

# Lab01: The Basics of Web Technology

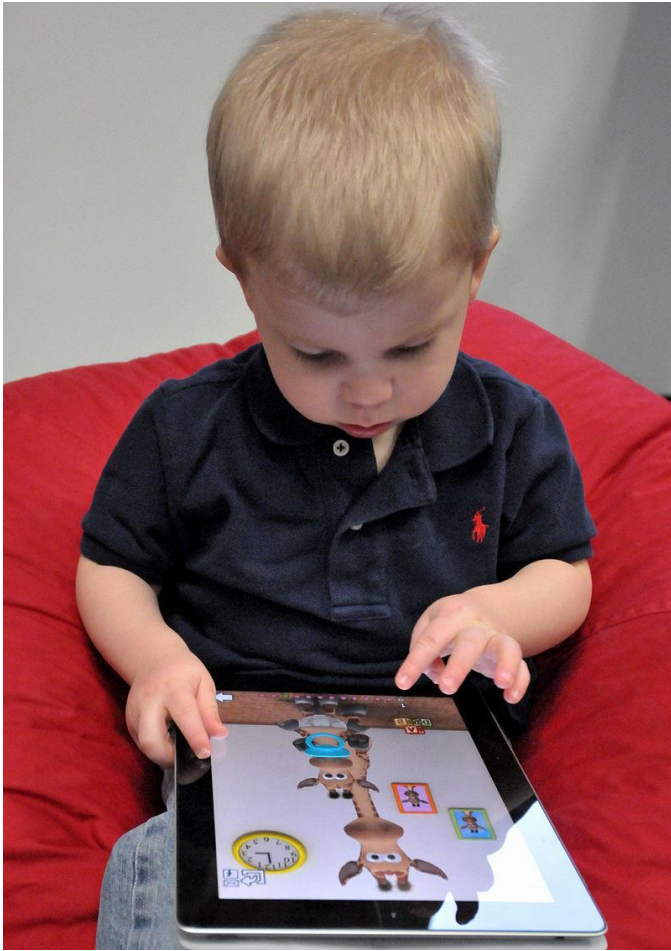
ESM2014-41

객체지향프로그래밍 및 실습

SKKU 시스템경영공학과

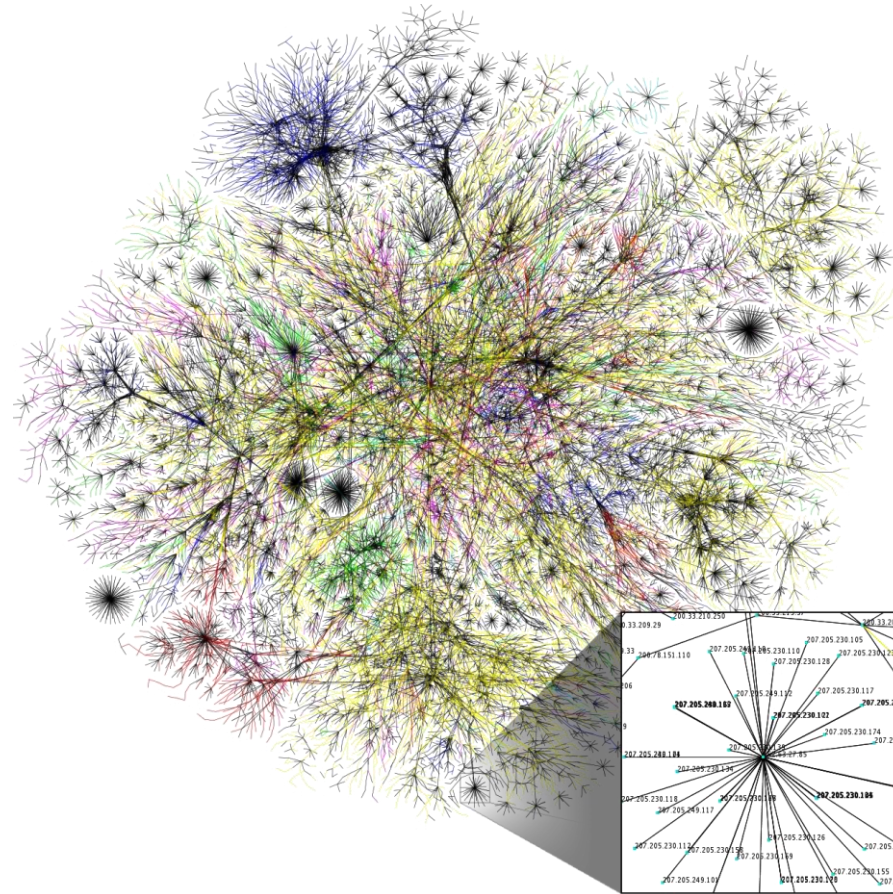
조영일

# Why we should study the Internet?



- Digital Native
  - 1980 ~ : We are the digital natives.
  - 태어날 때부터 디지털 환경에 노출되어 온 세대
  - 2008 ~ : ‘스마트폰을 쥐고 태어난 세대’
  - 모국어보다도 인터넷 환경을 먼저 접하게 되는 세상
  - ‘인터넷’ 기술은 현대 세상에서 가장 중요한 기저(basis)이다.

