

WEB PROGRAMMING

11/27 18:00 PM

WEB PROGRAMMING

- ✓ 웹을 기반으로 작동하는 웹 어플리케이션을 구현
- ✓ 웹: 1개 이상의 사이트가 연결되어 있는 인터넷 서비스의 형태
- ✓ 인터넷: 전세계의 컴퓨터 네트워크들을 하나로 묶어서 이루어진 네트워크
 - 표준 인터넷 프로토콜 (TCP/IP) 이용

➤ Protocol (프로토콜)

- 네트워크 상에서 약속한 통신 규약
- 현재 일반적으로 HTTP 가장 많이 사용
- ex) HTTP, FTP, SMTP, POP, DHCP

TCP/IP

➤ TCP (Transmission Control Protocol)

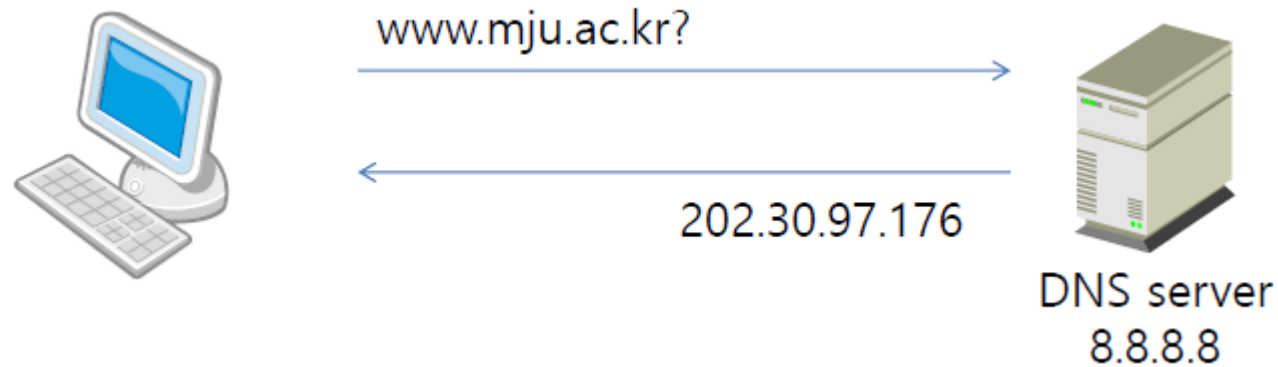
- IP 위에서 동작
- 데이터 송수신 및 순서 보장

➤ IP (Internet Protocol)

- 송수신 간 패킷 교환 프로토콜 (패킷: 데이터를 작게 자른 것)
- 중간 LAN에서 다음에 어디로 보낼지 모르고, 송수신 여부를 모름 => 비신뢰성
- IPv4, IPv6
- IP 주소: 0.0.0.0 ~ 255.255.255.255 (32비트)
- Domain Name: IP 주소에 대응 ex) www.naver.com

