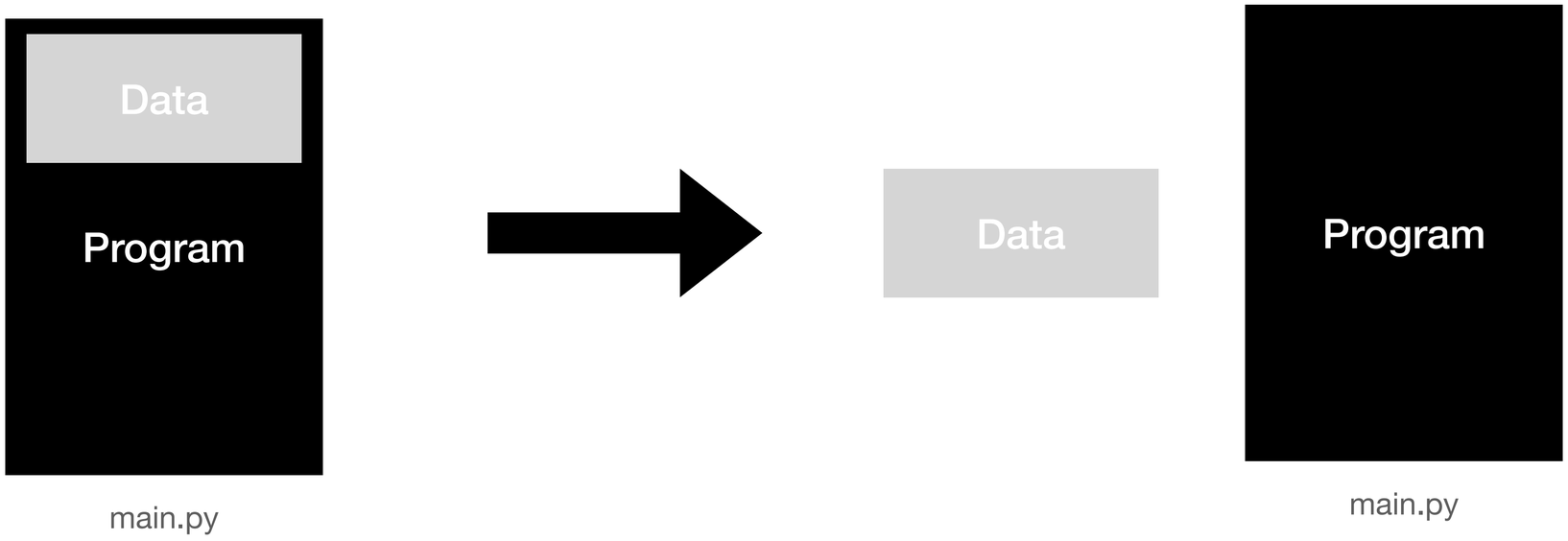
Big Data Analytics Programming

Week-06. Data

Jungwon Seo, 2021-Spring

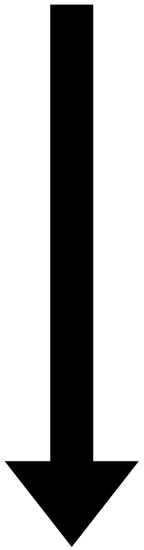
Back in the day..
데이터는 프로그램에 종속 되어있다



배울 내용

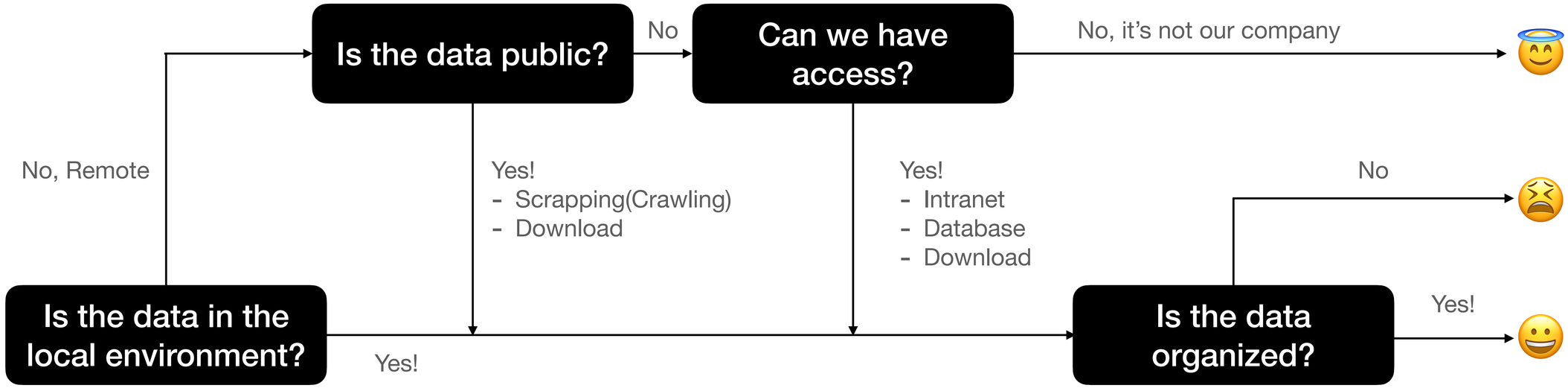
Week-06. Data

- Storage perspective
- Format perspective
- Data science perspective
- Programming language perspective
  - *From week-02*

