

Intro | 프로그래밍 잘 배우기

# 프로그래밍 '잘하는 방법' 배우기

# 진행하면서 수정해나가기

- Ver 0.0.0.1 만들고 업데이트 하기 / 처음부터 완벽하게 🧑‍💻
- 구현하는 과정에서 새롭게 얻은 정보 사용해 버전 업
- 가장 중요한 핵심은 무엇일까?
- Baby Step 점진적으로 / 잘 되는 방법으로
- 지금 상태를 개선해나가기



by ohahohah.dev

## 1. 레고 조립하기

큰 모듈을 작은 모듈로. 모듈을 조립하기

## 2. 에러 마음껏 내보기

에러를 고치는 것은 프로그래밍의 일부! 고치면 됩니다!

## 3. 질문하고 답변하기

바보 같은 질문은 없습니다. 나의 지식을 **우리** 지식으로!

## 4. 내 지식으로 만들기

내가 하고 싶은 프로젝트에 적용하기



## 강의 | 프로그래밍 잘 배우기



Fritz Grobe and Stephen Voltz perform the "Diet Coke and Mentos Experiment"

## 목표 | 클라우드에서 배포



Photo by Nick Seagrave on Unsplash

# Programming

- 하나 이상의 추상 알고리즘을  
특정한 프로그래밍 언어를 이용해 구체적인 컴퓨터 프로그램으로 구현하는 기술\*
- 문제를 해결하기 위해 알고리즘과 프로그래밍 언어를 사용해 프로그램을 만드는 것

---

- \* 컴퓨터 프로그래밍. Wikipedia 한국. Retrieved 17 October 2020. [https://ko.wikipedia.org/wiki/%EC%BB%B4%ED%93%A8%ED%84%B0\\_%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D](https://ko.wikipedia.org/wiki/%EC%BB%B4%ED%93%A8%ED%84%B0_%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D)

# Keyword Map

- Cloud
- Deployment
- IP, Domain
- CDN
- DNS, NameServer
- Nginx 웹 서버
- HTTPS
- Firewall(WAF)
- SQLInjection

# 웹 서비스 개념

---

강사 오시영



by ohahohah.dev

# 웹 사이트에 접속한다는 것?

- URL (예. [www.naver.com](http://www.naver.com))  
접속하면 어디서나 같은 웹 페이지  
보여짐.
- 서버 Server 가 역할을 하고 있다!



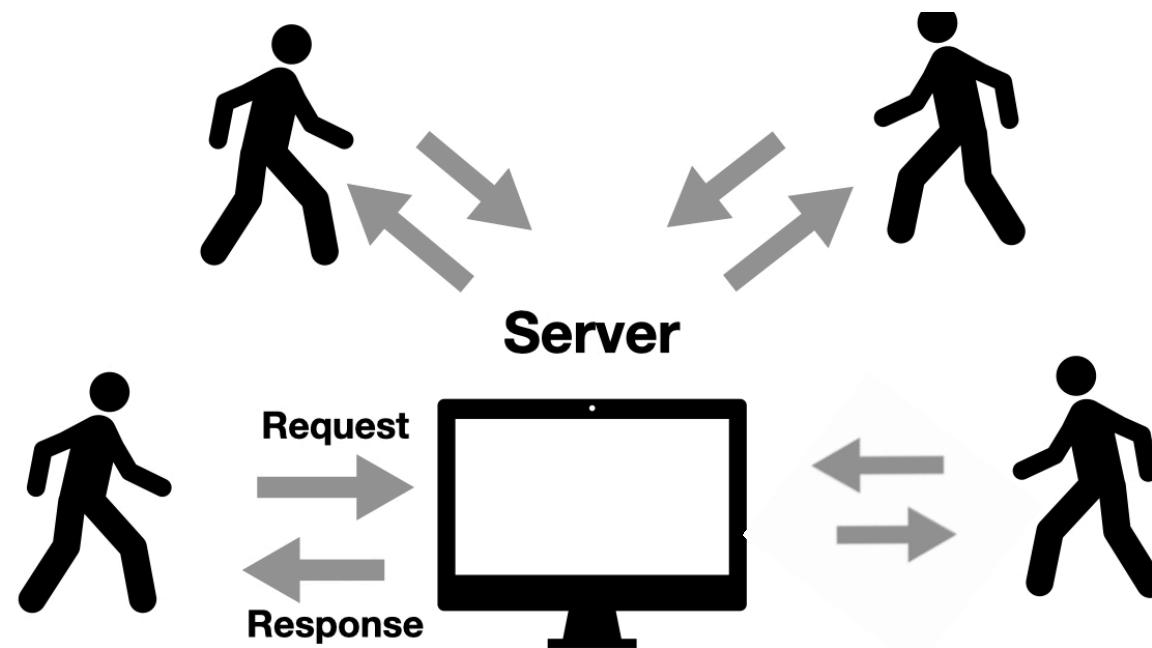
Photo by [imgix](#) on [Unsplash](#)

