



Python 기초 – Why and How Python?



1. Python이란?



1. Python 이란?

1. Python의 특징
2. Data 분석 분야의 Python
3. Jupyter Notebook 설치하기
4. Jupyter Notebook 살펴보기

1) Python의 특징들

✓ 오픈 소스

- ✓ 무료로 이용할 수 있다.
- ✓ 더 많은 유저들이 사용할 수 있기에 많은 피드백이 오간다.
- ✓ 따라서 발전 속도가 빠르고, 많은 Library가 생겨난다.
- ✓ SPSS, SAS 등 유료 소스와는 반대

✓ 크로스 플랫폼

- ✓ 여러 OS 에서 사용가능 하다.
- ✓ Windows, Mac OS, Linux를 모두 지원한다.
- ✓ Node-XL와 같은 프로그램과는 반대

✓ 다양한 회사에서 사용

- ✓ Google, Youtube, Instagram, Dropbox, NASA 등에서 사용

1) Python의 특징들

- ✓ 인간다운 직관적 언어
 - ✓ `if 4 in [1,2,3,4] : print('4가 있습니다')`
 - ✓ 위 코드를 통해서 보듯이, 머릿속에서 생각한 대로 적으면 됨
- ✓ 간결함
 - ✓ 다른 언어에서는 `void`, `var char` 등 자료의 형태를 직접 설정해야 함
 - ✓ Python에서는 자동으로 자료의 형태를 저장해줌

2) 최상위 인기도

Position	Language	Last stable release	Designed by	Influenced by
#1	Python	Python 3.9.7	Guido van Rossum	ABC, Ada, ALGOL 68, APL, C, C++, CLU, Dylan, Haskell, Icon, Java, Lips, Modula-3, Perl, Standard ML
#2	Java	Java SE 16.0.2	James Gosling	Ada 83, C#, C++, CLU, Eiffel, Lisp, Mesa, Modula-3, Oberon, Object Pascal, Objective-C, Smalltalk, Simula67, UCSD Pascal
#3	JavaScript (JS)	ECMAScript 2021	Brendan Eich	AWK, HyperTalk, Java, Scheme
#4	C# (C Sharp)	C# 9.0	Anders Hejlsberg	C++, Cω, Eiffel, F#, Haskell, Icon, J#, J++, Java, ML, Modula-3, Object Pascal, VB

3) Python의 단점

- ✓ 느린 속도
- ✓ 모바일 앱 개발에서 사용하기 어려움
- ✓ 버전 2.xx와 버전 3.xx 사이의 호환성이 낮은편

4) Python의 프로그래밍

- ✓ Programing
 - ✓ 프로그래밍은 "순서대로 완성되어야 하는 단계들의 연결" 이다.
- ✓ Sequential Step
 - ✓ 앞선 순서가 진행되어야 진행할 수 있는 것들
 - ✓ $x = 2$
 - ✓ $x = x + 2$
- ✓ Conditional Step
 - ✓ 조건에 따른 순서가 진행됨
 - ✓ "제어문"에서 다룰 예정
- ✓ Repeated Step
 - ✓ 특정 환경에서 반복해서 하는 단계들
 - ✓ 반복문 or 제어문에서 다룰 예정

1) Why Python?

- ✓ 빠른 개발 속도
 - ✓ 다른 Language에 비해 빠른 속도로 발전된다는 특징을 지님
- ✓ Numpy
 - ✓ 고차원 행렬 계산을 도와주는 패키지
 - ✓ 기존 파이썬의 "계산 속도" 문제를 해결
 - ✓ 데이터 분석 분야에 빠른 발전을 도모
- ✓ Pandas
 - ✓ 다양한 형태의 데이터(csv, xlsx, txt 등)를 읽고 처리할 수 있음
 - ✓ 다양한 자료 구조의 데이터가 하나에 입력 되고 처리할 수 있음
 - ✓ 데이터를 관리하는 측면에서 빠른 발전을 도모

1) Why Python?

✓ Matplotlib

- ✓ 데이터 시각화 프로그램
- ✓ 원래는 외부 프로그램이나, Python에 하나의 Library로 내장됨
- ✓ 그래프, 파이 차트, bar chart 등을 효과적으로 그려줌

✓ Scipy

- ✓ 통계적 처리를 도와주는 Library
- ✓ 통계 검증 처리를 Python에서 할 수 있게 되며 인기도 향상

✓ Scikit-learn

- ✓ Python 내부의 머신러닝의 권위자
- ✓ 머신러닝 알고리즘의 많은 부분을 내장하고 있음
- ✓ 모두 동일한 형태로 정리해두어 해석하기 편하게 정리해둠