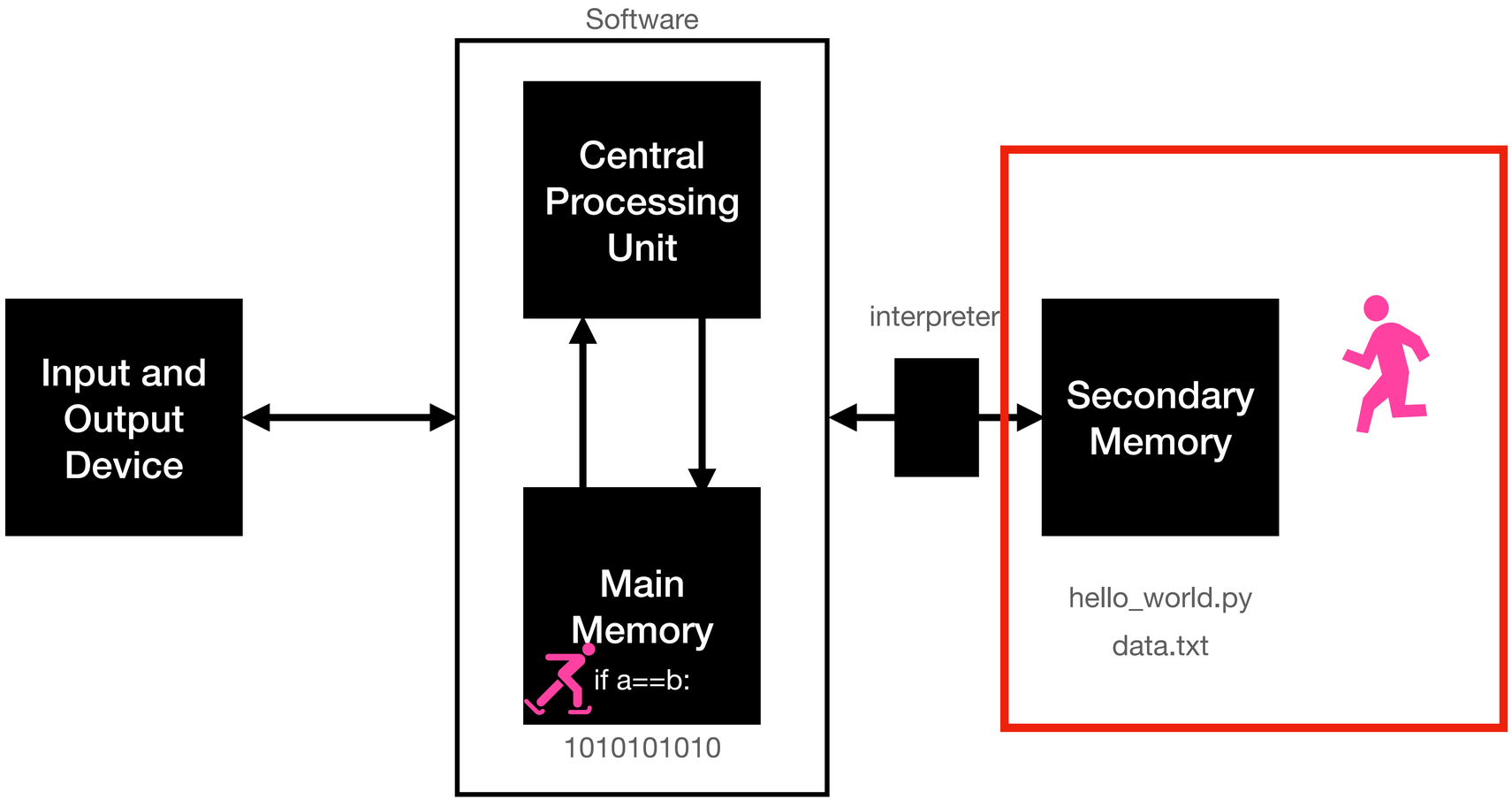


저장소의 관점에서의 데이터

데이터 수집 과정


Is the data public?
No
Can we have access?
No, it’s not our company
No, Remote
Yes!
- Scrapping(Crawling)
- Download
Yes!
- Intranet
- Database
- Download
No
Is the data in the local environment?
Yes!
Is the data organized?
Yes!

Recall. 프로그램 실행과정



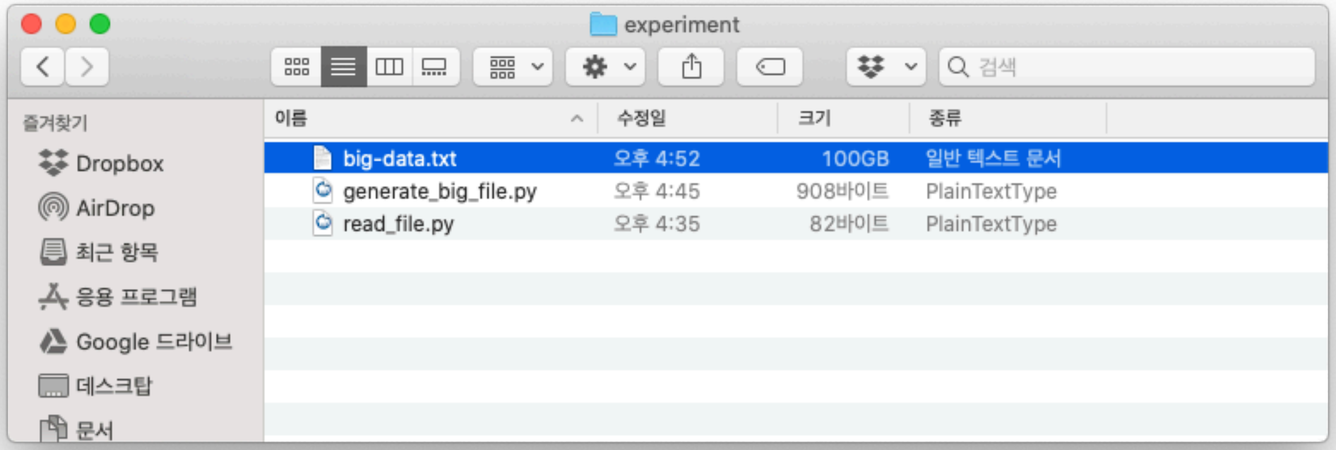
데이터 저장소

데이터가 로컬환경(본인의 컴퓨터)에 있을 때

- 데이터가 본인의 Secondary 메모리에 위치해 있는 상태
- File IO (input-and-output)
  - Python을 활용하여, secondary memory에 있는 파일을 main memory로 불러 들임
  - 예) f = open("data.txt", "r")

- Future Issue
  - 데이터의 크기가 Main Memory의 사이즈보다 크다면?
    - 부분적으로 읽는다 (partially)
  - 모든 데이터가 필요하다면?
    - Big-Data!!

Experiment!




```python
with open("big-data.txt", "r") as f:
    data = f.readlines()
    print(data[:10])
```

```
➜  experiment python read_file.py
[1]    66051 killed     python read_file.py
➜  experiment
```

데이터 저장소

데이터가 로컬환경(본인의 컴퓨터)에 있지 않을때

- 데이터가 외부에 존재하는 경우 (외부 서버라고 가정) 네트워크 통신을 활용하여 로컬 환경에 저장

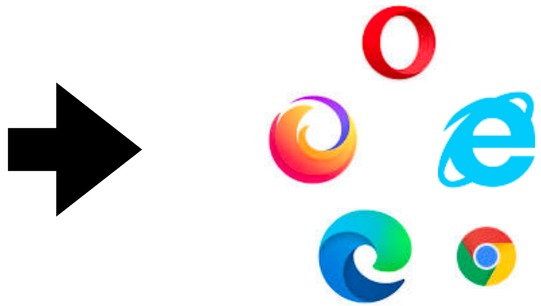
- Possible Locations
  - 외부 서버 (예: 연구실 서버)
  - 클라우드 저장소(구글 드라이브, 드랍박스 등)
  - 웹페이지 내에
  - 데이터베이스
  - ..

