

LESSON

웹 크롤링에 대한 이해

```
animLength = toTime - fromTime

# Ask user for directory
filePath = c4d.storage.SaveDialog()
filePath, objName = os.path.split(filePath)
objName = objName + ". "
filePath = filePath + "\\ "

# Ask for confirmation
questionDialogText = "Obj Sequence will be saved as:\n\n" \
    "" + filePath + objName + "####.obj\n\n" \
    "From frame " + str(fromTime) + " to " + str(toTime) + " "
proceedBool = c4d.gui.QuestionDialog(questionDialogText)

if proceedBool == True:

    # Loop through animation and export frames
    for x in range(0, animLength):

        # change frame, redraw view
        moveTime = c4d.BaseTime(fromTime, docFps) + c4d.BaseTime(x, docFps)
        doc.SetTime(moveTime)
        c4d.EventAdd(c4d.EVENT_FORCEREDRAW)
        c4d.DrawViews(c4d.DRAWFLAGS_FORCEFULLREDRAW)

        # update status bar
        c4d.StatusSetText("Exporting " + str(x) + " of " + str(animLength))
        c4d.StatusSetBar(100.0*x/animLength)

        # add buffer 0.001
        bufferedNumber = str(doc.GetTime().GetFrame(docFps))
        if len(bufferedNumber) < 3:
```

웹 크롤링이란?

- 웹 스크래핑(Web Scraping)이라고도 하며
컴퓨터 소프트웨어 기술로 각종 웹 사이트들에서
원하는 정보를 추출하는 것을 의미
- 웹 크롤러란 인터넷에 있는 웹 페이지를 방문해서
자료를 수집하는 일을 하는 프로그램
- 크롤링을 위한 선행 학습
 - 웹(Web)의 개념
 - HTML, CSS, JavaScript 구조 및 태그
 - 파이썬 기초

웹 크롤링의 기법

- HTML 페이지를 가져와서 **HTML/CSS등을 파싱**하고, 필요한 데이터만 추출하는 기법
- **Open API(Rest API)**를 제공하는 서비스에 Open API를 호출해서, 받은 데이터 중 필요한 데이터만 추출하는 기법
- 브라우저를 **프로그래밍으로 조작**해서 필요한 데이터만 추출하는 기법



웹 크롤링 예시

• 구글 검색엔진

▶ 구글에서는 크롤링 봇을 통해 수집한 결과를 검색 결과로 보여줌

• 쇼핑몰

▶ 경쟁사의 제품 정보 등을 크롤링하여 사용

• API 외의 정보

▶ API에서 제공하는 정보 외에 필요한 정보들을 크롤링하여 사용

크롤링 합법? 불법?

- ♣ 모든 크롤링이 불법은 아님
 - 사이트 운영자의 의사에 상관없이 무단으로 크롤링하는 것은 불법
- ♣ robots.txt 에서 크롤링 여부 확인 가능
 - 검색엔진 크롤링 봇들이 해당 파일을 열어본 뒤 수집 유무를 결정
- ♣ 네이버 로봇은 검색 허용, 야후 로봇은 검색 불가

User-agent: 네이버
Allow : /

User-Agent: 야후
Disallow: /

크롤링 합법? 불법?

❖ 4차 산업 혁명 시대에 데이터는 곧 ‘돈’