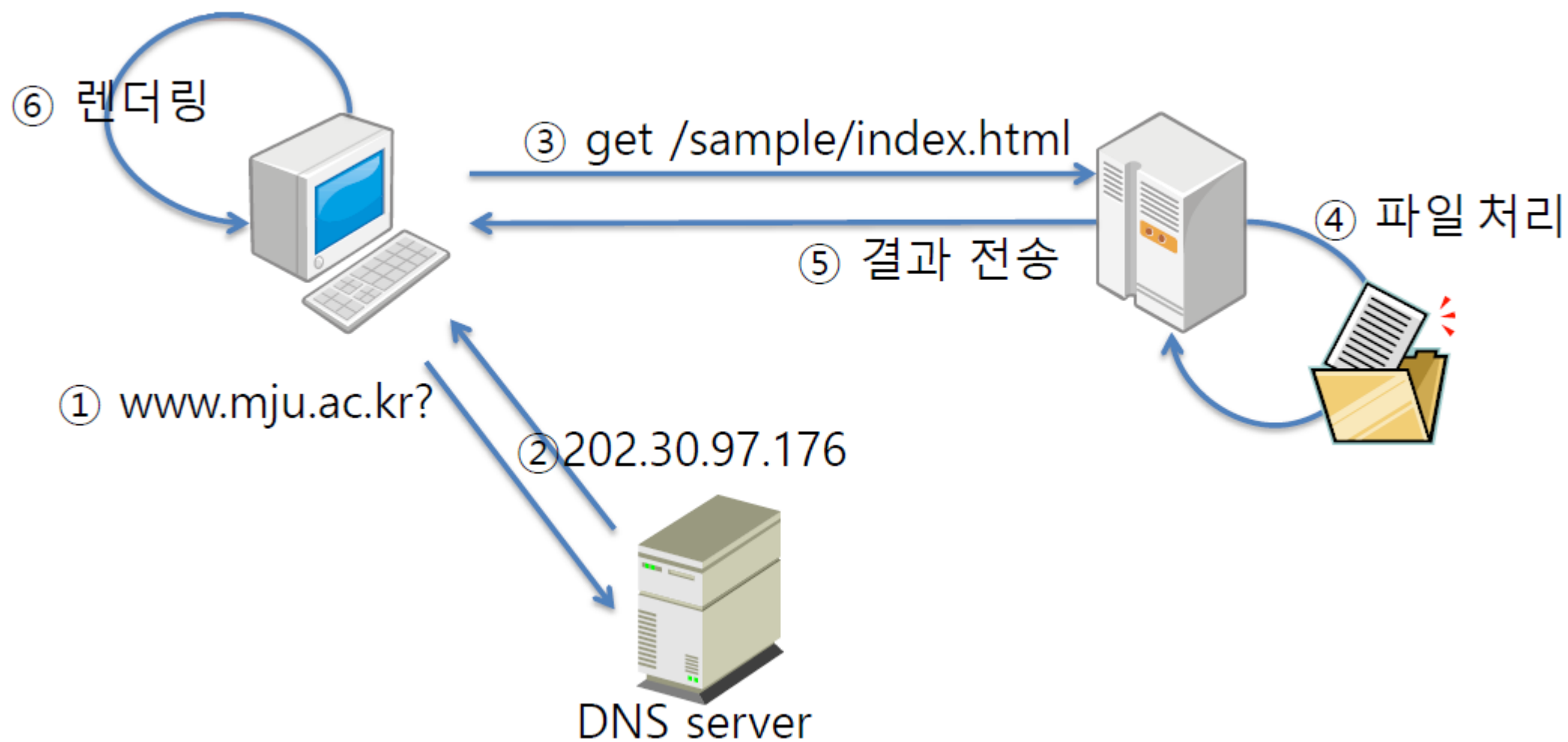
➤ DNS

- Domain Name Service
- Domain Name을 받아 IP 주소를 알려 줌



➤ Port 번호

- 해당 기기에서 구동되고 있는 프로그램을 구분할 수 있도록 하는 번호



CF) 무선 공유기

- ✓ 무선 공유기는 하나의 IP 주소를 여러 기기와 공유 (::IP 주소는 한정적!)
- ✓ 무선 공유기에 연결하고 있는 기기마다 (아이폰1, 아이폰2, 삼성1...) 가상 IP 주소 부여
- ✓ DHCP: 기기가 네트워크에 접속하고자 할 때마다 IP를 동적으로 할당
- ✓ 외부로 데이터 교환 시 (ex- 유튜브 영상 시청), 실제 IP 주소로 감싸고 까는 작업 수행

WEB

: Hypertext (링크 걸기) 접근 위한 시스템

➤ HTTP

- Hypertext 접근 프로토콜
- 웹 서버와 브라우저 사이 정보 교환 방법 (Request, Response)
- TCP/IP 위에서 동작
- Stateless - 접속이 유지되지 않음 => Cookie, Session 사용

클라이언트에서 요청을 보내면 (Request)



서버는 요청을 처리해서 응답 (Response).

➤ HTML

- Hypertext 마킹 위한 언어
- 웹 브라우저를 만드는 회사들이 (Google-Chrome, MS-Explorer) 표준 태그 만들기 위해 W3C에 가입해 있음

➤ URL

- 원하는 정보 자원의 위치와 종류를 나타내는 일련의 규칙

`http://www.mju.ac.kr/sample/index.html`

프로토콜

Host 주소

경로

파일명

➤ Web Server

- HTTP Request (요청) 처리하는 서비스 제공
- 정적/동적 HTML 처리
- 납작한 하드웨어에서 이를 처리하는 소프트웨어

➤ Web Browser

- HTTP Request 전송하고, Response를 받는 매개체
- HTML 렌더링 (text를 화면으로 만들어 줌)



HTTP Request / Response

➤ HTTP Request

- HTTP 클라이언트는 서버에 Request를 전송
- Request Line - Request Method (명령), URL (요청), HTTP 버전 명시
- Request Method : 클라이언트가 서버에 실행을 요구하는 명령
- Body는 비어 있어도 됨



