

파이썬을 이용한 데이터수집 및 스마트공장 견학

Python 소개 및 설치
Python의 기초

2023년 1월 09일
안재관

반갑습니다!

- 주식회사 스위트앤데이터 CEO
- 서울대학교 산업공학과 데이터마이닝 연구실 박사과정
- 더럼대학교 (Durham) 경제학 학사/ 런던 정경대 (LSE) 경제학 석사
- 관심분야: 딥러닝을 이용한 컴퓨터비전, 생성모델, 강화학습 시계열분석
- 연락처: jk.ahn7@dm.snu.ac.kr



강사 소개

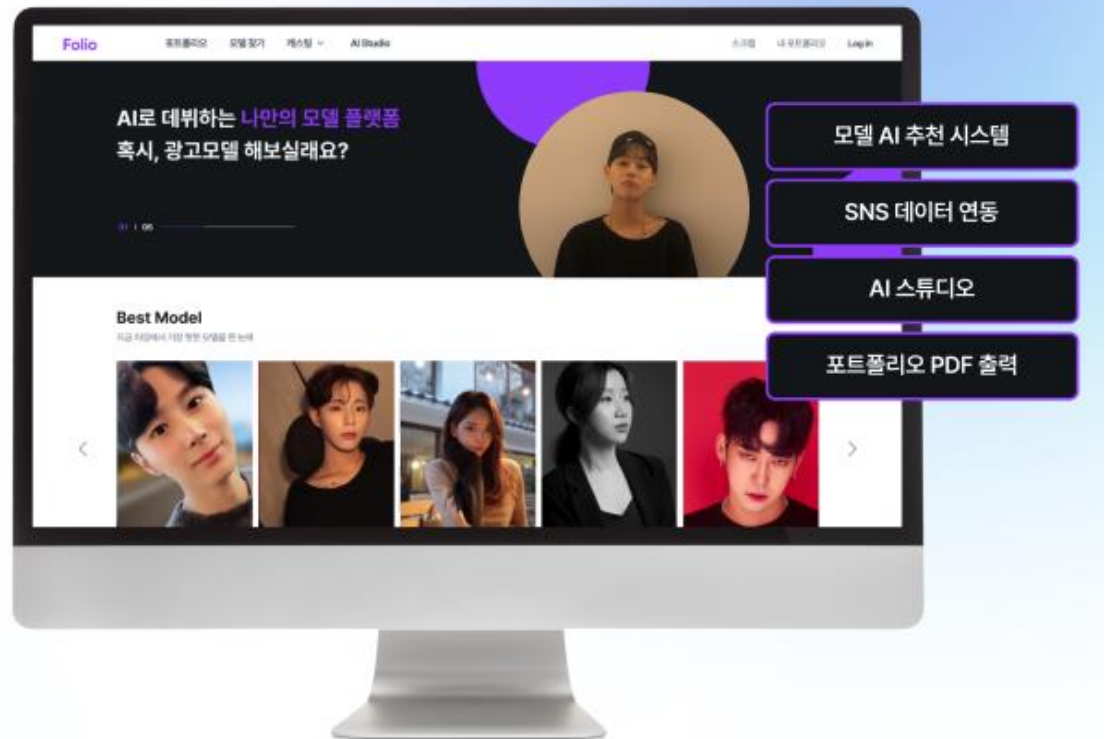
Solution

UI(Web)

웹-앱 최적화를 이룬 포트폴리오 플랫폼 서비스

데이터 기반의 모델 서칭 서비스부터, 해외 진출을
꿈꾸는 모델들을 위한 글로벌 컴카드 솔루션까지

차밍 AI는 엔터 시장에서의 수많은 공급과 수요들이
최적의 조건을 통해 연결되고 빠르게 해답을 찾도록
플랫폼, 그 이상의 솔루션을 제공합니다.



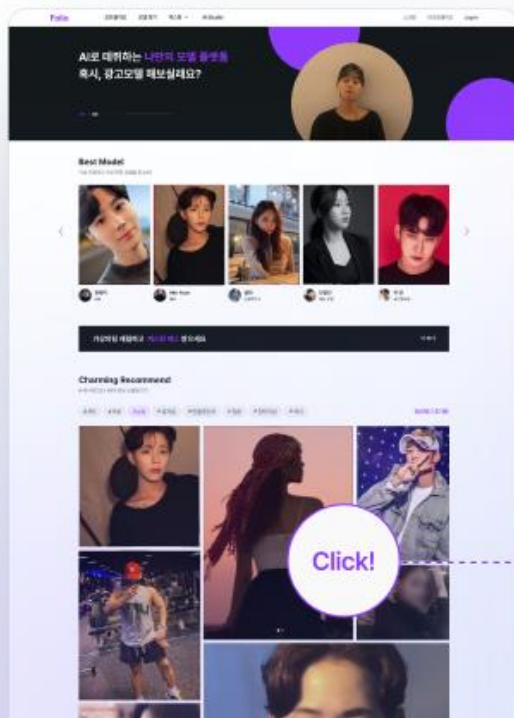
강사 소개

Solution

UI(Web)

SNS/이력 통합 포트폴리오 관리 플랫폼

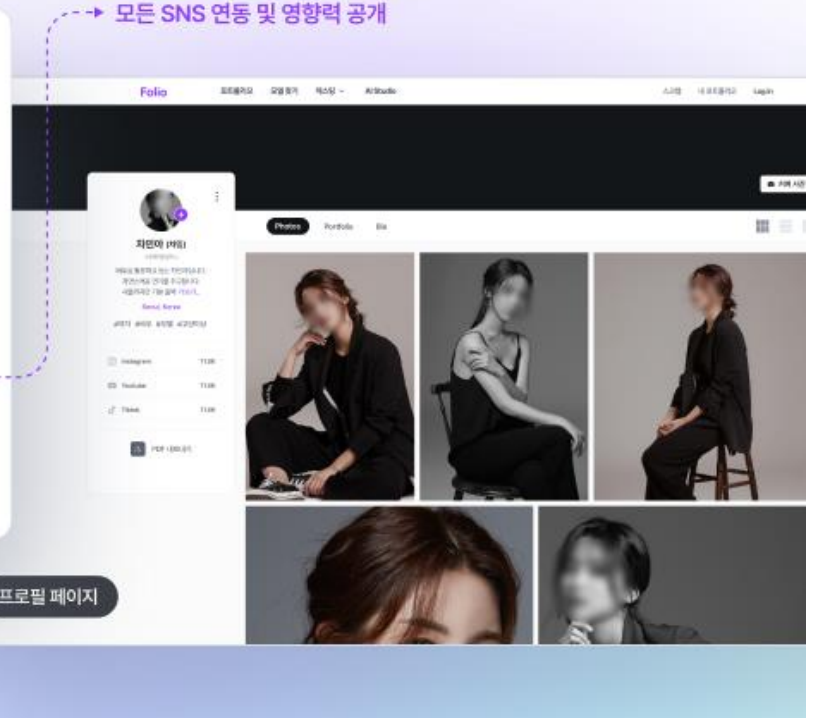
메인 랜딩페이지



프로필 상세 정보



모든 SNS 연동 및 영향력 공개



프로필 페이지

PDF 출력 가능

강사 소개

한 눈에 볼 수 있는 데이터 통합 **중앙 집중형** 포트폴리오



● Introduction

- 빅데이터와 크롤링
- 프로그래밍 언어의 소개
- 파이썬 소개

● 파이썬 설치

- Anaconda 설치
- Jupyter Notebook과 VS code 설치 및 적응
- 파이썬 가상환경 구성
- 파이썬 패키지 설치

● 파이썬 맛보기

- 첫 Python 프로그램 만들기
- 변수를 사용하여 구구단 출력하기
- 키보드 입력을 받아 변수에 저장 및 활용하기
- 문자열 관련 함수들을 응용하기

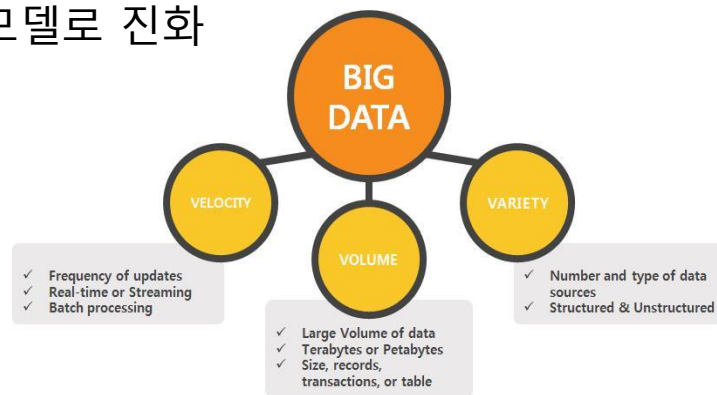
2017년 이후는 빅데이터 시대

● 빅데이터란

- 보통 수십에서 수천 테라바이트(terabyte: TB) 정도의 거대한 크기를 갖고 여러 가지 비정형 데이터를 포함하고 있으며, 생성-유통-소비가 몇 초에서 몇 시간 단위로 일어나 기존의 방식으로는 관리와 분석이 매우 어려운 데이터 집합

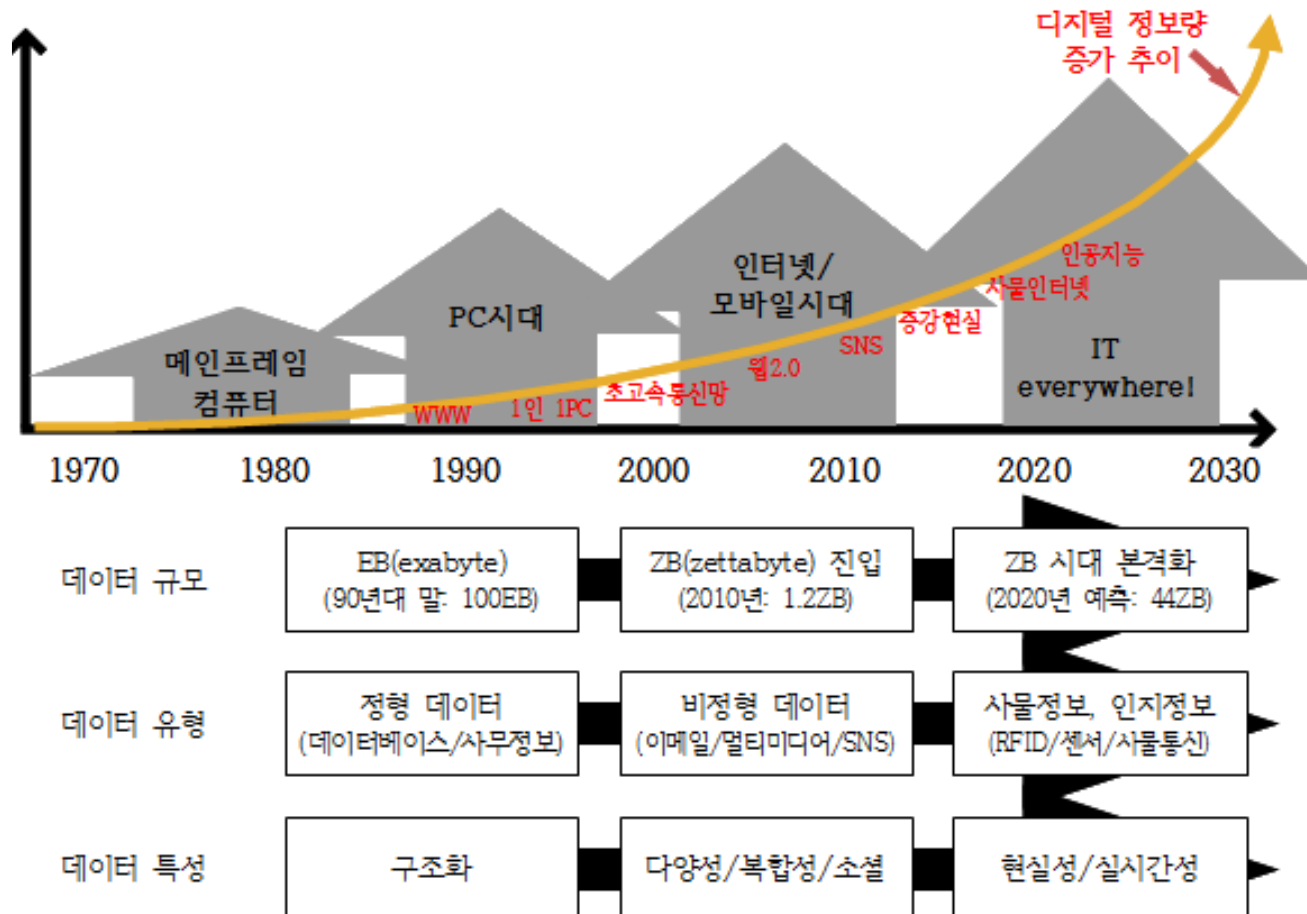
● 빅데이터의 특징

- 01년 메타그룹의 Doug Laney는 학문적인 관점에서 3V라고 하는 빅 데이터의 세가지 특징(Velocity, Volume, Variety)을 그림과 같이 정의
- 그 후, 추가로 Complexity 까지 포함하여 3V1C 모델로 진화



2017년 이후는 빅데이터 시대

● 디지털 트랜스포메이션으로 인한 데이터 량의 증가



2017년 이후는 빅데이터 시대

● 디지털 트랜스포메이션으로 인한 데이터 량의 증가

구분	PC시대	인터넷시대	모바일시대	스마트시대
패러다임 변화	전산화, 디지털화	정보화, 온라인화	소셜화, 모바일화	지능화, 개인화, 사물정보화
IT 비전	1인 1PC	클릭 e-Korea	손 안의 PC, 소통	IT everywhere, 신가치창출
IT 이슈	PC, PC통신, 데이터베이스	초고속인터넷, WWW, 웹서버	모바일 인터넷, 스마트폰	빅데이터, 차세대 PC, 사물통신(M2M)
핵심분야 (서비스)	PC, OS	포털, 검색엔진, 웹2.0	스마트폰, 앱, SNS	미래전망, 상황인식, 개인맞춤형 서비스
대표기업	IBM, 마이크로소프트 등	구글, 네이버, 유튜브 등	애플, 페이스북 등	?

2017년 이후는 빅데이터 시대

● G-Globe.com이 발표한 1분 동안 인터넷에서 엄청난 양의 데이터가 생성

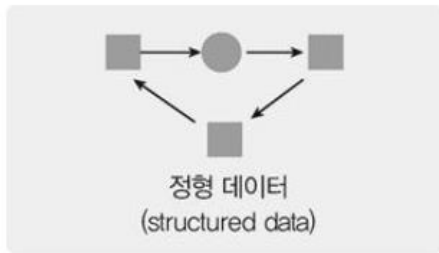
- 트위터의 경우, 98,000개의 트윗이 생성
- 페이스북의 경우 695,000개의 글 또는 사진이 게시
- 1억 6천 8백야 만개의 이메일이 송수신
- 구글의 경우, 698,445 검색이 이루어짐
- 1,820TB 데이터가 생성

→ 이 추세로 볼 때, 2022년에는 데이터 사용량이 약 50ZB가 될 것으로 추정



2017년 이후는 빅데이터 시대

● 빅데이터의 종류



- 틀이 잡혀 있는 데이터, 체계화된 데이터
- 높은 안정성, 유연하지 못한 구조
- 금융, 제조 등 대부분 기업의 업무용 데이터베이스



- 틀이 잡혀 있지 않고 사전 정의가 없는 데이터
- 다양하고 방대한 양의 데이터, 별도의 분석 처리 기술이 필요
- 텍스트, 이미지, 음원 데이터, 빅데이터

정형데이터



	-	-
1	-	-
2	-	-

반정형데이터



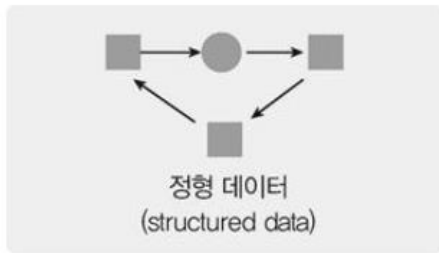
비정형데이터



데이터의 분류

2017년 이후는 빅데이터 시대

● 빅데이터의 종류



- 틀이 잡혀 있는 데이터, 체계화된 데이터
- 높은 안정성, 유연하지 못한 구조
- 금융, 제조 등 대부분 기업의 업무용 데이터베이스



- 틀이 잡혀 있지 않고 사전 정의가 없는 데이터
- 다양하고 방대한 양의 데이터, 별도의 분석 처리 기술이 필요
- 텍스트, 이미지, 음원 데이터, 빅데이터

정형데이터



	-	-
1	-	-
2	-	-

?