# Definition of Internet



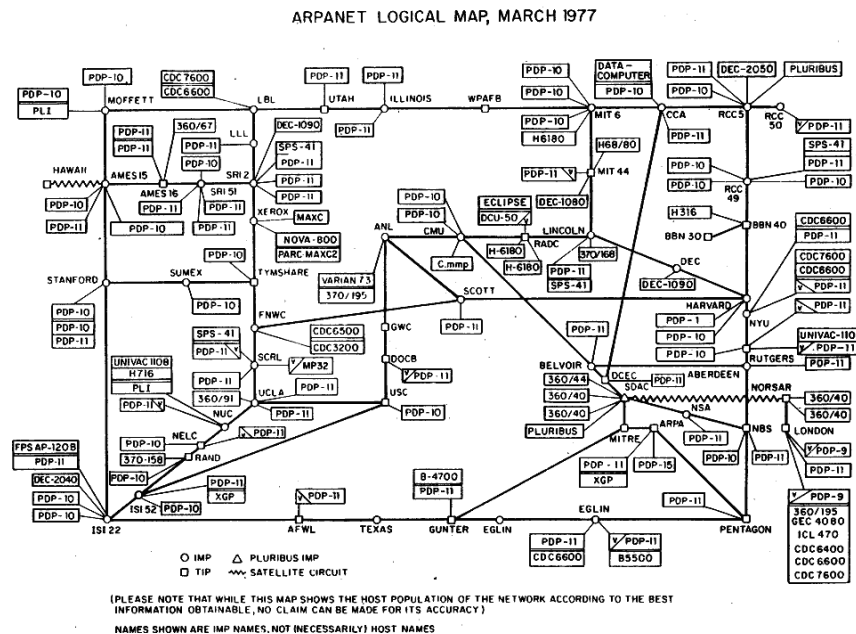
<https://www.submarinecablemap.com/>  
<https://global-internet-map-2018.telegeography.com/>

# Definition of Internet

“The Internet is the global system of interconnected computer networks that use the Internet protocol suite (TCP/IP) to link devices worldwide.”

- Network of Networks
- 세계 최대의 컴퓨터 통신망

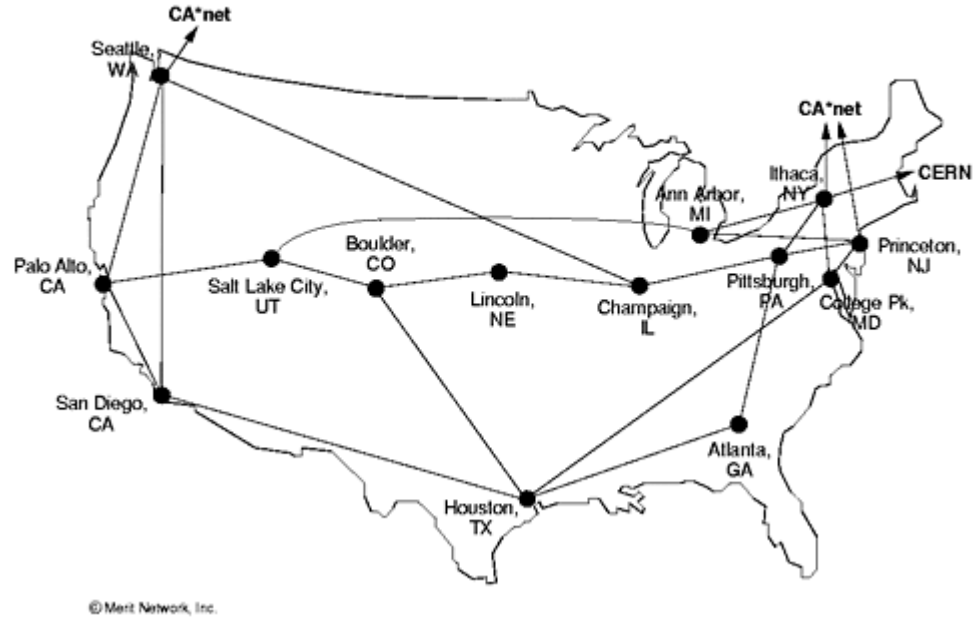
# History of Internet



- ARPANET
  - 1969년 미국 국방성에 의해 시작된 세계 최초의 패킷 스위칭 네트워크
  - 인터넷의 원형
  - UCLA / 캘리포니아 대학교 산타바바라 / 스탠퍼드 연구소 / 유타 대학교 네개의 네트워크간 연결

# History of Internet

NSFNET T1 Network 1991



- NSFNET
  - National Science Foundation Network
  - 1990년대 부터 본격적으로 인터넷이 보급되며 전 세계 컴퓨터는 이 NSFNET과 연결되기 시작함
  - 1995년 부터는 NSFNET이 사라지고 전 세계 각 지역에 유사한 네트워크가 등장하기 시작함

