

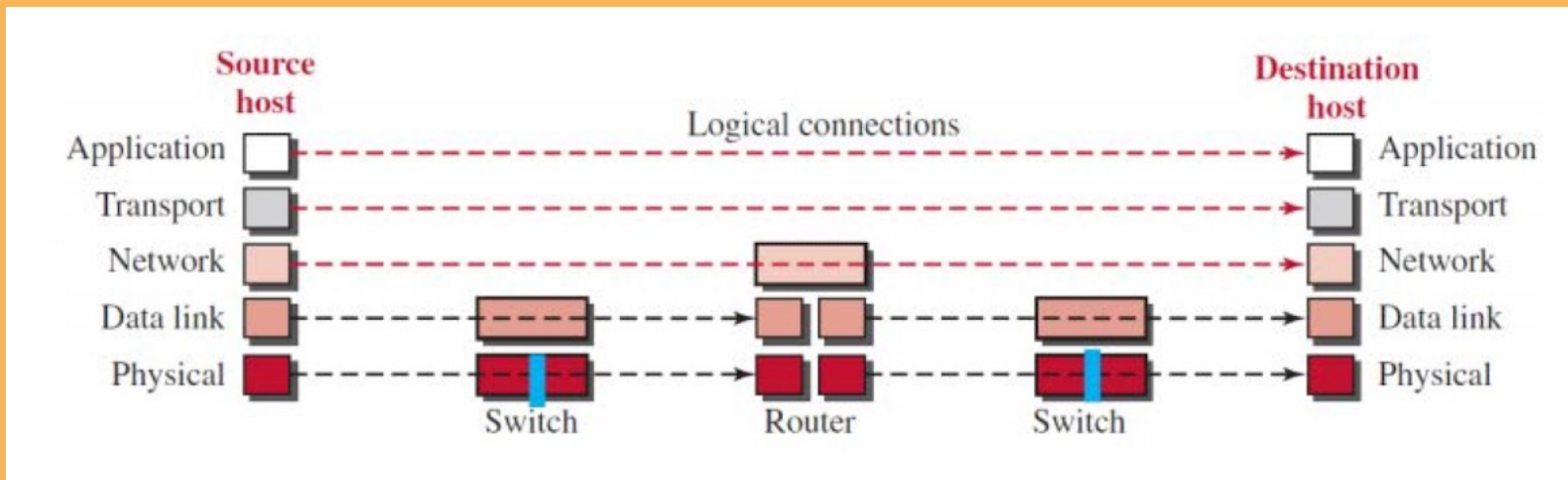
DEVKOR

Backend Study

Internet

개요

- 인터넷은 가장 유명한 컴퓨터 네트워크
- 기업, 기관 등의 조직이 각자의 네트워크를 구성
- 인터넷은 이런 네트워크를 잇는 네트워크들의 네트워크.
- IP = Internet Protocol을 통해 네트워크가 이어짐
- 우리가 사용하는 HTTP 등의 통신 기술(규약) 역시 runs on IP
- HTTP를 이용한 웹 기술은 가장 유명한 Internet Application 사례



WWW

- application layer -> 네트워크 간의 정보 전달을 활용하는 기술
- WEB = 거미줄. hyper text : 한 문서에서 다른 문서로 접근할 수 있는 텍스트
- hyper text를 이용하여 거미줄처럼 짜여진 웹을 돌아다니며 정보를 얻을 수 있음
- 웹 문서 = 웹 페이지를 어떻게 주고받을 것인가?
 - > HTTP : Hyper Text Transfer Protocol
 - 사용자와 웹 서버가 웹 페이지를 주고받는 기술
 - connectionless-> 그저 유저가 서버에 요청을 하면, 서버는 응답을 보내는 것

HTTP

- HTTP : 요청과 응답으로 나뉨
- Client - Server Protocol, Client는 요청을, Server는 응답을 보냄
- 요청이 포함하는 것
 - HTTP version
 - URL
 - HTTP method
 - header
 - body (optional)

<https://www.cloudflare.com/en-gb/learning/ddos/glossary/hypertext-transfer-protocol-http/>