반정형데이터



데이터의 분류

비정형데이터



2017년 이후는 빅데이터 시대

● 비정형데이터에 대한 처리: 임베딩

임베딩이란 기계가 이해할 수 있는 vector로 바꾼 결과 혹은 그 일련의 과정 전체

컴퓨터가 보는 문자

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

컴퓨터는 ASCII, 유니코드, UTF-8 encoding 등으로 문자를 표현하고 저장한다.

컴퓨터가 보는 단어

단어	1	2	3	4
love	0x6C	0x6F	0x76	0x65
live	0x6C	0x69	0x76	0x65
like	0x6C	0x69	0x6B	0x65

love는 live보다 like와 더 유사하다. 하지만 컴퓨터가 보기에는 love는 live와 더 비슷해 보인다.

2017년 이후는 빅데이터 시대

● 비정형데이터에 대한 처리: 단어 임베딩

One-hot Encoding

- 단어마다 별도의 차원을 부여

나	너	는	학교	집	에	갔다
1	0	0	0	0	0	0
0	1	0	0	0	0	0
0	0	1	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	1	0	0
0	0	0	0	0	1	0
0	0	0	0	0	0	1

- 나는 학교에 갔다
- 너는 집에 갔다

빈도수 기반 임베딩

- 나는 학교에 갔다

1 0 1 1 0 1 1

- 너는 집에 갔다

0 1 1 0 1 1 1

- 나는 집에 갔다

1 0 1 0 1 1 1

- 나는 집에, 너는 학교에 갔다

1 1 2 1 1 2 1

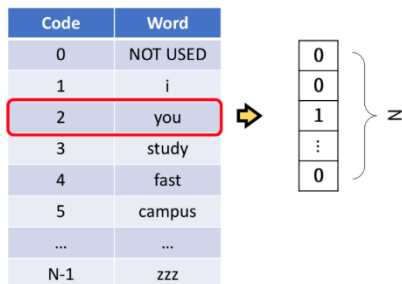
- 나는 학교에, 너는 집에 갔다

1 1 2 1 1 2 1

2017년 이후는 빅데이터 시대

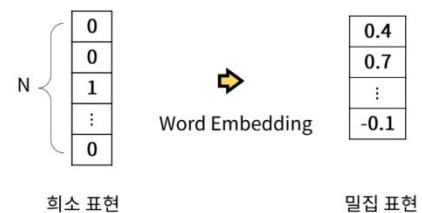
● 비정형데이터에 대한 처리: 단어 임베딩

One-Hot Encoding



N개의 단어를 좌측의 코드로 표현하면 **희소 표현(Sparse representation)**이라고 하며, 우측의 벡터로 표현할 경우 **One-Hot Encoding**이라고 한다.

밀집 표현 Dense Representation



희소 표현된 단어를 임의의 길이의 실수 벡터로 표현할 경우, 이를 **밀집 표현(Dense Representation)**이라고 한다. 이 과정을 Word Embedding이라고 하며, 밀집 표현된 결과를 **임베딩 벡터(Embedding Vector)**라고 부른다.

2017년 이후는 빅데이터 시대

● 비정형데이터에 대한 처리: 단어 임베딩

나는	[1, 0, 0, 0, 0]	나는	[0.123, -0.411, 0.345, 0.198, 0.757]
임베딩	[0, 1, 0, 0, 0]	임베딩	[0.841, 0.742, 0.956, 0.312, 0.149]
공부를	[0, 0, 1, 0, 0]	공부를	[2.314, 0.742, 1.233, 0.534, -0.015]
하고	[0, 0, 0, 1, 0]	하고	[-1.434, 0.742, 0.231, 0.864, 0.508]
있다	[0, 0, 0, 0, 1]	있다	[0.389, 0.742, 1.412, 1.218, -1.113]

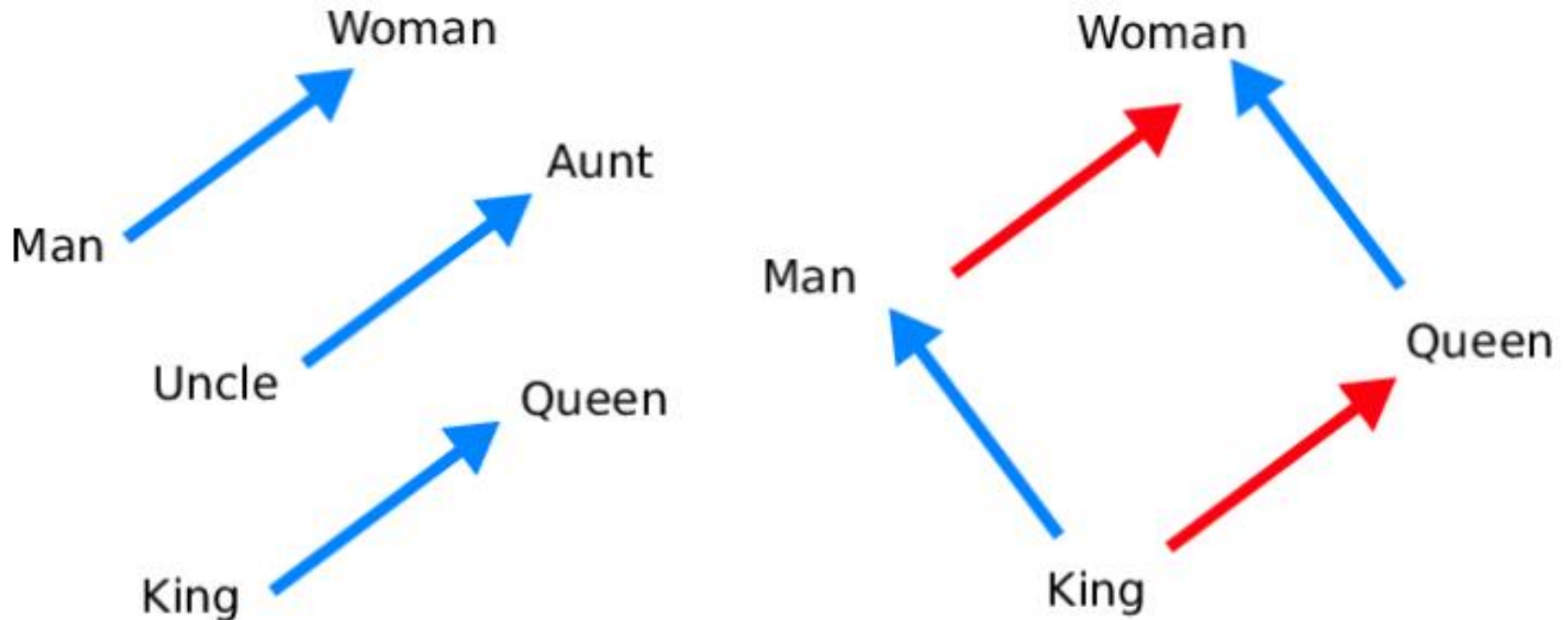
<그림1> (좌) 희소 표현 (우) 밀집 표현

	원-핫 벡터	임베딩 벡터
차원	고차원(단어 집합의 크기)	저차원
표현 방식	희소 표현	밀집 표현
표현 방법	수동	훈련 데이터로 학습
속성 값	1과 0	실수

<그림2> 표현 방법 비교 (출처: 딥러닝을 이용한 자연어 처리 입문)

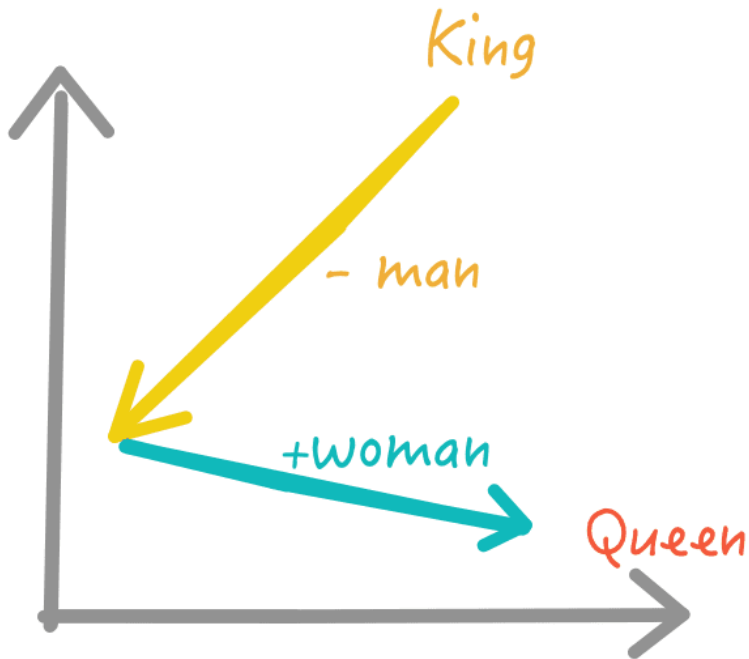
2017년 이후는 빅데이터 시대

- 비정형데이터에 대한 처리: 단어 임베딩

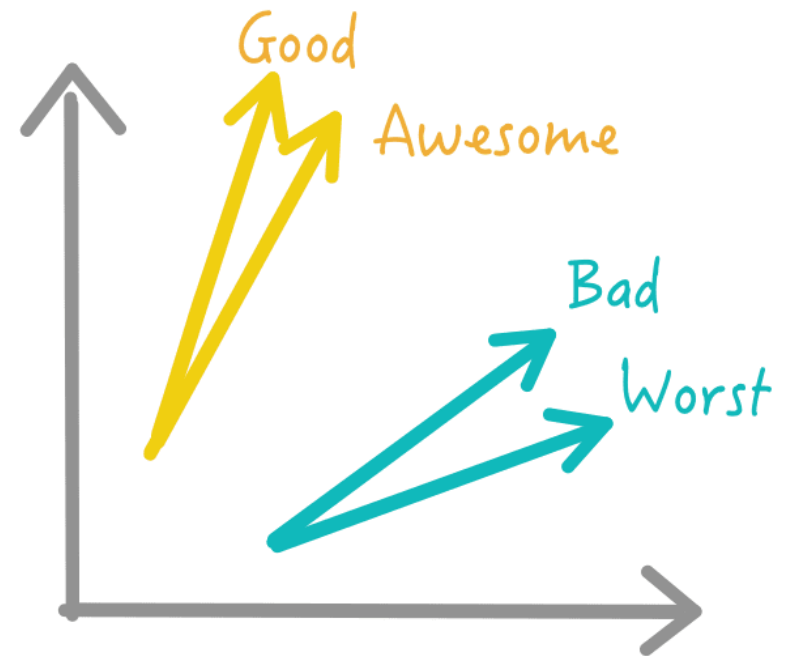


2017년 이후는 빅데이터 시대

● 비정형데이터에 대한 처리: 단어 임베딩



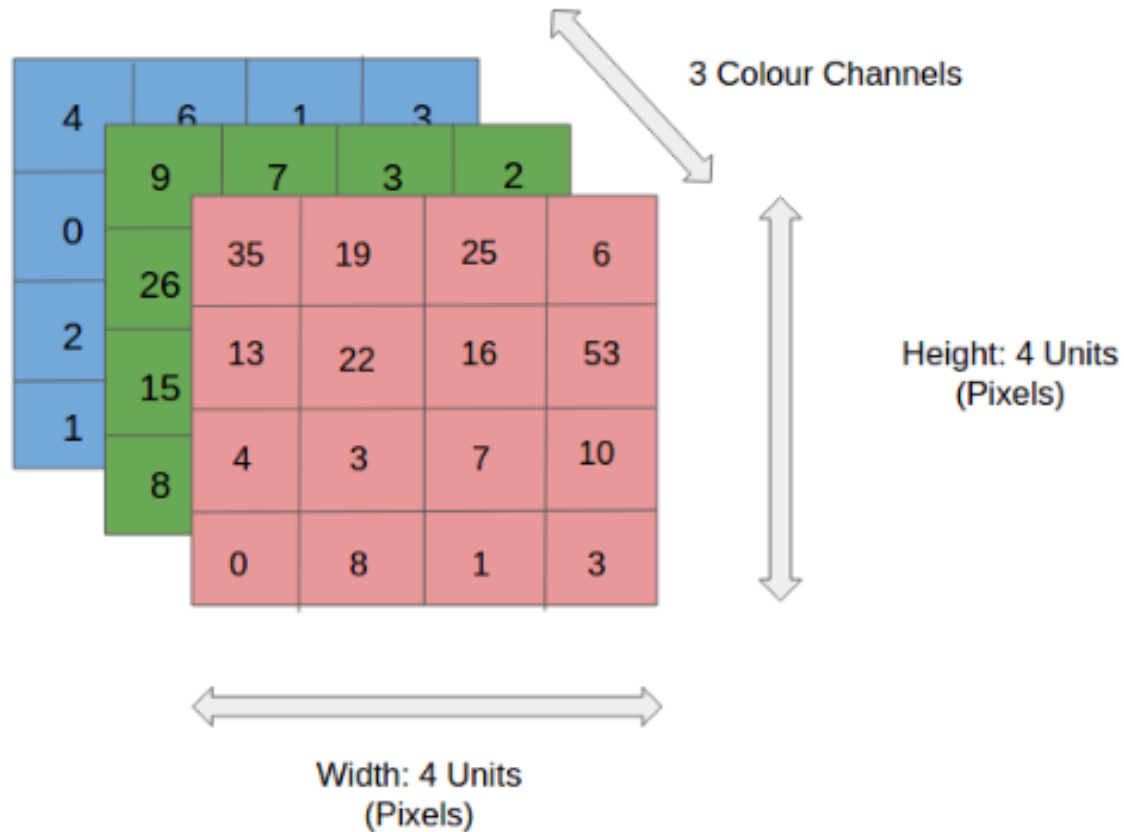
a) Learns Analogy



b) Similar Words have same angles

2017년 이후는 빅데이터 시대

- 비정형데이터에 대한 처리: 이미지 데이터(3차원의 데이터 형태; x,y, rgb)

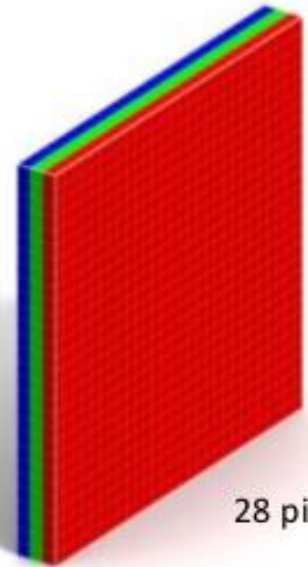
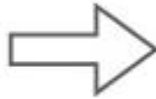