# History of Internet

- 1969 : ARPANET 등장
- 1972 : Email 탄생
- 1974 : Internet 용어 등장
- 1982 : TCP/IP 도입
- 1984 : DNS 제시
- 1986 : NSFNET 시작
- 1991 : WWW(World Wide Web) 개발(팀버너스리)
- 1993 : 브라우저 Mosaic 등장
- 1996 : 브라우저 MS Internet Explorer 등장
- 1998 : 세계 인터넷 이용자수 1억명 돌파, Google 창업
- 2004 : Facebook 창업
- 2007 : Apple iPhone 발표, 모바일 인터넷 사용자수 급증

# Definition of Network

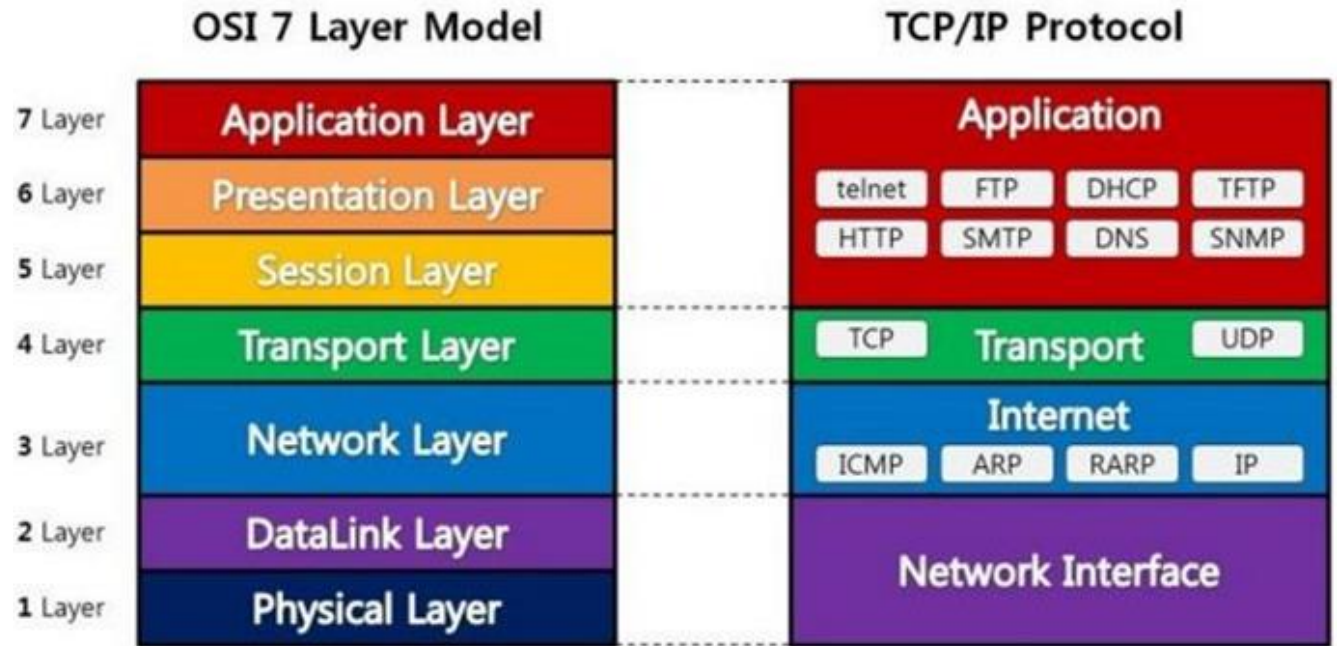
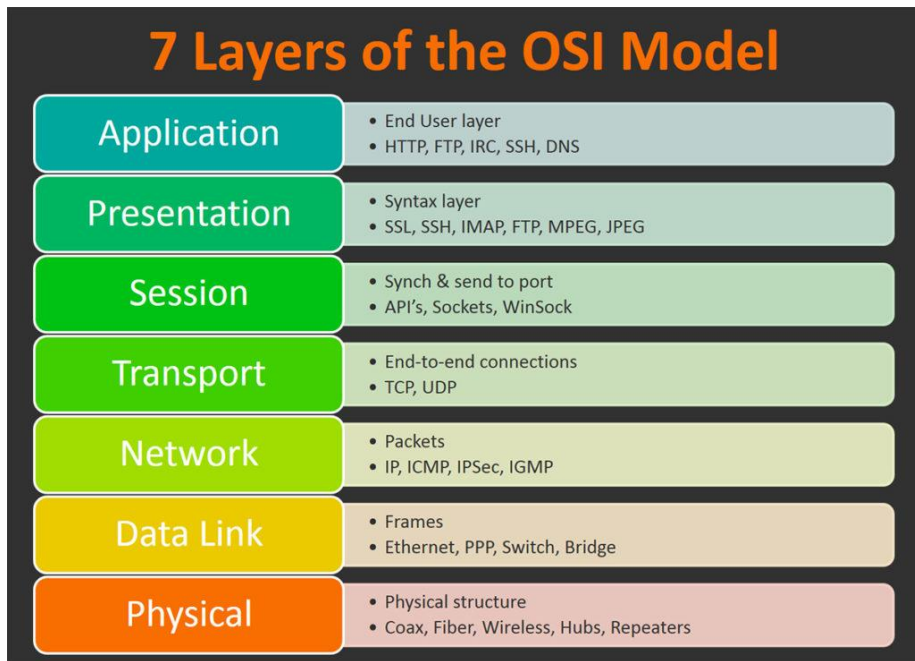
- Computer Network : A computer network is a digital telecommunications network which allows nodes to share resources.
- Telecommunications Communication : A telecommunications network is a collection of terminal nodes in which links are connected so as to enable telecommunication between the terminals.
- 통신(Communication)을 위해 필요한 3요소
  - 통신의 주체(Node) : 송신자, 수신자
  - 매체(Link) : 전화선, 광케이블 등
  - 공통 규약(Protocol) : 서로 신호를 주고 받기위한 언어 및 규약