HTTP

- Client : user agent. (almost 웹 브라우저)
- 웹 브라우저는 하이퍼링크를 통해 서버에 요청한 웹 페이지를 보여줌
- 웹 페이지는 HTML, CSS, Javascript로 이루어짐
 - HTML = HyperText Markup Language
 - 웹 페이지의 구조를 서술하는 언어
 - not 프로그래밍 언어
 - CSS = Cascading Style Sheet
 - HTML로 표시되는 요소들을 꾸미는 언어
 - Javascript
 - HTML, CSS 요소들을 바꿀 수 있는, 프로그래밍 언어

URL, DNS

- URL = Uniform Resource Locator
- 컴퓨터 네트워크 상에서 리소스 = 자원(웹 페이지 등)이 어디에 있는지 나타내는 주소
- URL을 통해 웹 사이트에 접속 가능
 - == 즉, 서버에 HTTP 요청을 보낼 수 있음
- 주소 = ip 주소 (+ 프로토콜, 포트 등..)
 - www.naver.com
 - = 223.130.200.104
 - DNS . Domain Name System 을 통하여 길고 외우기 어려운 ip주소 대신, 쉬운 도메인 이름을 사용

HTTP Method

- 요청을 보낸 서버에 바라는 행동
- GET, POST, PUT, DELETE ...
- 그 중에서도 GET, POST를 많이 쓴다

- GET
 - request data (요청 body는 담길 수 없음)
 - POST
 - 데이터 쓰기
 - PUT
 - 데이터 수정
 - DELETE
 - 데이터 지우기
- (PUT, DELETE도 잘 쓰지 않음)

보통 처음 만드는 서버의 기능들,, CRUD Create, Read, Update, Delete

HTTP headers

- key - value (이름과 값) pair의 형태로 정보를 나타냄
- method, user-agent, host, cookie 등 잡다한 정보 들이 전부 표시됨
- 개발자도구 네트워크로 확인해보아요

Request Headers

```
:authority: www.google.com
:method: GET
:path: /
:scheme: https
accept: text/html
accept-encoding: gzip, deflate, br
accept-language: en-US,en;q=0.9
upgrade-insecure-requests: 1
user-agent: Mozilla/5.0
```


HTTP body

- request 로보내고자 하는 정보
- data. password+id , ... etc

HTTP response

- 서버가 요청에 대한 응답으로서 클라이언트에게 보내줌
- status code
 - 404, 200 등 서버에러 , 성공 여부 등을 나타내는 코드번호
 - <https://developer.mozilla.org/ko/docs/Web/HTTP/Status>
- response header
 - request와 거의 비슷함
- body (option)
 - 요청했던 정보가 담겨서 들어옴

1. 1xx Informational
2. 2xx Success
3. 3xx Redirection
4. 4xx Client Error
5. 5xx Server Error

The "xx" refers to different numbers between 00 and 99.

web application

요청과 응답을 통해 데이터를 클라이언트와 서버가 데이터를 교환하는 애플리케이션

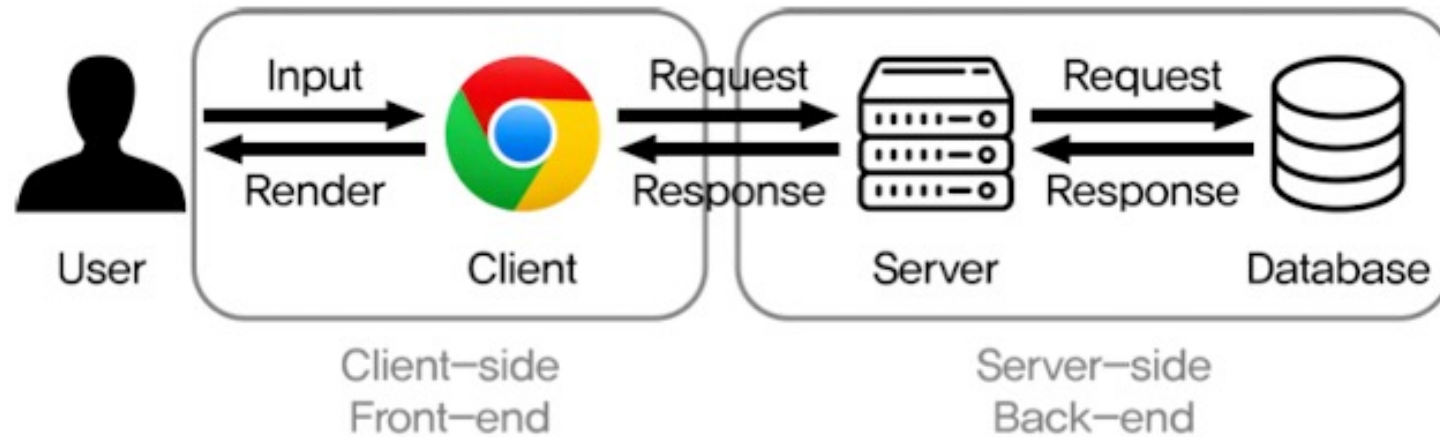


Figure 1.1 How Typical Web Application Works

HTTPS?