-데이터의 가치가 올라감에 따라 크롤링이 범죄가 되는 경우가 많이 발생

❖ 크롤링 분쟁

-후발 주자가 선행 주자를 빨리 따라잡기 위해 불법으로 데이터를 가져오는 행위들

잡 코리아 vs 사람 인

JOBKOREA 
vs. **saram****in**

야놀자 vs 여기어때

yanolja
vs. **여기어때.**

LESSON

파이썬에 대한 이해

```
animLength = toTime - fromTime

# Ask user for directory
filePath = c4d.storage.SaveDialog()
filePath, objName = os.path.split(filePath)
objName = objName + ". "
filePath = filePath + "\\ "

# Ask for confirmation
questionDialogText = "Obj Sequence will be saved as:\n\n"\
    "" + filePath + objName + "####.obj\n\n"\
    "From frame " + str(fromTime) + " to " + str(toTime) + " "
proceedBool = c4d.gui.QuestionDialog(questionDialogText)

if proceedBool == True:

    # Loop through animation and export frames
    for x in range(0,animLength):

        # change frame, redraw view
        moveTime = c4d.BaseTime(fromTime,docFps) + c4d.BaseTime(x)
        doc.SetTime(moveTime)
        c4d.EventAdd(c4d.EVENT_FORCEREDRAW)
        c4d.DrawViews(c4d.DRAWFLAGS_FORCEFULLREDRAW)

        # update status bar
        c4d.StatusSetText("Exporting " + str(x) + " of " + str(animLength))
        c4d.StatusSetBar(100.0*x/animLength)

        # add buffer 0.001
        bufferedNumber = str(doc.GetTime().GetFrame(docFps))
        if len(bufferedNumber) < 3:
```

파이썬의 사전적 의미

- ❖ 독이 없는 뱀으로 먹이를 몸으로 감아서 압사시키는 큰 뱀,
또는 그리스 신화에 나오는 악마



파이썬의 유래

- ❖ 1970년대 영국 BBC에서 방영한 “Monty Python’s Flying Circus” 라는 프로그램에서 유래됨
 - 이 프로그램은 매우 어색하고 이상한 행동을 보이는 사람들을 쇼 형태로 방영하며 많은 인기를 누렸다. 파이썬을 만든 Guido van Rossum이 이 프로그램의 엄청난 팬이었기에 파이썬이 되었다.
- ❖ 1989년 12월 크리스마스 휴가를 보내는 중이던 Guido van Rossum은 연휴 동안 무료함을 달래줄 무언가를 찾고 있었다. 마침 연휴라 연구실 또한 닫혀 있었고, 집에서 예전부터 생각했던 인터프리터를 만들어 보기로 결심한다.
 - 무료한 한 때를 보내기 위해 만들기 시작했던 언어가 오늘날 많은 사람들이 사용하는 파이썬(Python)이 되었다.

파이썬에 대한 이해

- ♣ 80년대 초 ABC라는 교육용 언어를 기반으로 설계, 발전한 언어
 - 효율적이고, 읽기 쉽고 되도록 코드를 간단하게 쓸 수 있도록 하려는 철학이 깊이 자리잡고 있음
- ♣ 파이썬의 핵심 철학



“아름다운 게 추한 것보다 낫다.”

“명시적인 것이 암시적인 것보다 낫다.”

“단순함이 복잡함 보다 낫다.”

“복잡함이 난해한 것보다 낫다.”

“가독성은 중요하다.”



파이썬에 대한 이해

❁ 스크립트(Scripts) 언어란?

- 연극의 대사 등이 적혀 있는 스크립트에서 유래
- 응용 소프트웨어를 제어하는 컴퓨터 프로그래밍 언어
- 소스코드를 한 줄 한 줄 읽어 바로 바로 실행하는 인터프리터 방식
 - 소스코드를 컴파일(Compile)하지 않고도 실행 할 수 있는 프로그래밍 언어

‘해석기’ 라는 뜻으로
프로그래밍 언어의 소스코드를
바로 실행하는 컴퓨터 프로그램
또는 환경을 말한다.

❁ 스크립트 언어 활용

- 파이썬(Python) 프로그래밍에 활용
- 리눅스의 셸(Shell), 윈도우즈의 배치(Batch) 등 시스템에 사용
- HTML, JavaScript 등의 웹 어플리케이션에 사용

파이썬에 대한 이해

♣ 컴파일 언어와 스크립트(scripts) 언어의 차이

컴파일 언어	구분	스크립트 언어
일괄 번역, 일괄 수행	수행과정	줄 단위로 번역 및 수행
번역 후 저장을 위해 필요	목적 프로그램	불필요
빠르다	실행 시간	느리다
많이 필요	메모리	적게 필요