# Network Protocol

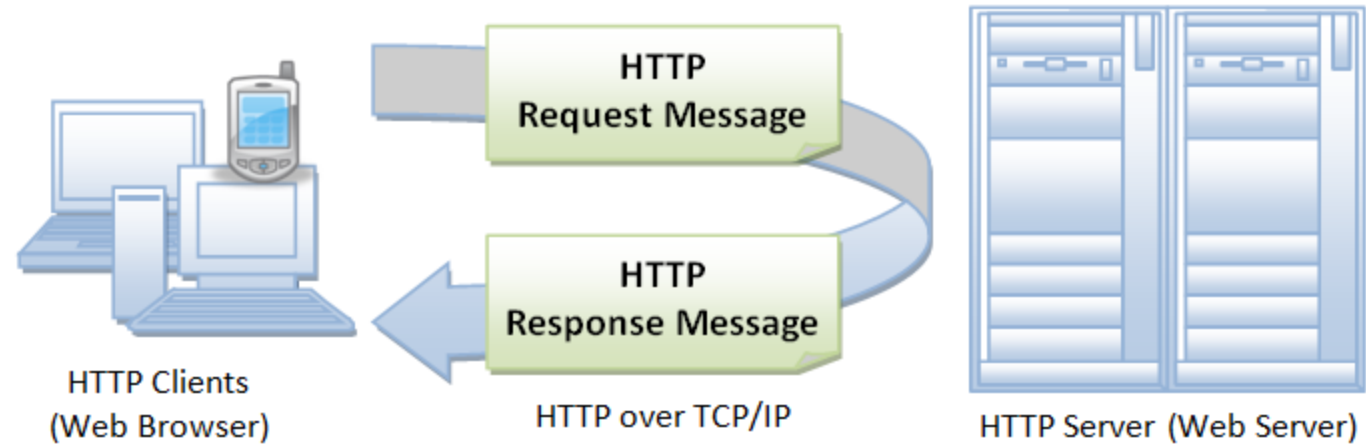
- 프로토콜은 통신을 위해 사람들이 설계한 규약으로 서로 다른 장치나 컴퓨터가 동일한 방식으로 데이터 통신을 할 수 있도록 함
- 대표적인 프로토콜 : TCP/IP, HTTP
- 인터넷의 기반이 되는 프로토콜 : TCP/IP(Transmission Control Protocol/Internet Protocol)
  - TCP : 전송하고자 하는 데이터를 패킷 단위로 분해하고 목적지에 도착 할 경우 전송된 패킷들을 검사하고 다시 재결합 하여 원래의 데이터로 변환하는 프로토콜
  - IP : 패킷을 특정 목적지로 전송하기 위한 역할을 수행하는 프로토콜

# OSI 7 Layer vs TCP/IP Protocol



# Network Protocol

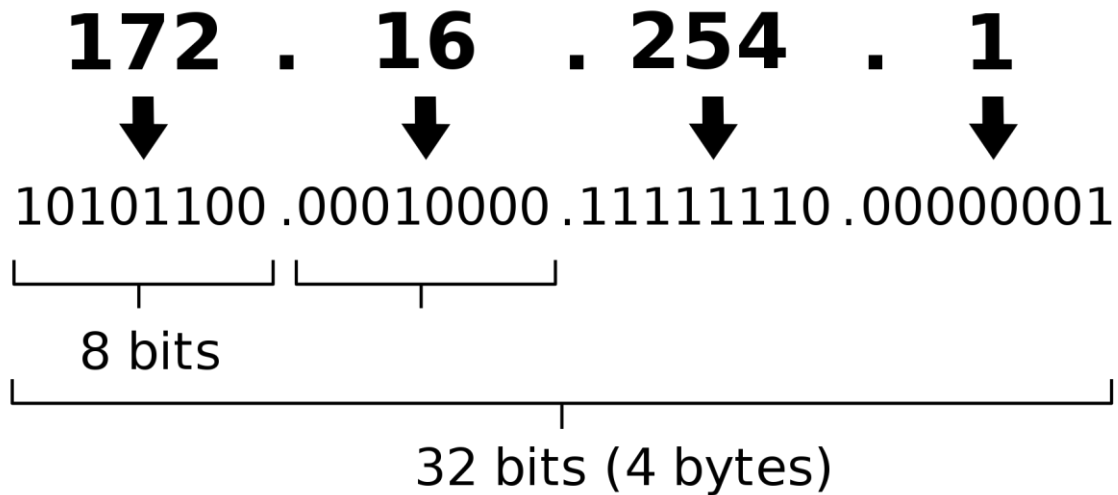
- HTTP(HyperText Transfer Protocol)
  - HTML 로 작성된 문서들의 송수신을 위해 사용되는 프로토콜
  - HTTP는 TCP/IP 보다 상위에서 동작하는 Application Protocol임



