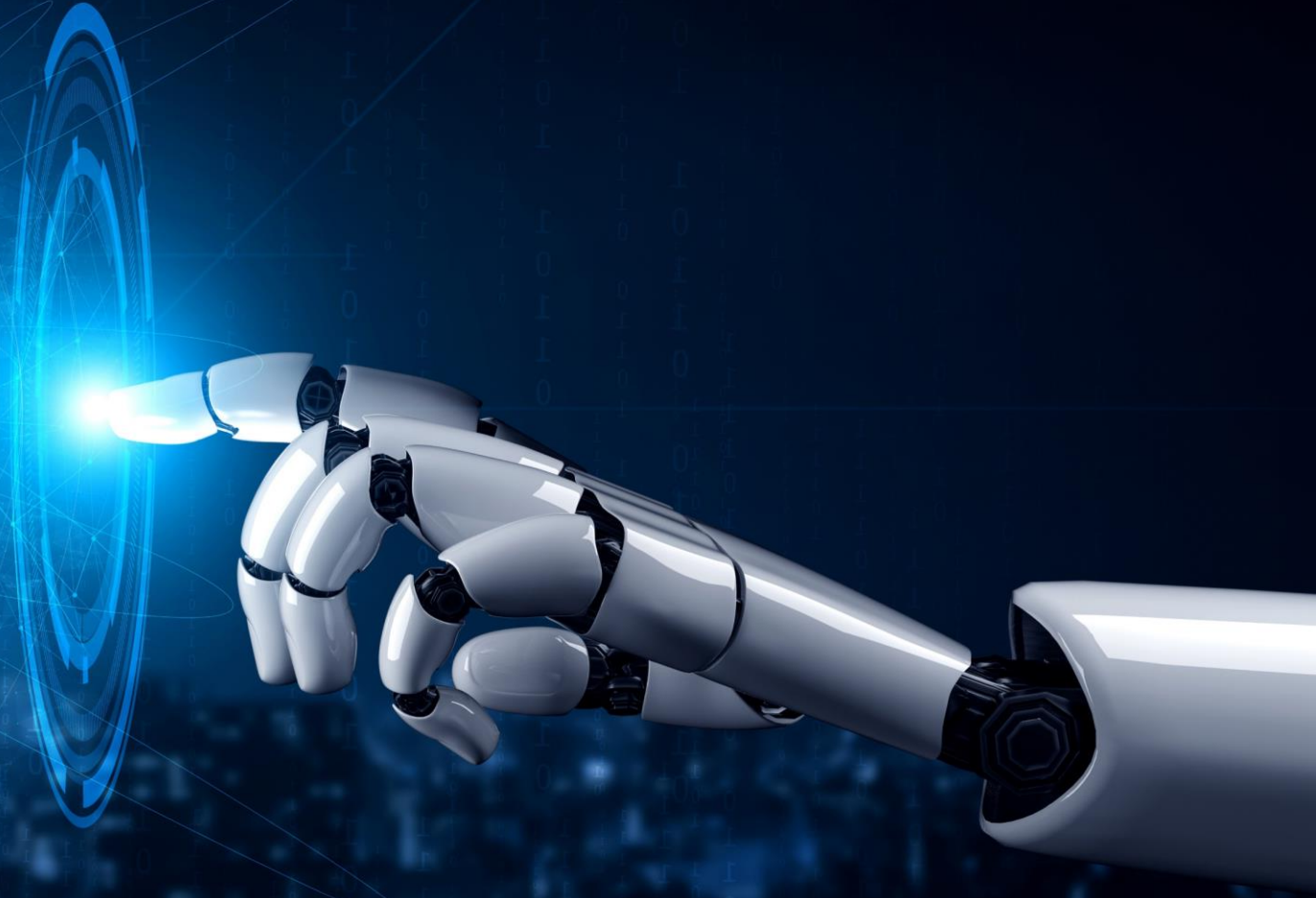


# 파이선의 기초 I

충북대학교 소프트웨어학과  
류관희



# 목 차

## ❖ Part 1 파이썬 개요 및 환경구축

- Python 개요
- Python 프로그램 환경구축

## ❖ Part 2 파이썬 실행하기 및 간단한 프로그래밍

- Python 실행하기
- Jupyter를 통한 프로그램 생성

## ❖ Part 3 파이썬 언어의 기본 기능

- Python 언어의 기본기능



01

- Python 개요
- Python 프로그램  
환경구축

02

- Python 실행하  
기
- Jupyter를 통한  
프로그램 생성

03

- Python 언어의  
기본기능

# 파이썬(Python) 이란

- 1990년 암스테르담의 **귀도 반 로섬(Guido Van Rossum)**에 의해 만들어진 **인터프리터 언어**
  - 사전적 의미 : 고대 신화 속의 파르나수스(Parnassus) 산의 동굴에 살던 큰 뱀
- 사용 예
  - 구글에서 만들어진 소프트웨어의 50%이상이 파이썬으로 제작됨
  - Dropbox(파일 동기화 서비스) 등
- 공동작업과 유지보수가 매우 쉽고 편하기 때문에 이미 다른 언어로 작성된 많은 프로그램과 모듈들이 Python으로 다시 재구성되고 있는 상황

## 파이썬 언어의 특징

- 인터프리터 언어
- 객체지향 지원 / 독립적 실행환경 지원
- 인간친화적 언어
  - if 4 in [1,2,3,4]: print("4가 있습니다")
  - 만일 1,2,3,4중에 4가 있으면 “4가 있습니다”를 출력하라.
- 쉬운 문법 / 간결성
- 강력함
  - 시스템 프로그래밍, 하드웨어 제어, 매우 복잡하고 많은 반복연산 등을 제외한 거의 모든 작업을 할 수 있음
  - 다른 언어의 모듈을 Python에 포함할 수 있음
    - (예) 뼈대는 Python, 속도를 요하는 부분은 C언어