➤ HTTP Response

- HTTP 클라이언트로부터 Request Message를 받은 서버는 해당 Request 처리 후, 그에 따른 결과를 Response Message로 회신
- Status Line - HTTP 버전, Status Code 명시
 - Status Code (응답 코드) : Request 처리한 결과로, 트랜잭션 이해할 수 있는 가장 쉬운 방법 제공
- Body - 요청한 내용에 대한 결과

```
      HTTP 버전   Status Code
      ↓           ↓
Status Line { HTTP/1.1 200 OK }
Response Header { Date: Sun, 16 Sept 2018 14:02:45 GMT
                  Server: Apache Tomcat/1.1
                  Content-Type: text/html;charset=UTF-8
                  Content-Length: 350
                  }
Body {          공백 한 줄
      { <!DOCTYPE html>
        <html>
        ...
        </html>
      }
    }
```

HTTP Request Methods

➤ GET

- 클라이언트가 서버에게 URL이 가리키는 웹 문서의 내용 전송하도록 요청
- 데이터(parameter)가 URL 주소(query string)에 묻어서 감
ex) `www.mypage.com/index.jsp?id=abc1234&pw=123123`
- 전송했던 데이터는 브라우저의 히스토리에 접속했던 주소와 함께 남아 있음 => 보안성에 취약
- 서버의 정보 가지고 올 때 사용 (ex: 게시판 글 조회, 검색)
- 전송할 수 있는 최대 크기 정해져 있음

➤ POST

- 클라이언트가 서버에게 데이터를 전송하도록 함
- 전송되는 데이터가 URL에 묻어 나가지 않고, 전송 객체의 Body 통해 전달됨
- 브라우저에 전달되는 데이터가 남지 않음 => 보안성 강함
- private한 데이터를 서버에 전송할 때 주로 사용 (ex: 비밀번호, 주민등록번호)
- 데이터 양 제한 없음 -> 대량 전송 가능

HTTP Response Status Code

- ✓ **200 OK** : 서버가 Request를 성공적으로 수행했다는 뜻으로, 주로 요청한 웹 페이지를 제공했다는 의미로 사용
- ✓ **400 Bad Request** : 서버가 클라이언트의 에러로 인해 Request를 수행하지 못함
(ex: 클라이언트 에러로는 malformed request syntax, invalid request...)
- ✓ **401 Unauthorized** : 인증이 필요한 페이지를 요청한 경우나 인증이 실패했을 때 사용. 이때 서버는 클라이언트에게 인증 요구 Response를 반환
- ✓ **403 Forbidden** : 서버에 의해 Request가 거부되었음을 뜻한다.
- ✓ **404 Not Found** : 서버가 Request의 URL을 찾을 수 없음.
(ex: 예를 들어 서버에 존재하지 않는 페이지에 대한 요청이 있을 경우)
- ✓ **500 Internal Server Error** : 서버에 Request를 처리할 수 없게 만드는 오류가 발생

HTML

: Hyper Text Markup Language

- ✓ Hyper Text: 링크로 서로 연결되어 있는 문서
- ✓ Markup: Tag 이용하여 문서의 메타 정보를 표현
- ✓ 웹에서 보여질 문서의 내용과 모양 표현하는 정적 페이지 만듦
- ✓ 문서 작업 가능한 프로그램에서 모두 작성 가능 (확장자 .html)
- ✓ HTML 문서는 tag (태그)라는 명령어로 구성되어 있음

<태그이름>문서의 내용</태그이름>



여는 태그



닫는 태그

CF)

<http://www.w3schools.com/tags/default.asp>
<https://developer.mozilla.org/ko/docs/Web/HTML>

HTML Basic

✓ 기본 구조

- `<html>` : root element
- `<head>` : 헤더 - title, meta 정보, script, css 등 포함
- `<body>` : 본문 - 실제 화면에 보여지는 부분