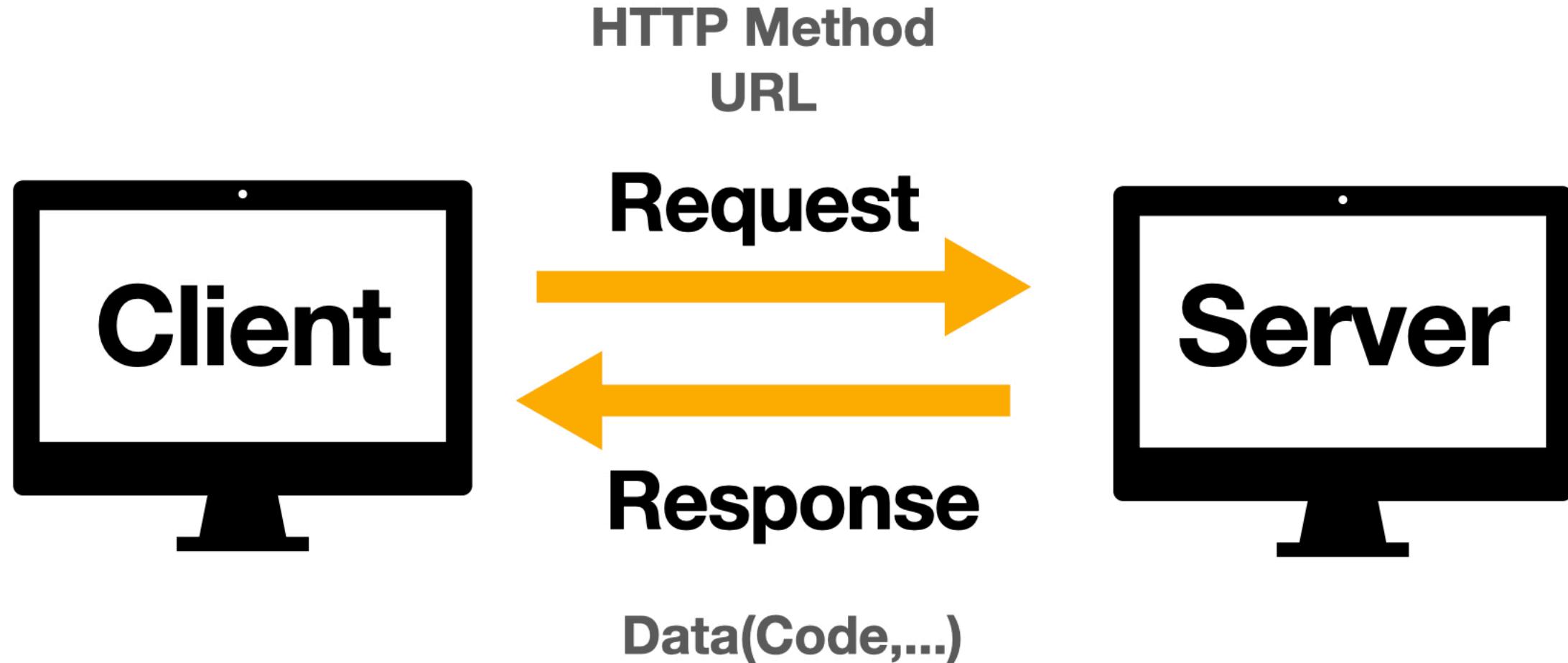


by ohahohah.dev

## 웹 사이트에 접속한다는 것?

- URL (예. [www.naver.com](http://www.naver.com)) 로 접속하면 어디서나 같은 웹 페이지(코드)를 보내줌.





# URL

**https://www.naver.com**

**HyperText** 하이パーテ스트

**Transfer** 전송

**Protocol** 규약

+

**Secure** 보안



by ohahohah.dev

## GET <https://www.naver.com>

- **HTTP** 라는 통신 규약을 따른다
- **HTTP Request Method** : HTTP 요청할 때 원하는 작업을 나타냄
- SW 기능 개발 기본 단위 - CRUD

**Create** → POST

**Read** → GET / 웹 페이지 요청 (웹 브라우저 주소창 사용)

**Update** → PUT

**Delete** → DELETE

- **API**에 따라 적절하게 사용함





Flask 사용

Frontend 코드 (HTML, CSS, JavaScript)

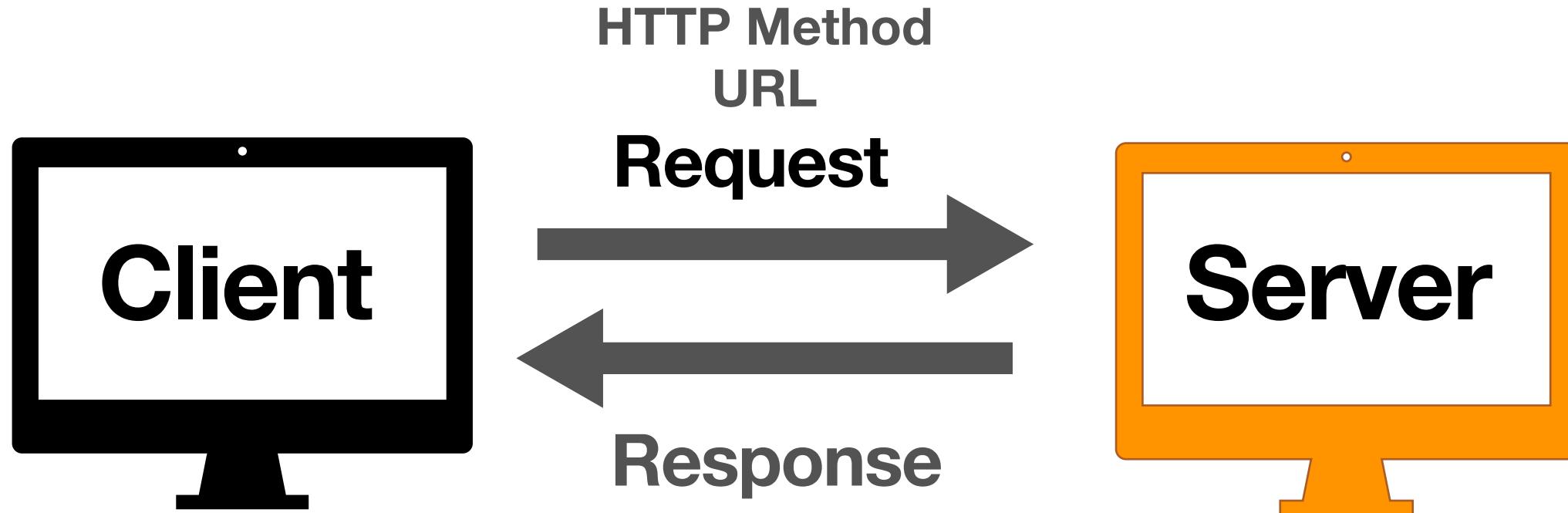
API 를 사용하는 코드

Backend 코드

- API 로 동작하는 코드

- 데이터를 관리하는 코드 (수집, 저장,...)