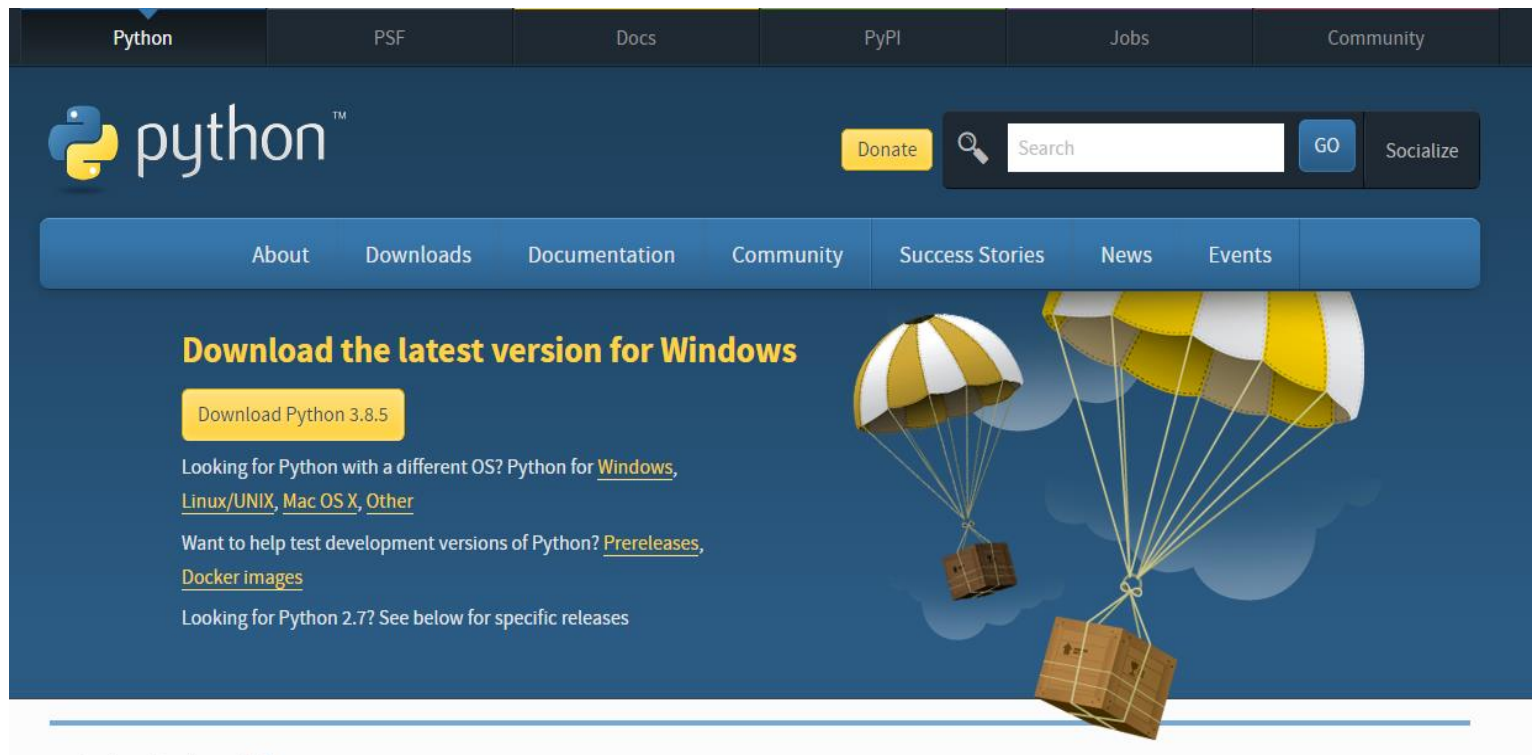
# Python 응용분야

- 할 수 있는 일
  - 시스템 유틸리티
    - OS의 시스템 명령어 사용 가능한 도구 제공
  - GUI프로그래밍
    - Tkinter
  - C/C++언어와의 결합
  - Web 프로그래밍
  - 수치연산 프로그래밍
    - Numerical Python 모듈 제공
  - 데이터베이스 프로그래밍
    - DB연결, pickle
- 할 수 없는 일
  - OS제작
  - 많은 반복과 연산을 필요로 하는 프로그램
    - 데이터압축 알고리즘 등
  - 속도를 요하는 프로그램
  - H/W를 직접 제어하는 프로그램



# 구현환경 구축하기(1)

- <https://www.python.org/downloads/>
- python-3.8.5.exe
- 64비트 환경, python 3.7.4 버전



# 파이썬 설치





## 파이썬 설치

- interpreter 설정
  - 경로설정
  - C:\Users\사용자명\AppData\Local\Programs\Python\Python38\Python.exe

## 문제풀이

- 파이썬은 언제 만들어 졌나요?
- 파이썬 이름의 어떻게 명명되었나요?
- 파이썬 언어가 주로 사용되는 분야는 ?

## 요약

- 파이선의 유래와 명명화에 대해 소개
- 파이선 환경 구축 방법
- 파이선 설치

01

- Python 개요
- Python 프로그램  
환경구축

02

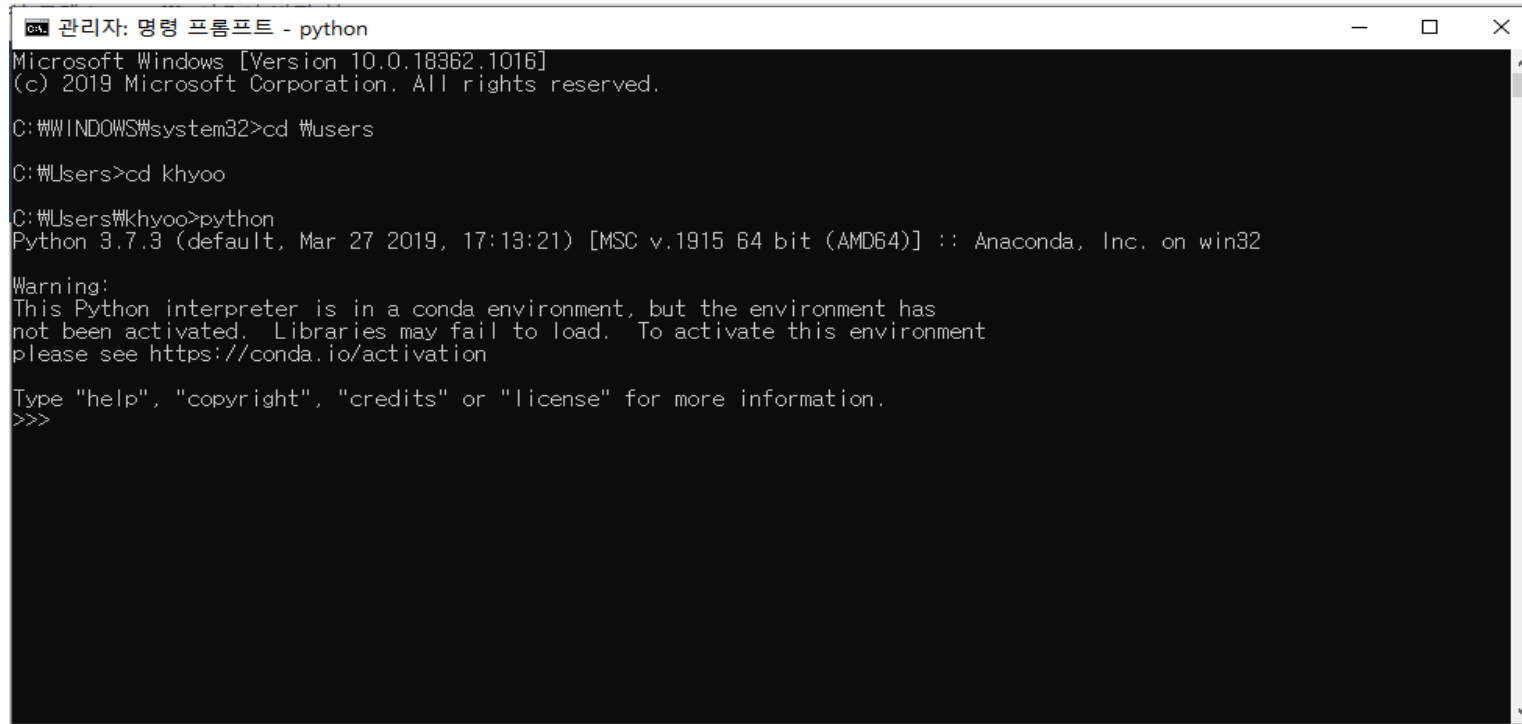
- Python 실행하기
- Jupyter를 통한  
프로그램 생성

03

- Python 언어의  
기본기능

## 파이썬 Python 실행하기(콘솔)

- 대화형 인터프리터
  - 시작 -> 모든프로그램 -> Python 3.8.5 -> Python(Command Line)
- 종료 : Ctrl+Z