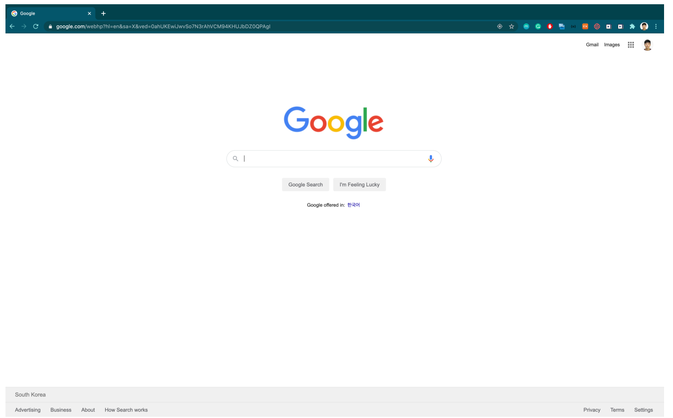
약간의 배경지식을 위해..

컴퓨터 간의 통신

웹 브라우저, 홈페이지, 웹사이트, 인터넷?

- 우리가 크롬/사파리와 같은 웹 브라우저를 켜서 구글에 접속하는 과정은?

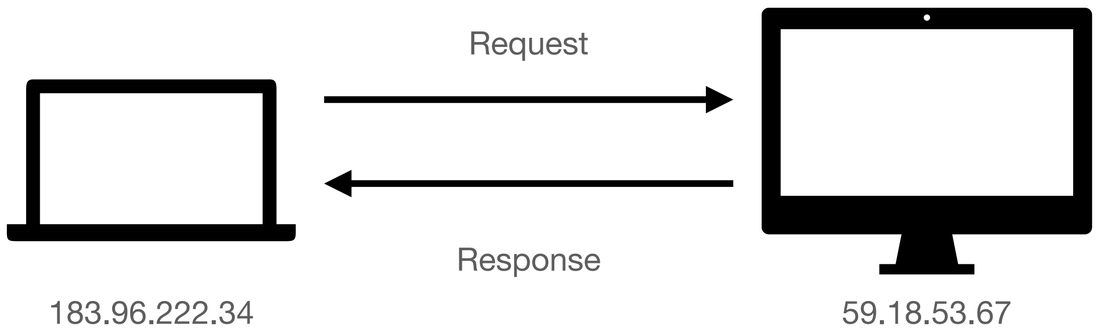




컴퓨터 간의 통신

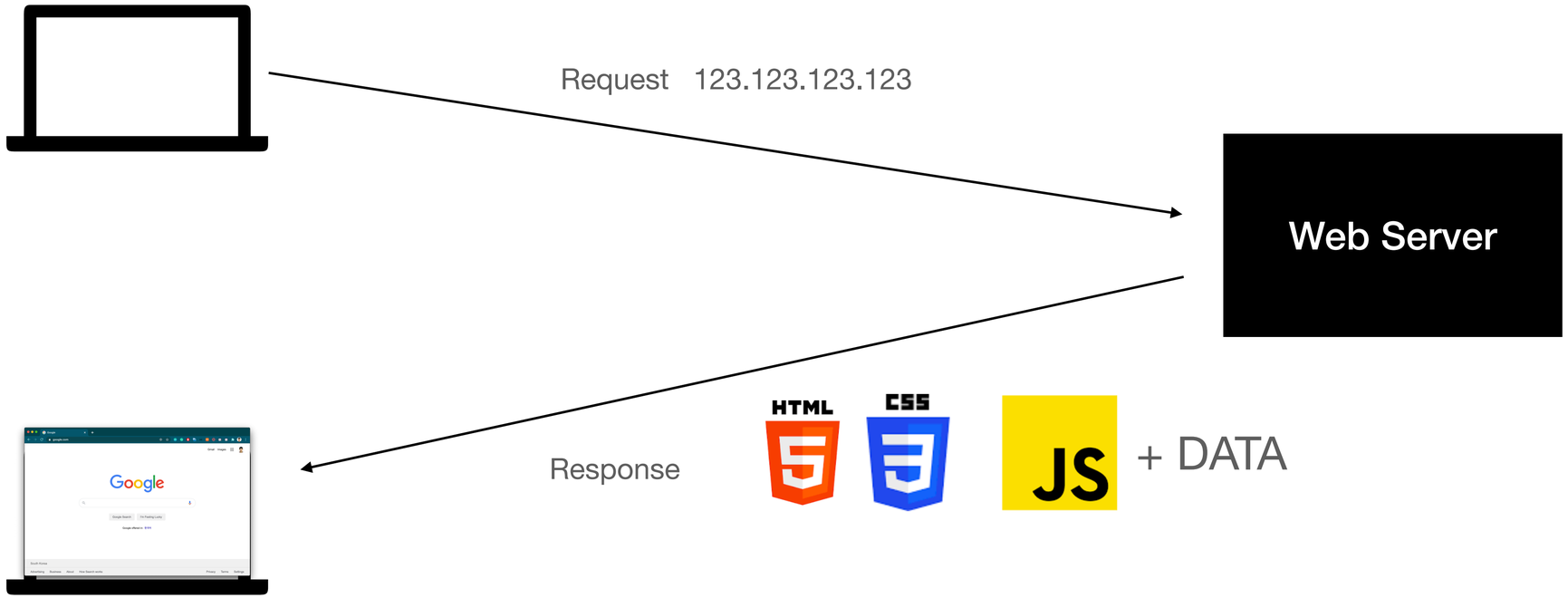
IP address, Domain Name, URL

- IP address
  - 비유를 하자면 위도와 경도 같이 Numerical 한 표기 방법
  - 인터넷을 통해 Device간의 통신을 하려면 IP주소가 필요함
    - 받는 사람도, 보내는 사람도
- Domain Name
  - 텍스트 기반의 주소(IP 주소를 외우고 다니기 힘들기 때문에)
  - 예) 서울시 서대문구 연세대학교
  - google.com, naver.com
  - Domain Name Server에서 IP와 Mapping
- URL (Uniform Resource Locator)
  - Full 주소
  - https://example.com/user/133
  - [프로토콜]://호스트주소:[포트]/[경로/쿼리스트링]



컴퓨터 간의 통신

TCP/HTTP Communication

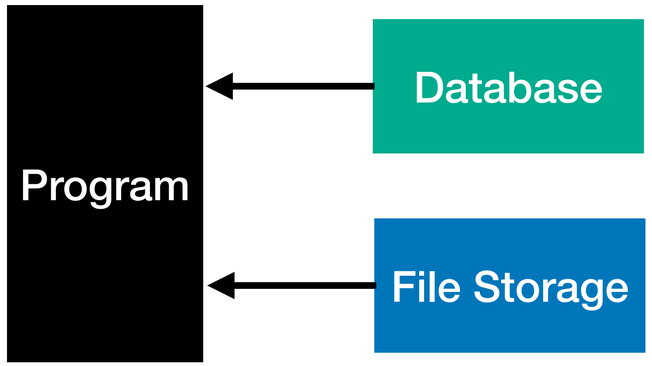


다시 돌아와서

데이터 저장소 (실습에서 자세히)