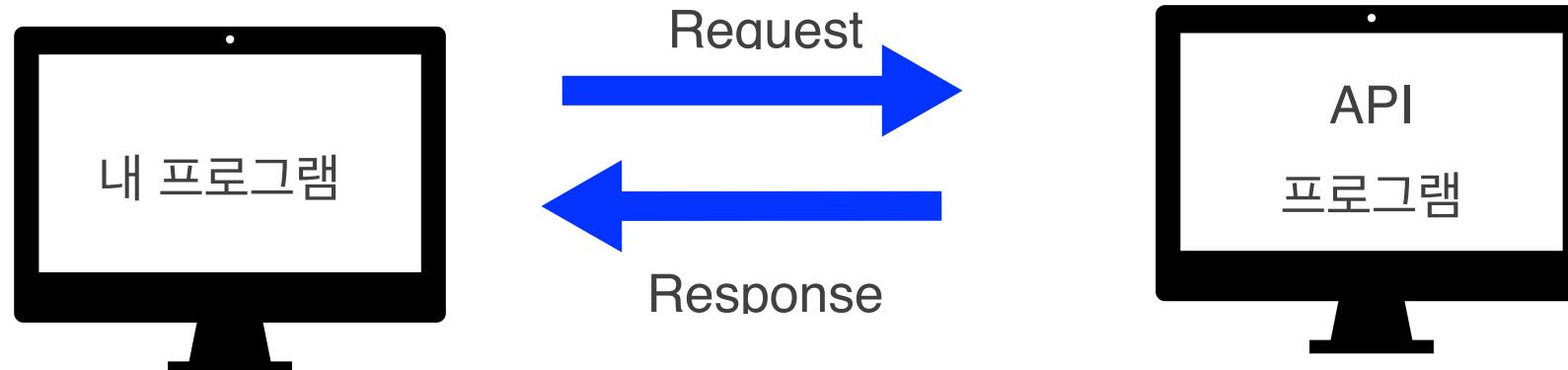
- 데이터를 가공하는 코드



요청에 맞게 실행한 후 결과를 보내줌  
Data(화면 Code / 데이터 JSON,... )

# API

- HTTP API 는 API 중 한 종류
- UI (User **Interface**) : User 가 Program 을 사용하기 위한 인터페이스(화면으로  
상호작용)
- API (Application Programming Interface)  
하나의 program 이 다른 program(기능) 을 사용할 수 있게 하는 Interface



# API 는 contract!

- API (Application Programming Interface)

특정 기능을 사용하기 위해서 꼭 지켜야하는 약속(contract)

