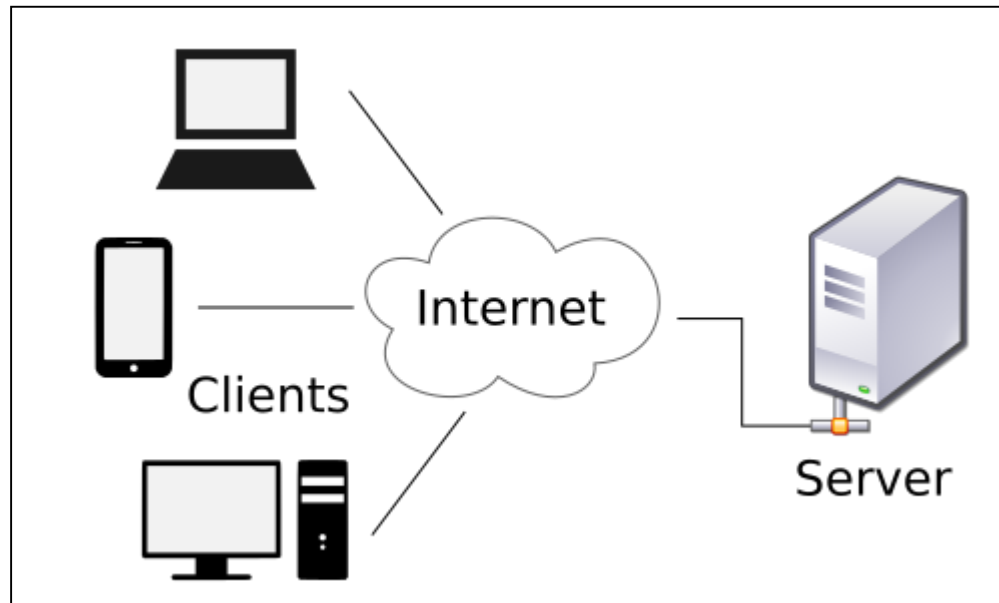


## □ 서버(Server)와 클라이언트(Client)

○ 클라이언트: 요청하는 쪽



○ 서버: 서비스를 제공하는 쪽

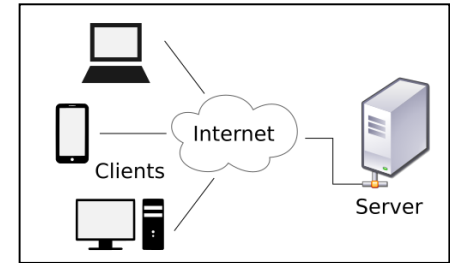


# 웹 프로그래밍의 필수 용어



## □ 서버(Server)

- 월드 와이드 웹(www)
- 구글, 페이스북, 네이버, 배달의 민족



## □ 클라이언트(Client)

- Chrome, IE, Firefox 등의 브라우저
- 아이폰, 안드로이드와 같은 스마트폰



## □ 요청(Request)와 응답(Response)

○ 요청: 클라이언트에서 서버쪽으로 요구하는 행위

- 접근하고자 하는 페이지(URL)
- HTTP 메소드: GET, POST 등이 존재

○ 응답: 서버에서 결과를 클라이언트쪽으로 보여주는 행위

- 상태코드: 성공여부
- 컨텐츠 타입: 텍스트, 그림, HTML
- 컨텐츠: HTML 코드, 이미지



## □ 프로토콜

○ 규약, 규칙, 약속

○ 컴퓨터 등 간의 정보교환에서 통신을 위해 미리 정해놓은 것

○ 네트워크, 웹, FTP 등 각각 다른 프로토콜을 가진다.

– TCP/IP, SMTP, HTTP, FTP 등... 



## □ HTTP (Hypertext Transfer Protocol)

○ 헤더(header)와 바디(body)로 구성됨



○ TCP/IP 기반의 웹용 프로토콜

- tcp: 한쪽 노드에서 다른 쪽 노드로 파일을 보내기 위한 프로토콜
- ip : 한 호스트에서 목적지 호스트까지 패킷을 옮기고 이동하기 위한 프로토콜

○ 웹 서버에서 서버-클라이언트 사이에 요청/응답이 가능하도록 만든 규약





## □ HTTP 메소드(Method)



○ HTTP 프로토콜을 사용하여 서버로 요청을 보내는 방법

- GET: URL을 이용하여 자원을 요청
- POST: Request에 첨부한 Body 정보를 서버로 전송
- HEAD: 헤더정보만 요청
- TRACE: 루프백 테스트 요청
- PUT, DELETE, OPTION, PATCH 등 존재