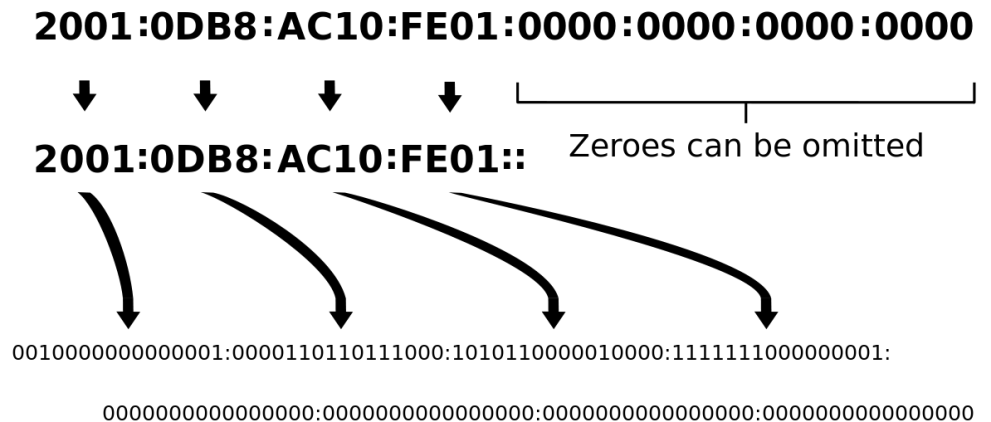
# IP Address

- IP Address
  - 인터넷에 연결된 컴퓨터를 식별하기 위한 고유의 주소
  - 인터넷에 연결된 모든 장치들은 고유의 IP Address를 가지고 있음
  - 현재는 IPv4 로 감당 가능한 장치 수를 넘어감, IPv6의 등장

IPv4 address in dotted-decimal notation



An IPv6 address (in hexadecimal)



# Domain Name / DNS

- IP Address 는 사람이 쉽게 기억하기 어려움
- 사람들이 기억하기 쉽고 사용하기 편리하기 위해 인터넷에서는 Domain Name 이라는 또 다른 주소를 제공
- DNS – Domain Name Server 는 Domain Name과 IP Address 사이의 변환을 담당함
- 장치가 속한 기관이나 국가에 따라 계층적 구조를 가짐
- [장치/서비스 이름].[기관이름].[기관종류].[국가이름] 의 형태
  - lib.skku.edu – [도서관].[성균관대학교].[교육기관]
  - www.intakefoods.co.kr – [월드와이드웹].[회사명].[회사].[한국]
- 마지막 [국가이름]의 경우 생략되는 경우도 있음. 과거 미국의 경우만 생략되었으나, 사실상 도메인 네임은 국경과 상관 없이 구매가능 하므로 현재는 그 의미가 불분명함

# Definition of Web

- 웹 출현 이전의 인터넷
  - 일부 전문가들이 외부 컴퓨터에 접속하여 정보를 공유하는 수단
  - console i/o를 통해 내 컴퓨터가 아닌 원격 컴퓨터에 명령어를 수행하기 위한 수단
  - 일반인에겐 어려운 문제
- Tim Berners-Lee
  - CERN(유럽 입자 물리학 연구소) 연구원
  - 하이퍼링크(Hyperlink) 기반의 문서 구조를 제안
  - \*하이퍼링크(Hyperlink) : 문서에서 링크를 통해 다른 문서로 연결하는 혁신적인 개념
  - 하이퍼링크를 지원하는 HTTP(Hypertext Transfer Protocol)에 대한 초안 제출
  - 하이퍼링크와 HTTP는 오늘날 웹이 있게 한 인터넷에서 가장 중요한 기술 중 하나

# Definition of Web

- Web 은 현재 모든 IT 기술의 기반
- 웹의 용도
  - 웹사이트 : 검색/SNS/쇼핑/행정 ...
  - 모바일 : Mobile Device 상의 대부분의 서비스들이 웹 기술을 사용하여 구현 됨
  - IoT : Internet of Things - 요즘 만들어지는 거의 대부분의 장비에는 인터넷 연결 기능이 탑재됨(e.g. 따릉이, 자동차, 냉장고, TV, 프린터, 공기청정기 등)