설계된 약속을 기록해둔 것- API 문서(Documentation, Doc), Spec

A. 기능

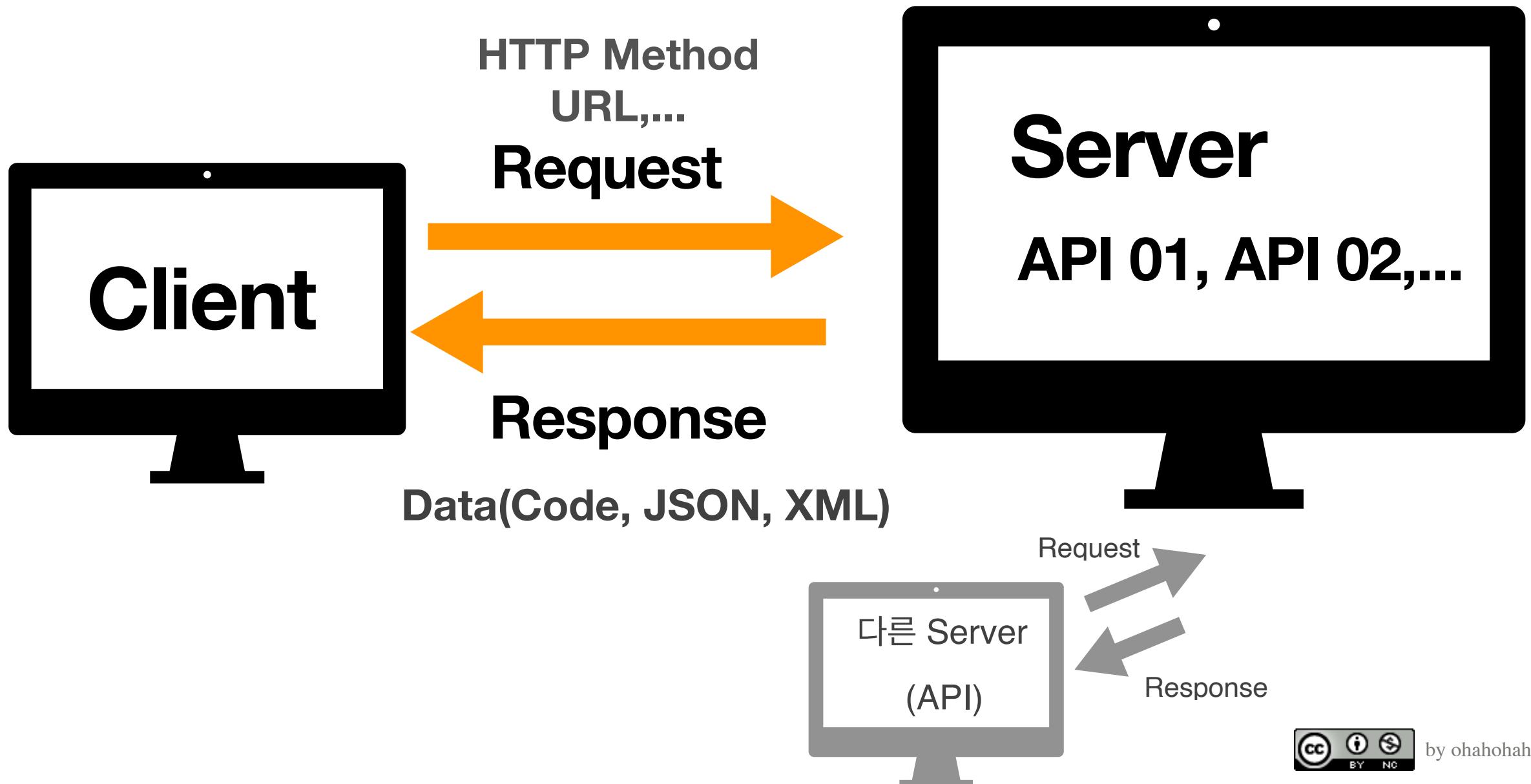
- 특정 데이터를 가져오기

B. 정해진대로 요청하고 - 특정 URL, 특정 Method, 인증키 Authentication, 데이터

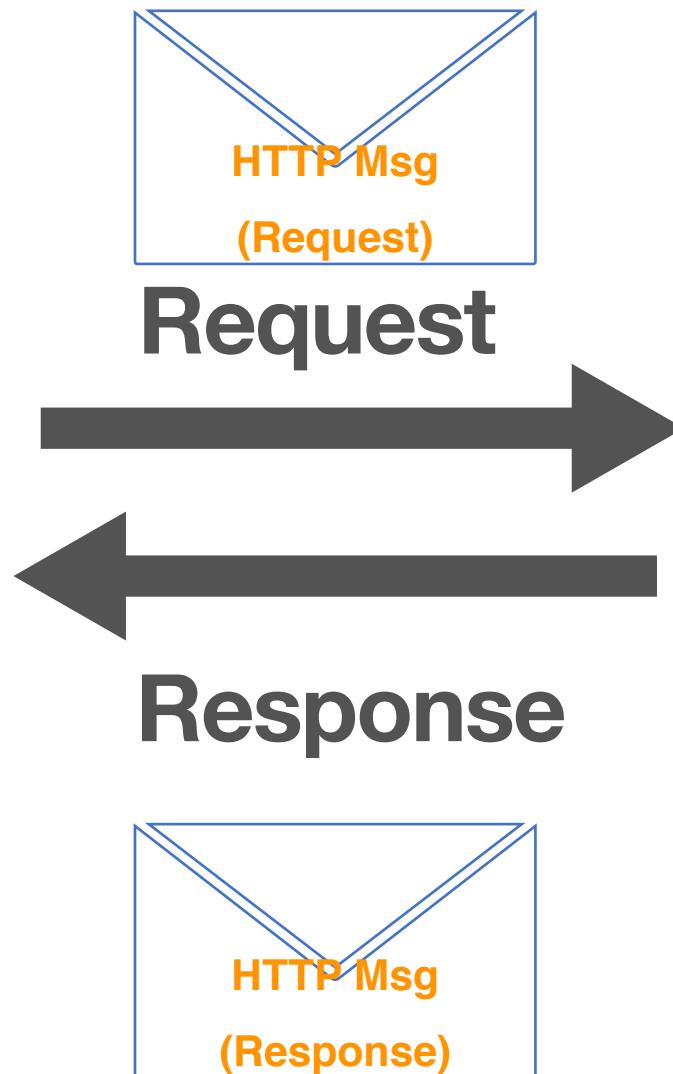
C. 정해진 응답결과를 받음 - JSON, 항목과 같은 데이터 형식



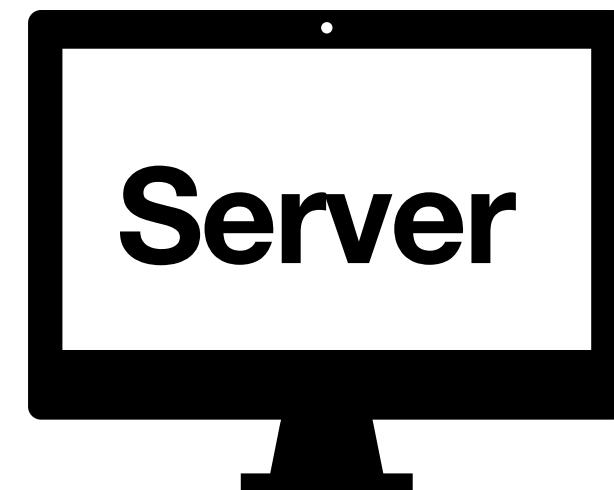
by ohahohah.dev



**IP Address \***  
(e.g,102.141.75.30)