파이썬 언어의 특징

☛ 대화 기능의 인터프리터 언어

```
In [2]: print(1) #컴퓨터와 개발자가 대화하듯
```

```
1
```

```
In [3]: print(10) #한 줄씩 주고받으며 프로그래밍
```

```
10
```

```
In [4]: print(3)
```

```
3
```

파이썬 언어의 특징

[illegible]

```
<class 'int'> <class 'str'> <class 'float'> <class 'bool'> <class 'int'>
```

파이썬 언어의 특징

```
a = "안녕"  
b = "하세요"
```

```
print (a+b)
```

안녕하세요

```
a = [1,2,3]  
b = [4,5,6]
```

```
print(a + b)
```

[1, 2, 3, 4, 5, 6]

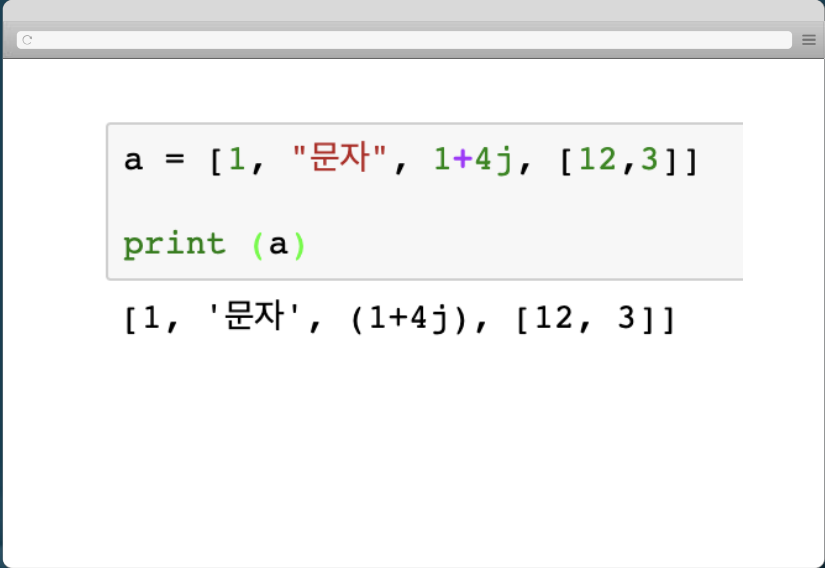
```
a = 1  
b = 2
```

```
print(a + b)
```

3

파이썬 언어의 특징

- 플랫폼 독립적 언어
 - 똑같은 코드를 Windows, Mac OS, Linux에서 별도의 과정 없이 완벽하게 실행 가능
- 개발 기간 단축에 초점을 둔 언어
- 간단하고 쉬운 문법
- 고수준의 내장 객체 자료형 제공



```
a = [1, "문자", 1+4j, [12, 3]]  
print (a)  
[1, '문자', (1+4j), [12, 3]]
```