1) Jupyter Notebook 설치하기

✓ Python 설치

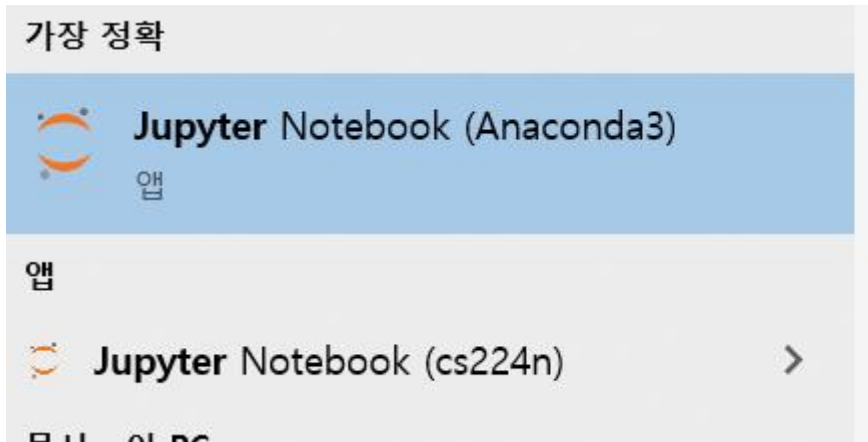
- ✓ Microsoft Store 이동
- ✓ Python 3.9 설치
- ✓ 명령 프롬프트 관리자 권한으로 실행
- ✓ `python --version` 을 입력해 버전 확인이 되는지 체크

✓ Jupyter Notebook 설치

- ✓ <https://www.anaconda.com/products/individual>
- ✓ 위 사이트에서 다운로드 받아 설치

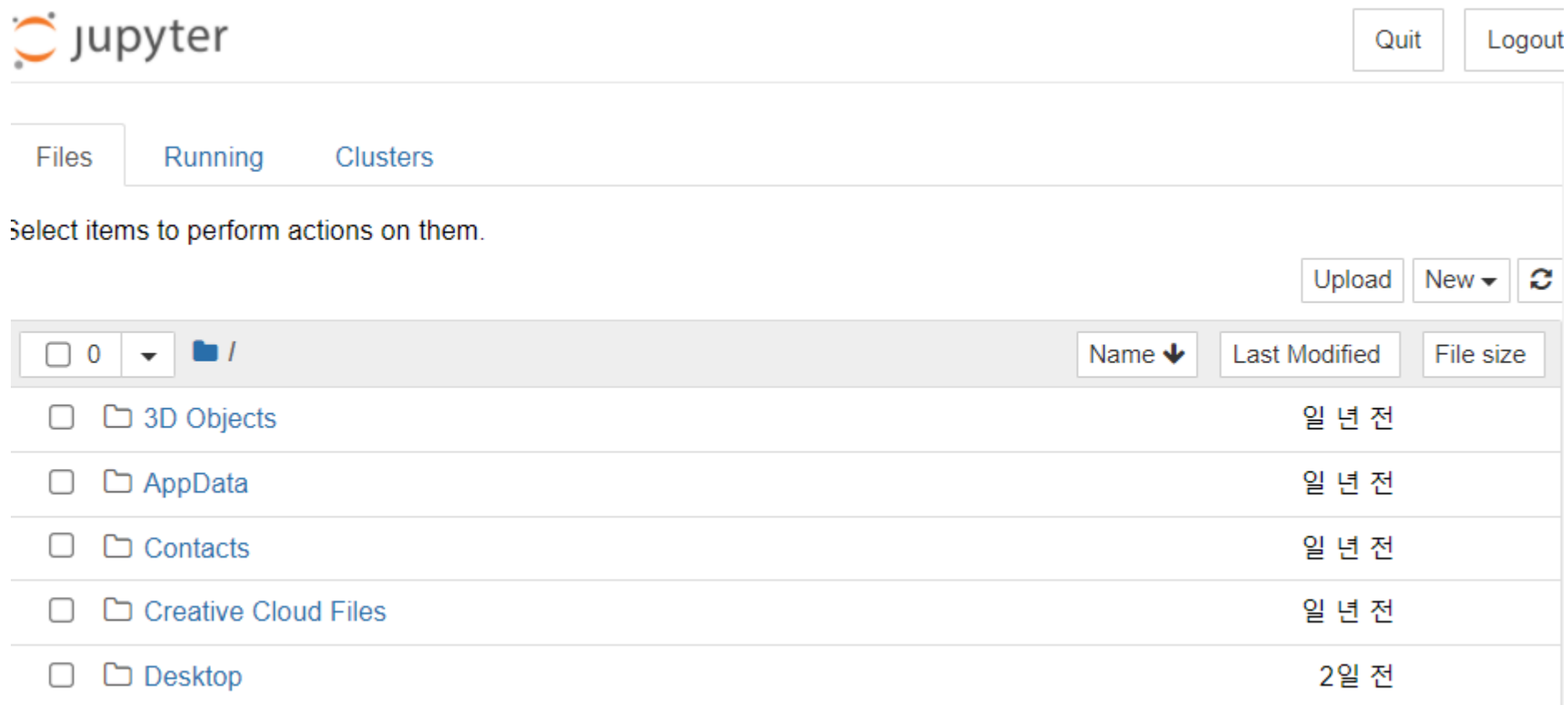


1) Jupyter Notebook - open



2) Jupyter Notebook - 초기 상태

✓ 사용자 이름으로 된 폴더가 시작 화면입니다.