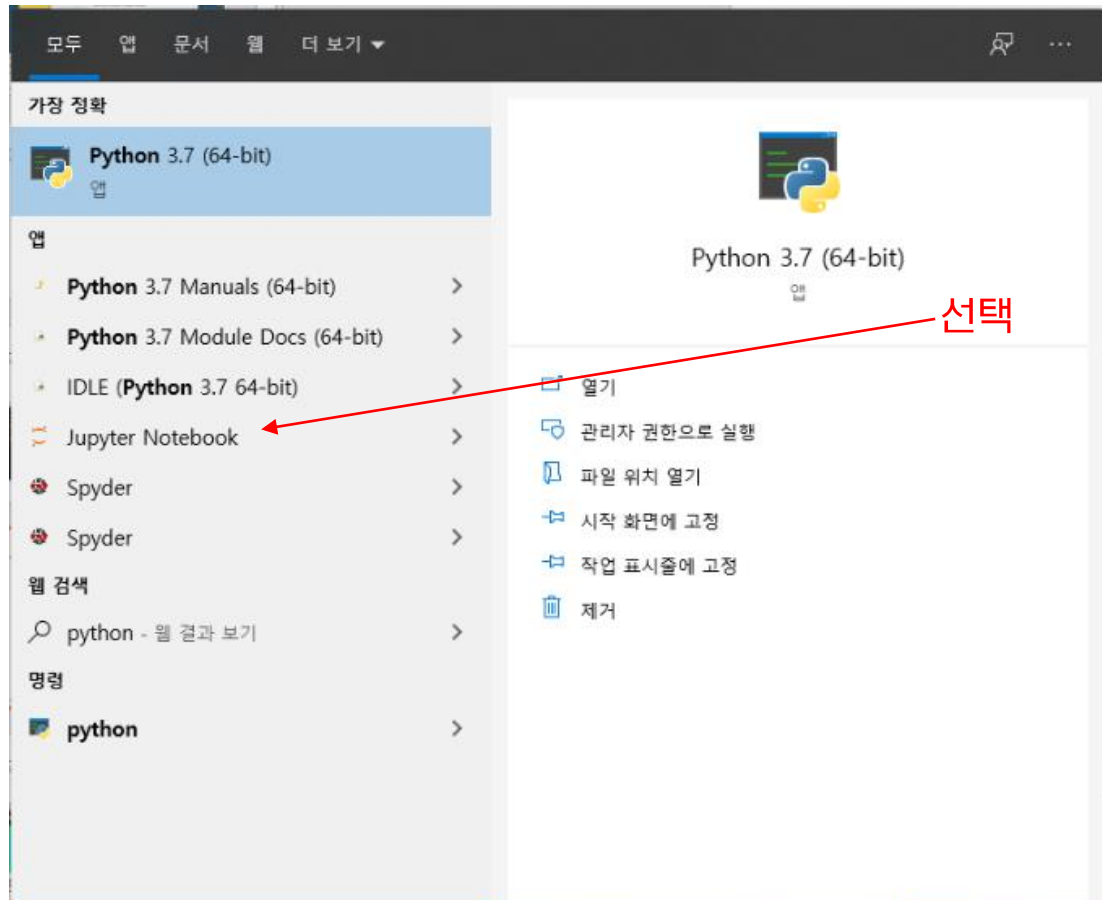
```
관리자: 명령 프롬프트 - python
Microsoft Windows [Version 10.0.18362.1016]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd %Users%
C:\Users>cd khyoo
C:\Users\khyoo>python
Python 3.7.3 (default, Mar 27 2019, 17:13:21) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>>
```

# 파이썬 실행 IDE





# 파이썬 실행 IDE

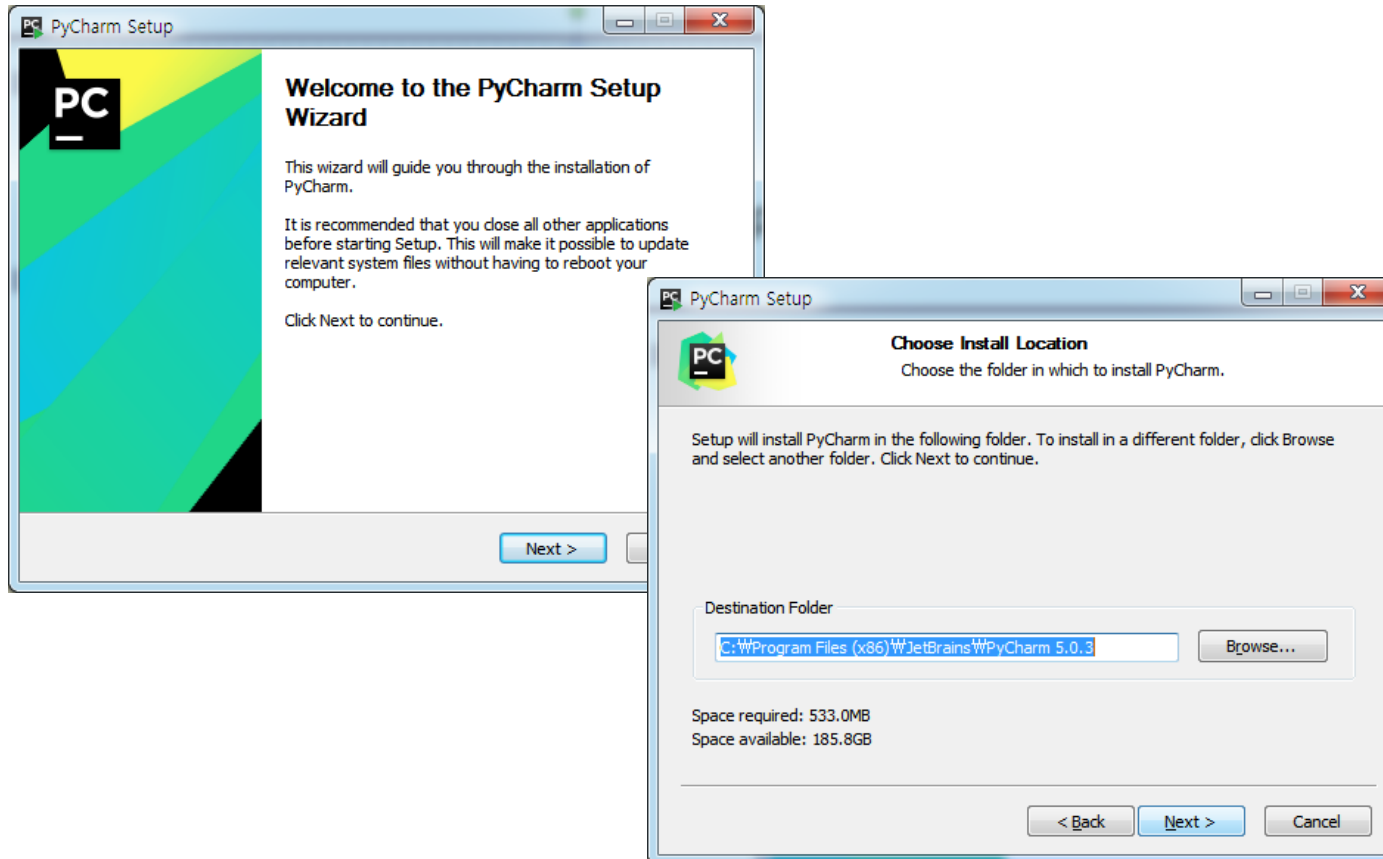
```
Python 3.7.5 Shell
File Edit Shell Debug Options Window Help
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/khyoo/Documents/kk.py =====
10
>>> i=10
>>> j=20
>>> print(i+j)
30
>>> |
```