## □ HTTP GET

- 간단한 요청을 보낼 때 사용
- 단순히 서버에게 자원을 요청함
- URL 뒤에 전송할 데이터를 붙여서 보냄
- 데이터가 노출됨
- 내용의 크기가 제한적





## □ HTTP POST



- 복잡한 요청을 보낼 때 사용
- 전송할 데이터를 바디에 넣어서 요청함
- 데이터가 바디에 붙어서 노출되지 않음
- 내용의 크기에 제한 없음



## □ HTTP Response

- 서버가 클라이언트로 보내는 응답
- 헤더와 몸체로 구성
- 헤더: 성공, 실패, 프로토콜 내용 등
- 몸체: 콘텐츠



## □ URL(Uniform Resource Locator)

- 네트워크 상에서 자원이 어디 있는지를 알려주기 위한 규약
- `http://www.google.com:80/path/jsp/jsp_01.html`
- `http://`: 프로토콜
- [www.google.com](http://www.google.com): 도메인, 서버주소, 인터넷 상의 유일한 주소
- 80: port번호, 80은 생략 가능
- `/path/jsp`: 서버에서 자원의 위치
- `jsp_01.html`: 자원, 요청한 컨텐츠 이름



## □ TCP 포트

- 서버 하드웨어 상에서 돌아가는 SW를 구분하기 위한 16비트 숫자 값
- 사용 가능 범위는 0 ~ 65535
- 0번 ~ 1023번까지는 이미 알려진 서비스를 위해 예약되어 있음.
- FTP(21), Telnet(23), SMTP(25), HTTP(80), POP3(110), HTTPS(443)



## □ 웹서버

- 클라이언트로부터 요청을 받아 요청한 것을 응답해주는 일을 함
- 웹서버는 정적인 페이지(html, javascript, css, 이미지)만 서비스함
- Apache가 가장 유명한 오픈 소스 웹서버
- 동적인 콘텐츠 생성이 불가능하다. (JSP를 서비스 못 함)



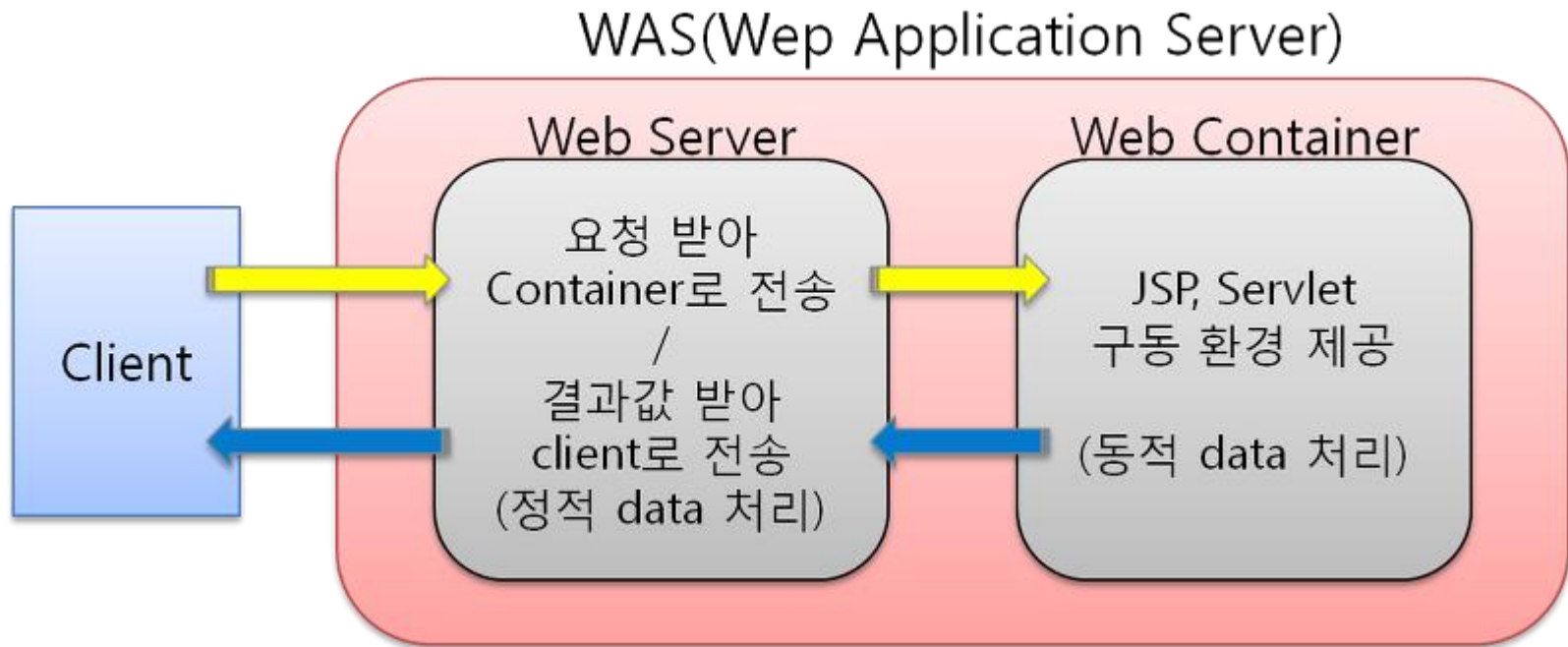
## □ 웹 어플리케이션 서버(WAS)