Database vs File Storage

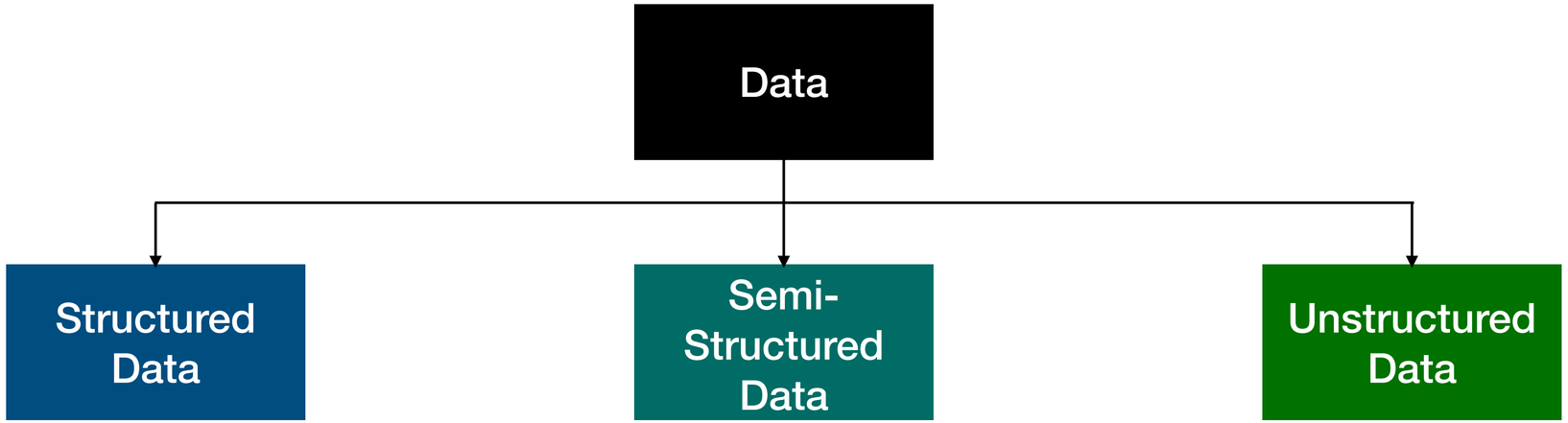
- 일반적으로 데이터를 저장할 수 있는 방식은 두가지
- 데이터베이스(Database)
  - 정형 데이터의 대표적인 예로, 테이블과 컬럼으로 구성
  - 테이블간의 관계가 표현되어있는 데이터베이스
    - 관계형데이터 베이스(Relational Database): MySQL
  - TCP로 연결, SQL로 데이터 요청
- 파일 저장소(File Storage)
  - 외부 서버의 Secondary Disk에 저장된 파일이 네트워크 요청에 의해 전달 가능
  - 얼마나 많이 저장해서 얼마나 빠르게 여러명에게 전달 할 수 있느냐가 관건
  - 주로 이미지, 비디오, 오디오와 같은 비정형 데이터들을 저장



파일 포맷의 관점에서의 데이터

데이터 파일 포맷

데이터 포맷의 큰 범주



	Structured Data	Semi-Structured Data	Unstructured Data
Difficulty of Preprocessing	Low	Medium	High
Usability	High	Medium	Low
Among all the data in the world	5-10%	5-10%	80%
Examples	Relational Database	CSV, JSON, XML, NoSQL	txt, log, image, audio, HTML(?)

데이터 파일 포맷

확장자가 반드시 그 파일의 포맷을 결정하는 것은 아니다!

- 왜 다른 파일 포맷들이 존재할까?
  - 파일의 목적에 따라 컴퓨터에 저장될 시 encoding 방식이 다르다.(불러올때는 decoding 방식)
    - 극단적인 예로, 일반 텍스트, 이미지, 비디오 등의 인코딩 방식이 각각의 목적에 따라 다름!
  - 몇몇 포맷은 오픈되어있지만, 일부 포맷들은 상업적으로 사용됨 (예 hwp)
- 그렇다면 파일 확장자(.jpeg, .csv, .docx)가 중요한가!?

> Windows file names have two parts; the file's name, then a period followed by the extension (suffix). The extension is a three- or four-letter abbreviation that signifies the file type. For example, in **letter.docx** the filename is **letter** and the extension is **docx**. Extensions are important because they tell your computer what icon to use for the file, and what application can open the file. For example, the **doc** extension tells your computer that the file is a Microsoft Word file.

  - 우리는 개발자이기 때문에, 안의 내용물까지 확인을 해야한다.
- 앞으로 자주 접하게 될 데이터 파일 포맷
  - .txt, .log, .csv, .sql, .json, .html, .xml