# PyCharm 설치

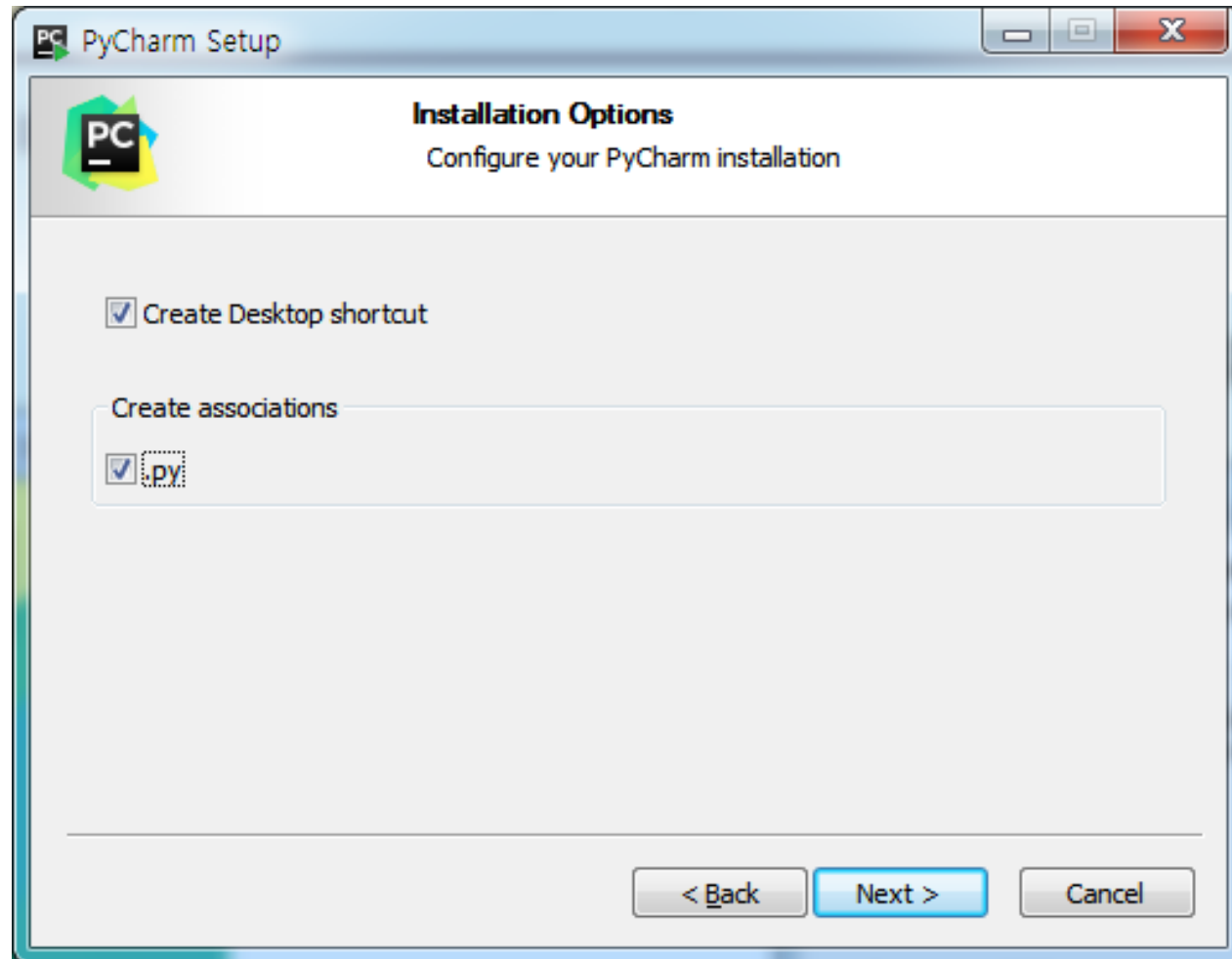
PyCharm

Python IDE : [pycharm-community-5.0.3.exe](#)