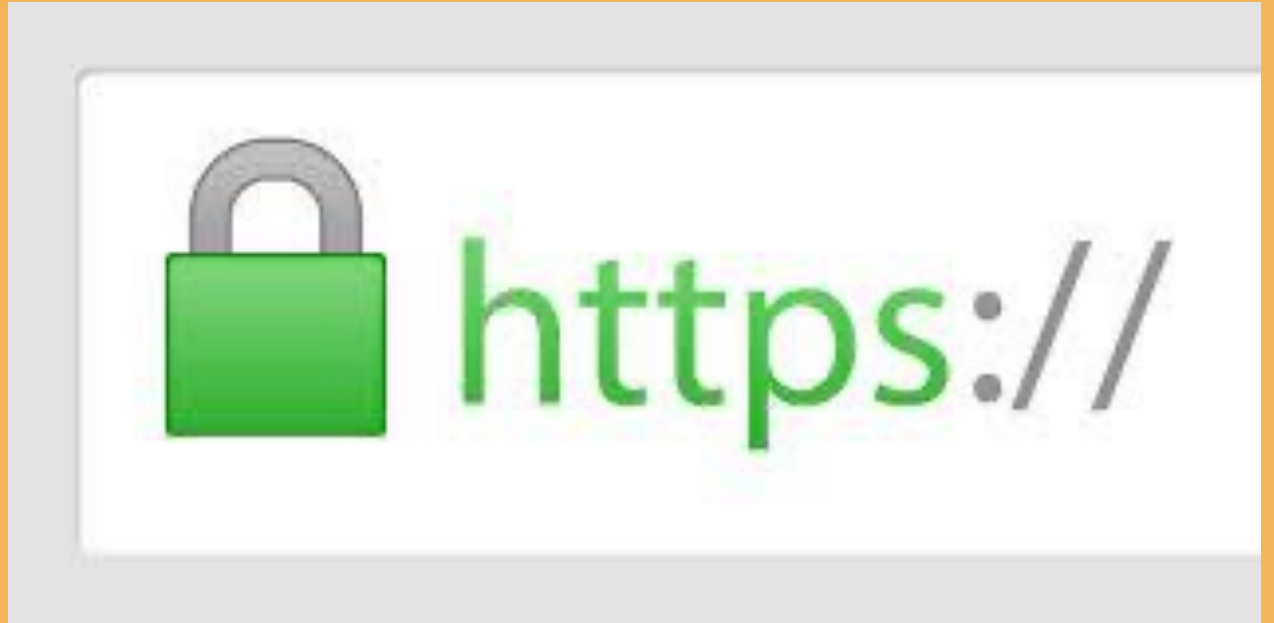
HTTP + TLS

HTTP on TLS : 암호화된 TCP 연결

= Secure Sockets Layer = Transport Layer Security

인터넷 패킷은 모두가 볼 수 있으므로..

제 3자가 그것을 가로채어 수정하지 못하도록 함



<https://www.google.com>에
접속할 때 일어나는 일

개발자 도구로 확인하기

Insomnia

브라우저와 같은 user-agent.