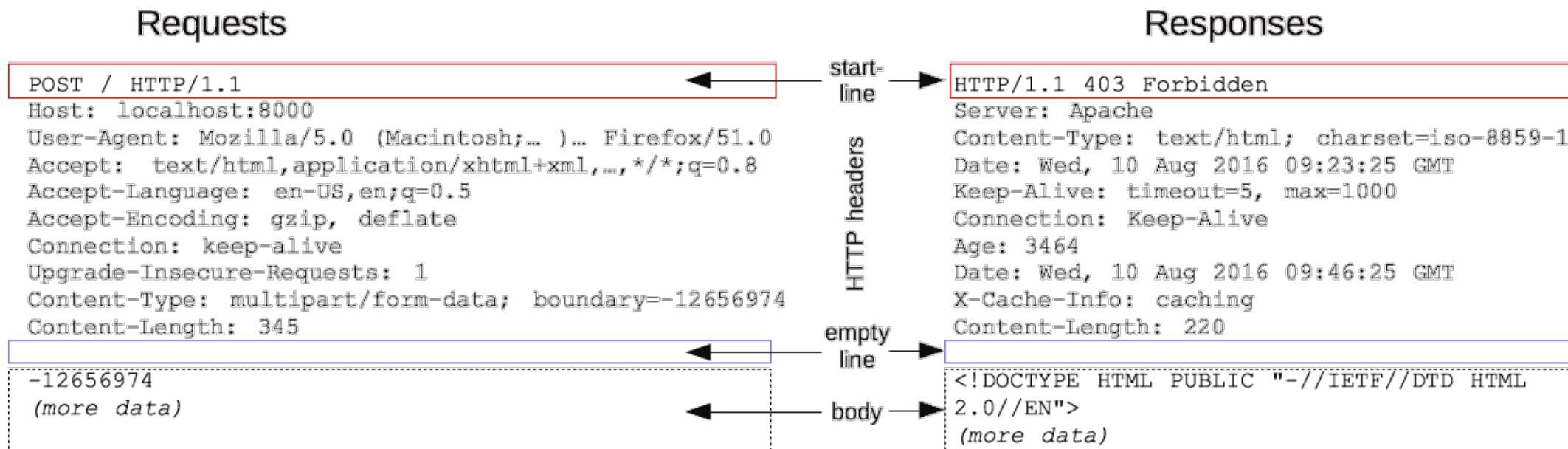


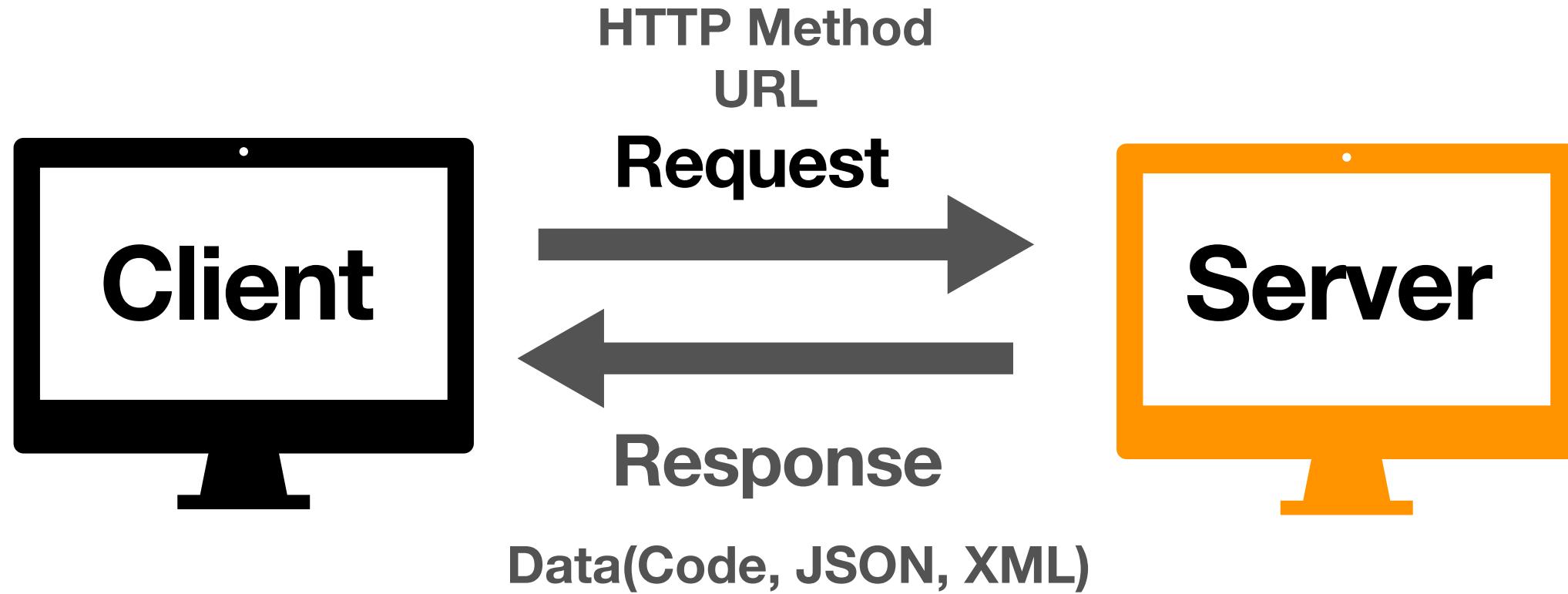
**IP Address \***  
(e.g,112.31.25.174)



# HTTP Message

- Request-Response 과정에서 교환되는 형식을 가진 데이터(Text)
- Request Msg: 요청 URL / Header: client 정보(Host 주소, 사용환경-브라우저 외),... / Body :Request Data
- Response Msg : 요청 Status(결과) / Header: Server 정보,... / Body: Response Data