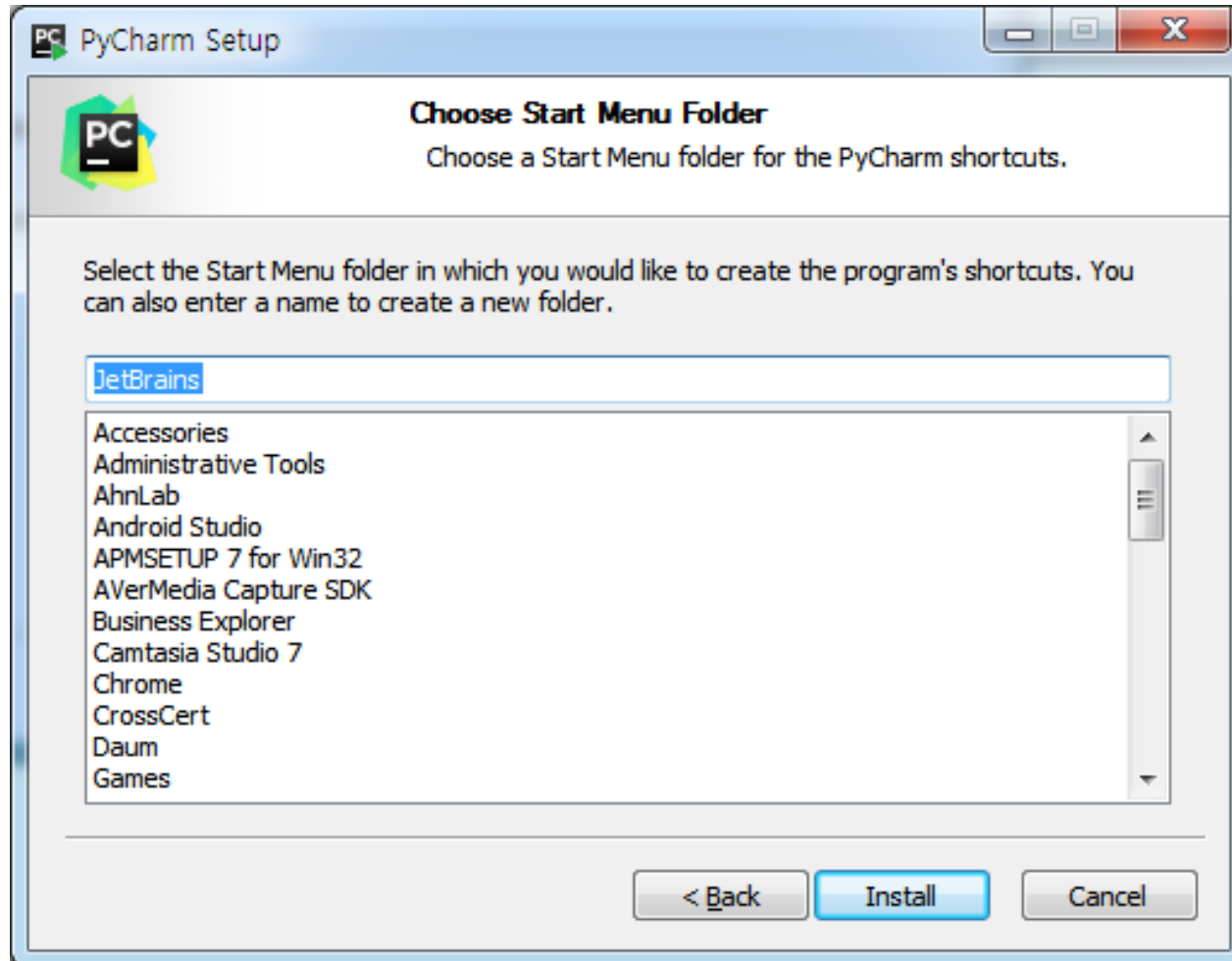
<http://www.jetbrains.com/pycharm/>



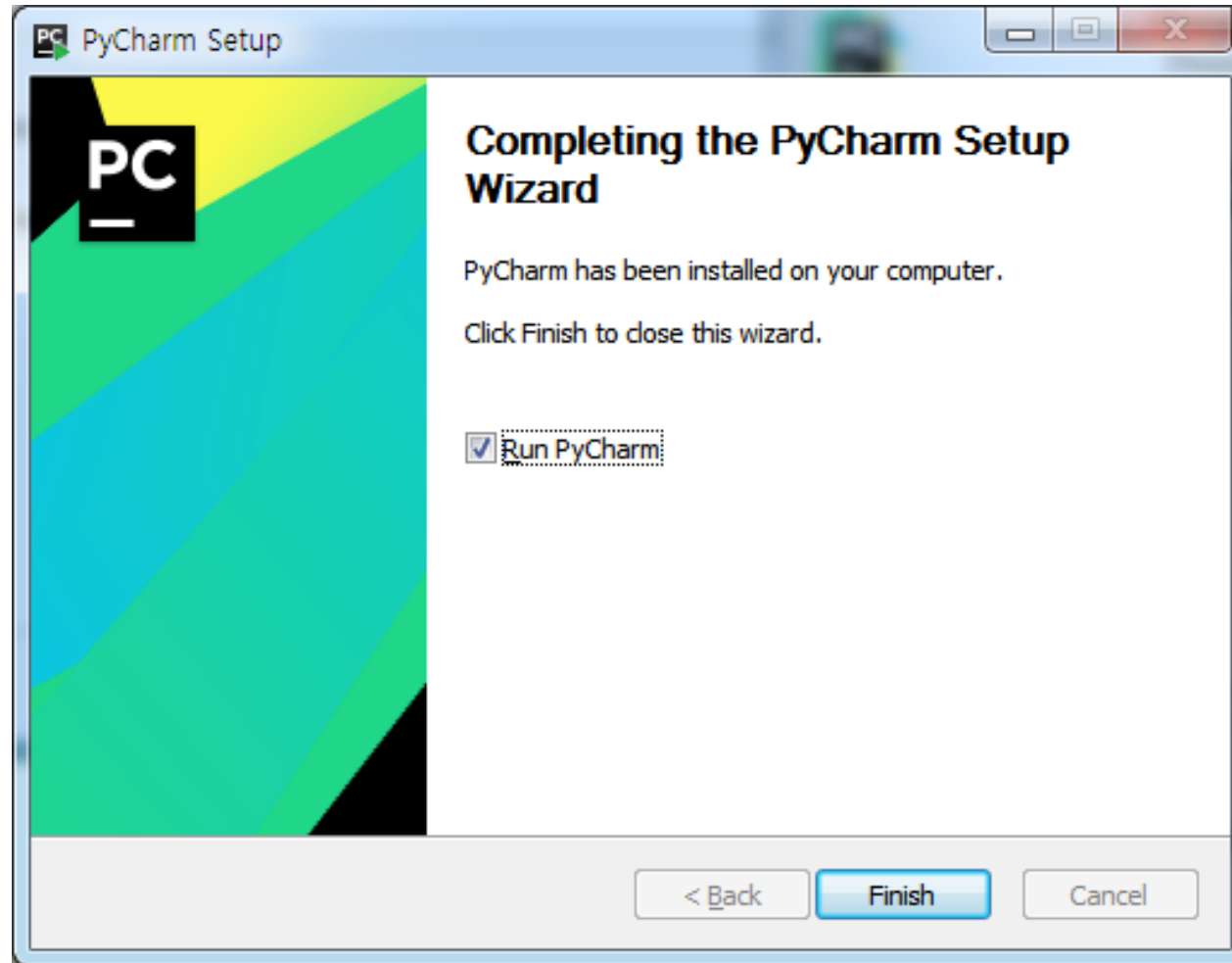
# 파이썬 실행 IDE



# 파이썬 실행 IDE

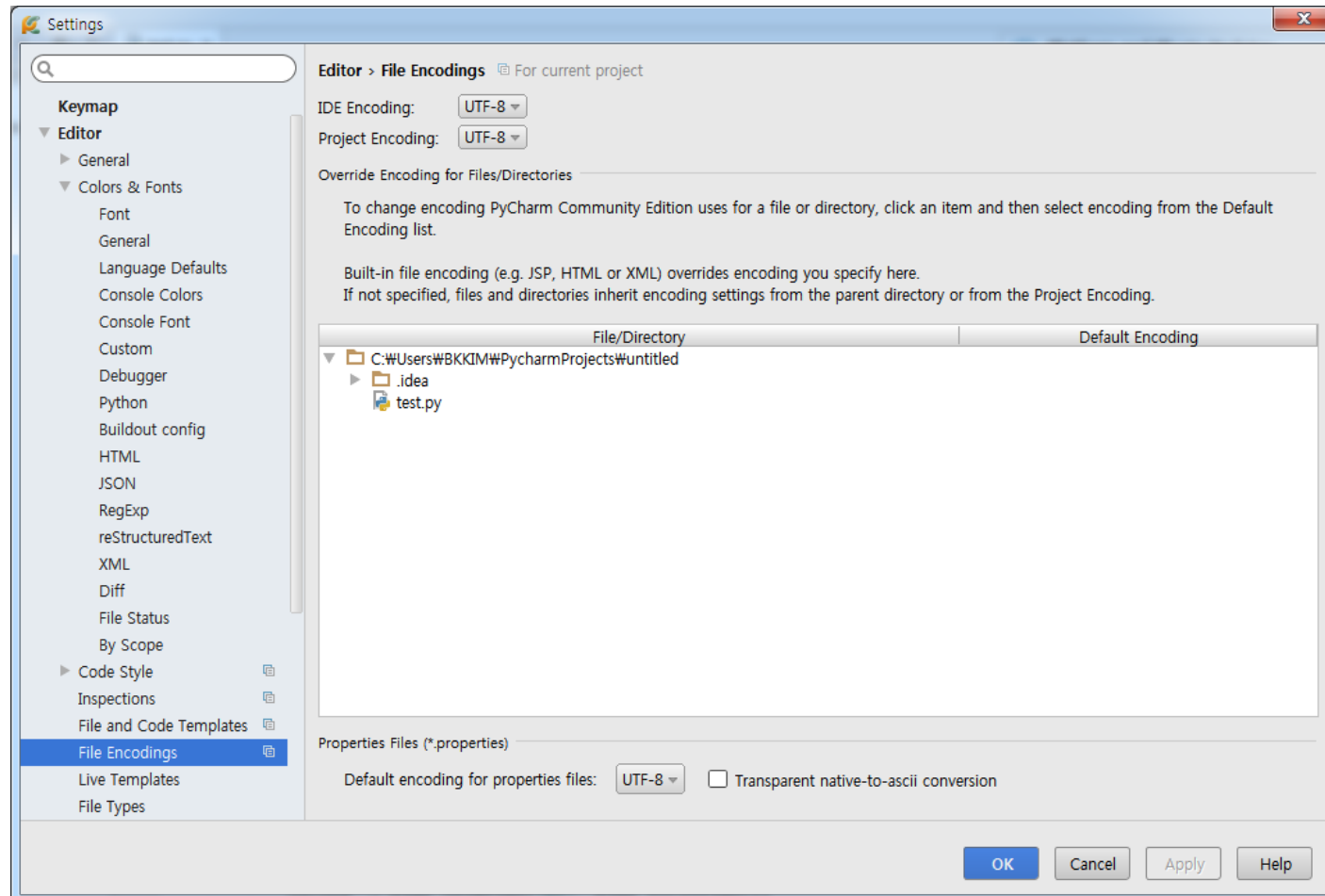


## 파이썬 실행 IDE



# PyCharm IDE

Settings...





