

## OSI 7계층 (TCP/IP 4계층)

↳ 다양한 프로토콜들이 존재

그중에서 가장 유명한 것이 HTTP

## 웹의 구성

- HTTP
  - HTML ⇒ 웹 페이지를 채운 내용
  - JavaScript ⇒ 웹페이지에 들어갈 기능
  - CSS ⇒ 웹페이지의 디자인
  - ASP/ASP.NET
  - JSP
  - DB
  - Python
  - Spring
  - JQuery
  - Ajax
- 클라이언트가 조작 가능
- 서버 사이드
- 필수
- 선택

## HTTP 프로토콜

Hyper Text Transfer Protocol

↳ HTML, JS, CSS 같은 파일을 웹서버에 요청/응답하는 프로토콜

X HTTP 1.0은 비효율적이라서 대체해서 1.1 사용

3way Handshake를 2번 하기 때문

## HTTP 요청 프로토콜의 구조

X 다른 계층과는 다르게 "문자"로 작성

Request Line
Headers (옵션)
공백 1줄
Body

### Request Line

요청타입	공백	URI	공백	HTTP 버전
------	----	-----	----	---------

### 요청타입

GET, POST ... 등이 있음

↳ 둘다 서버에서 문서를 가져오거나 서버에 문서를 보낼 수 있음

X GET과 POST의 차이

↳ GET 방식은 서버로 데이터를 보낼 때 URI에 포함

POST는 Body에 포함시킬 것

EX) ID, PW 등을 전송할 때는 POST로

### URI

# Uniform Resource Identifier

↳ 인터넷 상에서 특정 자원을 나타내는 유일한 주소

Scheme : // host [:Port] [/path] [*?query*]

ex) ftp : // IP주소 : 포트 / 파일을

http : // IP주소 : 포트 / 폴더이름 / 파일을

도메인주소 쿼리스트링이 있어서 변경 가능

*\* query*

↳ 사용자가 입력 데이터를 전달하는 영역

ex) http : // 127.0.0.1 : 443 / ABC / calculation.jsp ? num1=10 & num2=20

⇒ calculation.jsp에 쿼리 받은 두 값을 더하는 코드가 있다면

num1과 num2의 값을 더하여 이 페이지에 띄운다

즉, 쿼리는 추가적 정보를 서버에 전달해준다

## HTTP 응답 프로토콜의 구조

Status Line
Headers
공백
Body

## Status Line

HTTP 버전	공백	상태코드	공백	상태문구
---------	----	------	----	------

## 상태코드 + 상태문구

상태코드	상태문구	해석
⋮	⋮	
200 ~ 299	OK	정상
400 ~ 499	Forbidden / Not Found	클라이언트 문제
500 ~ 599	Internal Server Error / Service Unavailable	서버 문제