## 강의 | Cloud 이해하기

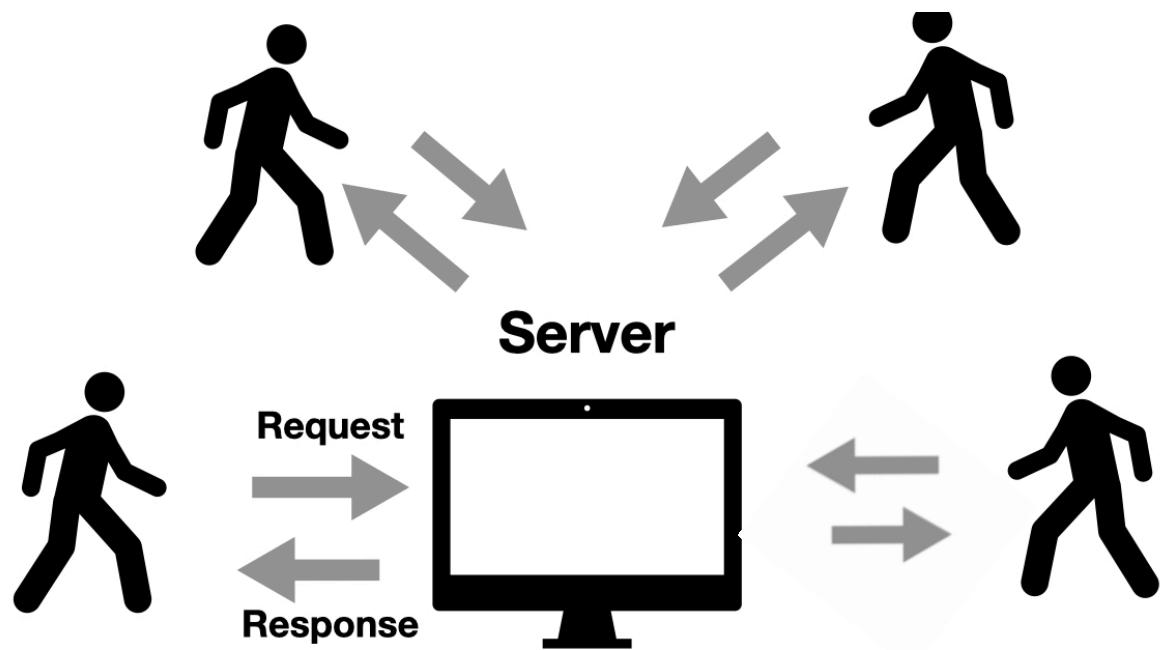


Photo by [imgix](#) on [Unsplash](#)

직접 물리적인 컴퓨터를 세팅하거나  
관리되는 원격 컴퓨터를 사용하거나

# Cloud

- Datacenter(IDC) 에서 **Cloud** 로.
- 외부에서도 데이터에 접근하고 서버를 관리할 수 있다면?  
공간의 제약이 사라짐.
- 클라우드는 인터넷을 통해 컴퓨터의 리소스를 사용하는 것.  
여기서 컴퓨터의 리소스는 컴퓨터를 이루고 있는 메모리, 저장장치(하드디스크, SSD), CPU 등
- 예. 외장하드 대신 네이버 클라우드에 파일을 저장하고 읽어오기  
= 인터넷을 통해 저장장치에 접근하는 것 -> 클라우드!

# Cloud

- 물리적 컴퓨터를 관리하지 않더라도 인터넷으로 편하게 사용할 수 있다!
- 미국 국립표준연구소(NIST) 가 정의한 클라우드 컴퓨팅 특징
  - A. Essential Characteristics
  - B. Service Models
  - C. Deployment Models

# Cloud

## A. Essential Characteristics

- On-demand self-service: consumer가 컴퓨팅 자원을 요구하는 즉시 자동으로 제공
- Broad newtwork Access: 어디 있든 인터넷을 통해 리소스에 액세스
- Resource pooling: provider는 리소스 풀을 확보해서 multi-tenant model로 제공. 규모의 경제. 고객은 리소스 위치에 대해 신경쓸 필요없음.
- Rapid elasticity: 탄력적으로 리소스를 줄이거나 늘릴 수 있음(scale up and down)
- Measured service: 리소스 사용량이 측정되어서 쓴 만큼만 지불함. 투명성, 리소스 모니터링, 제어 및 보고 가능

# Cloud

## B. Service Models

- Software as a Service (SaaS) : 소프트웨어처럼 바로 사용할 수 있음.  
예를 들면, MS 오피스 365, 구글 클라우드, 네이버 클라우드
- Function as a Service (FaaS): 함수 단위 등으로 실행  
예를 들면, Lambda
- Platform as a Service (PaaS) : 응용 프로그램(Application)을 작성하고 실행할 수 있는 환경을 제공하는 것.  
예를 들면, Google App Engine, Heroku
- Infrastructure as a Service (IaaS) : 사용할 수 있는 인프라를 제공하는 것.
- Everything as a Service (EaaS)

# More.Cloud

## C. Deployment Models

- Private cloud : 비공개 클라우드. 단일 조직에서만 독점적으로 사용하는 것.
- Community cloud : 특정 용도로만 제한된 조직에서만 사용하는 것.
- Public cloud : 공개 클라우드. 일반인이 공개적으로 사용할 수 있는 것. 대부분의 클라우드 서비스
- Hybrid cloud : 두 종류 이상의 클라우드로 구성된 것.