# Project Creation & execution

File -> New Project

interpreter 설정

C:\Users\사용자명

\AppData\Local\Programs\Python\Python3

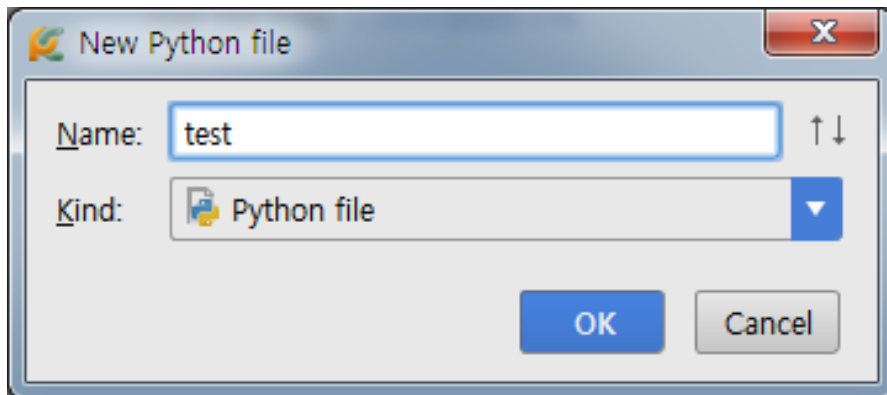
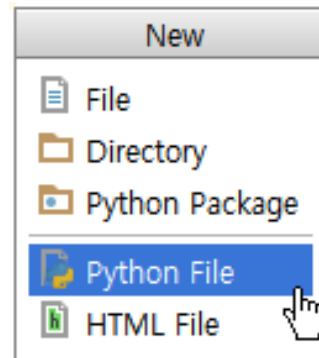
5\Python.exe

New

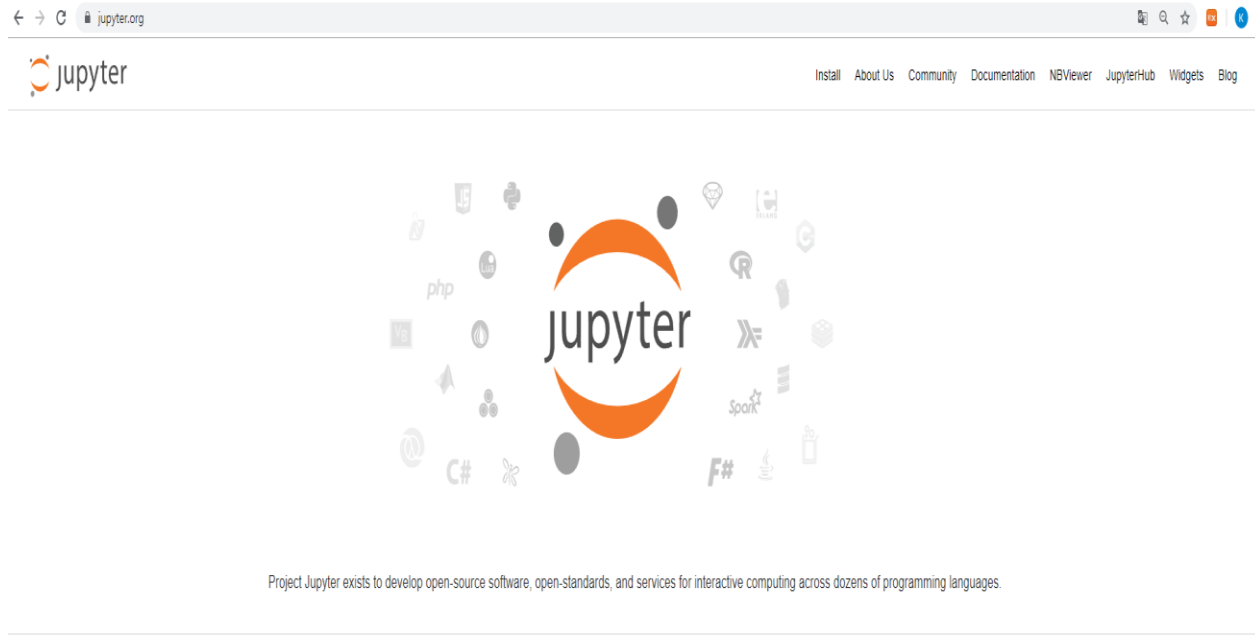
print("Hello world!")

Shift+F10

Ctrl + F5



# Jupyter notebook



# Jupyter notebook

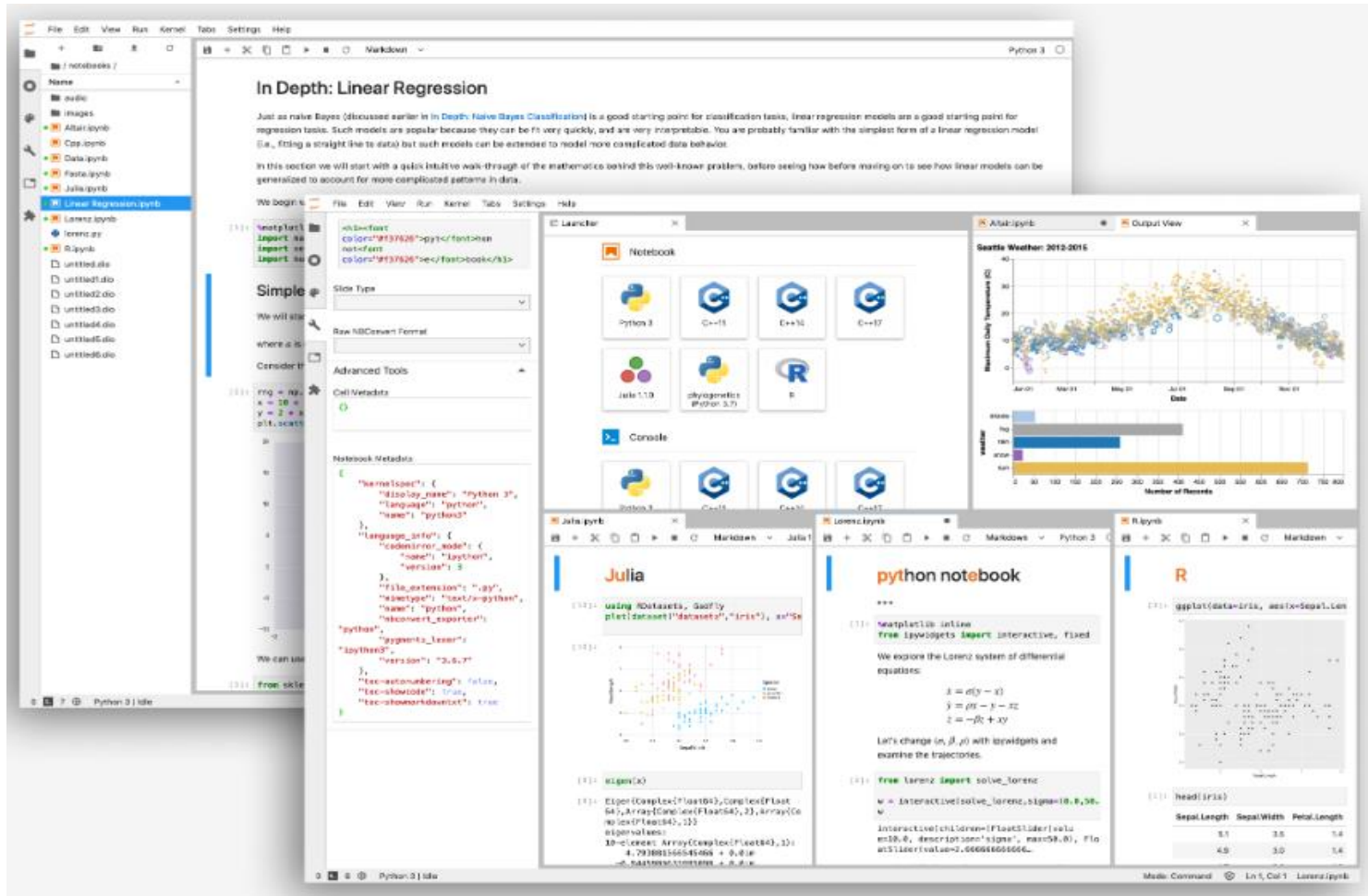
## JupyterLab 1.0: Jupyter's Next-Generation Notebook Interface

JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data.

JupyterLab is flexible: configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning.

JupyterLab is extensible and modular: write plugins that add new components and integrate with existing ones.

# Jupyter notebook



The screenshot displays the JupyterLab environment with several open notebooks:

- In Depth: Linear Regression**: A notebook with text explaining linear regression models and their applications.
- Simple**: A notebook showing a simple linear regression model with a scatter plot and a fitted line.
- Seattle Weather: 2012-2015**: A notebook displaying a scatter plot of monthly temperature and a bar chart of monthly precipitation.
- Julia**: A notebook showing a scatter plot of data points.
- python notebook**: A notebook showing a system of differential equations and their solutions.
- R**: A notebook showing a scatter plot of data points.