https://it.nmu.edu/docs/common-windows-file-extensions

데이터 파일 포맷

txt, log

- 비정형 데이터의 대표적인 포맷
- 보통 줄 단위로 데이터가 구성되어 있기 때문에, 부분 추출이 가능함
- txt
  - 가장 기본이 되는 텍스트 데이터의 포맷
  - 원시 데이터 형태로, 유의미한 분석을 하기 위해서는 추가적인 전처리가 요구됨
- log
  - log 포맷의 경우 기본적으로 확장자외에는 텍스트와 다르지 않음
  - 웹서버와 같은 곳에서 외부에서 들어오는 요청들을 줄단위로 저장해 놓은 데이터

```
172.31.13.177 - - [09/May/2020:08:50:24 +0000] "OPTIONS /api/v2/videos/ HTTP/1.1" 200 0 "https://slid.cc/lectures" "Mozilla/5.0 (Windows
NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.129 Safari/537.36"
172.31.13.177 - - [09/May/2020:08:50:24 +0000] "POST /api/v2/videos/ HTTP/1.1" 200 1184 "https://slid.cc/lectures" "Mozilla/5.0 (Windows
NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.129 Safari/537.36"
172.31.13.177 - - [09/May/2020:08:50:25 +0000] "POST /api/v2/videos/ HTTP/1.1" 200 928 "https://slid.cc/lectures" "Mozilla/5.0 (Windows NT
10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36"
172.31.17.148 - - [09/May/2020:08:50:25 +0000] "POST /api/v2/videos/ HTTP/1.1" 200 928 "https://slid.cc/video_url?
url=https%3A%2F%2Feclass2.ajou.ac.kr%2Fwebapps%2Fblackboard%2Fexecute%2Fcontent%2FblankPage%3Fcmd%3Dview%26content_id%3D_461891_1%26course
_id%3D_51383_1" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36"
172.31.17.148 - - [09/May/2020:08:50:25 +0000] "POST /api/v2/videos/ HTTP/1.1" 200 928 "https://slid.cc/video_url?
url=https%3A%2F%2Feclass2.ajou.ac.kr%2Fwebapps%2Fblackboard%2Fexecute%2Fcontent%2FblankPage%3Fcmd%3Dview%26content_id%3D_461891_1%26course
_id%3D_51383_1" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36"
172.31.17.148 - - [09/May/2020:08:50:25 +0000] "POST /api/v2/videos/ HTTP/1.1" 200 928 "https://slid.cc/video_url?
url=https%3A%2F%2Feclass2.ajou.ac.kr%2Fwebapps%2Fblackboard%2Fexecute%2Fcontent%2FblankPage%3Fcmd%3Dview%26content_id%3D_461891_1%26course
_id%3D_51383_1" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36"
172.31.17.148 - - [09/May/2020:08:50:25 +0000] "OPTIONS /api/v1/clips/110501/ HTTP/1.1" 200 0
"https://slid.cc/lectures/be511d7040924dfbb03b7ce2e6dde575" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/81.0.4044.138 Safari/537.36"
172.31.17.148 - - [09/May/2020:08:50:25 +0000] "DELETE /api/v1/clips/110501/ HTTP/1.1" 200 10
"https://slid.cc/lectures/be511d7040924dfbb03b7ce2e6dde575" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/81.0.4044.138 Safari/537.36"
172.31.13.177 - - [09/May/2020:08:50:25 +0000] "OPTIONS /api/v2/videos/ HTTP/1.1" 200 0 "https://slid.cc/video_url?
url=https%3A%2F%2Feclass3.cau.ac.kr%2Fcourses%2F30696%2Fexternal_tools%2F2" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36"
172.31.13.177 - - [09/May/2020:08:50:26 +0000] "POST /api/v2/videos/ HTTP/1.1" 200 800 "https://slid.cc/video_url?
url=https%3A%2F%2Feclass3.cau.ac.kr%2Fcourses%2F30696%2Fexternal_tools%2F2" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36"
172.31.17.148 - - [09/May/2020:08:50:26 +0000] "OPTIONS /api/v1/clips/177936/ HTTP/1.1" 200 0
"https://slid.cc/lectures/51833fc3089f411e89d80c6a52bb22e0" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/81.0.4044.138 Safari/537.36"
172.31.17.148 - - [09/May/2020:08:50:26 +0000] "POST /api/v2/videos/ HTTP/1.1" 200 928 "https://slid.cc/video_url?
url=https%3A%2F%2Feclass2.ajou.ac.kr%2Fwebapps%2Fblackboard%2Fexecute%2Fcontent%2FblankPage%3Fcmd%3Dview%26content_id%3D_461891_1%26course
_id%3D_51383_1" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36"
172.31.17.148 - - [09/May/2020:08:50:26 +0000] "PUT /api/v1/clips/177936/ HTTP/1.1" 200 555
"https://slid.cc/lectures/51833fc3089f411e89d80c6a52bb22e0" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/81.0.4044.138 Safari/537.36"
172.31.17.148 - - [09/May/2020:08:50:26 +0000] "OPTIONS /api/v2/videos/ HTTP/1.1" 200 0 "https://slid.cc/lectures" "Mozilla/5.0
(Macintosh; Intel Mac OS X 10_15_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.129 Safari/537.36"
172.31.17.148 - - [09/May/2020:08:50:26 +0000] "GET /api/v2/videos/ HTTP/1.1" 200 33730 "https://slid.cc/lectures" "Mozilla/5.0
(Macintosh; Intel Mac OS X 10_15_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.129 Safari/537.36"
172.31.17.148 - - [09/May/2020:08:50:27 +0000] "POST /api/v2/videos/ HTTP/1.1" 200 928 "https://slid.cc/video_url?
url=https%3A%2F%2Feclass2.ajou.ac.kr%2Fwebapps%2Fblackboard%2Fexecute%2Fcontent%2FblankPage%3Fcmd%3Dview%26content_id%3D_461891_1%26course
_id%3D_51383_1" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36"
```