# Definition of Web

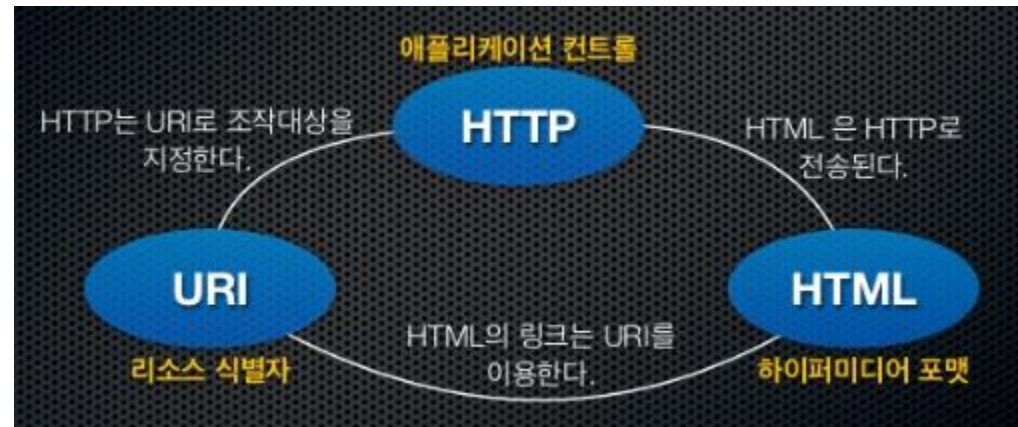
- WWW(World Wide Web)

- 1990년에 발표된 인터넷에 연결된 컴퓨터를 이용해 전 세계 사람들과 정보를 공유할 수 있는 거미줄(Web)처럼 얽힌 공간
- 하이퍼텍스트(Hypertext) 기반으로 연결된 문서들의 집합체

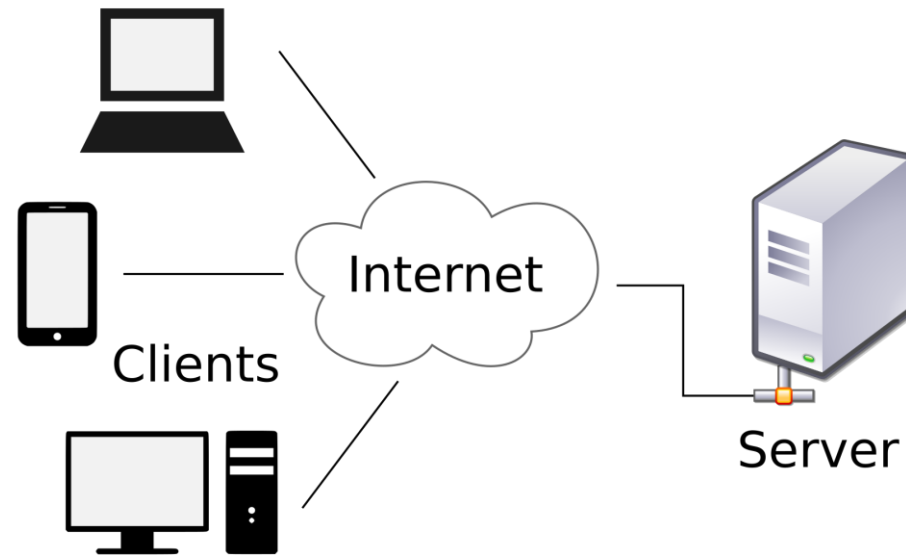


# 3 Component of Web

- HTTP : Hyper Text Transfer Protocol – 하이퍼텍스트 문서를 전송하기 위한 통신 규약
- URI : Uniform Resource Identifier – 인터넷 상에서 문서의 위치를 나타내기 위한 수단
- HTML : Hyper Text Markup Language – 하이퍼텍스트 문서를 작성하기 위한 언어