브라우저는 웹 서핑에 특화되어있지만, 그와 다르게
개발할 때, 서버를 테스트할 때 유용하도록

HTTP 요청을 수정하고 응답을 확인하기 쉽게 만들어진 프로그램

<https://insomnia.rest/download>

서버를 만들고 테스트할 때 쓸 예정

간단히 프론트엔드 배우기

HTML

- HyperText Markup Language
- 요소-element로 구성됨.
- 브라우저가 어떻게 페이지를 렌더링하면 되는지 묘사

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

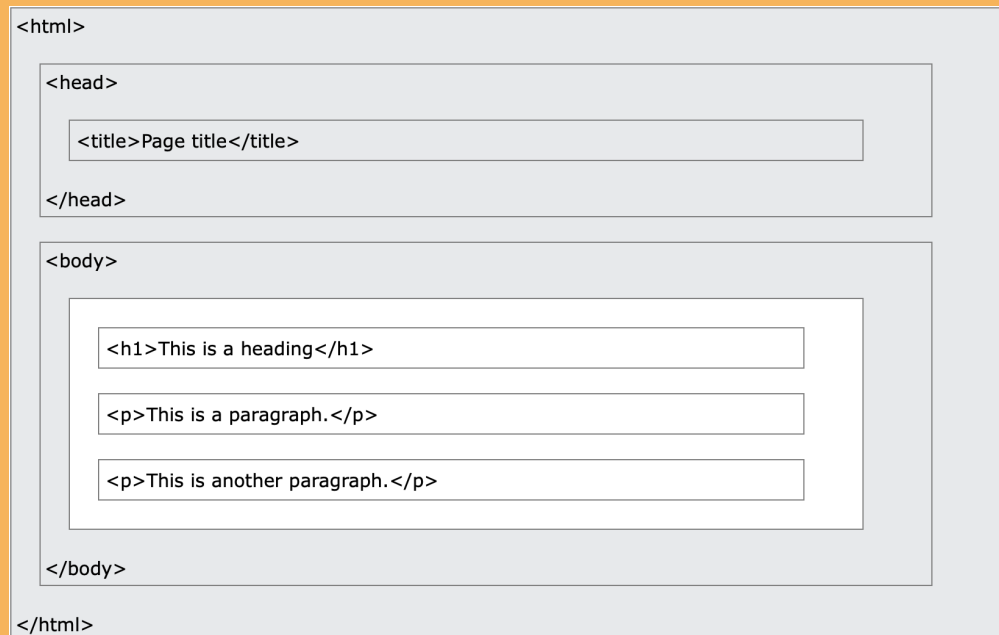

간단히 프론트엔드 배우기

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

- The `<!DOCTYPE html>` declaration defines that this document is an HTML5 document
- The `<html>` element is the root element of an HTML page
- The `<head>` element contains meta information about the HTML page
- The `<title>` element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
- The `<body>` element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- The `<h1>` element defines a large heading
- The `<p>` element defines a paragraph



간단히 프론트엔드 배우기

CSS

- Cascading Style Sheet
- HTML로 표현된 구조가, 실제로 어떻게 표시되는지 기술
- 각 HTML 요소 (태그, 클래스, id로 선택)의 스타일을 정의

```
p {  
  font-size: 110%;  
  font-family: garamond, sans-serif;  
}  
h2 {  
  color: red;  
  background: white;  
}  
.highlight {  
  color: red;  
  background: yellow;  
  font-weight: bold;  
}  
#test_id {  
  color: blue;  
  background: white;  
}
```

간단히 프론트엔드 배우기

JavaScript

- HTML 요소를 바꿀 수 있는 프로그래밍 언어
- 브라우저 (프론트엔드) 단에서 좀 더 계산적인 문제들을 풀 수 있도록 개발
 - 계산 문제 = 컴퓨터가 풀 수 있는 문제들 -> if, for문 등 ..
 - HTML요소에 따른 분기, 반복 등등등이 가능하도록
 - 구조 서술과 스타일에 그치지 않고, 복잡한 일을 하도록

https://www.w3schools.com/js/tryit.asp?filename=tryjs_intro_lightbulb

- 즉, 브라우저에서 돌아가는 프로그래밍 언어 !
 - 웹 브라우저의 javascript 엔진에서 돌아가는 언어

프론트와 백엔드

HTML, CSS, JS 문서를 HTTP 프로토콜로 주고받는 것은 맞지만,
이것은 프론트엔드 배포의 영역 !

백엔드는 프론트엔드에서 다루는 데이터들을 저장 (DB),
처리하여 프론트엔드단으로 보내주는 로직을 만듦!

Node JS

브라우저 상에서 돌아가던 JS는 프론트엔드에서만 쓰이던 언어였음

그러나 크롬의 자바스크립트 엔진인 V8로 빌드된 자바스크립트 런타임
즉, 브라우저 밖에서 JS 코드를 실행할 수 있게 해주는
Node.JS의 등장으로 자바스크립트로 서버를 구축하는 등 여러 활용이 가능

입출력에 대한 성능, 확장 용이성, 쉬운 접근성으로 점유율이 높음

과제

Node.js 설치해오기!

<https://nodejs.org/ko/>