2017년 이후는 빅데이터 시대

- 비정형데이터에 대한 처리: 이미지 데이터(3차원의 데이터 형태; x,y, rgb)



Color Image
(RGB)



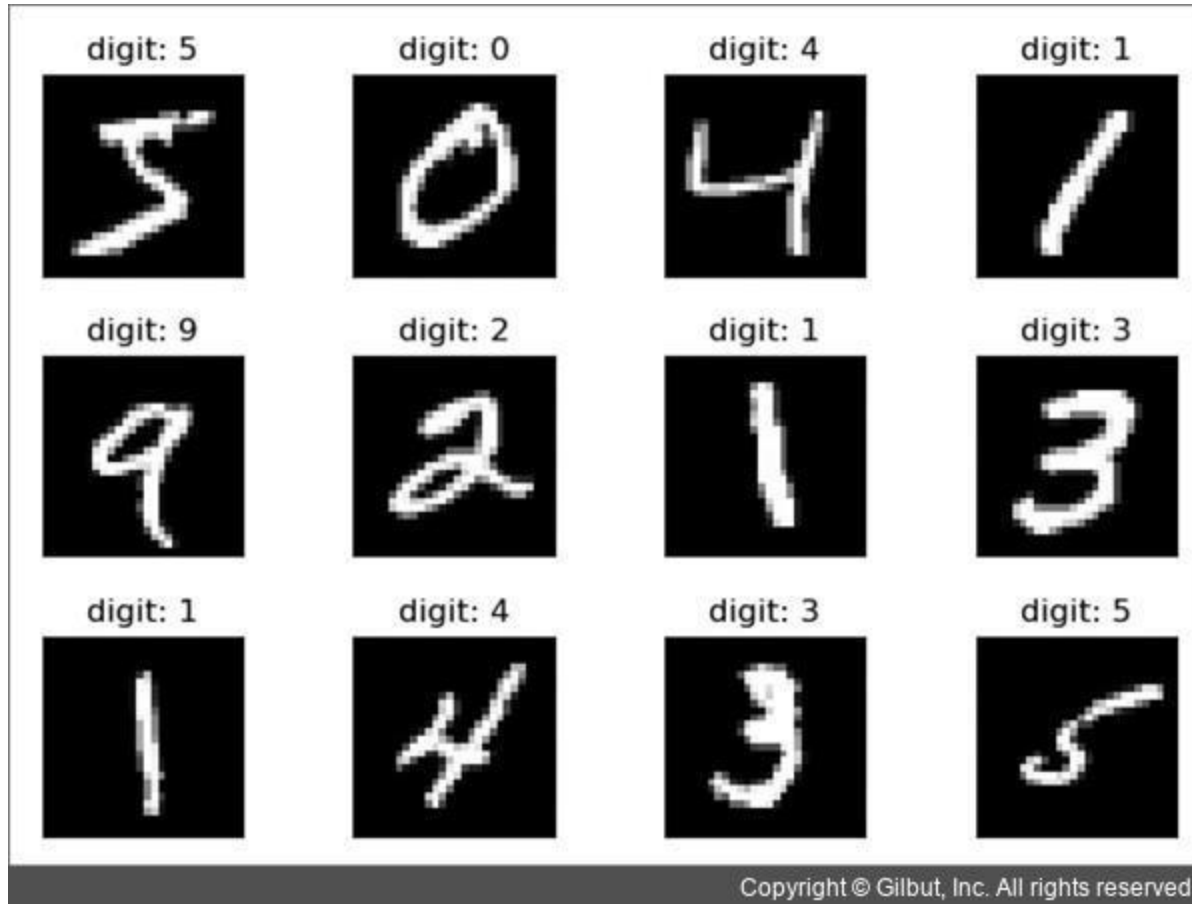
3 channels
(RGB)

28 pixels (height)

28 pixels (width)

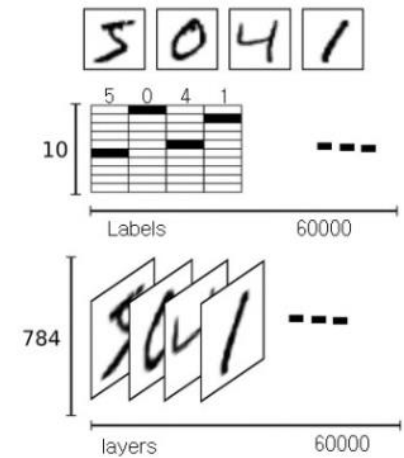
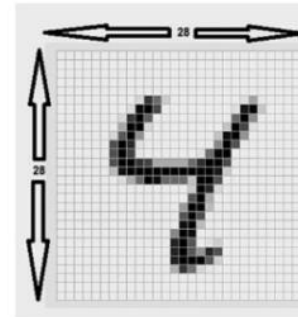
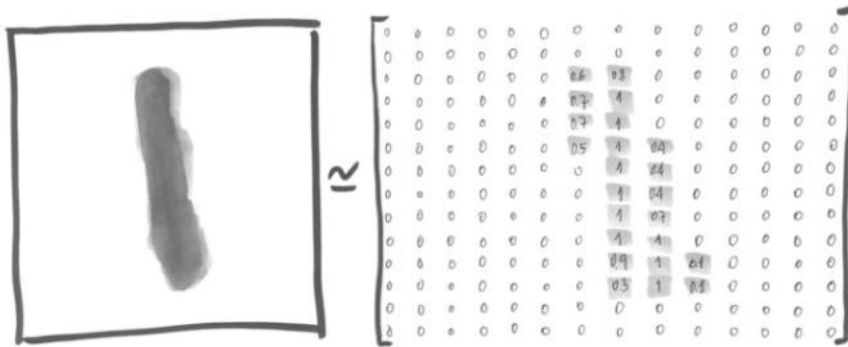
2017년 이후는 빅데이터 시대

- 비정형데이터에 대한 처리: 대표적인 이미지 데이터 MNIST



2017년 이후는 빅데이터 시대

● 비정형데이터에 대한 처리: 대표적인 이미지 데이터 MNIST

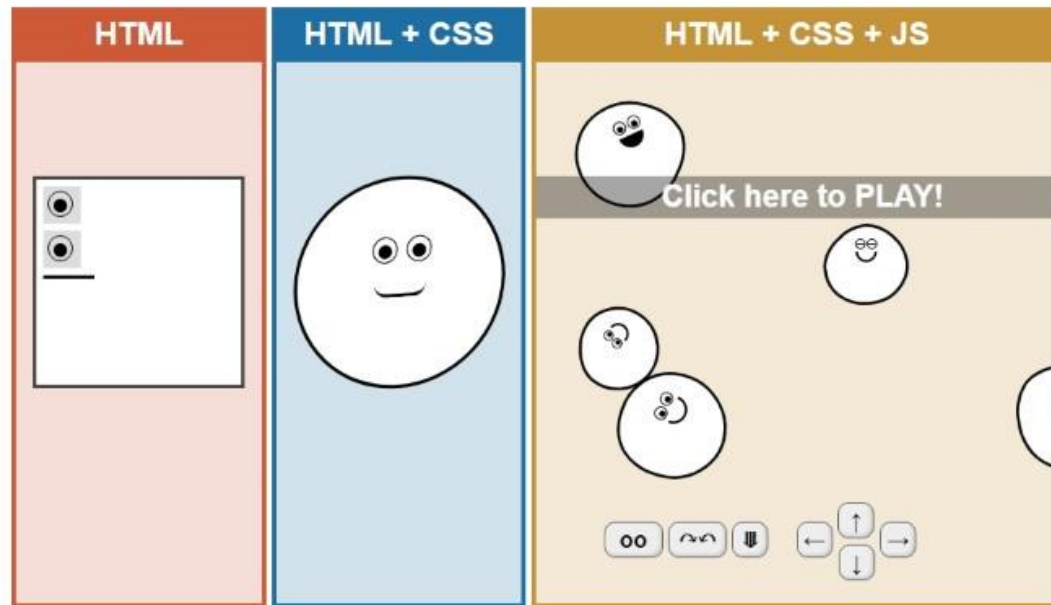


크롤링이란?

- 빅데이터 시대에 정보들을 우리가 분석하기 쉽고 활용하기 쉽게끔 데이터를 수집하는 행위를 크롤링(Crawling) 이라고 하한다.
- 원하는 데이터를 추출하는 스크래핑(Scraping)과 개념이 혼동되기도 하는데 사실 크롤링의 정확한 정의는 다양한 웹사이트의 페이지를 브라우징하는 작업을 말한다. 그런데 사실상 정보를 수집하기 위해선 브라우징만 하지 않기 때문에 페이지 안에 있는 데이터를 추출해서 가공하는게 대부분 최종 목표이다.
- 결국 크롤링은 스크래핑의 과정을 포괄적으로 내포한다.
- 이를 하기 위해서 프로그래밍 언어를 알아야 한다.

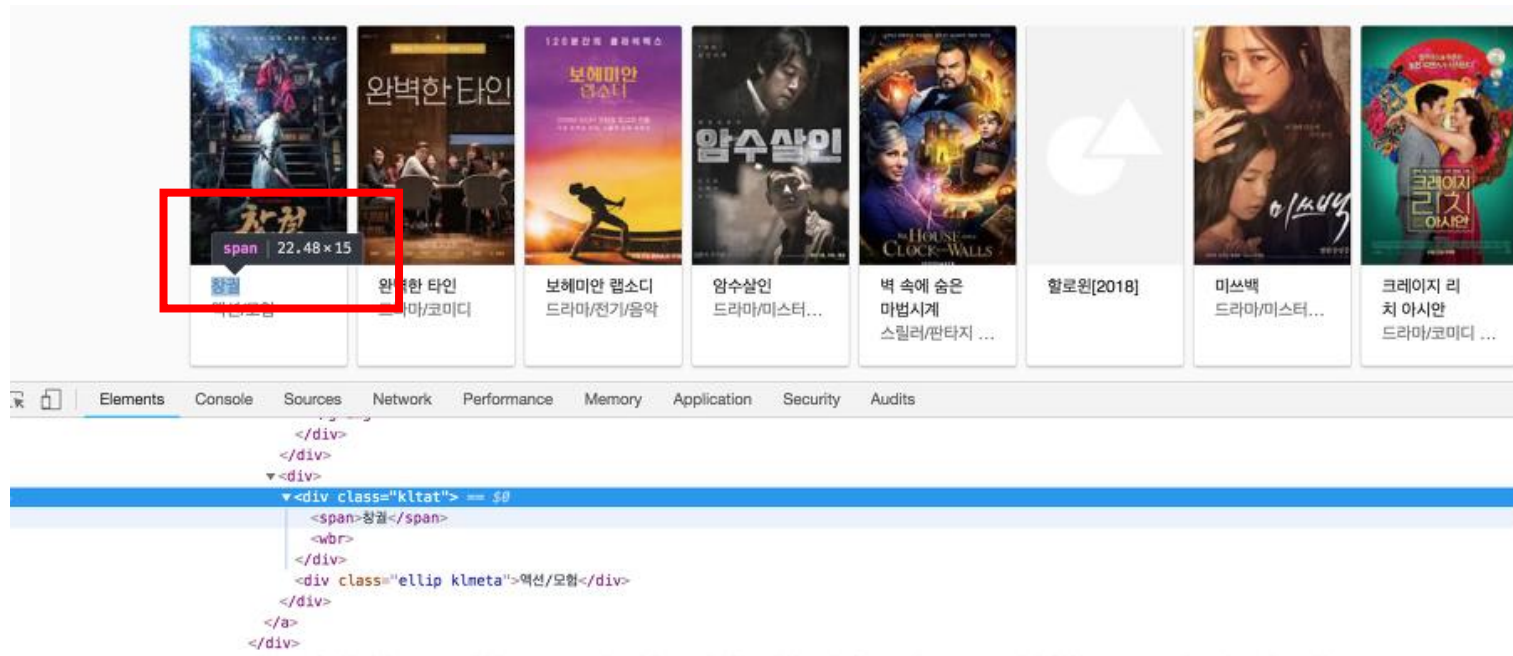
크롤링의 원리

- 크롤링의 원리를 알기 전에 먼저 웹사이트의 구성을 알 필요가 있다.
- 웹사이트는 **HTML** 문서로 작성되어 있으며, 이 문서에는 인터페이스를 참조할 수 있는 **CSS**파일과 페이지 상호작용을 위한 **JavaScript**파일을 참조하며 상호작용을 한다.



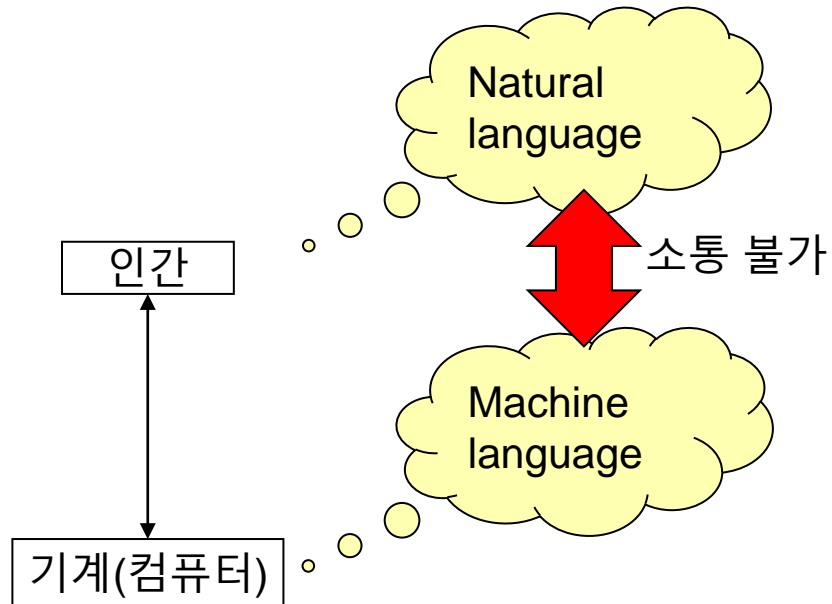
크롤링의 원리

- 모든 웹사이트의 **HTML** 문서를 확인할 수 있는데. 그리고 **HTML** 문서에 어떤 **CSS** 문서가 참조 되었는지, 어떤 내용이 들어가 있는지 확인할 수 있다.
- 우리는 크롤러를 만들 때, **HTML** 문서에서 태그 등을 찾아서 원하는 데이터를 추출한다.



프로그래밍 언어

● 인간과 기계의 소통



● 프로그래밍 언어

- 인간과 컴퓨터 사이에서 의사 소통을 가능하게 하는 인공적인 언어

프로그래밍 언어

- 컴퓨터는 0, 1 기계어 (binary code)로만 명령을 받는다.
- 초기엔 프로그램을 기계어를 통하여 프로그래밍을 했다.
 - 생산성, 기기 간 호환성, 디버깅 등 모든 면에서 비효율적.
- 컴파일러는 프로그래밍 언어로 쓰여진 소스 파일을 다른 언어로 바꾸어 주는 번역기.
- 프로그래밍 언어는 명령을 쓸 수 있는 방법론
- 프로그래밍 언어 자체도 “또 다른 프로그램”



프로그래밍 언어의 특징

- **문법 / Syntax**

어떠한 텍스트가 허용 가능한지 아닌 지에 대한 규칙

- **어떻게 컴파일 되는가?**

Compiler

Interpreter

- **Type System (Data Structure)**

Dynamic vs Static: Variable에 Type이 Associate되어 있는가?

Strong vs Weak: 메모리에 있는 모든 값에 타입이 Associate되어 있는가?