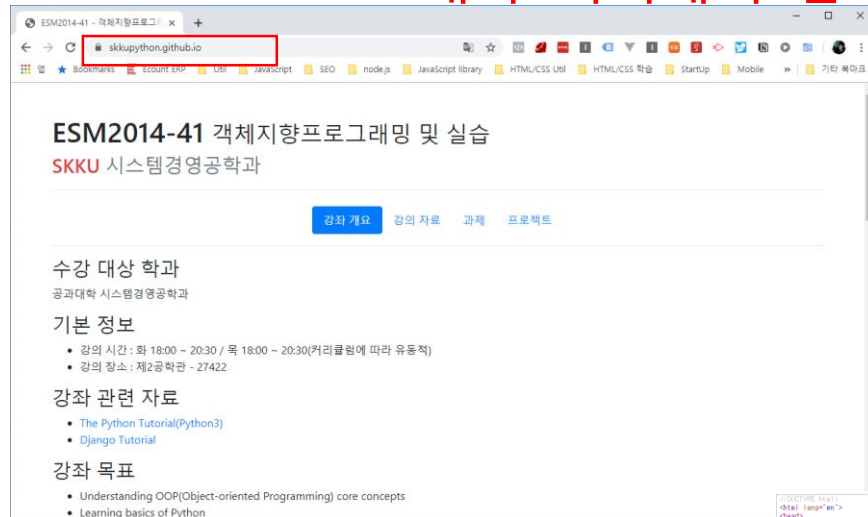


# Client / Server Model



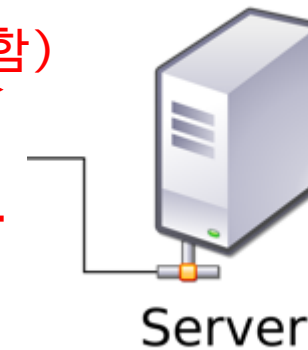
# Client / Server Model

URI : 내가 서버에서 탐색하고자 하는 자원의 고유한 위치



HTTP Request(URI 정보를 포함)

HTTP Response(HTML을 포함)

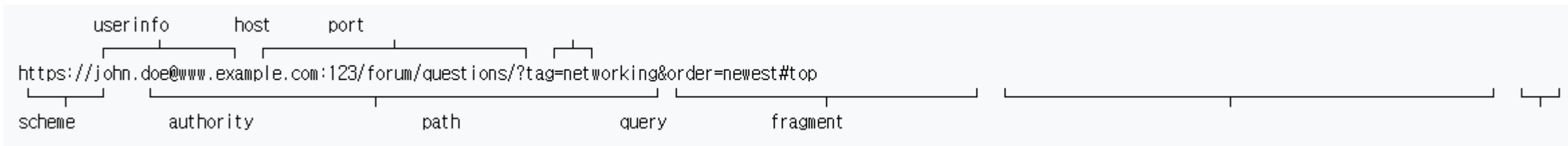
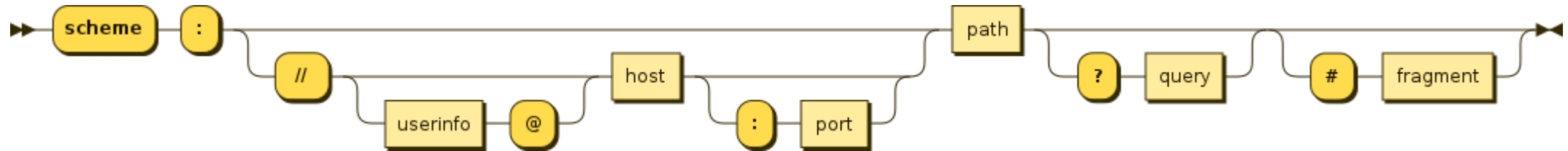


Web Browser : HTML 문서와 그림, 멀티미디어 파일등 WWW을 기반으로 인터넷의 콘텐츠를 탐색하기 위한 프로그램 (e.g. MS Edge, MS Internet Explorer, Google Chrome)

```
<DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <title>ESM2014-41 - 객체지향프로그래밍 및 실습</title>
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-gppwRj9Vlh3P0Pv6+8tQ32RejY7Wf0/J6LHk876l8X2N11" crossorigin="anonymous">
  <link rel="stylesheet" href="assets/css/style.css">
</head>
<body>
  <div class="row">
    <div class="col-md-12">
      <div class="page-header">
        <h1>ESM2014-41 객체지향프로그래밍 및 실습</h1>
        <h2>SKKU 시스템경영공학과</h2>
      </div>
      <div class="row">
        <div class="col-md-12">
          <div class="main-content">
            <div class="nav-item">
              <a href="#">강좌 개요</a>
            </div>
            <div class="nav-item">
              <a href="#">강의 자료</a>
            </div>
            <div class="nav-item">
              <a href="#">강의 과제</a>
            </div>
            <div class="nav-item">
              <a href="#">강의 프로젝트</a>
            </div>
          </div>
          <div class="tab-content">
            <div class="tab-pane fade show active">
              <div class="summary">
                <h3>수강 대상 학과</h3>
                <p>공과대학 시스템경영공학과</p>
                <h3>기본 정보</h3>
                <p><strong>강의 시간</strong> : 화 18:00 ~ 20:30 / 목 18:00 ~ 20:30(커리큘럼에 따라 유동적)</p>
                <p><strong>강의 장소</strong> : 제2공학관 - 27422</p>
                <h3>강좌 관련 자료</h3>
                <p><a href="https://docs.python.org/3/tutorial/index.html" target="_blank">The Python Tutorial(Python3)</a></p>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

HTML

# URI / URL / URN 의 개념



- URI는 인터넷상에서 특정 문서의 위치를 나타내기 위한 표기법
  - URI는 URN(Uniform Resource Name – 리소스 자체의 이름을 나타내기 위한 방법), URL(Uniform Resource Locator – 리소스 자체의 위치를 나타내기 위한 방법)로 구성되어 있음
  - URN은 실제적으로 사용하는 경우가 드물다, 결국 우리 URI 중 URL을 대부분 사용하고 있다.

# History of HTTP

- HTTP 0.9 – HTTP의 탄생
  - 팀 버너스 리가 최초의 웹을 발명했을 때 사용한 프로토콜
  - 현재의 HTTP와 다르게 헤더가 없고, GET 메서드만 존재한다. 현재 사용되지 않음.
- HTTP 1.0 – HTTP 최초의 표준화
  - IETF에서 표준화 하여 1993년 Draft, 1996년 최종 버전(RFC 1945) 공개
  - 헤더의 도입, GET 이외의 메서드 추가
- HTTP 1.1 – HTTP의 완성
  - 1997년 RFC 2068에서 개정, 1999년 RFC 2616 발행.
  - 채널전송, Accept Header Contents Negotiation, Cache-Control, Keep-Alive 등 추가
- HTTP 2.0 – HTTP 1.1이 가지고 있는 여러가지 문제를 해결
  - 2009년 Google 이 발표한 SPDY프로토콜에 기반하여 개발
  - HTTP 1.1 이 가지는 구조적인 문제점에서 기인하는 성능 문제를 해결하는 것에 초점

# Basics of HTTP

- 클라이언트가 서버에 접속하여 요청(Request)를 보내면 서버는 요청을 처리해서 응답(Response)를 전달
- 동기형(Synchronous) 프로토콜 - 요청을 위해 서버에 접속하고 응답이 올때까지 대기
- 비연결형(Connectionless) 프로토콜 - 응답이 오고 나면 연결을 즉시 끊어 버림
- 비상태형(Stateless) 프로토콜 - 서버와 클라이언트는 서로의 상태를 기억하고 있지 않음

# HTTP Message



- Request 와 Response 모두 동일한 구조로 정의되어 있음
- Header : 요청/응답의 정보를 담기 위한 Section
- Body : 요청/응답의 본문을 담기 위한 Section

# HTTP Message : Request