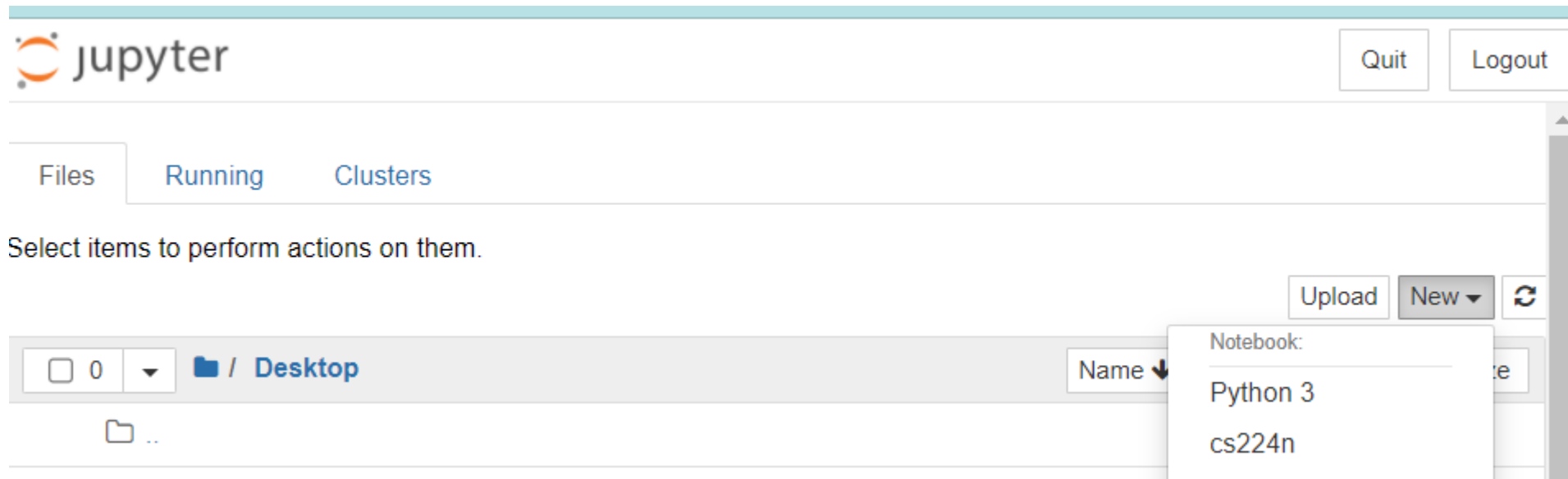


The screenshot shows the Jupyter Notebook interface. At the top left is the Jupyter logo. To the right are 'Quit' and 'Logout' buttons. Below the logo are tabs for 'Files', 'Running', and 'Clusters'. A message says 'Select items to perform actions on them.' To the right of this message are 'Upload', 'New', and a refresh icon. Below this is a table with columns for 'Name', 'Last Modified', and 'File size'. The table lists several folders: '3D Objects', 'AppData', 'Contacts', 'Creative Cloud Files', and 'Desktop'. Each folder has a checkbox on the left and a timestamp on the right.

	Name	Last Modified	File size
<input type="checkbox"/>	0		
<input type="checkbox"/>	3D Objects	일 년 전	
<input type="checkbox"/>	AppData	일 년 전	
<input type="checkbox"/>	Contacts	일 년 전	
<input type="checkbox"/>	Creative Cloud Files	일 년 전	
<input type="checkbox"/>	Desktop	2일 전	

3) Jupyter Notebook – new file

- ✓ Desktop 폴더로 이동 후 우측 상단 “New- Python 3”를 클릭해줍니다.
- ✓ 그러면 바탕화면에 Python 파일이 하나 생깁니다.



4) Jupyter Notebook – 살펴보기

- ✓ Jupyter Notebook은 코드를 “Cell” 단위로 입력할 수 있습니다.
- ✓ “Cell”은 독립적인 코드로 존재할 수 있습니다.
- ✓ Help -> Keyboard Shortcuts에서 단축키를 확인할 수 있습니다.
- ✓ 자주 쓰는 단축키는 수업에서 다루겠습니다.