데이터 파일 포맷

CSV- Comma-separated values

- 가장 일반적으로 쓰이는 포맷
  - 순수 데이터 외에는 쉼표(,) 및 line breaker(\n)만 추가공간을 차지함
  - 만약 컴퓨터에 Excel이나 Numbers같은 앱이 설치되어 있으면, 일반 엑셀 처럼 행렬로 데이터가 보임
- 데이터가 표현하고자하는 최소 단위가 쉼표(,)로 분리되어 있음
  - Jungwon,31,70,174,M
- 자연어 처리에서는 텍스트 데이터 내에 쉼표가 존재하는 경우가 빈번해 TSV 형식으로 저장
  - Tab-separated values: 쉼표(,) 대신 탭(tab)으로 값들을 분리

```
self previous shot,player position,home game,location x,opponent previous shot,home team,shot type,points,away team,location y,time,date,shoot player,time from last shot,quarter,current shot
outcome
MISSED,PF,No,676.0,MISSED,NOP,Jump Shot,2,CHI,225.0,10:48,2017-04-02,Bobby Portis,9.0,1,SCORED
MISSED,PG,Yes,59.0,SCORED,LAL,Layup,2,DAL,230.0,1:50,2016-12-29,D'Angelo Russell,45.0,1,MISSED
SCORED,C,No,50.0,SCORED,BRO,Layup,2,CHA,269.0,4:06,2016-11-04,Cody Zeller,46.0,3,SCORED
SCORED,SG,No,194.0,MISSED,LAL,Pullup Jump Shot,2,POR,357.0,0:13,2017-01-10,Allen Crabbe,,4,SCORED
SCORED,PF,No,35.0,SCORED,MEM,Jump Shot,2,NYK,449.0,11:38,2017-04-07,Kyle O'Quinn,27.0,3,MISSED
MISSED,SF,No,226.0,MISSED,GSW,Jump Shot,3,HOU,434.0,2:20,2017-03-31,Sam Dekker,44.0,4,SCORED
MISSED,PG,Yes,288.0,SCORED,CHA,Jump Shot,3,PHI,205.0,2:16,2017-02-13,Kemba Walker,26.0,1,MISSED
SCORED,PF,Yes,54.0,MISSED,DET,Reverse Layup,2,NYK,259.0,4:14,2016-11-01,Jon Leuer,19.0,2,SCORED
MISSED,SG,Yes,768.0,MISSED,BOS,Pullup Jump Shot,2,DEN,324.0,0:12,2016-11-06,Avery Bradley,,3,SCORED
MISSED,SG,Yes,35.0,MISSED,SAS,Pullup Jump Shot,2,ATL,405.0,2:05,2017-03-13,Danny Green,31.0,1,MISSED
MISSED,C,Yes,67.0,SCORED,MIN,Layup,2,CHA,269.0,4:45,2016-11-15,Gorgui Dieng,26.0,1,SCORED
MISSED,PG,Yes,108.0,MISSED,NYK,Jump Shot,2,MIN,113.0,1:16,2016-12-02,Derrick Rose,6.0,1,MISSED
SCORED,PF,No,851.0,SCORED,NYK,Jump Shot,2,MIA,240.0,7:04,2017-03-29,James Johnson,37.0,1,MISSED
MISSED,SF,No,141.0,MISSED,NYK,Morris 14' Fadeaway Jumper,2,DET,351.0,8:55,2016-11-16,Marcus Morris,34.0,4,MISSED
MISSED,SG,Yes,141.0,MISSED,TOR,Driving Floating Jump Shot,2,DET,212.0,3:57,2016-10-26,Cory Joseph,23.0,2,MISSED
MISSED,SG,Yes,887.0,SCORED,PHX,Layup,2,UTA,222.0,1:53,2017-01-16,Devin Booker,13.0,3,MISSED
SCORED,PG,Yes,182.0,MISSED,OKL,Pullup Jump Shot,2,NOP,341.0,8:05,2017-02-26,Russell Westbrook,54.0,1,SCORED
SCORED,PG,No,809.0,SCORED,NOP,Jump Shot,2,POR,107.0,3:54,2016-11-18,Damian Lillard,29.0,1,MISSED
MISSED,PG,Yes,740.0,SCORED,PHX,Pullup Jump Shot,2,DEN,230.0,9:06,2016-11-27,Eric Bledsoe,19.0,4,MISSED
SCORED,SF,No,52.0,BLOCKED,ATL,Driving Layup,2,ORL,250.0,10:12,2016-12-13,Evan Fournier,62.0,4,SCORED
MISSED,PG,No,637.0,SCORED,DAL,Jump Shot,3,IND,295.0,3:37,2016-12-09,Aaron Brooks,25.0,2,SCORED
SCORED,PF,No,900.0,SCORED,LAL,Step Back Jump Shot,2,DET,369.0,11:13,2017-01-15,Tobias Harris,34.0,2,SCORED
MISSED,PF,Yes,79.0,SCORED,NOP,Jump Bank Shot,2,LAL,151.0,9:37,2016-11-12,Anthony Davis,24.0,1,MISSED
MISSED,C,Yes,642.0,SCORED,MEM,Jump Shot,3,SAC,290.0,0:56,2016-12-16,Marc Gasol,,3,MISSED
MISSED,PF,No,52.0,SCORED,MIL,Driving Dunk,2,HOU,250.0,7:17,2017-01-23,Nene Hilario,80.0,3,SCORED
SCORED,C,No,51.0,SCORED,UTA,Reverse Layup,2,SAC,230.0,7:59,2016-12-21,DeMarcus Cousins,46.0,3,MISSED
SCORED,SF,Yes,298.0,MISSED,IND,Jump Shot,3,CHI,161.0,8:59,2016-12-30,Paul George,29.0,2,MISSED
SCORED,SG,Yes,873.0,MISSED,SAS,Driving Layup,2,DEN,231.0,2:39,2017-02-04,Jonathon Simmons,37.0,4,SCORED
SCORED,PF,Yes,52.0,SCORED,DEN,Cutting Dunk Shot,2,ORL,250.0,3:22,2017-01-16,Kenneth Faried,32.0,1,SCORED
MISSED,SG,Yes,714.0,MISSED,WAS,Jump Shot,3,PHX,449.0,5:28,2016-11-21,Bradley Beal,6.0,3,SCORED
MISSED,SG,No,809.0,SCORED,MIN,Jump Shot,3,PHI,25.0,9:00,2016-11-17,Nik Stauskas,6.0,1,SCORED
MISSED,PG,No,735.0,MISSED,GSW,Floating Jump Shot,2,NOP,254.0,6:54,2017-04-08,Jrue Holiday,44.0,1,SCORED
MISSED,PG,Yes,882.0,SCORED,OKL,Driving Layup,2,NYK,266.0,8:59,2017-02-15,Russell Westbrook,57.0,3,SCORED
SCORED,C,Yes,161.0,BLOCKED,BRO,Jump Shot,2,SAS,264.0,5:02,2017-01-23,Brook Lopez,23.0,1,SCORED
MISSED,C,Yes,62.0,MISSED,OKL,Tip Layup Shot,2,NOP,254.0,10:22,2016-12-04,Joffrey Lauvergne,17.0,2,MISSED
MISSED,PG,No,269.0,SCORED,GSW,Jump Shot,3,NYK,365.0,8:00,2016-12-15,Brandon Jennings,21.0,3,MISSED
SCORED,SG,Yes,743.0,SCORED,POR,Pullup Jump Shot,2,LAL,347.0,4:18,2017-01-05,Evan Turner,61.0,4,SCORED
MISSED,PF,No,890.0,BLOCKED,WAS,Finger Roll Layup,2,SAS,254.0,10:15,2016-11-26,David Lee,56.0,1,SCORED
MISSED,SF,No,67.0,SCORED,ORL,Driving Layup,2,MIN,274.0,2:01,2016-11-09,Andrew Wiggins,25.0,4,MISSED
SCORED,SF,No,886.0,SCORED,CHA,Running Layup,2,MIN,238.0,12:00,2016-12-03,Andrew Wiggins,19.0,1,SCORED
SCORED,C,Yes,62.0,SCORED,BOS,Driving Bank Shot,2,UTA,301.0,10:44,2017-01-03,Al Horford,64.0,2,MISSED
SCORED,SG,No,709.0,SCORED,DAL,Jump Shot,3,BRO,424.0,5:28,2017-03-10,Randy Foye,36.0,2,MISSED
SCORED,SF,Yes,269.0,SCORED,WAS,26' 3PT Jump Shot,3,ORL,383.0,9:57,2017-03-05,Otto Porter Jr.,67.0,2,MISSED
MISSED,SF,No,180.0,SCORED,OKL,Pullup Jump Shot,2,CHI,378.0,4:55,2017-02-01,Doug McDermott,33.0,4,SCORED
MISSED,SF,Yes,50.0,MISSED,CLE,Jump Shot,3,MIL,478.0,5:38,2016-12-21,LeBron James,72.0,2,SCORED
SCORED,SG,Yes,288.0,SCORED,MIL,Jump Shot,3,NYK,344.0,7:32,2017-03-08,Khris Middleton,40.0,1,SCORED
MISSED,C,Yes,50.0,MISSED,UTA,Reverse Layup,2,LAL,265.0,2:29,2017-01-26,Rudy Gobert,3.0,2,SCORED
SCORED,PG,Yes,881.0,SCORED,MIA,Layup,2,BOS,230.0,0:52,2016-11-28,Goran Dragic,31.0,3,SCORED
MISSED,PG,No,882.0,SCORED,TOR,Driving Layup,2,BOS,259.0,3:22,2017-02-24,Isaiah Thomas,27.0,1,SCORED
MISSED,C,No,58.0,SCORED,CLE,Jump Shot,2,ORL,426.0,5:52,2016-10-29,Nikola Vucevic,28.0,3,MISSED
MISSED,SG,Yes,861.0,MISSED,NYK,Jump Shot,2,NOP,454.0,4:05,2017-01-09,Sasha Vujacic,38.0,4,MISSED
"NBA_TRAIN.csv" 105037L, 10397439C
```