프로그래밍 언어의 특징

- **Functional Language**
- **Object-Oriented Language**
- **인기: Community의 활성화**
- **목적:**

Database: SQL, Key-Value

Front-End Web: HTML, CSS, JavaScript

Back-End Web: PHP, Python, Ruby, Java, Rust 등등

Data-Science: Python, Matlab, R, Julia

Operating System: C

Application: C++, Java

나는 지금까지 어떤 프로그램 언어를 배웠는가?

- **고등학교 (교과목) : Excel에서의 기본 컴퓨터 언어**
- **학부 (경제) : Eviews, R(시계열 데이터와 횡단면 데이터 분석)**
- **석사 (수리경제) : MATLAB(금융공학 미 최적화, 시뮬레이션)**
- **연구실 (통계연구실, 금융연구소): R, Python(여러 정형 데이터 통계 모델링)**
- **박사 (데이터 사이언스) : Python(이미지, 텍스트와 같은 비정형 데이터)
Javascript(웹 프론트 개발)
Solidity(블록체인)**

프로그래밍 언어의 미래

개인적인 생각:

System: C , C++ → Rust

Web: JavaScript → TypeScript

Application: Java → Go

Data Science: Python, MATLAB, R → Julia

데이터 사이언티스트 vs 개발자?

● 데이터 사이언티스트

= 데이터 애널리스트, 데이터 엔지니어?

● 고전적 관점

데이터 애널리스트 (통계 전문가)

→ Excel, SPSS를 잘 다루고 데이터를 통해 문제 해결

→ 빅데이터를 다루거나 알고리즘 모델을 능력 X

데이터 엔지니어

→ 데이터베이스를 잘 다루고 분석가를 위해 핵심 데이터를 추출

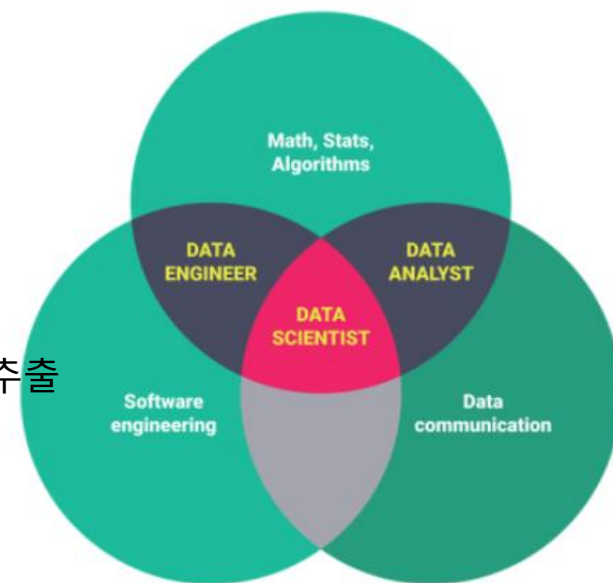
→ 데이터 분석이나 머신러닝 능력 X

● 최신 트렌트

데이터 과학자

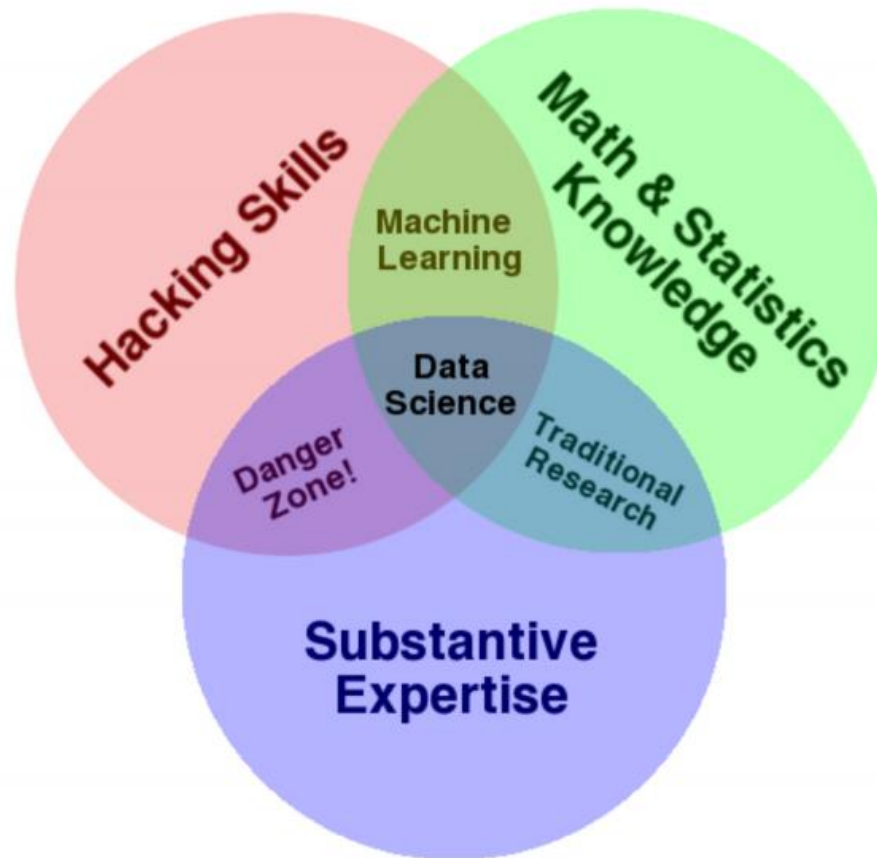
→ 파이썬 및 다른 컴퓨터 언어를 다룰 줄 알고 머신러닝과 통계에 능숙

→ 다양한 스킬셋 보유



2017년 이후는 빅데이터 시대

인공지능(AI), 머신러닝 그리고 딥러닝 - Drew Conway's Venn Diagram(2010)



2017년 이후는 빅데이터 시대

인공지능(AI), 머신러닝 그리고 딥러닝 - Ullman's Venn Diagram(2010)

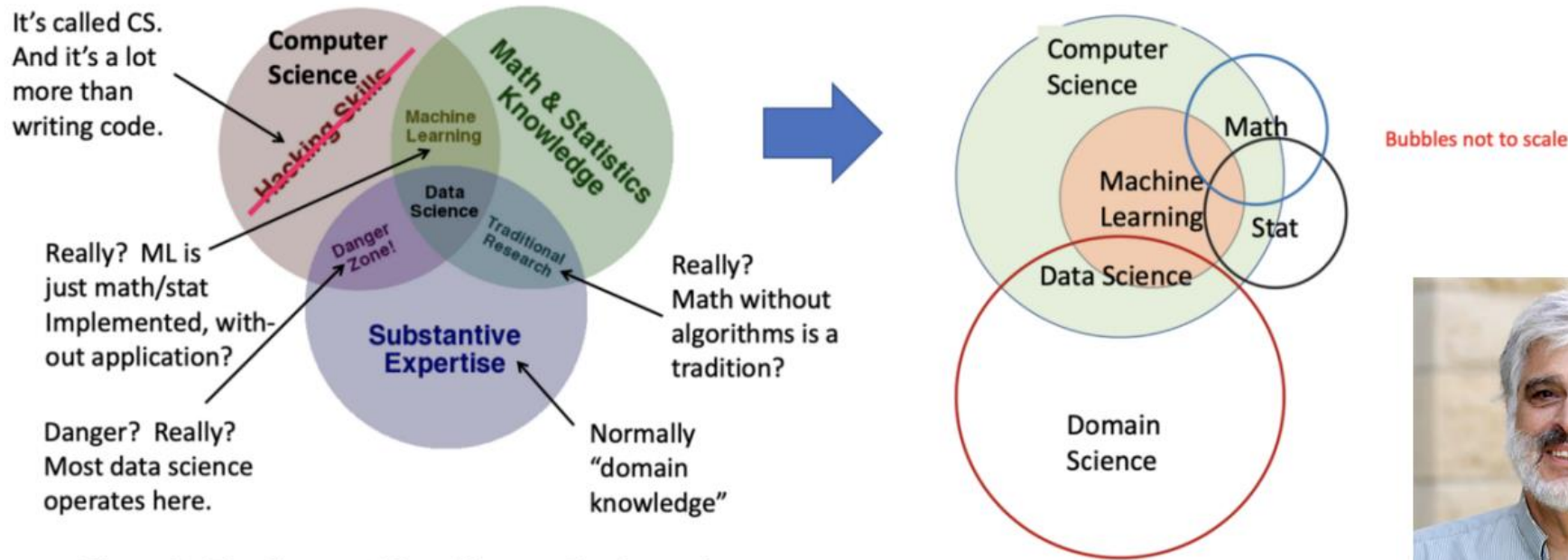
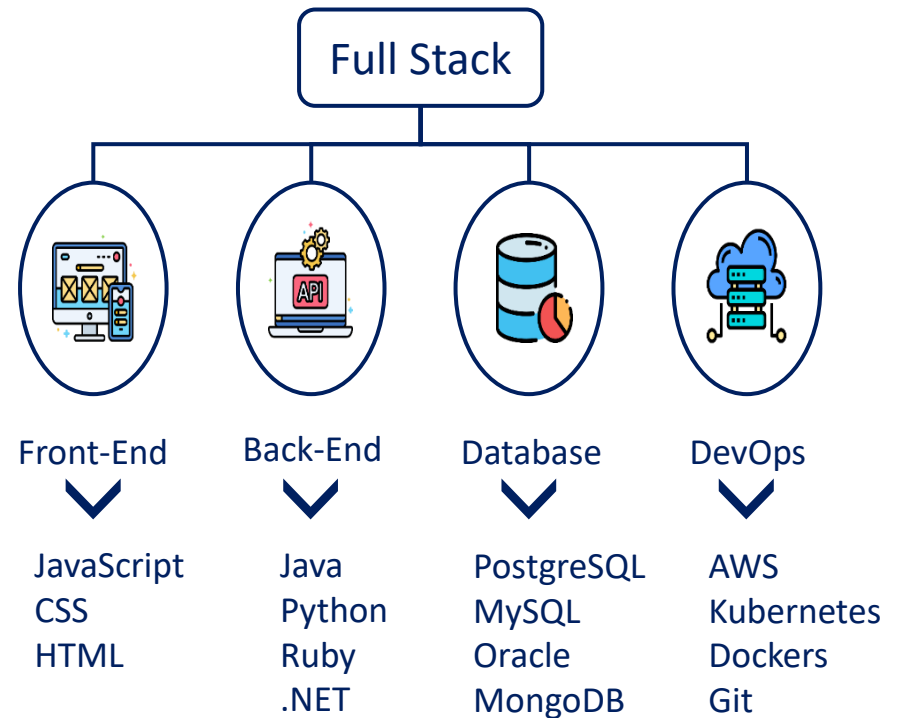
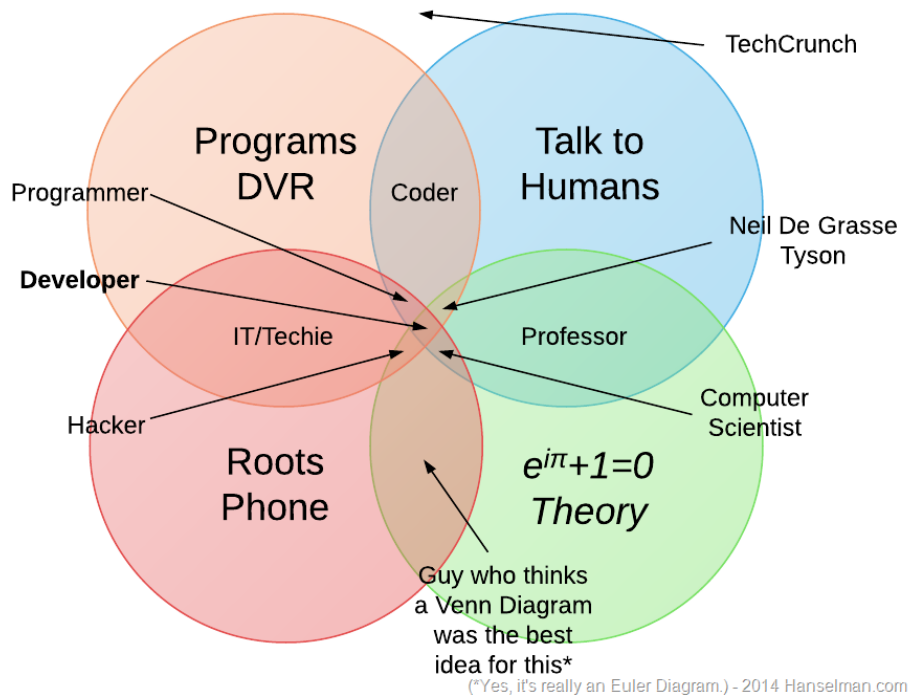


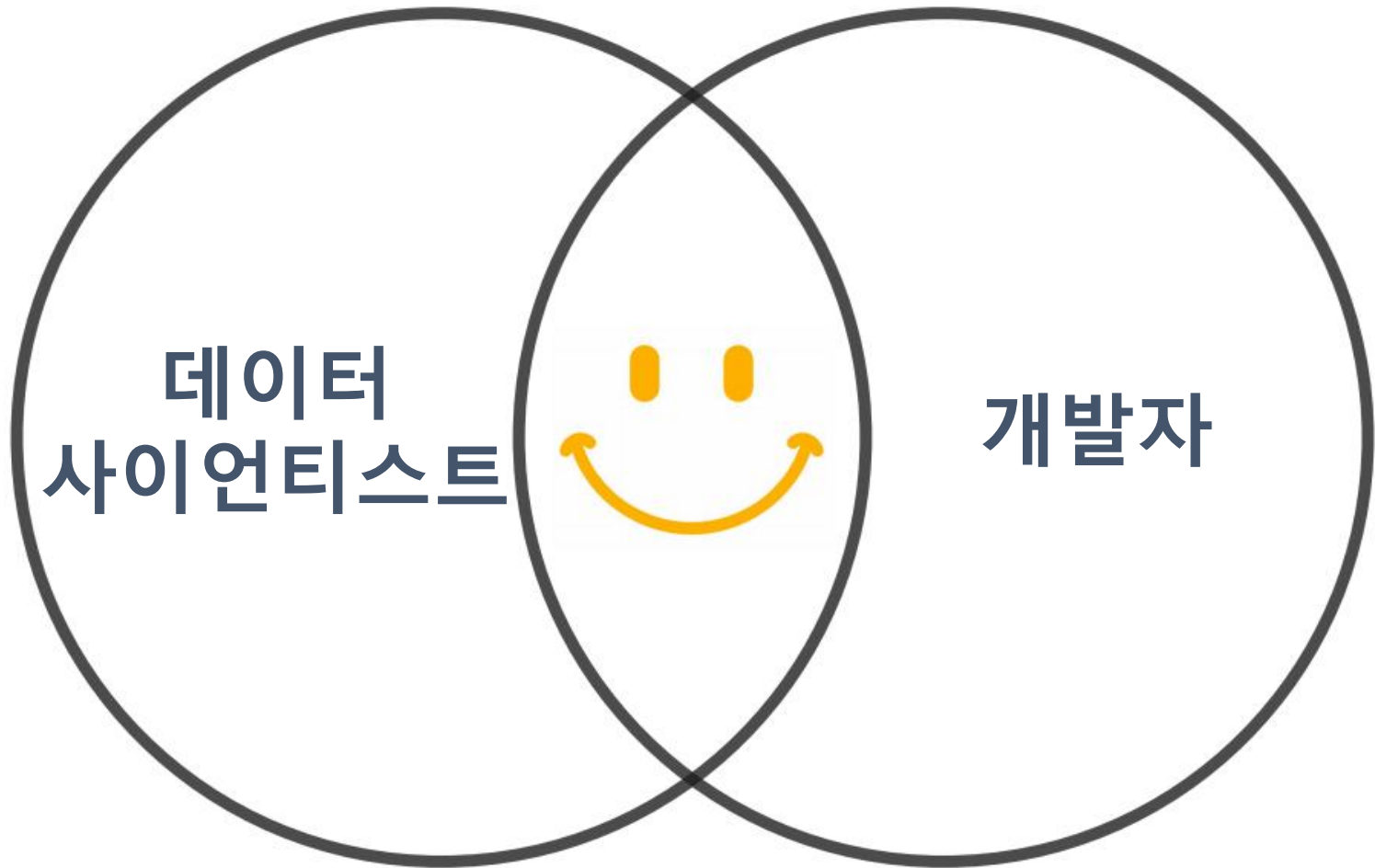
Figure 1: The Conway Venn diagram for data science



데이터 사이언티스트 vs 개발자?