# Cloud

- 가상화
  - 물리적 장치가 아닌 소프트웨어로 리소스를 사용가능하게 함.
  - 리소스를 탄력적으로 설정할 수 있고, 이중화 등에 유리
- 분산처리
  - 트래픽 부하 분산처리 등

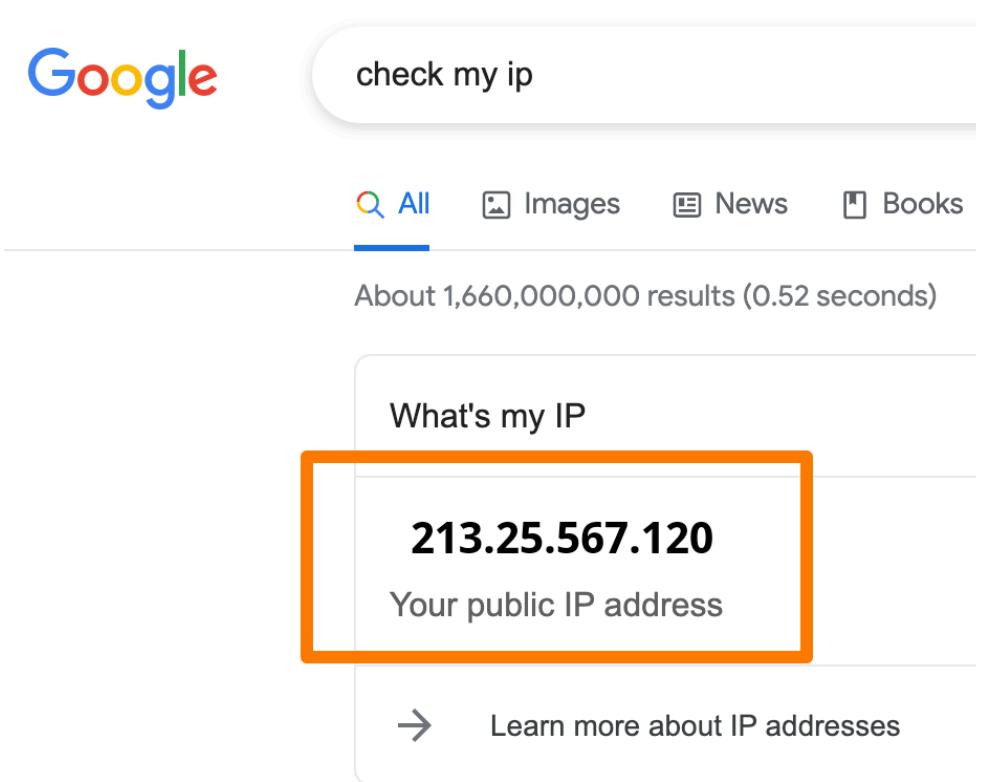
# 네트워크

---

강사 오시영

# IP Address

- IP(Internet Protocol) Address. 줄여서 IP
- 인터넷 사용이 가능한 기기(LAN card)마다 IP 있음.
- 구글에 [check my ip 검색](#)
- URL 은 IP 를 사람이 알아보기 쉽게 영어로 나타낸 것  
(전화번호부 저장 이름 - 전화번호 관계와 유사)
  - <https://google.com> == <https://172.217.175.14>
  - <https://google.com> == <https://142.251.2.101>



# Port

- 통신의 종단점 End Point
- 하나의 IP 주소에 여러 port 로 서비스를 할 수 있음.
- HTTP 의 기본 port : 80
  - <http://000.000.000.000:80>
  - 웹 브라우저에서는 자동으로 '80' 을 붙여줌.
  - 굳이 '주소:80' 이라고 적지 않아도 됨!



by ohahohah.dev

# DNS / 네임 서버

- Name Server - IP 주소가 특정 URL 과 연결됨을 알려주는 서버!
- Domain Name System - 도메인 이름을 관리하는 시스템