The interface includes a file browser on the left, a central workspace for editing code, and an output view on the right showing the results of the code execution.

# Jupyter notebook

## Getting started with JupyterLab

### Installation

JupyterLab can be installed using `conda` or `pip`. For more detailed instructions, consult the [installation guide](#).

#### conda

If you use `conda`, you can install it with:

```
conda install -c conda-forge jupyterlab
```

#### pip

If you use `pip`, you can install it with:

```
pip install jupyterlab
```

If installing using `pip install --user`, you must add the user-level `bin` directory to your `PATH` environment variable in order to launch `jupyter lab`.

# Jupyter notebook

## Getting started with the classic Jupyter Notebook

### Prerequisite: Python

While Jupyter runs code in many programming languages, Python is a requirement (Python 3.3 or greater, or Python 2.7) for installing the JupyterLab or the classic Jupyter Notebook.

### Installing Jupyter Notebook using Anaconda

We **strongly recommend** installing Python and Jupyter using the [Anaconda Distribution](#), which includes Python, the Jupyter Notebook, and other commonly used packages for scientific computing and data science.

First, download [Anaconda](#). We recommend downloading Anaconda's latest Python 3 version.

Second, install the version of Anaconda which you downloaded, following the instructions on the download page.

Congratulations, you have installed Jupyter Notebook! To run the notebook, run the following command at the Terminal (Mac/Linux) or Command Prompt (Windows):

```
jupyter notebook
```

See [Running the Notebook](#) for more details.

### Installing Jupyter Notebook with pip

As an existing or experienced Python user, you may wish to install Jupyter using Python's package manager, pip, instead of Anaconda.

If you have Python 3 installed (which is recommended):

```
python3 -m pip install --upgrade pip
python3 -m pip install jupyter
```

If you have Python 2 installed:

```
python -m pip install --upgrade pip
python -m pip install jupyter
```

Congratulations, you have installed Jupyter Notebook! To run the notebook, run the following command at the Terminal (Mac/Linux) or Command Prompt (Windows):

```
jupyter notebook
```

See [Running the Notebook](#) for more details.



# Jupyter를 통한 python 프로그램

관리자: 명령 프롬프트

```
Microsoft Windows [Version 10.0.18362.1016]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd %users%

C:\Users>cd khyoo

C:\Users\khyoo>python
Python 3.7.3 (default, Mar 27 2019, 17:13:21) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>>

C:\Users\khyoo>jupyter notebook
```

## 문제풀이

- 파이썬 IDLE를 이용한 실행 방법은?
- Jupyter notebook을 이용한 실행 방법은?

## 요약

- 파이썬 Concole, IDLE를 통한 파이썬 프로그래밍하기
- PyCharm을 이용한 파이썬 프로그래밍하기
- Jupyter notebook을 이용한 파이썬 프로그래밍하기

01

- Python 개요
- Python 프로그램  
환경구축

02

- Python 실행하  
기
- Jupyter를 통한  
프로그램 생성

03

- Python 언어의  
기본기능

# Jupyter를 통한 프로그램 생성

jupyter

Quit Logout

Files Running Clusters

Select items to perform actions on them.

Upload New

Python3 선택

Upload New

Notebook:

PyTorch

Python 3

Other:

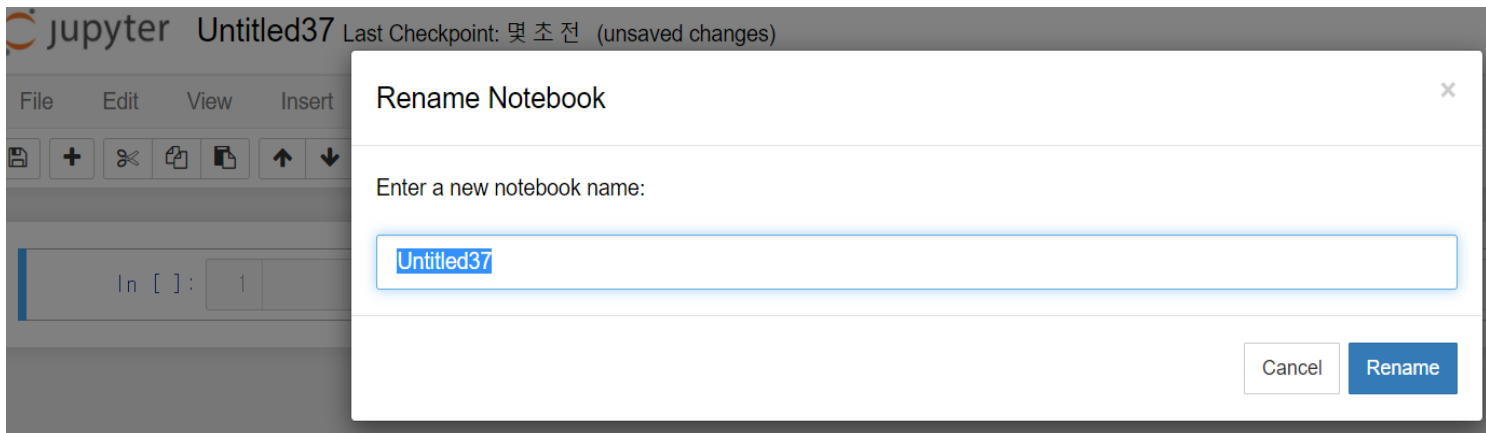
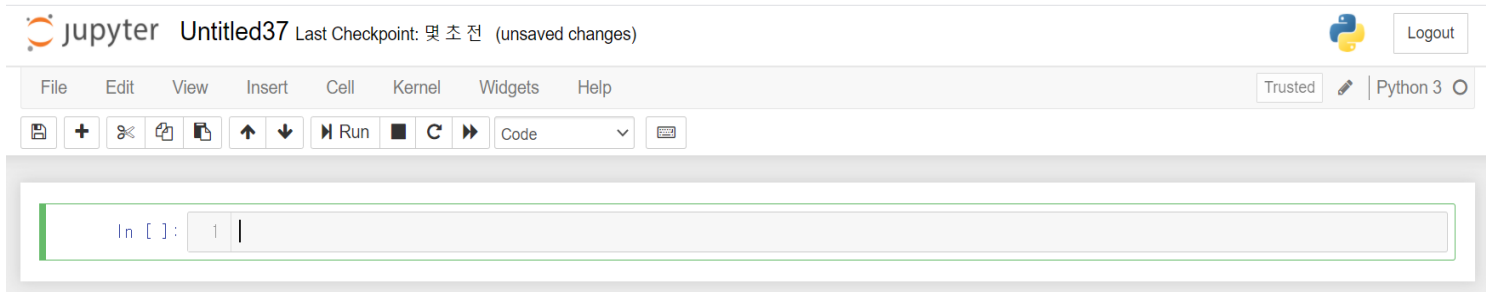
Text File