파이썬 언어의 특징

메모리
자동 관리

쉬운
유지 보수

많은 수의
라이브러리
제공

높은
확장성

짧아지는
코드

들여쓰기
필수

LESSON

개발 환경 구축

```
animLength = toTime - fromTime

# Ask user for directory
filePath = c4d.storage.SaveDialog()
filePath, objName = os.path.split(filePath)
objName = objName + "-"
filePath = filePath + "\\ "

# Ask for confirmation
questionDialogText = "Obj Sequence will be saved as:\n\n"\
    "" + filePath + objName + "###.obj\n\n"\
    "From frame " + str(fromTime) + " to " + str(toTime) + " "
proceedBool = c4d.gui.QuestionDialog(questionDialogText)

if proceedBool == True:

    # Loop through animation and export frames
    for x in range(0,animLength):

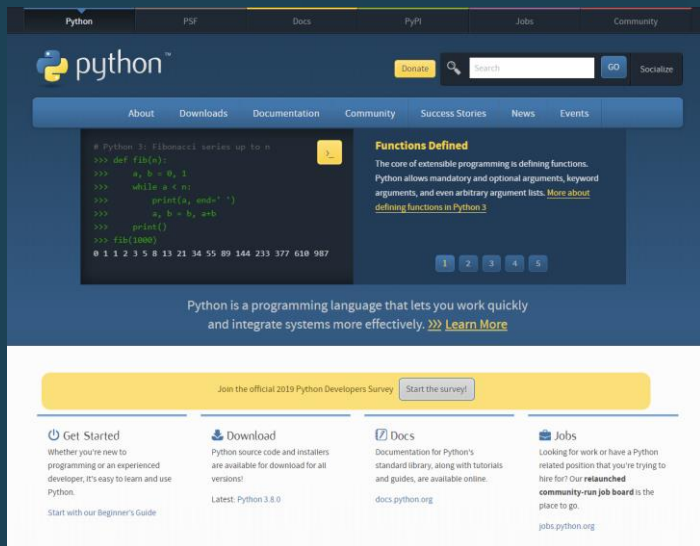
        # change frame, redraw view
        moveTime = c4d.BaseTime(fromTime,docFps) + c4d.BaseTime(x)
        doc.SetTime(moveTime)
        c4d.EventAdd(c4d.EVENT_FORCEREDRAW)
        c4d.DrawViews(c4d.DRAWFLAGS_FORCEFULLREDRAW)

        # update status bar
        c4d.StatusSetText("Exporting " + str(x) + " of " + str(animLength))
        c4d.StatusSetBar(100.0*x/animLength)

        # add buffer 0001
        bufferedNumber = str(doc.GetTime().GetFrame(docFps))
        if len(bufferedNumber) < 4:
```

파이썬 설치

🔗 파이썬 홈페이지에서 직접 파이썬 다운로드
(<https://www.python.org/>)



파이썬 통합 개발환경

Jupyter
notebook

이클립스

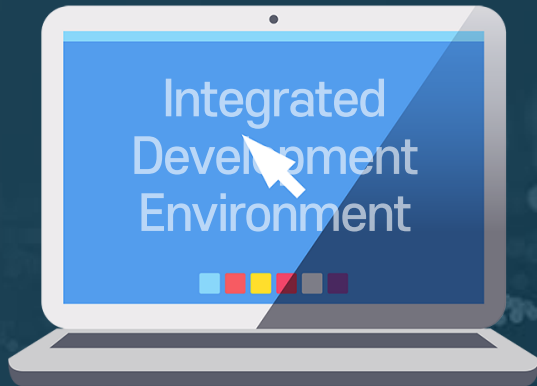
PyCharm

텍스트
에디터

비주얼
스튜디오

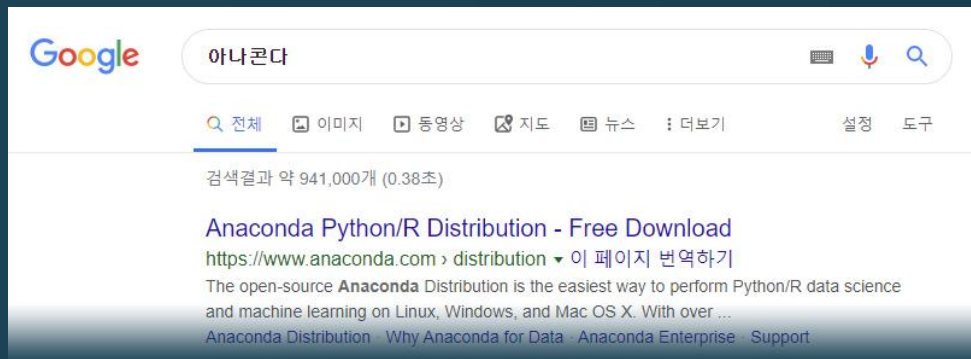
Integrated
Development
Environment

※ 메모장,
Sublime Text 등



파이썬 전용 Anaconda 환경 사용

- ❖ 수학·과학 분야에서 필요한 주요 라이브러리들(numpy, pandas ..)을 함께 설치
- ❖ 필요에 따라 Environment를 여러 개 만들어 각자 다른 환경에서 프로그램 가능
- ❖ 최근 4차산업혁명 관련해 IT분야에서 파이썬을 많이 활용하는데 Anaconda가 거의 정석처럼 사용되고 있는 추세임
- ❖ <https://www.anaconda.com/> (Anaconda 주소)