데이터 사이언티스트 vs 개발자?



Python의 역사



Guido van Rossum



Python 1: 1994 A middle ground between C and shell scripts.



Python 2: 2000



Python 3: 2008

이 강의에서는 Python 3을 사용합니다

Python의 역사

프로그램보다 프로그래머를 중요하게 생각하는 언어

“

가장 아름다운 하나의 답이 존재한다

”

Python의 철학

아름다운 것이 추한 것보다 낫다.

명시(explicit)적인 것이 암시(implicit)적인 것보다 낫다.

간결(simple)한 것이 복잡적(complex)인 것보다 낫다.

복합적(complex)인 것이 복잡한(complicated) 것보다 낫다.

수평적(flat)인 것이 내포된(nested) 것보다 낫다.

여유로운(sparse) 것이 밀집한(dense) 것보다 낫다.

가독성은 중요하다.

특별한 경우들은 규칙을 어길 정도로 특별하지 않다.

허나 실용성(practicality)은 순수성(purity)을 이긴다.

오류는 절대로 조용히 지나가지 않는다.

Python Summary

- **Python은 1990년 귀도 반 로섬이 개발**

- 플랫폼에 독립적인 인터프리터 언어
- 객체 지향적, 동적 타이핑 언어
- C언어로 구현



- **활용도가 높은 언어로, 다양한 분야 및 회사에서 사용되고 있음**

- 다양한 라이브러리가 존재하며, 사용하기 쉬움
- Numpy, pandas, scipy, scikit-learn, ...

- **문법이 쉽고 인간이 직관적으로 이해하기 쉬운 형태**

JAVA

```
for (int i=1; i<10; i++){  
    System.out.println(i);  
}
```

Python

```
for i in range(1,10):  
    print(i)
```


Python 설치 - Anaconda

● Anaconda

- Python에 추가적인 기능들을 합쳐놓은 프로그램
- Python만 설치할 경우 추가 기능들을 따로 다운로드 받아야 하는 불편함 있음
- 비유
 - Python만 설치 : 핸드폰을 공장초기화한 상태
 - Anaconda를 설치 : 공장초기화된 핸드폰에 유명한 앱들을 깔아놓은 상태
- 윈도우 32비트 or 64비트 확인
 - 탐색기 열기 : 윈도우키 + e
 - 내 컴퓨터 우클릭 - 속성

시스템

프로세서:	Intel(R) Core(TM) i5-4690 CPU @ 3.50GHz 3.50 GHz
설치된 메모리(RAM):	16.0GB
시스템 종류:	64비트 운영 체제, 64 기반 프로세서
펜 및 터치:	이 디스플레이에 사용할 수 있는 펜 또는 터치식 입력이

Python 설치 - Anaconda

● Anaconda

- <https://www.anaconda.com/products/individual>
- 다운로드 버튼을 누르거나, 페이지 맨 아래로 내려가면 다운 가능

Download

- Anaconda Installers에서 32비트 또는 64비트 선택하여 다운로드



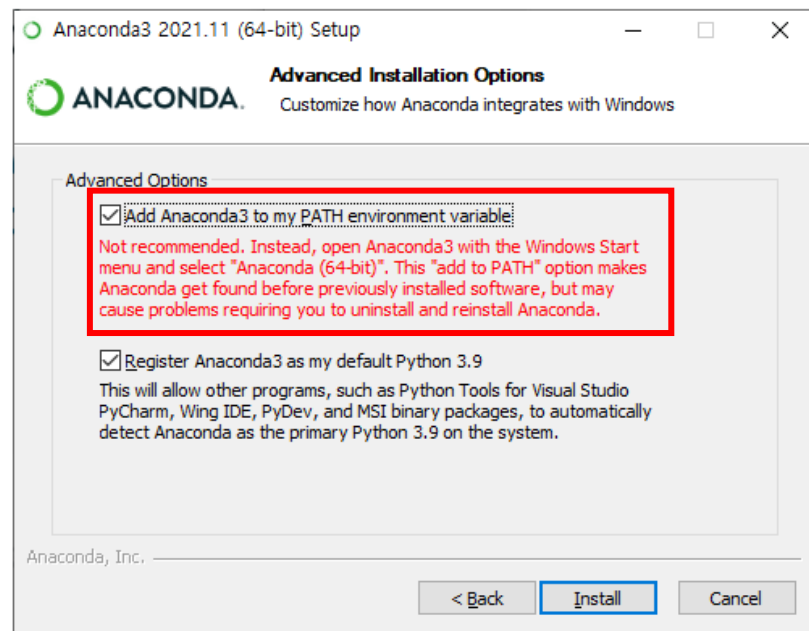
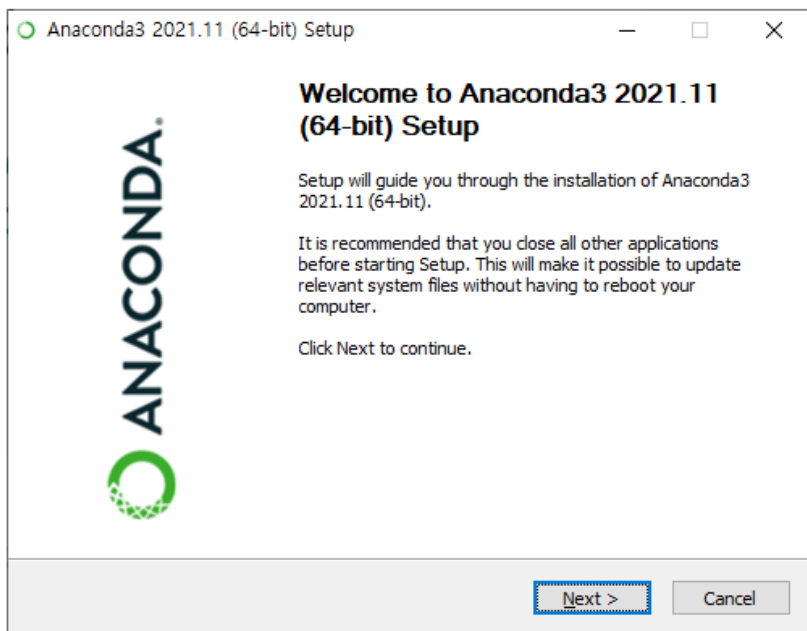
- 다운로드가 되면 더블클릭하여 설치

Python 설치 - Anaconda

● Anaconda

- Anaconda3~~~.exe 설치 시작

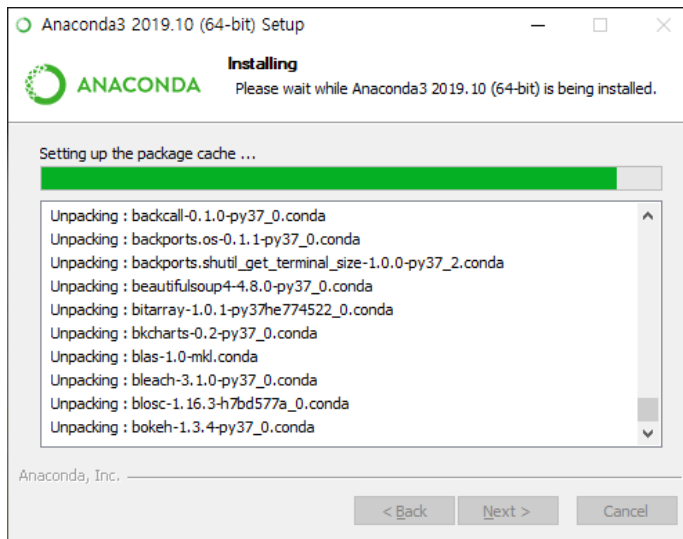
- 아래 체크박스 꼭 클릭!
나중에 가상환경 설치에 중요!



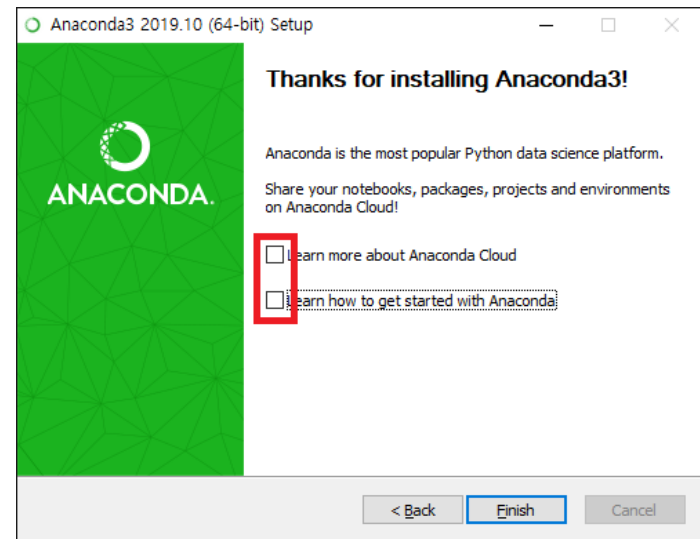
Python 설치 - Anaconda

● Anaconda

- 마지막 단계가 오래걸림

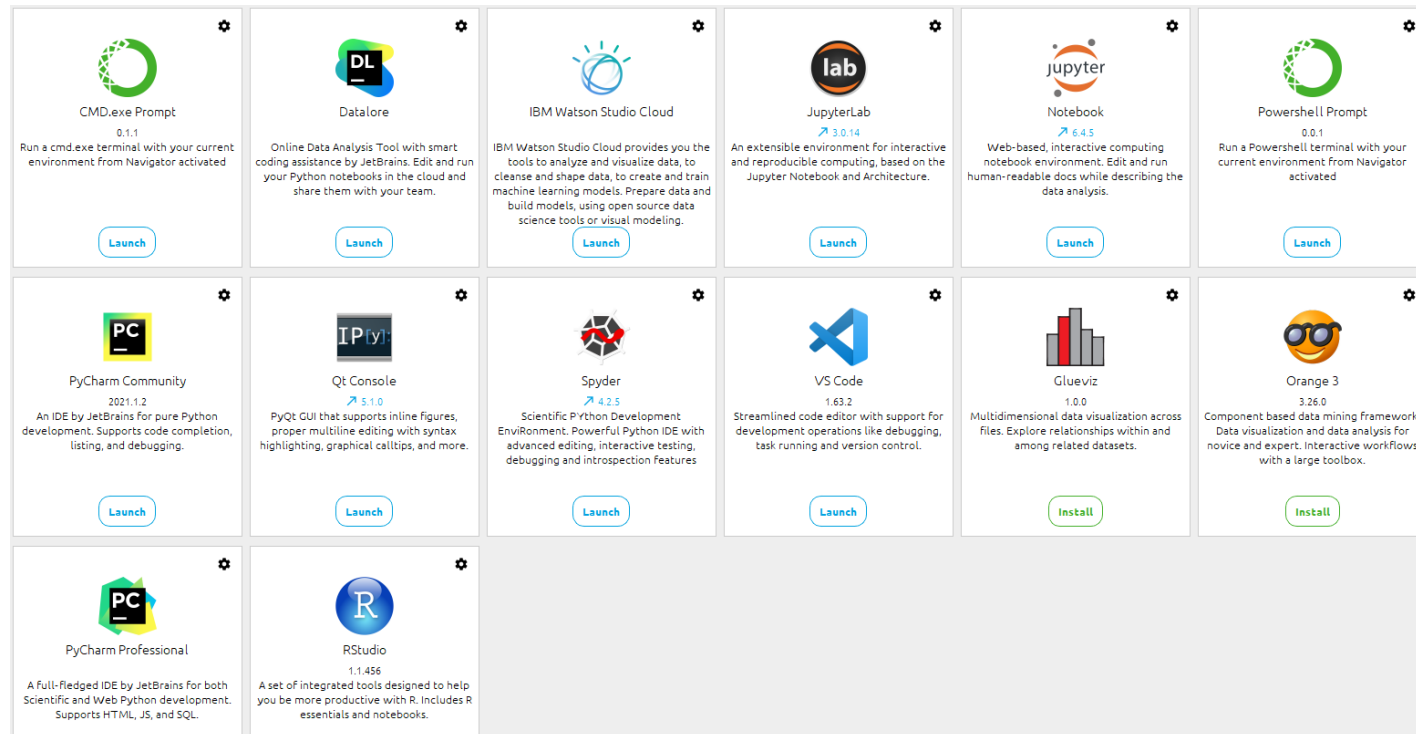


- 설치 완료 후에는 2개 박스에 체크 제거하고 Finish



Python IDE(Integrated Development Environment)

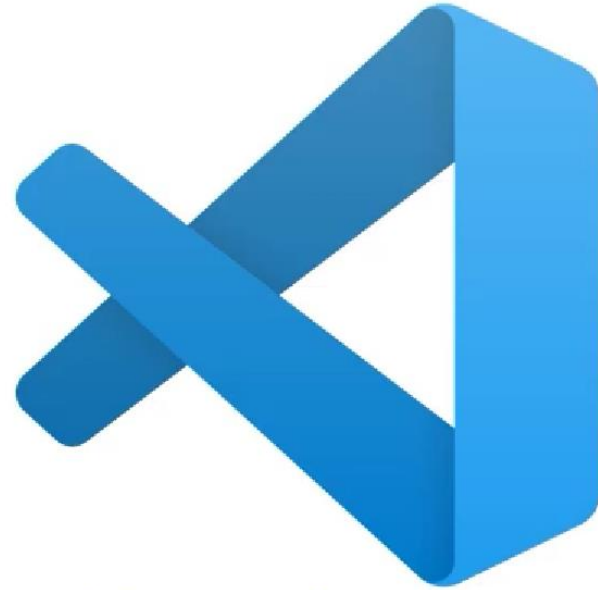
- **IDE(Integrated Development Environment)**란 통합된 개발 환경으로, 개발자를 위한 종합 프로그램을 말하며, 쉽게 말해 개발 도구 모음집이라 보면 된다.
- **Anaconda 실행 시 나오는 IDE 종류**