Folder

Terminal

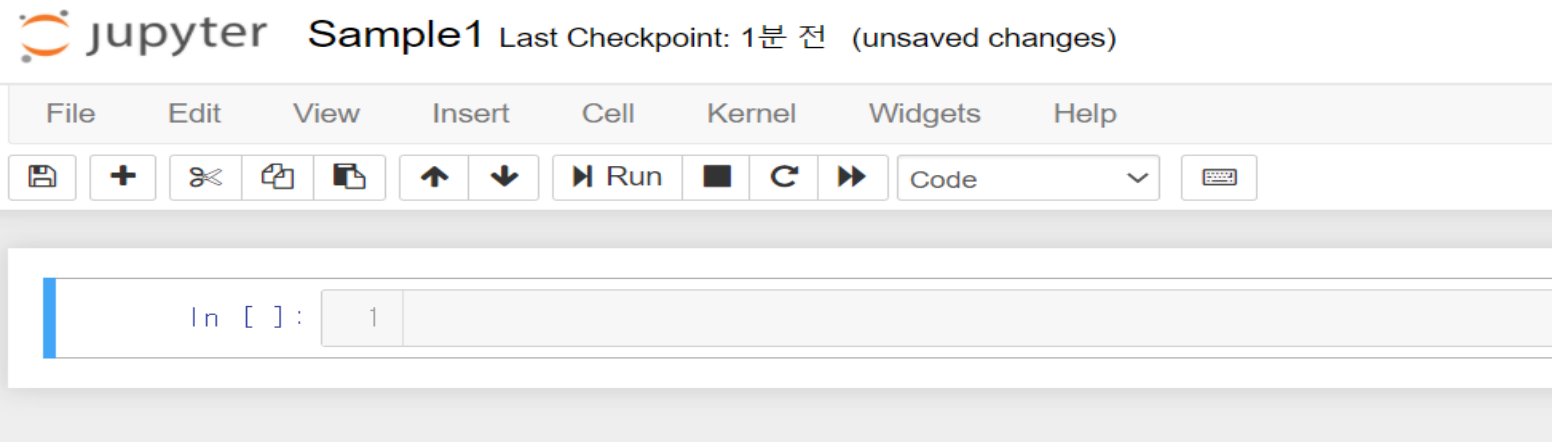
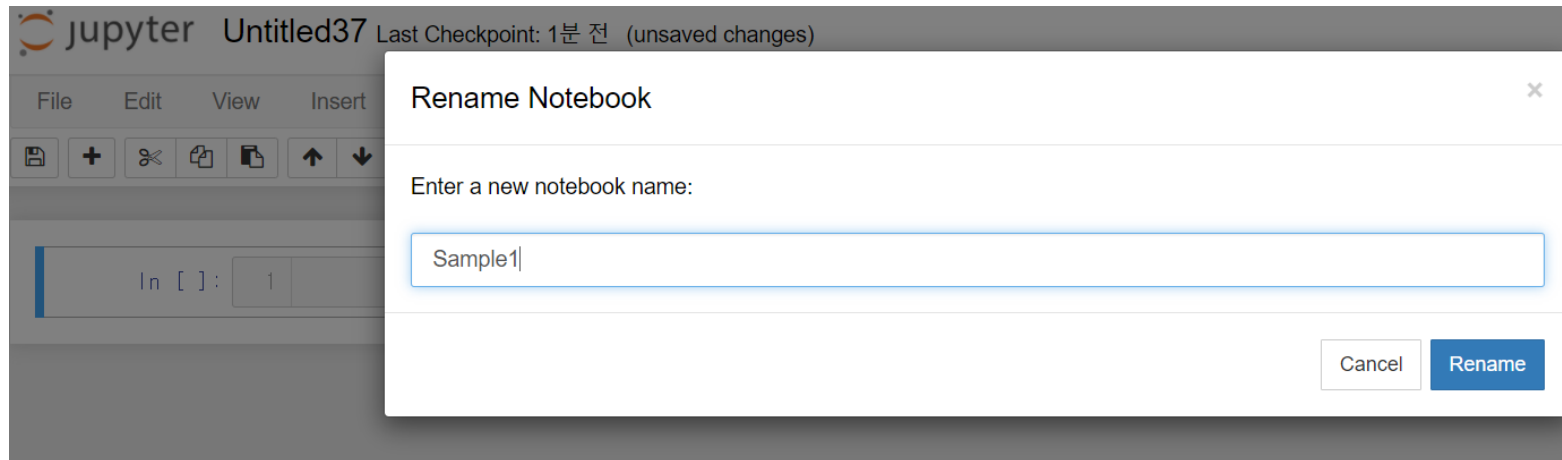
6달 전

# Python 프로그램 이름 변경







# Python 프로그램 이름 변경



# Python 프로그램 만들기

 jupyter Sample1 Last Checkpoint: 3분 전 (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

        Run    Code  

```
In [1]: 1 i=10
```

```
In [2]: 1 j=40
```

```
In [3]: 1 print(i,j, i+j)
```

```
10 40 50
```

# Python 언어의 기본 기능

- 라인주석
- 자료 유형
  - 문자열
  - 숫자(number)
  - 날짜
  - 리스트
  - 튜플
  - 딕셔너리
  - 집합
  - 기타
- 변수
- 명령문
  - 연산자
  - 제어문
  - 반복문

## 주석 방법

- 라인주석
  - #
- 블록주석
  - “” 와 “”로 감싸기
  - 주의사항
    - 중간에 “”가 있으면 안됨
    - 들여쓰기 레벨이 맞아야 함

## 자료 유형 종류

- String `" " ' ' " " " "`
  - 문자열
- Number (int, float, float32, float64)
  - 숫자형태로 이루어진 자료형
- List `[]`
  - 리스트 : 생성, 삭제, 수정 가능
- Tuple `()`
  - 튜플 : 생성, 삭제, 수정 불가
- Dictionary `{ : }`
  - key와 value의 쌍으로 이루어진 자료
- Sets `set()`
  - 집합
- 기타

## 문자열 String

문자열 만들기: “ ”, ‘ ’, """ """ , ''' ''' 사이에 문자열을 넣을 수 있음.

```
"안녕하세요."
```

```
'Python is fun'
```

```
"""Life is""Life is too short, You need python""too short,  
You need python"""
```

```
message= "안녕하세요. \n충북대 오신걸 환영합니다."
```

```
>>>string1= input("문자 입력 = ")
```

```
>>>print(message)
```

## 문자열 출력 지정 String

- `str = "I eat %d apples." % 3`  
`print(str)`
  - `=> I eat 3 apples.`
- `str = "I eat %s apples." % "five"`  
`print(str)`
  - `=> I eat five apples`
- `number = 10`  
`day = "three"`  
`str = "a = %d b = %s" % (number, day)`  
`print(str)`
  - `=> a = 10 b = three`

## 문자열 연산자

+ : 문자열 더하기

```
Str1 = "안녕하세요. "
```

```
Str2 = "홍길동 입니다."
```

```
Print(Str1+Str2)    # 안녕하세요. 홍길동입니다.
```

\* : 문자열 곱하기

```
print("hello" * 2)    # hellohello
```

```
Str1 = "안녕하세요. "
```

```
Print(Str1 * 2)
```



## 문자열 연산자

- indexing
  - `str = "Hello world"`  
`print(str[0])`           # H  
`print(str[-1])`       # d
- slicing
  - `str = "Hello world"`  
`print(str[0:5])`                   # Hello  
`print(str[:5])`               # Hello  
`print(str[6:11])`               # world  
`print(str[6:])`               # world

## 문자열 연산자

- upper, lower
  - Str1 = "This is Python Program"
  - Str1.upper()
  - Str1.lower()
- count
- find, index
- join
- lstrip, rstrip, strip
- split
- replace

## 문자열 연산자

- `str1 = "This is a Python program"`
- upper, lower
  - `str1.upper()` => THIS IS A PYTHON PROGRAM
  - `str1.lower()` => this is a python program
- count
  - `str1.count("is")` => 2
- len
  - `len(str1)` => 24
- find, index
  - `print(str1.find('P'))` => 10
  - `print(str1.index('T'))` => 0

## String Functions 연산자

- `ch = "충북" str = "대학교 소프트웨어학과 "`
- Join
  - `" ".join(str), "-".join(str), print(ch.join(str))`
- `lstrip, rstrip, strip`
  - `print(str.strip())`
- Split
  - `- print(str.split())`
- Replace
  - `Print(str.replace(" ", ","))`

## 문제풀이

- 두 문자열을 합치는 명령은 무엇인가?
- Join 명령을 설명하시오?
- Split 명령을 설명하시오?

## 요약

- Jupyter notebook을 이용한 파이썬 프로그래밍하기
- 문자열 처리 기법을 소개하였다.