IP  
**172.217.161**



Domain  
**google.com**



by ohahohah.dev

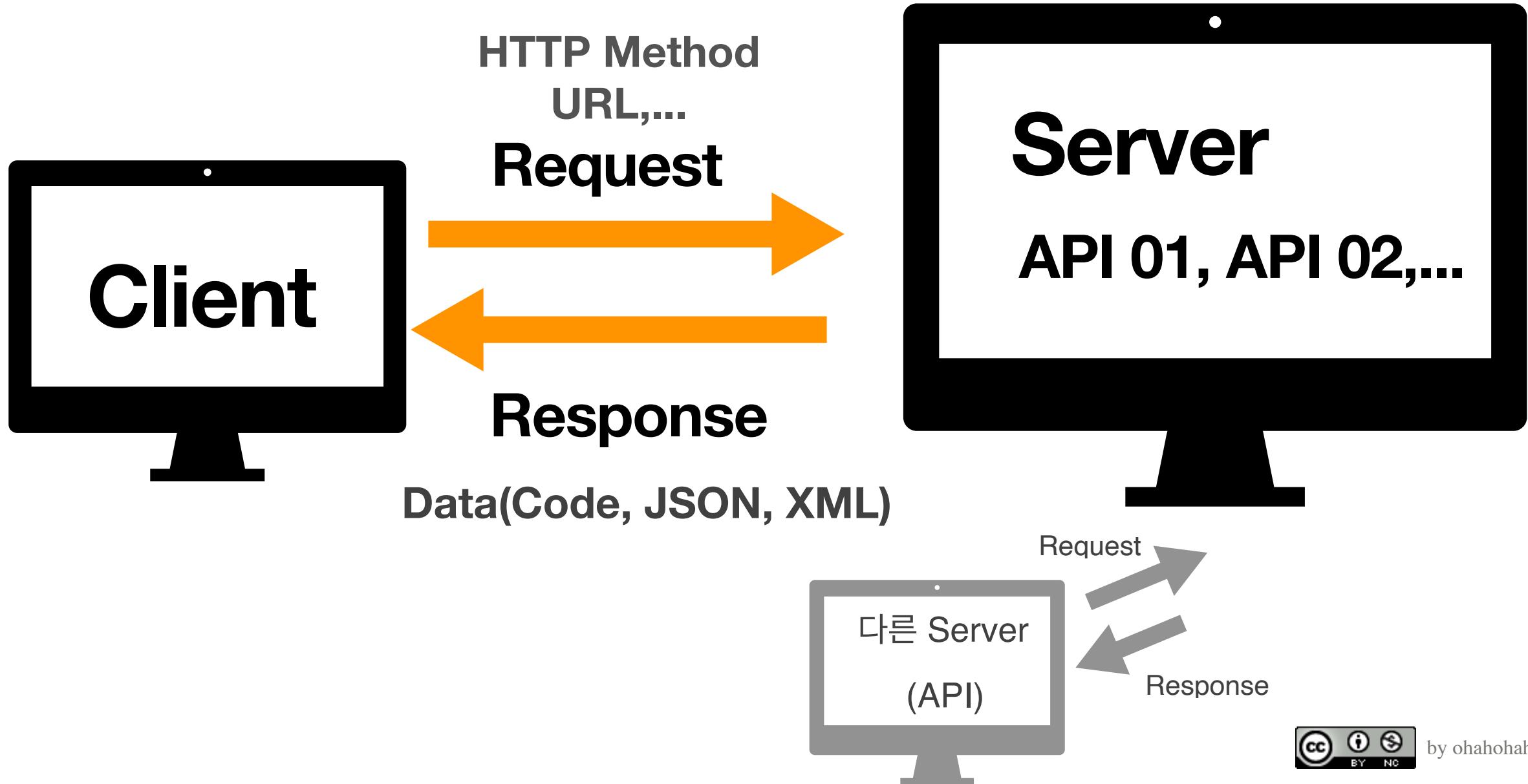
# 사전 체크

- [Windows 만] Git bash 설치되어있는지 다시 한번 확인하기
  - Server 접속하고 명령어 내릴 때 사용하는 도구
- Filezilla 다운로드 후 설치하기 - FTP 쉽게 사용하는 프로그램
  - <https://filezilla-project.org/download.php>
- Server 설정할 때 사용했던 keypair(\*.pem) 위치 확인하기
- [gabia.com](http://gabia.com) 로그인하기 - 유료 도메인 확인용

# DB 어떻게 사용할까?

---

강사 오시영



## Data를 관리하자

- 내 프로젝트에 API 결과 가져다쓰기
- 변하지 않는 정보를 매번 접속해서 가져오면 시간이 오래 걸리지 않나?
- 사용자가 보낸 정보를 저장하거나 수정하려면?
- **CRUD** (Create, Read, Update, Delete) 모두 가능!
- 엑셀에 데이터 저장해두고 꺼내쓰는 것처럼!



강의 | DB 가 무엇일까?

# DataBase DB

- CRUD (Create, Read, Update, Delete)
- Structured data (정형데이터), Unstructured data(비정형데이터)

Unstructured data	Semi-structured data	Structured data																								
<pre>The university has 5600 students. John's ID is number 1, he is 18 years old and already holds a B.Sc. degree. David's ID is number 2, he is 31 years old and holds a Ph.D. degree. Robert's ID is number 3, he is 51 years old and also holds the same degree as David, a Ph.D. degree.</pre>	<pre>&lt;University&gt; &lt;Student ID="1"&gt; &lt;Name&gt;John&lt;/Name&gt; &lt;Age&gt;18&lt;/Age&gt; &lt;Degree&gt;B.Sc.&lt;/Degree&gt; &lt;/Student&gt; &lt;Student ID="2"&gt; &lt;Name&gt;David&lt;/Name&gt; &lt;Age&gt;31&lt;/Age&gt; &lt;Degree&gt;Ph.D. &lt;/Degree&gt; &lt;/Student&gt; .... &lt;/University&gt;</pre>	<table border="1"><thead><tr><th>ID</th><th>Name</th><th>Age</th><th>Degree</th></tr></thead><tbody><tr><td>1</td><td>John</td><td>18</td><td>B.Sc.</td></tr><tr><td>2</td><td>David</td><td>31</td><td>Ph.D.</td></tr><tr><td>3</td><td>Robert</td><td>51</td><td>Ph.D.</td></tr><tr><td>4</td><td>Rick</td><td>26</td><td>M.Sc.</td></tr><tr><td>5</td><td>Michael</td><td>19</td><td>B.Sc.</td></tr></tbody></table>	ID	Name	Age	Degree	1	John	18	B.Sc.	2	David	31	Ph.D.	3	Robert	51	Ph.D.	4	Rick	26	M.Sc.	5	Michael	19	B.Sc.
ID	Name	Age	Degree																							
1	John	18	B.Sc.																							
2	David	31	Ph.D.																							
3	Robert	51	Ph.D.																							
4	Rick	26	M.Sc.																							
5	Michael	19	B.Sc.																							

- 대표적인 DB 종류 : RDB(Relational Database), No SQL

# Keyword Map

- Cloud
- Deployment
- IP, Domain
- CDN
- DNS, NameServer
- Nginx 웹 서버
- HTTPS
- Firewall(WAF)
- SQLInjection

# Extreme Programming

지금 상황과 상관없이 여러분은 언제나 더 나아질 수 있다.

더 나아지는 일은 언제나 스스로부터 시작할 수 있다.

더 나아지는 일은 언제나 오늘부터 시작할 수 있다.

No matter the circumstance you can always improve.

You can always start improving with yourself.

You can always start improving today.