✓ 기본 Tag

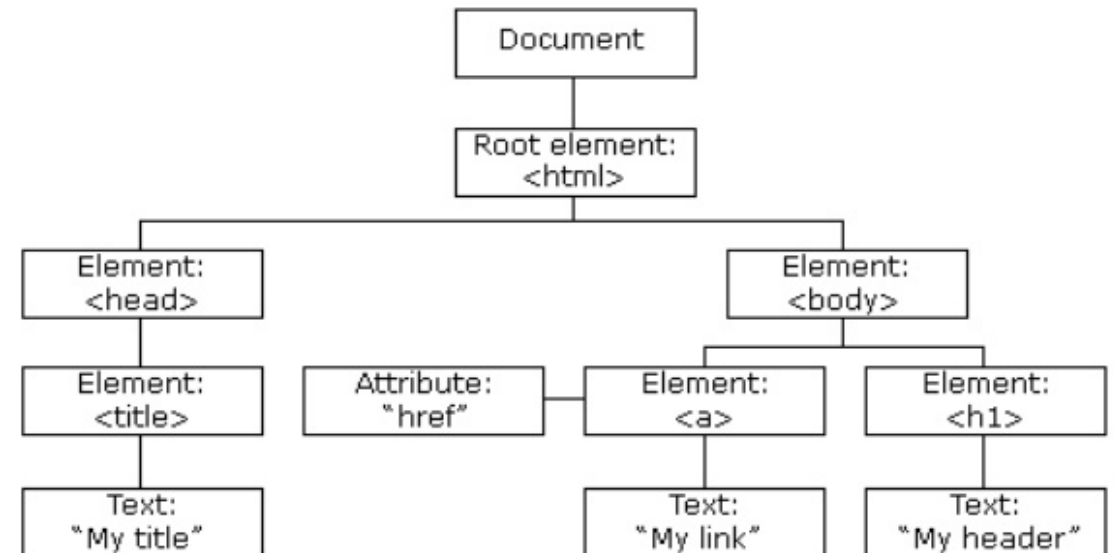
- `<h1></h1>` ~ `<h6></h6>` : header 태그 (1이 가장 크고 6이 가장 작음)
- `<p></p>` : 단락 구분
- `
` : 줄 바꿈
- `<hr>` : 가로 구분선
- `<div></div>` : 영역을 block으로 구분
- `` : 영역을 line으로 구분
- `<!-- -->` : 주석

HTML Element

- ✓ 여는 태그 <>와 닫는 태그</>로 둘러 쌓임
- ✓ 범위가 없으면 바로 닫을 수 있음
- ✓ 내부에 다른 element나 attribute를 중첩하여 포함할 수 있음

```
1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4      <meta charset="UTF-8">
5      <title>My title</title>
6  </head>
7  <body>
8      <a href="www.naver.com">My Link</a>
9      <h1>My Header</h1>
10 </body>
11 </html>
```

Dom tree



HTML Attribute

- ✓ Element의 속성을 지정
- ✓ 이름 = “값” 형태
- ✓ 하위 노드까지 속성값 전달됨
- ✓ Examples
 - id : element의 unique한 id
 - class: 용이하게 style 지정 가능하게 함
 - style: HTML 내부에서 inline style 지정
 - title: 부가 정보

```
1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4      <meta charset="utf-8" />
5      <title>Page Title</title>
6  </head>
7  <body>
8      <h1>My First HTML DOC</h1>
9      <p>
10         첫 번째 단락
11         
12         <span style="color: red">
13             빨간 글씨 속에 <b>두꺼운</b><i>이탤릭</i>글씨 넣기
14         </span>
15     </p>
16     <p>
17         두 번째 단락
18     </p>
19 </body>
20 </html>
```

HTML Form

✓ 데이터를 서버로 전송하기 위하여 사용

<form action="url" method="post/get"></form>

- action : 폼 전송할 url 주소
- method : get/post

```
18 <form action = "" method = "">
19     NAME : <input type = "text"> <br>
20     ID : <input type = "text"> <br>
21     PW : <input type = "password"> <br>
22     PW CHECK : <input type = "password"> <br>
23     INTRO : <textarea rows = "5" cols = "50"></textarea><br>
24     <!-- &nbsp; 는 스페이스바 기능 수행 -->
25     SEX : <input type = "radio" name = "gender"> M &nbsp; &nbsp; &nbsp;
26     <input type = "radio", name = "gender"> F <br>
27     AREA : <select>
28         <option>SEOUL</option>
29         <option>BUSAN</option>
30         <option>JEJU</option>
31         <option>OTHERS</option>
32     </select> <br>
33     HOBBY : <input type = "checkbox" name = "hobby"> BOOKS
34             <input type = "checkbox" name = "hobby"> MUSIC
35             <input type = "checkbox" name = "hobby"> MOVIES
36             <input type = "checkbox" name = "hobby"> OTHERS <br>
37     <!-- 버튼 안에 들어갈 문자는 value 속성에 적음 -->
38     <input type = "submit" value = "OK">
39     <input type = "reset" value = "CANCEL">
```

✓ **input tag** : 데이터 입력 위해 사용하는 태그로, 속성으로 태그 종류 지정

✓ **type** : 태그의 종류 지정

- text : 일반적인 text 데이터를 입력
- password : 로그인, 회원가입 페이지 등에서 비밀번호 입력
- checkbox : 데이터 값을 여러 개 전송
- radio : 여러 개의 데이터 값 중에 한 개의 값만 전송
- select : 리스트 형태의 데이터
- textarea : 여러 줄의 텍스트를 입력

속성 cols와 rows로 행과 열의 크기를 지정

- file : 파일 업로드 위해 사용
- button : 버튼 만들기 위해 사용

속성으로 value를 지정하여 버튼 안에 들어갈 문자열 입력

- submit : form 데이터를 서버로 전송
- reset : form 데이터 초기화