우측 상단의 Download

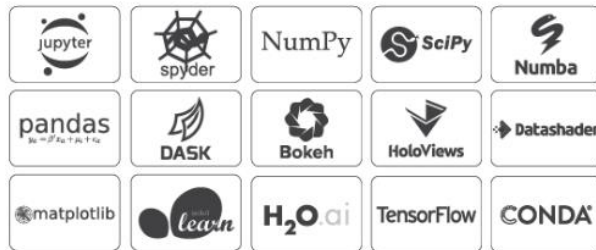
Anaconda Distribution

The World's Most Popular Python/R Data Science Platform

Download

The open-source [Anaconda Distribution](#) is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 15 million users worldwide, it is the industry standard for developing, testing, and training on a single machine, enabling *individual data scientists* to:

- Quickly download 1,500+ Python/R data science packages
- Manage libraries, dependencies, and environments with Conda
- Develop and train machine learning and deep learning models with scikit-learn, TensorFlow, and Theano
- Analyze data with scalability and performance with Dask, NumPy, pandas, and Numba
- Visualize results with Matplotlib, Bokeh, Datashader, and Holoviews





Windows



macOS



Linux

Windows, Mac OS, Linux 버전 제공

Anaconda 2019.10 for Windows Installer

Python 3.7 version

Download

64-Bit Graphical Installer (462 MB)

32-Bit Graphical Installer (410 MB)

Python 2.7 version

Download

64-Bit Graphical Installer (413 MB)

32-Bit Graphical Installer (356 MB)

추천 설치 옵션

🔌 All Users

Install for:

- ☐ Just Me (recommended)
- ☒ All Users (requires admin privileges)

주의사항 !

설치 경로에 한글이
있으면 제대로
설치가 되지 않을 수
있습니다.

🔌 Add PATH

Advanced Options

☒ Add Anaconda to my PATH environment variable

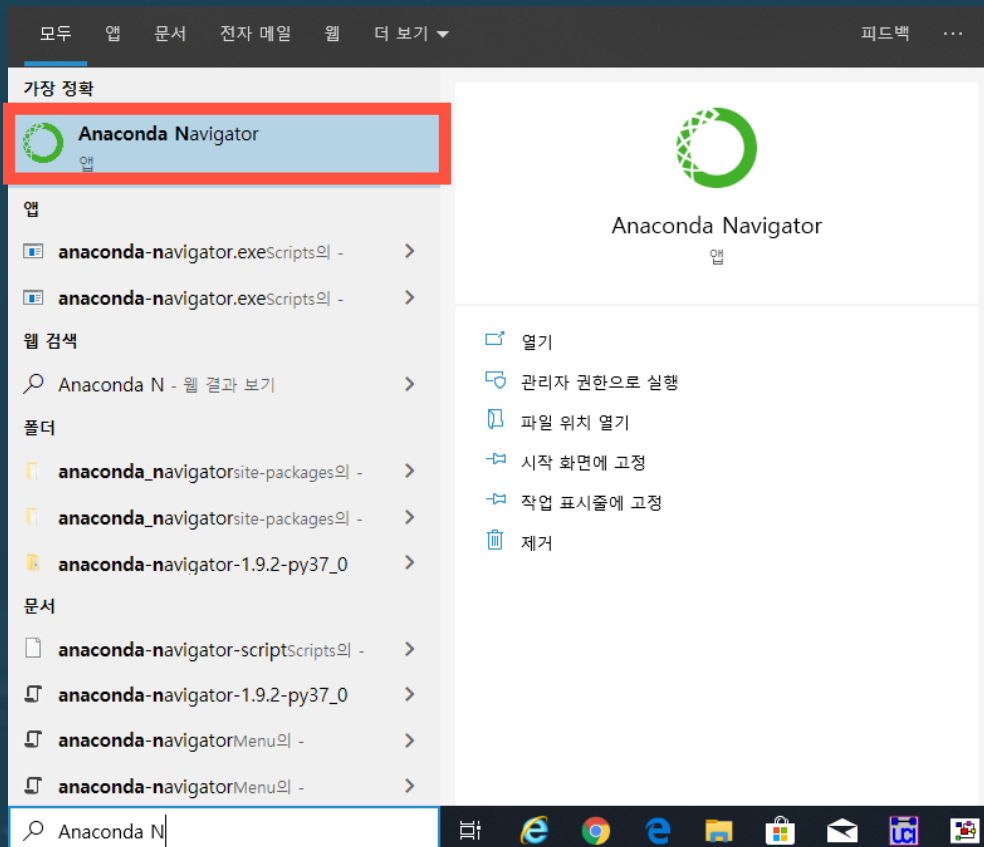
Not recommended. Instead, open Anaconda with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.