❤ 파이썬 IDE의 양대산맥



파이참

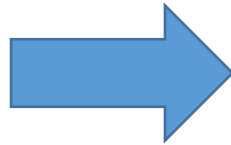


Vs code
비주얼 스튜디오 코드

❤ 파이썬 IDE의 양대산맥



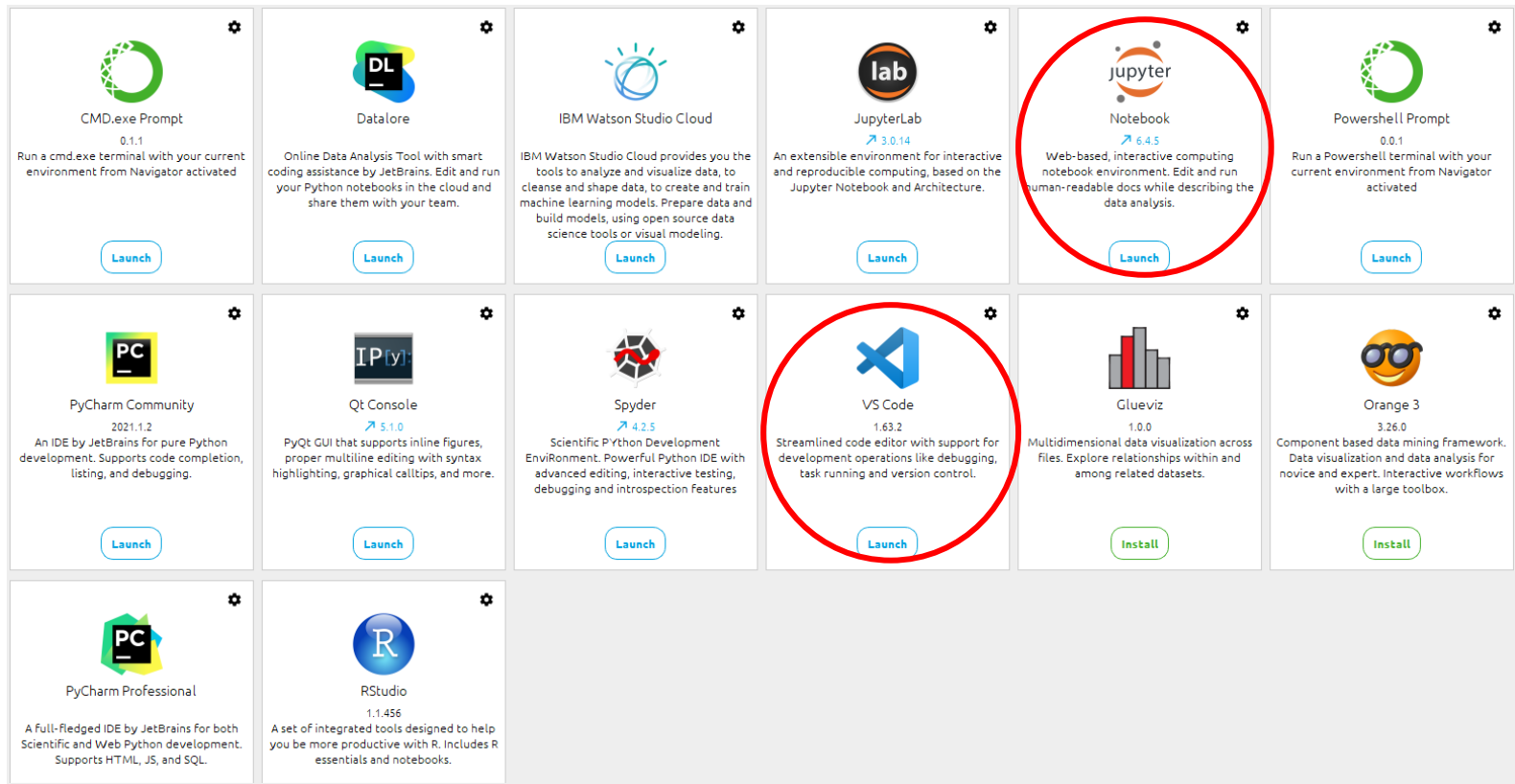
파이참



Vs code
비주얼 스튜디오 코드

Python IDE(Integrated Development Environment)

- 본 강의에서는 Jupyter Notebook과 VS code를 배운다.



Jupyter Notebook

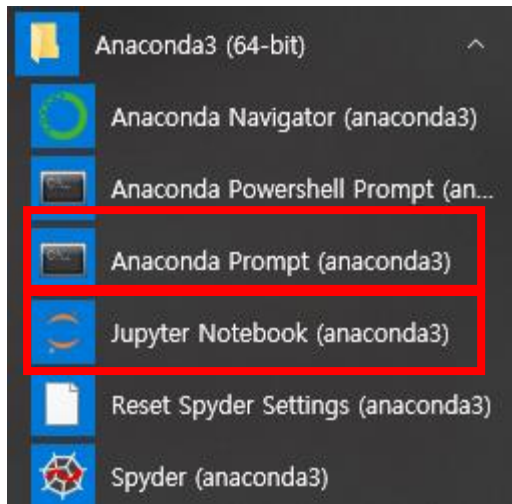
- 대화형 컴퓨팅을 위한 **command shell**
- 40여종 이상의 다양한 언어 지원
- 실시간 대화형 위젯
 - 간단한 코드의 작동 여부나 작동 원리, 결과를 실시간으로 확인하기 편리
- 브라우저 기반 노트북
 - 노트북 파일은 .ipynb 파일로 저장
 - 웹 브라우저로 .ipynb 파일 오픈 가능
- Anaconda 설치 시 자동 설치
- cmd창 실행 후 cd 명령어로 원하는 디렉터리로 이동한 후 jupyter notebook 입력 하거나 Anaconda Navigator 이용
 - 웹 브라우저 자동 실행
 - 자동 실행되지 않는 경우 웹 브라우저 실행 후 localhost:8888 입력
- 웹 브라우저를 닫아도 jupyter은 계속 실행, 종료하려면 cmd창에서 ctrl + C



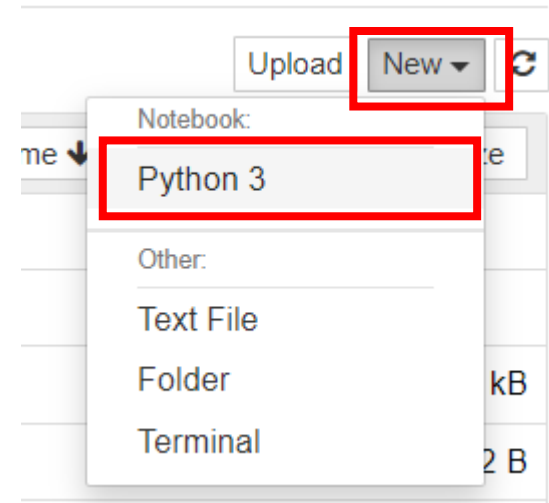
Jupyter Notebook

● Jupyter Notebook 사용하기

- Anaconda를 설치하면 Jupyter Notebook이란 프로그램도 설치됨
- 시작 – 프로그램 – Anaconda3 – Jupyter Notebook



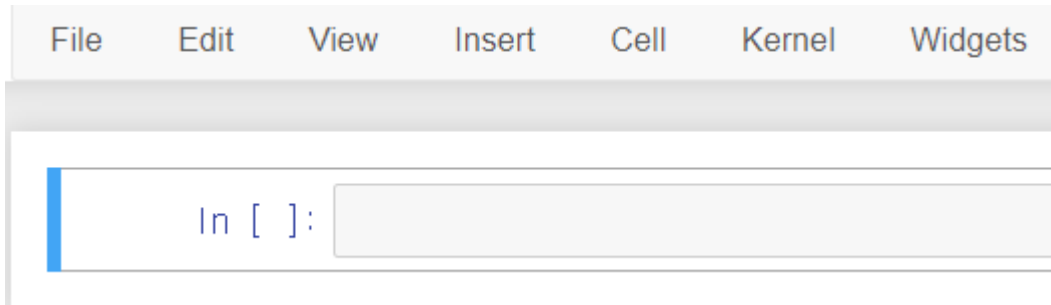
- 브라우저가 실행되며 파일 목록이 뜸
 - C:\Users\₩사용자명\폴더
- 원하는 폴더로 이동해서 New – Python 3 선택



Jupyter Notebook

● Jupyter Notebook 사용하기

- 다음과 비슷한 창이 뜸



- 파이썬 코드를 입력하고 Shift + Enter 를 입력하면 실행됨
- print를 할 때 print(~~~~~) 할 필요 없이 ~~~~~만 입력하면 출력됨
- 코딩 처음일 때, 한줄 한줄 출력해볼 때 유용

- * 유용한 단축키: - bb: 중간에 입력창 삽입
 - dd: 중간에 입력창 제거
 - m: 입력창 주석처리

VS Code

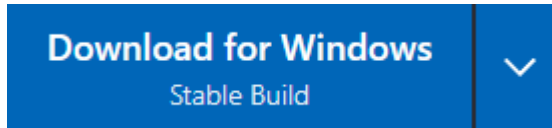
- PyCharm 대비 가볍고 서버로 띄워 리모트 작업할 때 용이
- 가상환경 설치 또한 간편
- 파이썬 뿐만 아니라 다른 언어를 사용할 때도 훨씬 유용
- 상대적으로 사용하는 비-파이썬 개발자가 많아
유용한 Extension도 많음
- Linux 기반 Terminal 창도 같이 켜서 쓸 수 있음



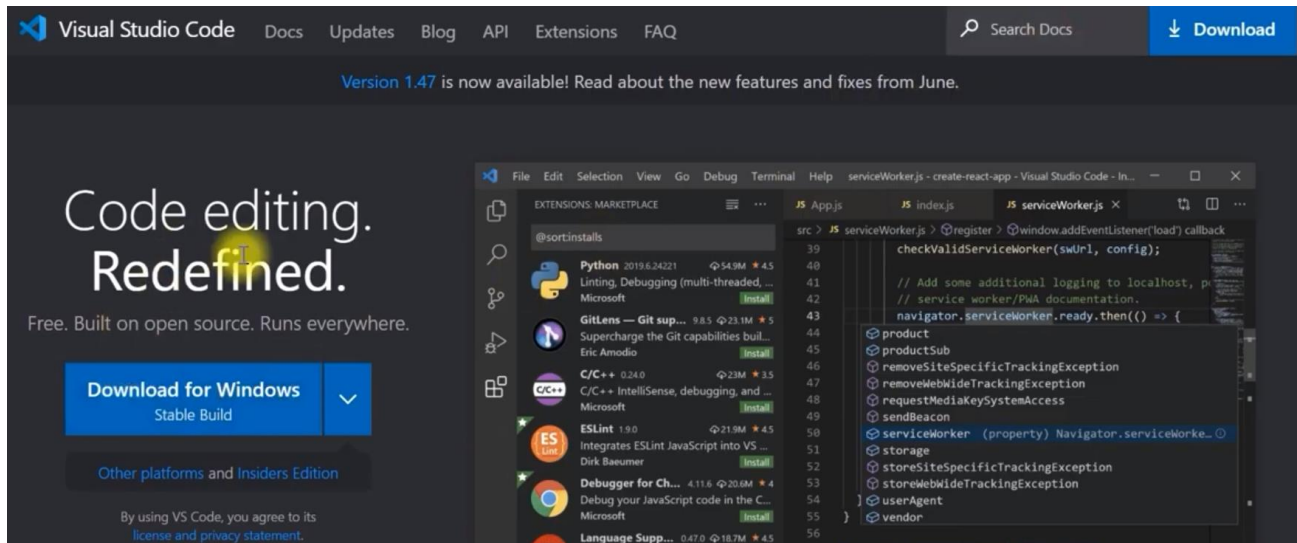
VS Code

● VS Code

- <https://code.visualstudio.com>
[Visual Studio Code - Code Editing. Redefined](#)
- 다운로드 버튼을 누르거나, 페이지 위 다운로드 버튼 통하고 다운 가능