- 동적인 부분을 담당함(JSP 등)
- 프로그램 실행 환경과 데이터베이스 접속 기능을 제공
- 여러 개의 트랜잭션을 관리
- 업무를 처리하는 비즈니스 로직을 수행
- 대표적으로 Tomcat이 있다.





## □ 웹서버와 WAS





## □ HTML(Hyper Text Markup Language)

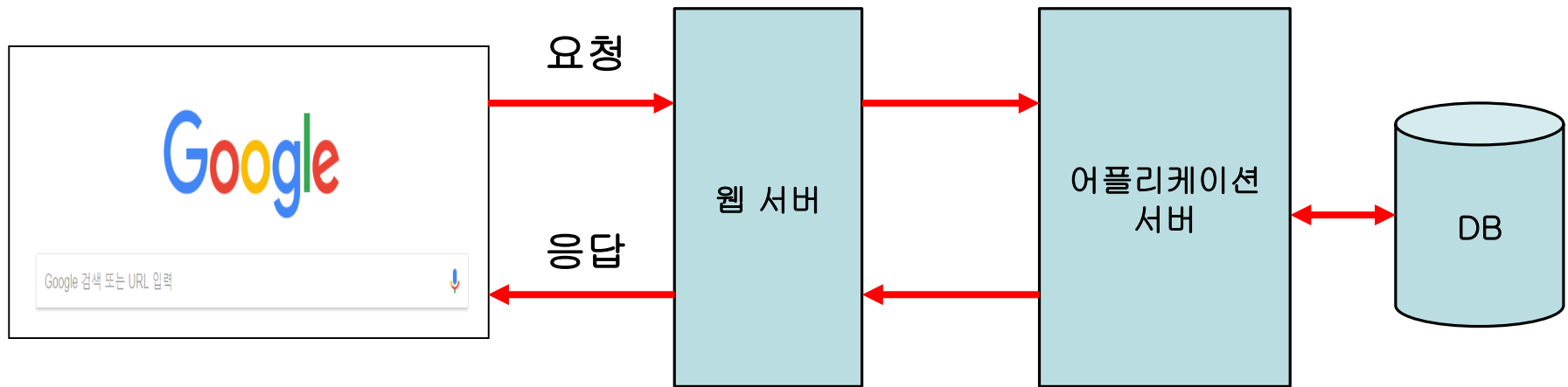
- HTML은 웹 페이지를 구성하는 핵심요소

- 마크업 언어: 문자를 이용하여 다른 문자의 특성을 표시하는 방법

- ex) `<b>hello world</b>` ➔ **hello world**

- 시작태그-컨텐츠-끝태그

# 웹 프로그래밍 동작 방법





- Java: JSP 프로그래밍을 위한 필수 선행 학습
- HTML: 웹어플리케이션을 위한 기본 언어
- Javascript: 웹브라우저 위에서 동작하는 언어
- CSS: 웹어플리케이션의 레이아웃 및 스타일 언어



## □ JSP(Java Server Page)

- HTML 페이지 내에 자바 코드를 삽입하여 웹 서버에서 동적으로 웹 페이지를 생성하여 클라이언트에 전달하는 언어
- .jsp 확장자이며 html코드와 java코드가 동시에 존재
- 클라이언트 요청에 동적으로 작동하여 응답은 html을 이용
- jsp는 서블릿으로 변환되어 실행
- MVC패턴에서 View로 이용됨