☒ Register Anaconda as my default Python 3.6

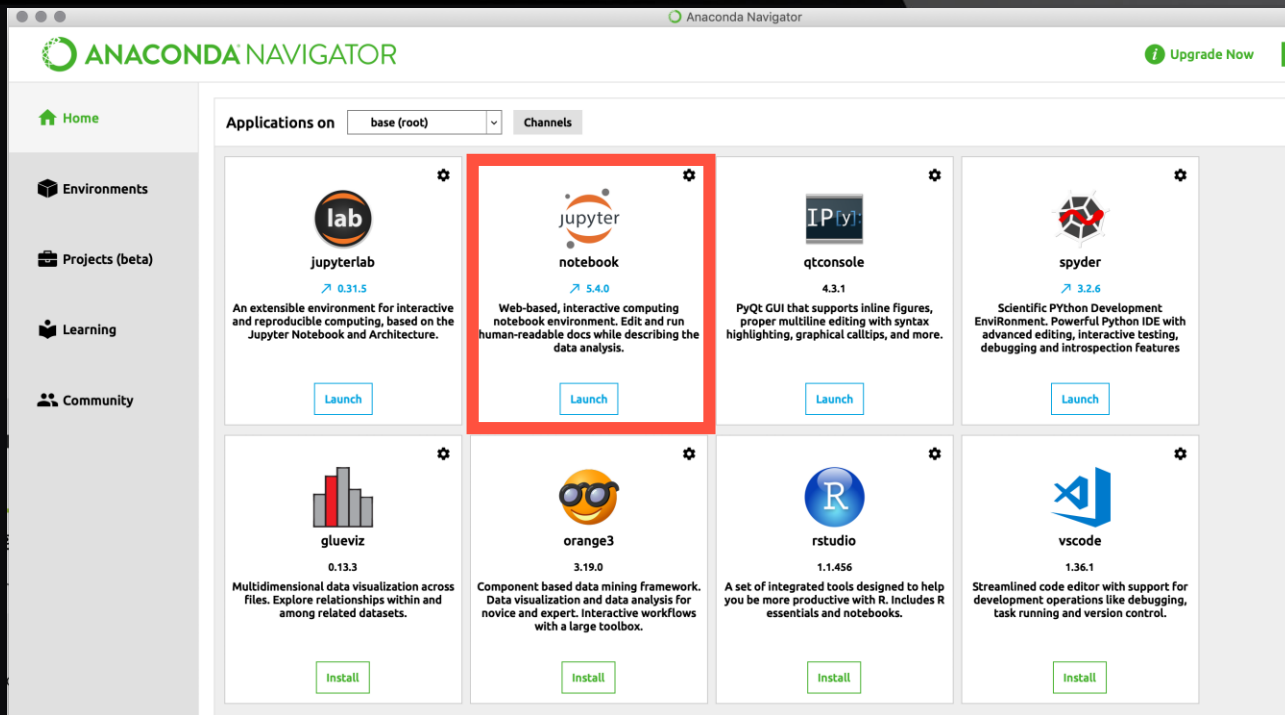
This will allow other programs, such as Python Tools for Visual Studio, PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.6 on the system.