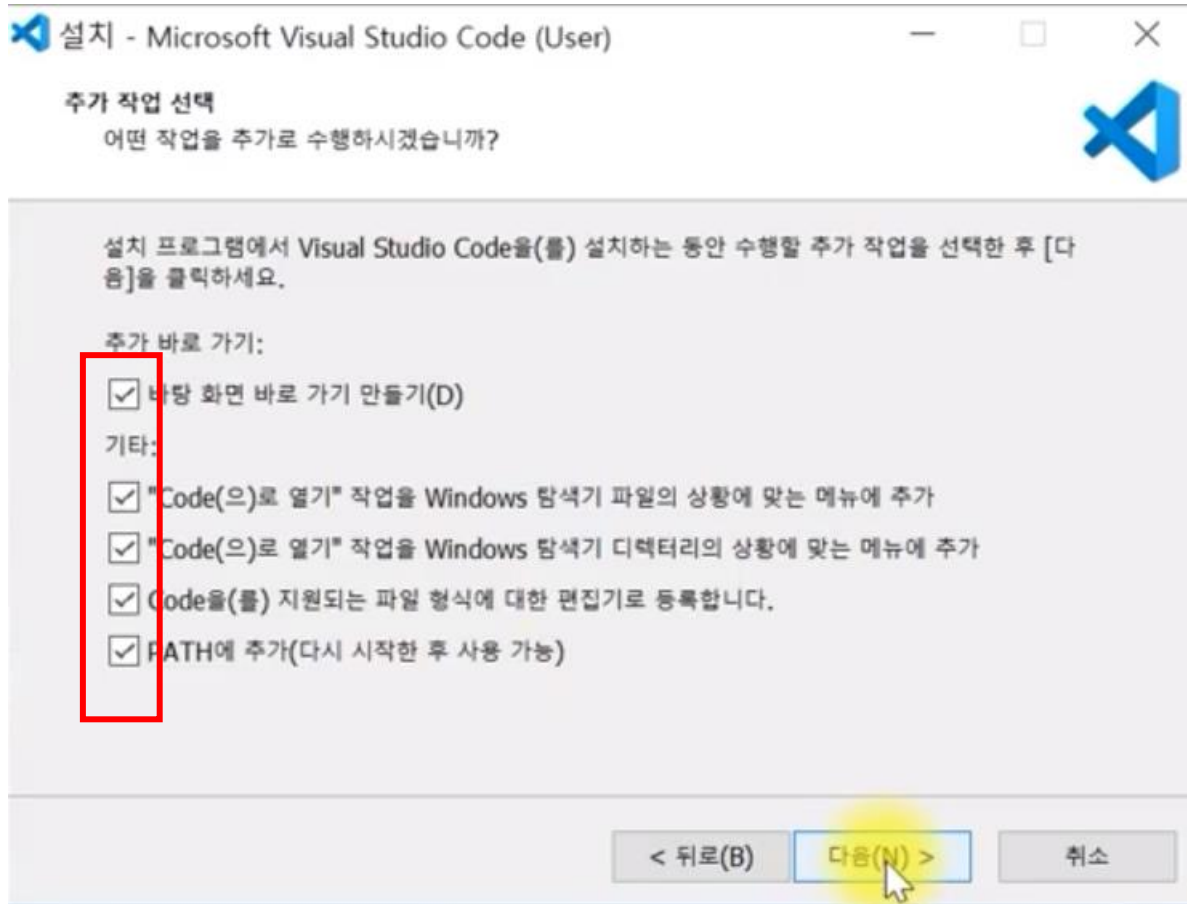


- 다운로드가 되면 더블클릭하여 설치



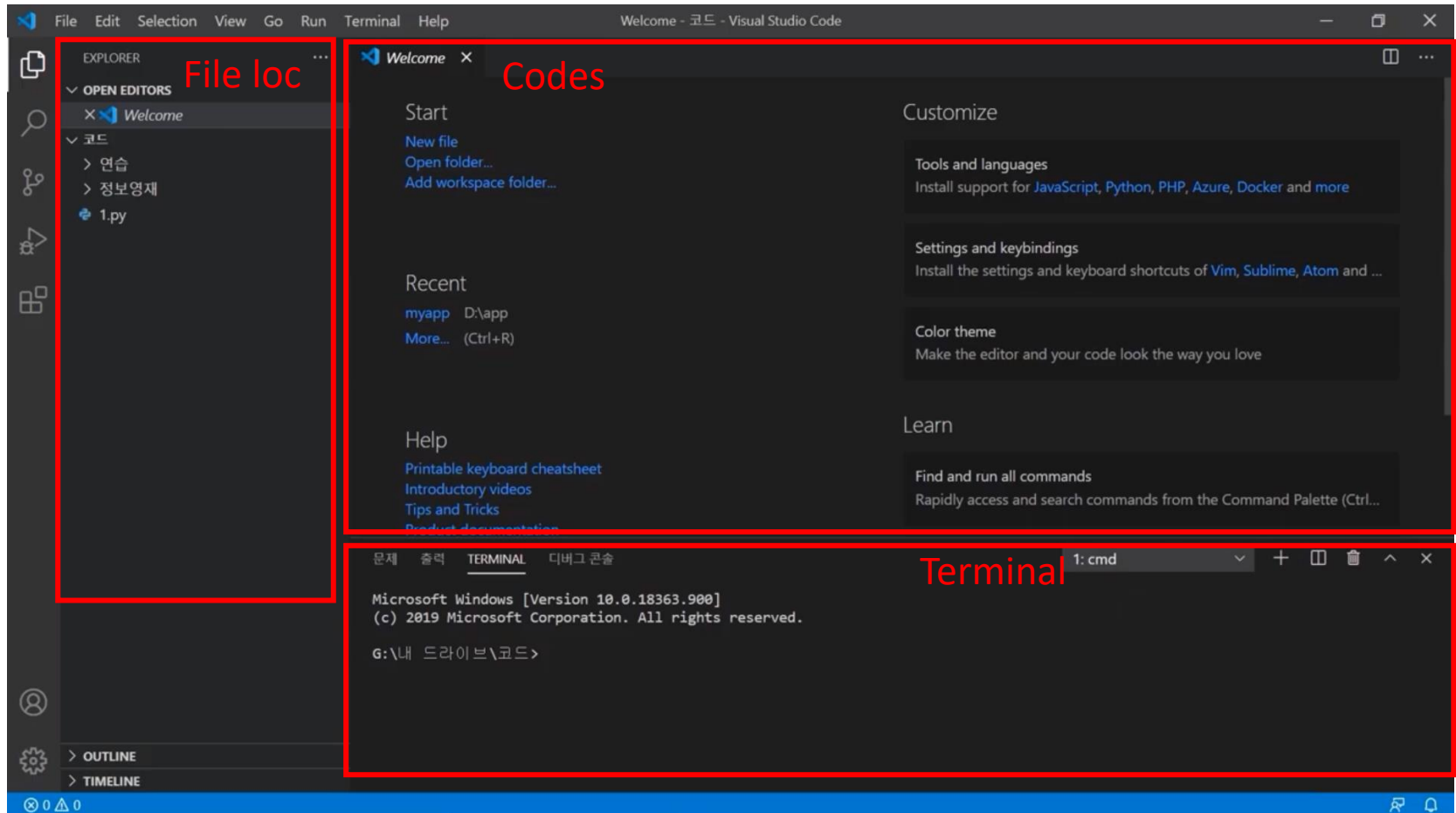
VS Code

● VS Code



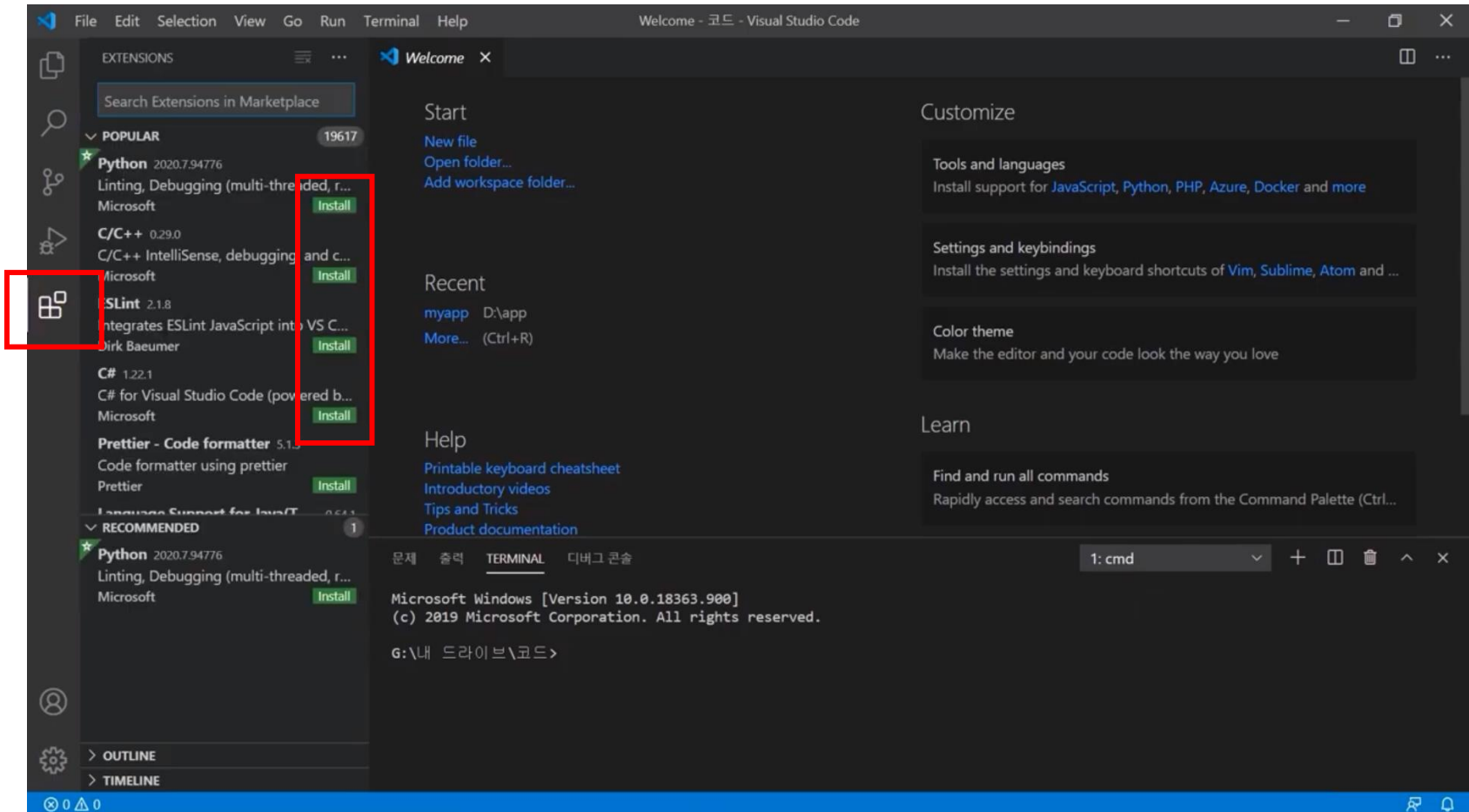
VS Code

● VS Code



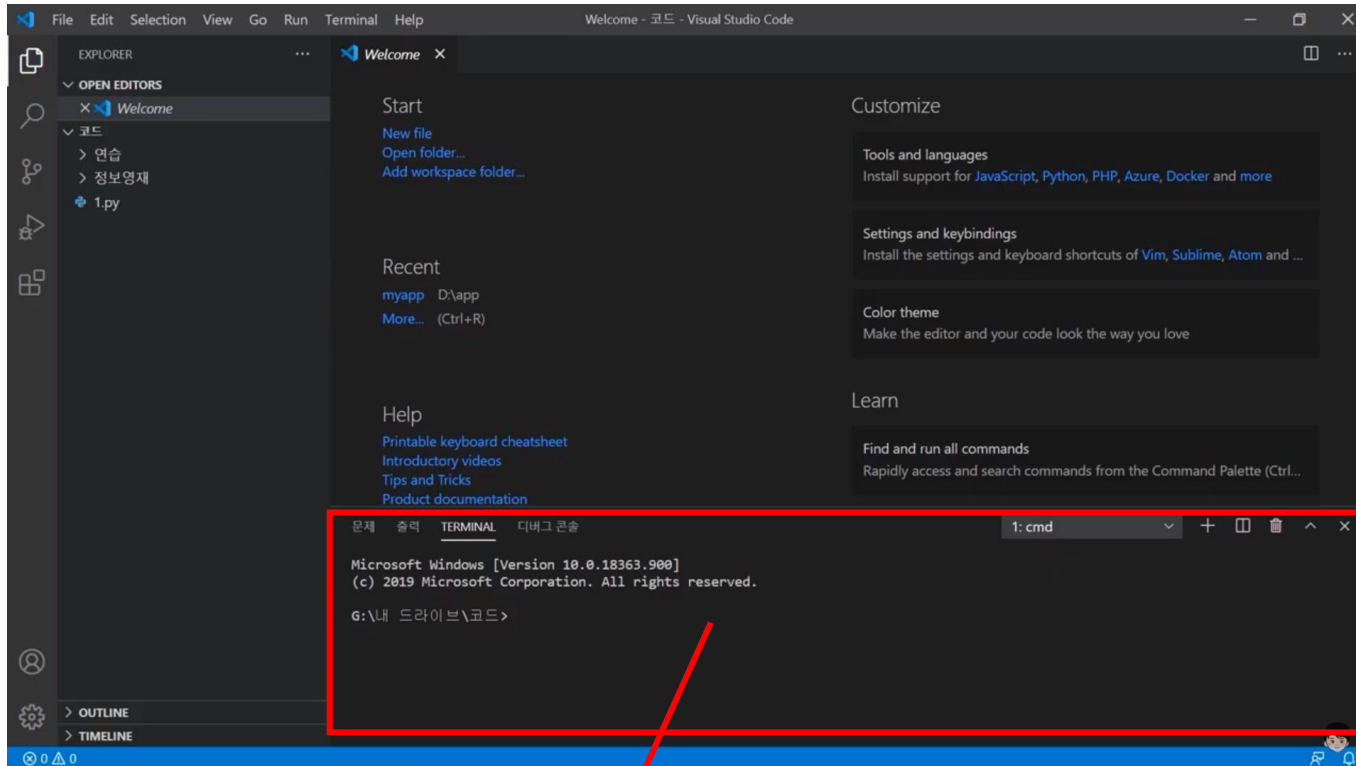
VS Code

● VS Code extension 설치



VS Code

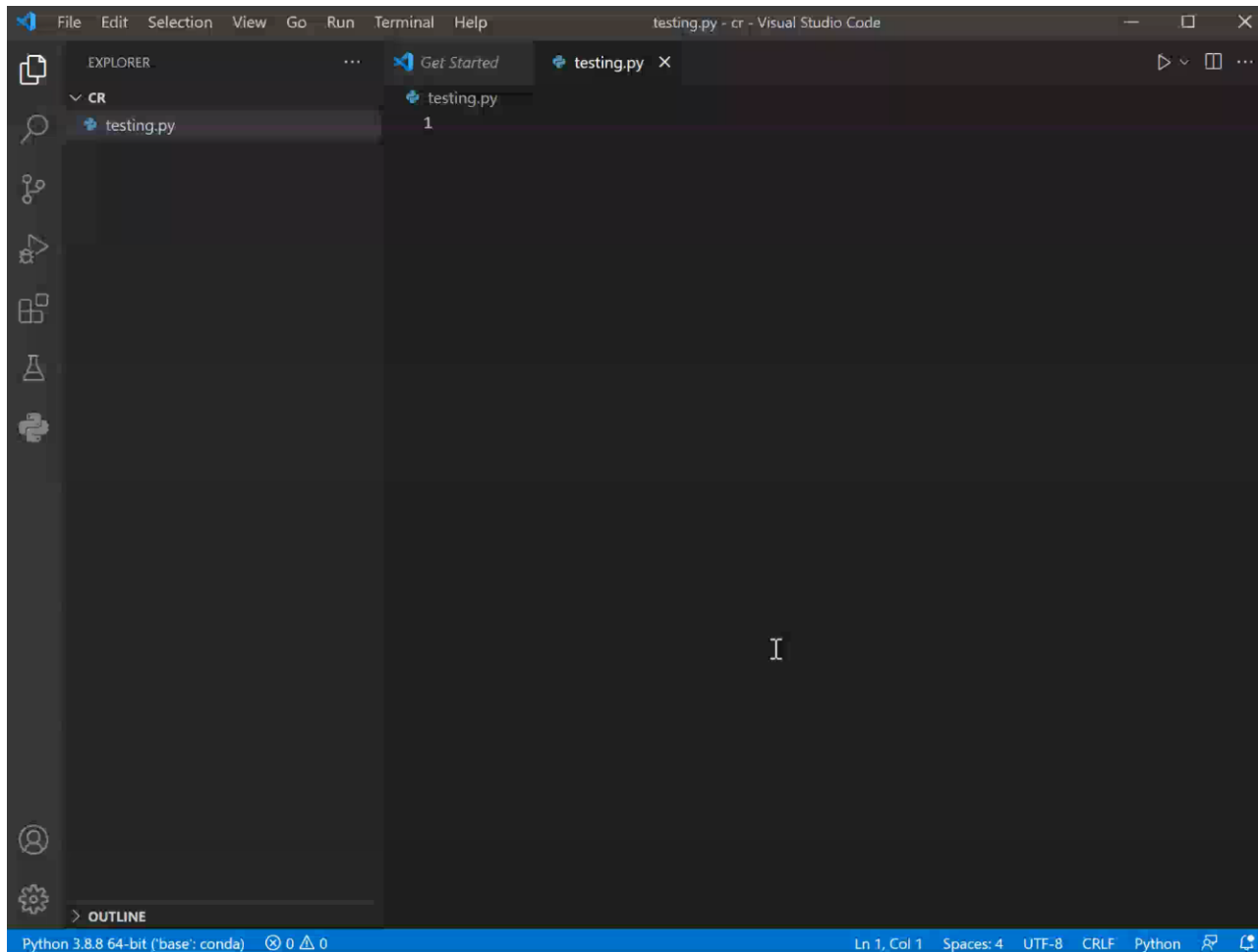
● VS Code conda 환경 활성화(activation)



```
C:\\<conda 설치 경로>\\Scripts\\activate
```

VS Code

● VS Code conda 환경 활성화(activation)



Google Colab

● (참고) Google Colab 사용하기

- <http://colab.research.google.com>
- Jupyter Notebook + Google Drive
- 소스코드 공유, 협업이 쉬움
- 컴퓨터에 파이썬이 전혀 깔려있지 않아도 괜찮음
- Google의 자원을 사용하기 때문에 사양 낮은 노트북 등에서도 복잡한 프로그래밍 가능
- Google의 GPU 제한적으로 무료 활용 가능 - 딥러닝 등

파이썬 가상환경(virtualenv) 구성

- 가상환경(virtualenv)은 여러 개의 파이썬 프로젝트가 하나의 컴퓨터에서 충돌을 일으키지 않고 존재할 수 있도록 하는 과정.
- 각 프로그램별로 완전히 독립적인 가상의 환경을 만들어서 각 프로그램별로 패키지/라이브러리/모듈 등의 버전을 별도로 지정
- Windows나 Ubuntu 환경 모두 사용 가능
 - 여기서는 windows 10 기준으로 설명
- Python IDE 설치 후 anaconda prompt 혹은 terminal 검색 및 실행
 - 명령어(<https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html> 참조)를 이용하여 가상환경 구성

```
conda update conda 입력하여 conda 업데이트 우선 진행  
conda create -n [환경이름] python=3.7 실행하여 가상환경 만들기  
conda activate [환경이름] 입력하여 가상환경 실행
```