데이터 파일 포맷

JSON-JavaScript Object Notation

- Key-Value, array 등과 같은 고차원적인 데이터를 저장하기 위해 사용되는 포맷
  - Python에서는 dictionary와 list의 조합과 비슷
  - CSV의 경우 보통 1 column 1value (저차원)
- 오픈 표준 파일 포맷으로 (open standard file format) 대부분의 언어가 json타입의 데이터에 대한 encoding-decoding 함수를 제공함
  - 파이썬 예) import json
- 괄호 ({ }, [ ])의 쌍으로 데이터가 감싸지기 때문에 데이터의 부분적인 추출이 불가능

```
[
  {
    "_id": "5f69cc07427393510d722df5",
    "index": 0,
    "guid": "7cbb38e8-b3bc-4bef-a117-f7ee107eaaa7",
    "isActive": false,
    "balance": "$2,818.90",
    "picture": "http://placehold.it/32x32",
    "age": 37,
    "eyeColor": "brown",
    "name": "Gilda Buchanan",
    "gender": "female",
    "company": "ZORROMOP",
    "email": "gildabuchanan@zorromop.com",
    "phone": "+1 (897) 547-2003",
    "address": "295 Story Street, Dargan, Kentucky, 5481",
    "about": "Ullamco ad consequat excepteur veniam quis est ad deserunt irure nulla velit exercitation aliquip. Velit eiusmod irure
commodo qui ipsum sunt labore ex veniam irure. Id est cupidatat nostrud dolore exercitation consequat ipsum nostrud quis anim pariatur
laboris quis eu. Minim cupidatat exercitation ullamco elit qui fugiat. Labore ut enim amet culpa laboris laborum.\r\n",
    "registered": "2020-07-17T04:08:12 -09:00",
    "latitude": -52.733468,
    "longitude": -125.645575,
    "tags": [
      "incididunt",
      "id",
      "ullamco",
      "adipisicing",
      "veniam",
      "anim",
      "est"
    ],
    "friends": [
      {
        "id": 0,
        "name": "Justine Cervantes"
      },
      {
        "id": 1,
        "name": "Sampson Lane"
      },
      {
        "id": 2,
        "name": "Bethany Collier"
```

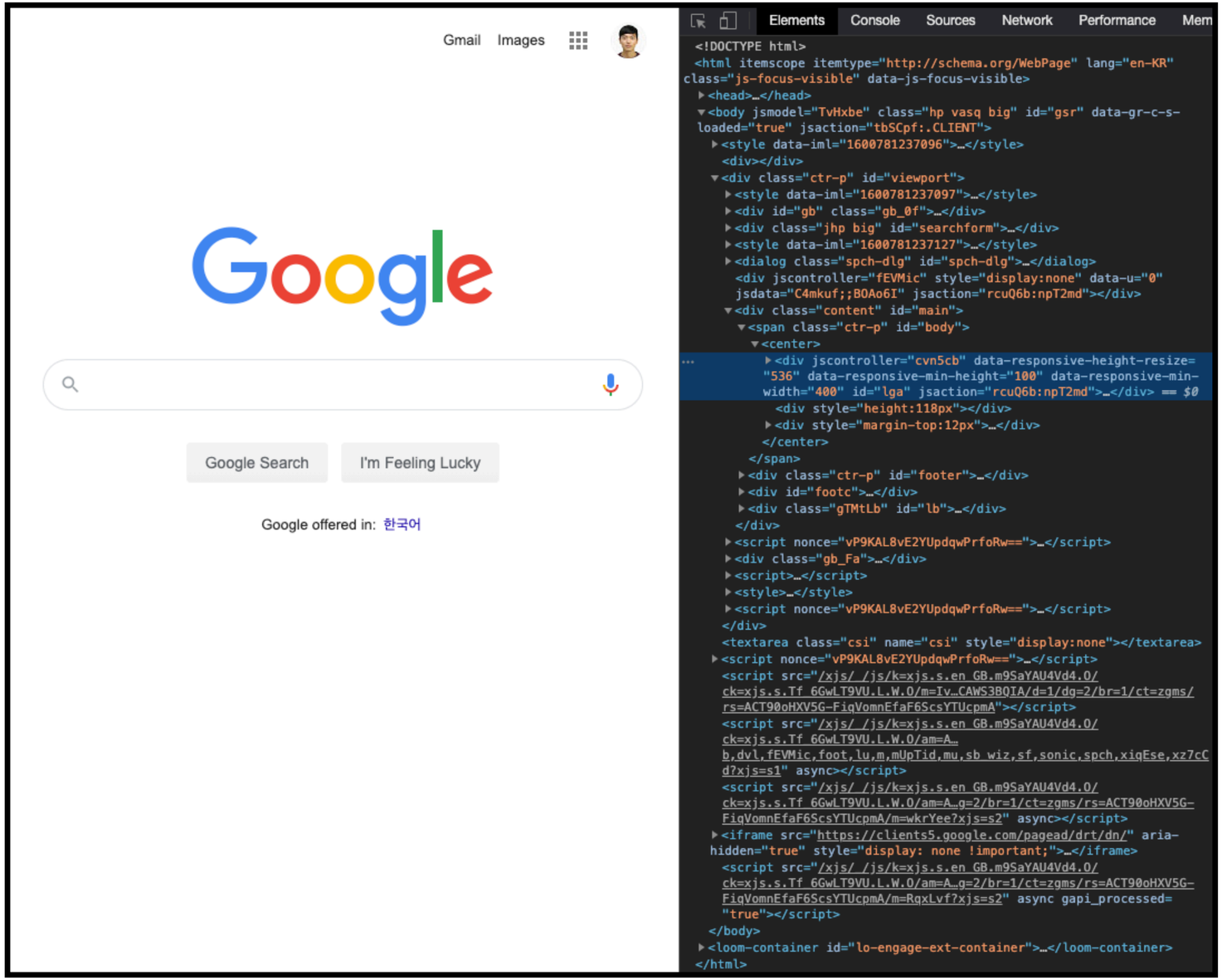
데이터 파일 포맷

Markup Language

- Markup Language
  - 문서에 태그나 특수기호를 사용하여 문서나 데이터의 구조를 표현하는 언어
  - 예) README.md
- HTML: HyperText Markup Language
  - 웹 브라우저에서 Presentation 을 목적으로 사용되는 언어
  - 전통적인 Server-side Rendering 웹 페이지의 경우 HTML 전체를 받아와서 Parsing을 하여 내부의 데이터를 추출
- XML: Extensible Markup Language
  - 다목적 Markup Language
  - 제한된 접근성의 HTML을 한계를 다른 종류의 시스템간의 데이터를 교환을 목적으로 개발
  - 단순 데이터 전달의 측면에서는 JSON에 비해 용량적으로 낭비스러움