파이썬 실행하기

- Anaconda Navigator
- 콘솔 창에서 실행



Anaconda Navigator



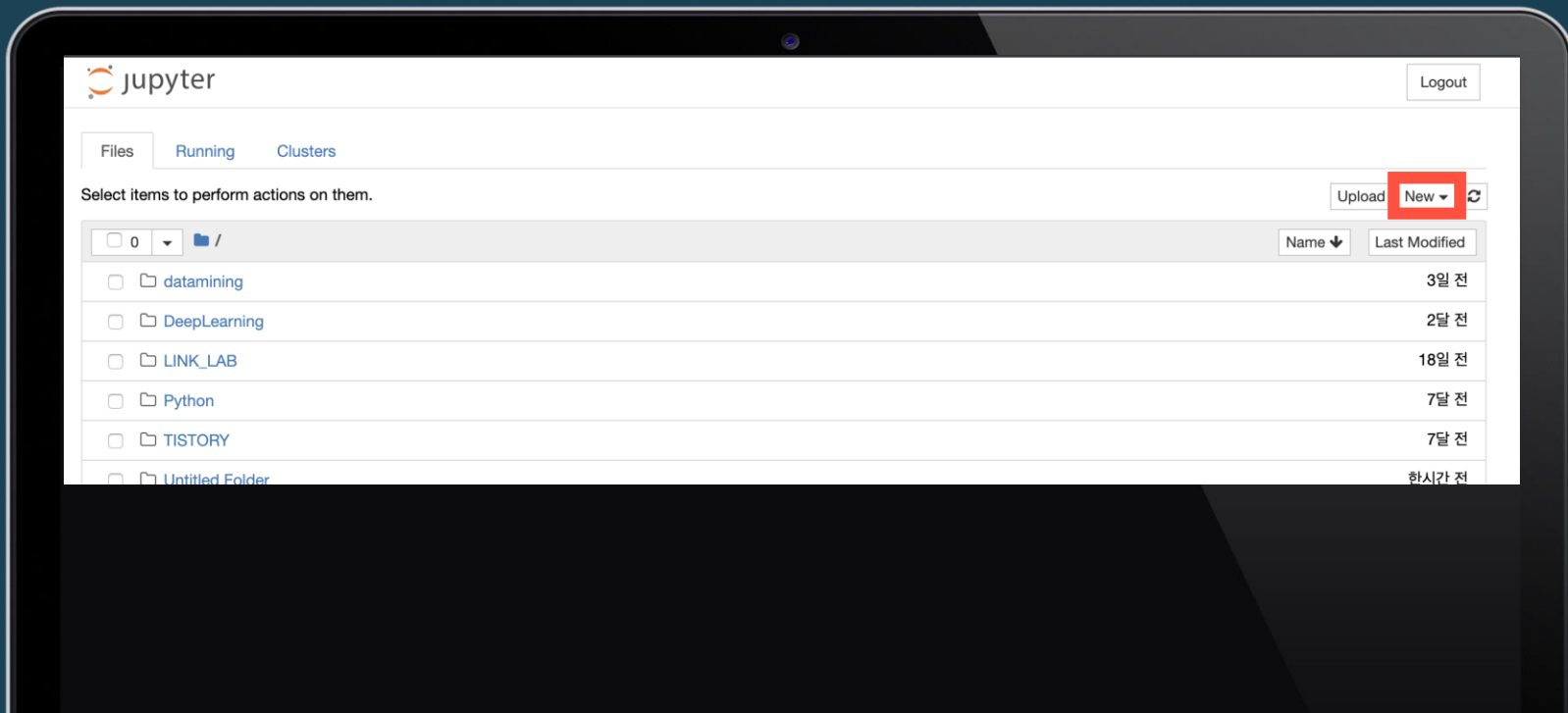
콘솔 창에서 실행

Anaconda Prompt

```
(base) C:\Users\zzzoost>jupyter notebook
```

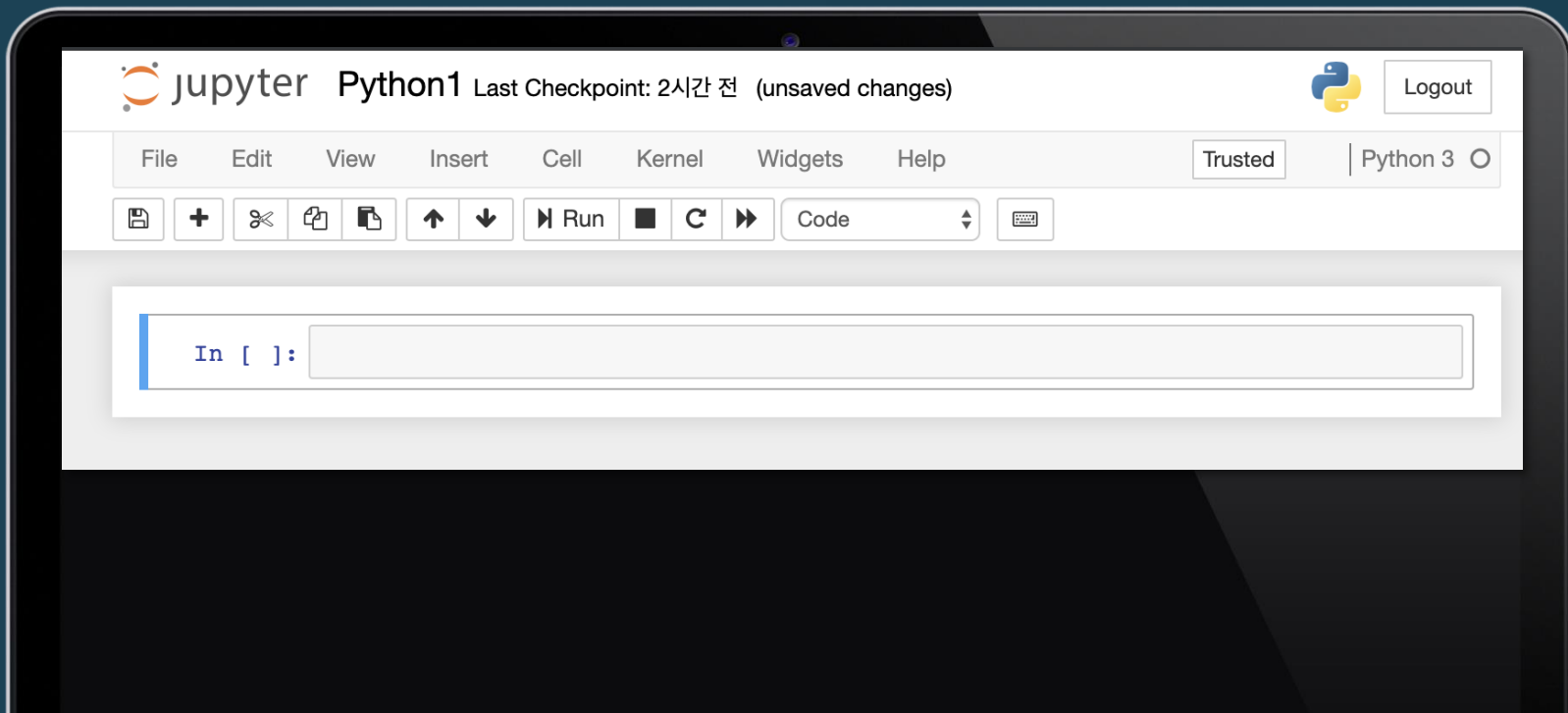
파이썬 실행하기

📁 우측 상단 New → Python 3 클릭



파이썬 실행하기

🔗 우측 상단 New → Python 3 클릭



그 외 추천 하는 코딩 환경 구성 방법

♣ PyCharm 설치

(<https://www.jetbrains.com/pycharm/download/>)

→ 무료 버전 다운로드 및 설치 (Community 버전)

♣ 무료 코드 에디터 설치

→ Sublime Text3 설치 (<https://www.sublimetext.com/3>)

→ 아톰, 브라켓, VS Code 등