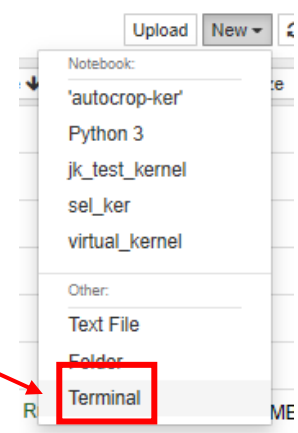
* Jupyter notebook에서는 추가 작업 필요:

```
conda install jupyter notebook  
python -m ipykernel install --user --name [환경이름] --display-name  
"[디스플레이 이름]"
```

파이썬 패키지 설치

- 필요한 패키지(라이브러리)를 Python 환경에 설치하는 명령어(cmd or terminal)
 - Python 3만 설치되어 있는 경우 : `pip install <package_name>`
 - Python 2와 Python 3이 같이 설치되어 있는 경우 : `pip3 install <package_name>`
 - 기존에 있는 패키지를 최신 버전으로 업데이트할 때 : `pip install --upgrade <package_name>`
 - 현재 설치된 패키지들 나열 : `pip(또는 pip3, conda) list`
 - 설치된 특정 패키지의 정보 확인 : `pip(또는 pip3) show <package_name>`

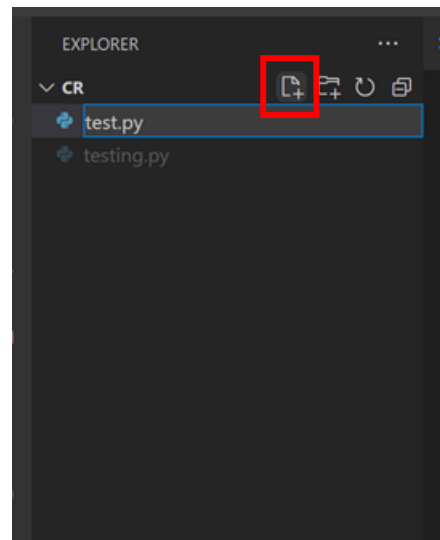
* Jupyter notebook에서는 Terminal을 키지 않고 "!"를 pip앞에 쓰면 IDE 자체에서 실행 가능



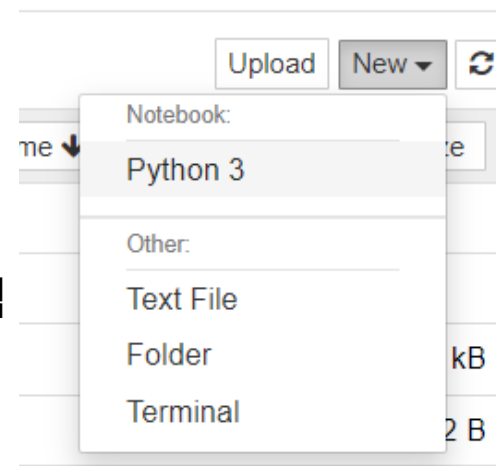
첫 Python 프로그램 만들기

● Python 파일 만들기

- Vscode 실행 후, file->open folder 원하는 폴더로 이동
- Explorer 사진과 같이 클릭
- 파일명.py 입력



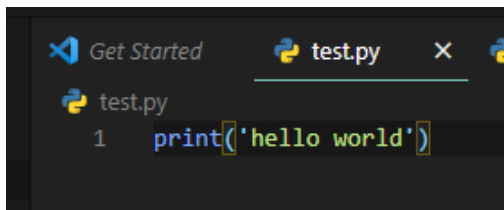
- Jupyter Notebook 실행
- 브라우저가 실행되며 파일 목록이 뜬
 - C:\Users\사용자명 폴더
- 원하는 폴더로 이동해서 New - Python 3 선택



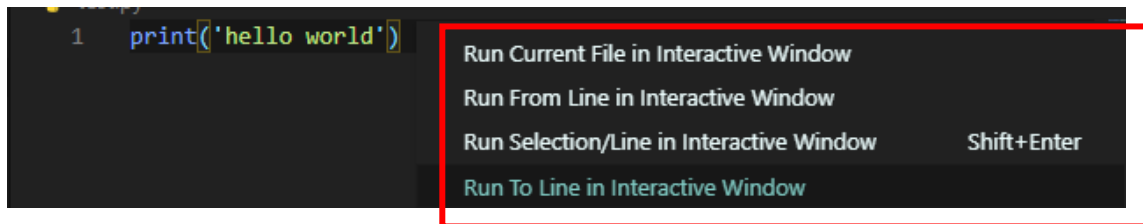
첫 Python 프로그램 만들기

● Python 파일 편집 및 실행하기

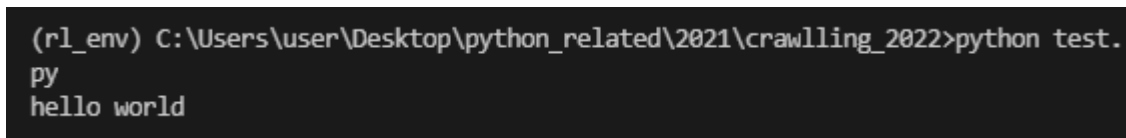
- 우측에 파일명.py로 편집기가 생김
- `print("쓰고싶은 말")` 입력



- Ctrl + s 로 저장
- 코드 마우스 오른쪽 클릭하면 다양한 실행 옵션 제공



- Terminal에 실행하려면 `python 파일명.py` 후 enter



리눅스 터미널 코드: `cd ls` 이용

* 주의사항: 터미널에서 실행하려면, 터미널 위치가 파일명.py가 존재하는 폴더위치

다양한 출력해보기

● print()

- 괄호 안의 내용을 출력하라는 함수
- 괄호 안에는 문자, 숫자, 변수들이 들어갈 수 있음
 - 문자 : 큰 따옴표 또는 작은 따옴표 안에 들어가야함

```
print("anything...234ㄹㅇfkadjsfdfklasdfjkl sdfjlk sdj")  
print('작은 따옴표도 가능')
```

- 숫자 : 따옴표 없이 숫자만 출력 가능

```
print(33453424)
```

```
anything...234ㄹㅇfkadjsfdfklasdfjkl sdfjlk sdj  
작은 따옴표도 가능  
33453424
```


다양한 출력해보기

● 구구단 출력하기

- 가장 원시적인 방법... : 90줄을 써야함

```
print("구구단 2 단을 출력합니다.")
print("2 x 2 = 4")
print("2 x 3 = 6")
print("2 x 4 = 8")
print("2 x 5 = 10")
print("2 x 6 = 12")
print("2 x 7 = 14")
print("2 x 8 = 16")
print("2 x 9 = 18")
...
print("구구단 9 단을 출력합니다.")
print("9 x 2 = 18")
print("9 x 3 = 27")
print("9 x 4 = 36")
print("9 x 5 = 45")
print("9 x 6 = 54")
print("9 x 7 = 63")
print("9 x 8 = 72")
print("9 x 9 = 81")
```

구구단 2 단을 출력합니다.

```
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
```

...

구구단 9 단을 출력합니다.

```
9 x 2 = 18
9 x 3 = 27
9 x 4 = 36
9 x 5 = 45
9 x 6 = 54
9 x 7 = 63
9 x 8 = 72
9 x 9 = 81
```

- 변수를 활용하면? : DIY

파이썬과 다른 프로그래밍 언어와의 차이

- Python은 scope를 indent로 정한다. 세미 콜론(:)을 생략해도 됨.

JAVA

```
public class test {  
    public static void main(String[] args) {  
        System.out.println(1);  
        System.out.println(2);  
    }  
}
```

python

```
def test():  
    print(1)  
    print(2)
```

- Python의 경우 자료형을 지정해주지 않아도 된다.

- Java의 경우 변수 정의 시 자료형을 함께 지정함으로써 기본 값으로 변수 정의 가능
- Python의 경우 자료형을 지정해주지 않아도 되는 대신, 값을 할당하지 않고 선언만 하는 것은 불가능함.
- 변수는 할당된 값에 따라 하나의 변수의 자료형이 바뀔 수 있음.

- type(변수 이름)을 통해 변수의 자료형 확인 가능

JAVA

```
int a = 1;  
String b = "abc";  
char c = "a";
```

python

```
a = 1  
b = "abc"  
c = "a"
```

식별자(Identifier)

● 식별자(Identifier)

- 변수나 함수들을 식별하기 위한 이름
- 대소문자 알파벳, underscore(_), 숫자만 활용 가능 / 숫자로 시작할 수 없음
- Reserved words들 사용 불가

and, as, assert, break, class, continue, def, del, elif, else, except, False, finally, for, from, global, if, import, in, is, lambda, None, nonlocal, not, or, pass, raise, return, True, try, while, with, yield

- 많이 사용하는 식별자 형태(필수 X)

스네이크 케이스(snake_case) : 단어들 사이에 _ 기호 사용, 주로 변수 및 함수에서 사용

캐멀 케이스(camelCase) : 단어의 첫 글자에 대문자 사용, 주로 클래스에서 사용

식별자(Identifier)

● 식별자(Identifier)의 value와 type

- **Value**는 문자나 숫자와 같이 프로그램이 작동하는 기본적인 것 중 하나이다. 지금까지 우리가 본 몇몇 **Value**들은 2, 42.0, 그리고 'Hello World!'이다.
- 이 **Value**들은 다른 **Type**에 속한다: 2는 **Integer**(정수), 42.0은 **Float**(실수), 그리고 'Hello, World!'는 **String**(문자열)이다.

```
>>> type(2)
<class 'int'>
>>> type(42.0)
<class 'float'>
>>> type('Hello, World!')
<class 'str'>
```

파이썬에서의 Type

Text Type: **str**

Numeric Types: **int**, **float**, complex

Sequence Types: **list**, tuple, range

Mapping Type: **dict**

Set Types: set, frozenset

Boolean Type: **bool**

Binary Types: bytes, bytearray, memoryview

변수

● 변수

- 특정 데이터를 저장하는 곳
- 변수 할당 방법 : 변수이름 = 저장하고싶은 데이터
- 변수는 따옴표 없이 사용
- 출력 예시

```
a = 1
print(a)
a = 4
print(a)
print("a")
```

- a에 처음에 1을 넣고 a를 출력했을 때는 1이 나옴
 - 그 후에 a에 4를 넣으면 1은 덮어쓰기되어 4가 새롭게 저장됨
 - a와 "a"의 차이점 알고 넘어가기
- 변수들끼리의 연산

```
a = 3
b = 5
print(a * b)
print(a - b)
print(a / b)
```

1

4

a

15

-2

0.6

변수

● 변수 이름

- 길이 제한 없음
- 글자 숫자 가능
- 숫자로 시작 불가능
- 대문자 사용 가능하지만 소문자만 하는 것이 관례
- 특수문자는 유일하게 _ (underscore) 만 가능

```
>>> 2020course = '컴퓨터의 개념 및 실습'
File "<input>", line 1
    2020course = '컴퓨터의 개념 및 실습'
      ^
SyntaxError: invalid syntax
```

```
>>> course$2020 = '컴퓨터의 개념 및 실습'
File "<input>", line 1
    course$2020 = '컴퓨터의 개념 및 실습'
      ^
SyntaxError: invalid syntax
>>> course_2020 = '컴퓨터의 개념 및 실습'
>>>
```

변수 출력해보기

● 변수와 문자 섞어서 출력하기

- 괄호 내에서 쉼표로 분리하여야 함
- 마지막으로 저장한 변수의 값들이 적용됨
 - a는 3, b는 5
- 변수들의 연산식을 적으면 연산결과가 출력됨

```
print("저는", b, "층에 살아요")  
print("차는 지하", a, "층에 있어요")  
print("꼭대기층은", a * b, "층이에요")
```

저는 5 층에 살아요
차는 지하 3 층에 있어요
꼭대기층은 15 층이에요

키보드로부터 입력받기: input() 함수

● input()

- 사용자에게 데이터를 키보드로 입력 받는 역할
- 입력을 받아서 보통 변수에 문자로 저장함
- 예시
 - 이름과 나이를 입력받아 한 문장에 정리해서 출력하기

```
name = input("이름 : ")
age = input("나이 : ")
print("제 이름은", name, "이고, ", age, "살입니다.")
```

- 유의사항 : 변수에 문자로 저장하기 때문에 나이를 숫자로 입력 받았지만 숫자 연산을 시도하면 에러가 발생

```
broAge = age - 2
print("제 동생은", broAge, "살입니다.")
```

Traceback (most recent call last):

File "[C:/Users/cnc4e/PycharmProjects/kpu/firstPython.py](#)", line 63,

broAge = age - 2

TypeError: unsupported operand type(s) for -: 'str' and 'int'

Process finished with exit code 1

- age는 문자이고(str) 2는 숫자인데(int) 문자에서 숫자를 빼려고 했다는 에러

키보드로부터 입력받기: int() 함수

● int()

- 괄호 안의 문자를 숫자로 바꿔주는 역할
- age가 아닌 int(age)를 사용하면 에러 없이 실행 가능
- 코드 수정 및 결과

```
broAge = int(age) - 2  
print("제 동생은", broAge, "살입니다.")
```

제 이름은 양용석 이고, 10 살입니다.
제 동생은 8 살입니다.

● DIY 1

- 섭씨(C) 온도를 입력받아 화씨(F) 온도로 출력하기
- 화씨 = 섭씨 x 1.8 + 32

문자열 다루기

● 문자열 (string)

- 문자들이 모인 것, 텍스트
- 인터넷에서 데이터 수집을 할 때 텍스트 데이터를 상당히 많이 접하게 됨
- 실제로 문자 하나하나씩 저장이 됨
- index(순서)는 1이 아닌 0부터 시작

index	0	1	2	3	4	5	6	7	8	9	10
-index	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
value	i	공백	a	m	공백	h	u	n	g	r	y

- 대괄호를 사용하여 문자열의 일부분을 표현 가능

```
str = "i am hungry"
print(str[-2])
print(str[7])
```

```
r
n
```

```
print(str[:3])
print(str[7:])
print(str[2:-3])
```

```
i a
ngry
```

문자열 다루기

● replace()

- 문자열을 바꾸는 함수
- 첫 인자 문자열을 두 번째 문자열로 바꿈
- 원래 문자열을 바꾸는 함수가 아니라 새로운 문자열을 만들어줌

```
str2 = str.replace("hu", "a")  
print(str2)  
print(str)
```

```
i am angry  
i am hungry
```

- 첫 인자를 찾지 못했을 경우 아무것도 바꾸지 않음

```
str3 = str.replace("gryt", "ting")  
print(str3)
```

```
i am hungry
```

● strip()

- 앞뒤의 공백이 있을시 제거한 새로운 문자열을 만들어줌

```
str4 = "    앞뒤로 공백이 있어요!"  
print(str4)  
print(str4.strip())  
print(str4)  
print(str4.replace(" ", ""))
```

```
"  
    앞뒤로 공백이 있어요!  
앞뒤로 공백이 있어요!  
    앞뒤로 공백이 있어요!  
앞뒤로공백이있어요!
```

문자열 다루기

● split()

- 특정 문자열로 문자열을 분리하는 함수

```
str5 = "#오늘#점심은#뭐먹을까"
print(str5[3])
splitResult = str5.split("#")
print(splitResult[0])
print(splitResult[1])
print(splitResult[2])
print(splitResult[3])
```

오늘
점심은
뭐먹을까

0 1 2 3

● DIY 2

- 다음 문자열이 주어졌을 때

diyString = "처음에 글이 있고 중간에는

공백이 길고, 끝부분에는 공백이 짧아요 "

- 1. 앞뒤 공백을 제거하고
- 2. "공백" 이라는 문자열로 분리를 하고 난 결과의 index가 1인 문자열의
- 3. index가 3인 문자는?