```
1 GET / HTTP/1.1
2 HOST: www.shopintake.com
3 accept: application/json
4 accept-encoding: gzip, deflate
5 accept-language: en-US,en;q=0.8
6 user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.120 Safari/537.36
7 cookie: csrftoken=wY2cHz7cbbJzVL83AwrikFcGKkK5DkwLL1d7QGpQ877fel6boc9rxbs3N0zQ5; sessionid=r2rk9z1z62fefe3dgtlfbegbm9khiyi
```

- 1 Start Line : 요청 HTTP Method , URI, 프로토콜 버전
- 2~7 Header(s) : 요청하는 클라이언트에 대한 정보 “이름: 값” 형태로 구성
- POST, PUT Method의 경우 Header 이후에 Body가 존재



# HTTP Message : Response

```
1 HTTP/1.1 200 OK
2 content-encoding: gzip
3 content-length: 18742
4 content-type: text/html; charset=utf-8
5 date: Mon, 28 Oct 2019 08:06:55 GMT
6 loadingtime: 1.29s
7 server: Apache/2.4.33 (Amazon) mod_wsgi/3.5 Python/2.7.14
8 set-cookie: sessionid=ilq3dz34o2mc5k2ba4319hqefx2rhaeb; expires=Mon, 28-Oct-2019 20:06:56 GMT; httponly; Max-Age=43200; Path=/

9 <!DOCTYPE html>
10 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ko" lang="ko">
11 <head>
...

```

- 1 Status Line : 요청 프로토콜, 요청에 대한 HTTP Status Code, Text  
구문
- 2~8 Headers : 응답에 대한 정보(contents encoding, length, type, cookie 등), “이름: 값”  
형태로 구성
- 9~ Body: 응답의 본문(보통은 HTML)

# HTTP Method

Method	용도	의미
GET	Read	리소스 취득
POST	Create	서브 리소스의 작성, 리소스 데이터의 추가, 그 밖의 처리
PUT	Update	리소스 갱신, 리소스 작성
DELETE	Delete	리소스 삭제
HEAD		리소스의 헤더(메타 데이터) 취득
OPTIONS		리소스가 서포트하는 메소드의 취득
TRACE		자기 앞으로 요청 메시지를 반환(루프 백) 시험
CONNECT		프록시 동작의 터널 접속으로 변경

# HTTP Status Code

HTTP Status Codes		
<b>Level 200 (Success)</b> 200 : OK 201 : Created 203 : Non-Authoritative Information 204 : No Content	<b>Level 400</b> 400 : Bad Request 401 : Unauthorized 403 : Forbidden 404 : Not Found 409 : Conflict	<b>Level 500</b> 500 : Internal Server Error 503 : Service Unavailable 501 : Not Implemented 504 : Gateway Timeout 599 : Network timeout 502 : Bad Gateway

# Study Topic

- 웹을 지탱하는 기술(<https://www.slideshare.net/guruguru/ss-14241987>)
- 기본 웹 기술 용어(<https://www.slideshare.net/kthcorp/kth-1-20120307>)
- Advanced Topic : MDN 개발자를 위한 웹 기술  
(<https://developer.mozilla.org/ko/docs/Web/HTTP/Overview>)