This XML file does not appear to have any style information assoc

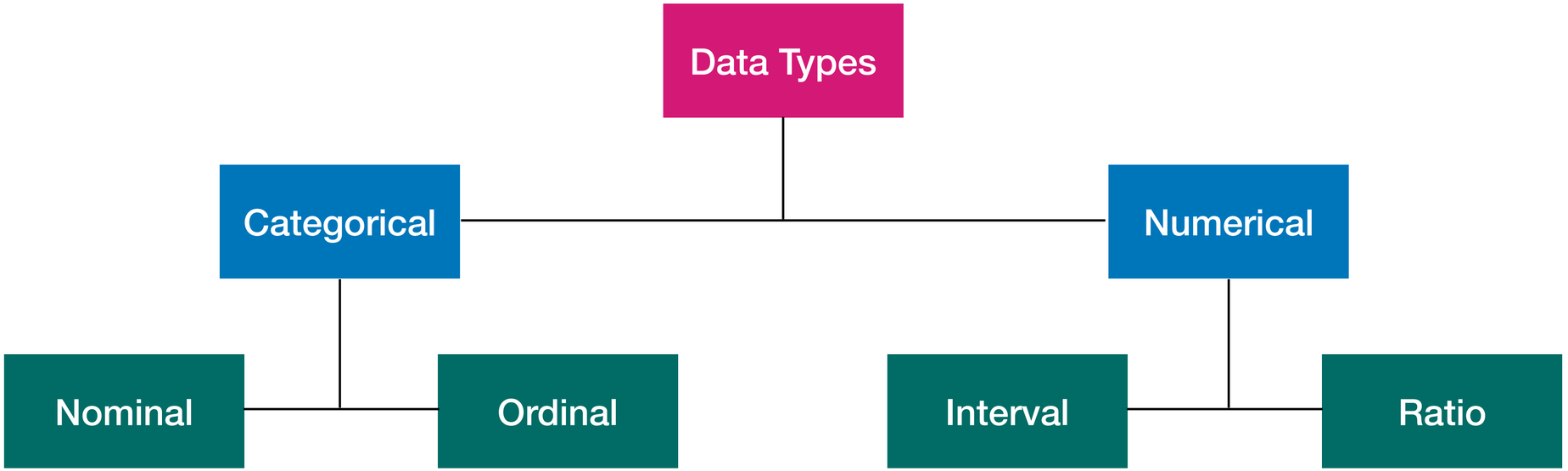
```
▼<CATALOG>
  ▼<CD>
      <TITLE>Empire Burlesque</TITLE>
      <ARTIST>Bob Dylan</ARTIST>
      <COUNTRY>USA</COUNTRY>
      <COMPANY>Columbia</COMPANY>
      <PRICE>10.90</PRICE>
      <YEAR>1985</YEAR>
    </CD>
  ▼<CD>
      <TITLE>Hide your heart</TITLE>
      <ARTIST>Bonnie Tyler</ARTIST>
      <COUNTRY>UK</COUNTRY>
      <COMPANY>CBS Records</COMPANY>
      <PRICE>9.90</PRICE>
      <YEAR>1988</YEAR>
    </CD>
  ▼<CD>
      <TITLE>Greatest Hits</TITLE>
      <ARTIST>Dolly Parton</ARTIST>
      <COUNTRY>USA</COUNTRY>
      <COMPANY>RCA</COMPANY>
      <PRICE>9.90</PRICE>
      <YEAR>1982</YEAR>
    </CD>
  ▼<CD>
      <TITLE>Still got the blues</TITLE>
      <ARTIST>Gary Moore</ARTIST>
      <COUNTRY>UK</COUNTRY>
      <COMPANY>Virgin records</COMPANY>
      <PRICE>10.20</PRICE>
      <YEAR>1990</YEAR>
    </CD>
  ▼<CD>
      <TITLE>Eros</TITLE>
      <ARTIST>Eros Ramazzotti</ARTIST>
      <COUNTRY>EU</COUNTRY>
      <COMPANY>BMG</COMPANY>
      <PRICE>9.90</PRICE>
      <YEAR>1997</YEAR>
    </CD>
  ▼<CD>
      <TITLE>One night only</TITLE>
      <ARTIST>Bee Gees</ARTIST>
      <COUNTRY>UK</COUNTRY>
      <COMPANY>Polydor</COMPANY>
      <PRICE>10.90</PRICE>
      <YEAR>1998</YEAR>
    </CD>
  ▼<CD>
      <TITLE>Sylvias Mother</TITLE>
      <ARTIST>Dr.Hook</ARTIST>
      <COUNTRY>UK</COUNTRY>
      <COMPANY>CBS</COMPANY>
      <PRICE>8.10</PRICE>
      <YEAR>1973</YEAR>
    </CD>
```



데이터 과학 관점에서의 데이터

데이터 타입

숫자라고 다 같은 숫자가 아니다!



데이터 타입

올바른 전처리/학습을 위해서는 데이터 타입 분류를 잘해야..!

- Categorical
  - **Nominal**: 명목자료
    - 비교 X, 차이 O
    - 예: 우편번호, 주민번호 뒷자리, 인종, 성별
  - **Ordinal**: 서열자료
    - 순서가 있는 명목자료
    - 예: 학점, 학년
- Numerical
  - **Interval**: 구간자료
    - True Zero가 존재하지 않음
    - 0도가 가장 낮은 온도인가? 1월 1일이 가장 이른 날짜인가?
  - **Ratio**: 비율자료
    - True Zero가 존재
    - 예) 절대온도, 길이, 무게, 횟수

데이터 타입

데이터 타입의 따른 요약통계의 차이

- Categorical
  - 빈도수: A형:1명, AB형:10명, B형:5명, O형:30명
  - 최빈값: O형

- Numerical
  - 평균, 중위값, 최대/최소값, 분산(표준편차)
  - 예) 대한민국 고등학생들의 키(height)에 대한 요약 통계

- 왜 학년(Grade)은 Numerical이 아니라 Categorical인가?
  - 과제에서 😅

프로그래밍 언어 관점에서의 데이터
Week-02 다시보기!

정리

데이터 확보 부터 Python에 Load하기 까지

1. 데이터 확보
    1. 데이터 베이스에 있나? 바로 3번으로
    2. 파일 형태로 저장되어 있나?
2. 데이터 포맷 확인
    1. Semi Structured인가? Unstructured인가?
    2. 각각에 맞는 전처리
3. 데이터 타입 확인
    1. Categorical? Numerical?
    2. Nominal, Ordinal, Interval, Ration?
4. 각각의 데이터 타입에 맞는 파이썬 변수 타입 설정
    1. Int? float? str?

E.O.D