

파이썬으로 시작하는 인공지능



Programming language



Machine language

CPU가 직접 해독하고 실행할 수 있는 비트 단위로 쓰인 컴퓨터 언어

Assembly language

기계어와 일대일 기호로 대응이 되는 컴퓨터 프로그래밍의 저급 언어

Assembly Language

```
mov ecx, ebx
mov esp, edx
mov edx, r9d
mov rax, rdx
```

Programmer

Assembler + Linker

Machine Language

```
100101011001
010011111011
111010101101
01010101010
```

Processor

High level programming language

사람이 이해하기 쉽게 작성된 프로그래밍 언어



Compiler vs Interpreter



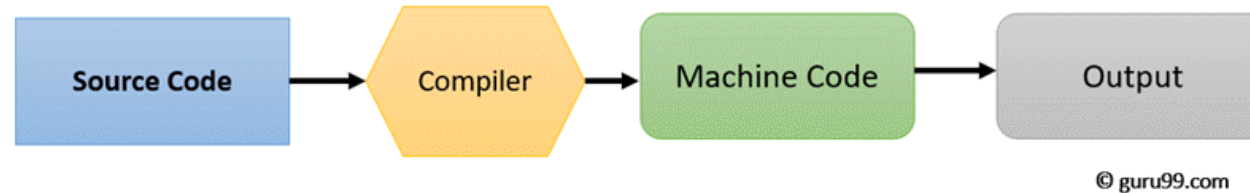
Compiler

특정 프로그래밍 언어로 쓰여 있는 문서를 다른 프로그래밍 언어로 옮기는 언어 번역 프로그램
고급 프로그래밍 언어를 실행 프로그램으로 만들기 위해 저급 프로그래밍 언어로 바꾸는 데 사용

Interpreter

고급 언어로 작성된 원시코드 명령어들을 한번에 한 줄씩 읽어 들어서 실행하는 프로그램

How Compiler Works



How Interpreter Works



고급언어로 작성된 프로그램들을 실행하는 데에는 두 가지 방법이 있다. 가장 일반적인 방법은 프로그램을 컴파일 하는 것이고, 다른 하나는 프로그램을 인터프리터에 통과시키는 방법이다. 인터프리터는 고급 명령어들을 중간 형태로 번역한 다음, 그것을 실행한다. 이와는 대조적으로, 컴파일러는 고급 명령어들을 직접 기계어로 번역한다.

Error



Syntax Error(구문적 오류)

프로그래밍 언어의 문법적인 에러
컴파일 과정에서 나오는 것들이며, 구문 오류, 컴파일 타임 에러
예를 들어, 키워드를 잘못 작성하거나, 구두점을 빠뜨리거나, 괄호를 연 후 닫지 않는 것과 같은 경우
문법 오류는 보통 컴파일러가 어떤 줄에서, 어떤 이유로 발생했는지 알려주기 때문에 해결하기 쉬운 편

Runtime Error(실행 오류)

프로그램 실행 중 발생하여 프로그램이 비정상적으로 종료되게 하는 오류
프로그램이 실행되는 중 실행할 수 없는 연산을 만나면 발생하는 오류
대표적인 런타임 에러로는 잘못된 입력이 있는 경우 발생하는 입력 오류가 있다.

Semantic Error(의미적 오류)

프로그래밍 언어의 문법은 정상적이지만 실행의 결과가 원하는 대로 나오지 않는 오류
문법적으로는 옳기 때문에 컴파일 오류는 나타나지 않지만, 의도하지 않은 결과가 나온다는 점에서 문제
컴파일러가 오류를 잡아주지 않아 사람이 검출해야 해서 다른 오류보다 수정이 어렵다.

What is the difference between syntax and semantic errors?

Debug



Debugging

컴퓨터 프로그램 개발 단계 중에 발생하는 시스템의 논리적인 오류나 비정상적 연산(버그)을 찾아내고 수정하는 작업 과정



Python

<https://www.python.org/>



파이썬의 특징

파이썬은 인간다운 언어이다
문법이 쉬워 빠르게 배울 수 있다
무료이지만 강력하다
간결하다
프로그래밍을 즐기게 해 준다
개발 속도가 빠르다

파이썬의 사용

서버 측 웹 개발
Python 스크립트를 사용한 자동화
데이터 과학 및 기계 학습
소프트웨어 개발
소프트웨어 테스트 자동화

Documentation

<https://docs.python.org>



IDE(**I**ntegrated **D**evelopment **E**nvironment)



프로그램 개발에 관련된 모든 작업을 하나의 프로그램 안에서 처리하는 환경을 제공하는 소프트웨어

- Text editor
- Debugger
- Compiler
- Code completion
- Programming language support
- Integrations and plugins



Cloud IDE vs Local IDE



Local or standalone IDE

Customization

Lots of free and open source options

No internet connection required

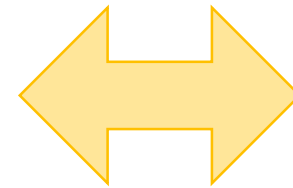
Benefits

Installing and configuring your environment

Configuration drift

Compute intensity

Pain point



Cloud IDE

Both local and cloud IDEs have distinctive advantages and disadvantages.
What works best for you depends upon your situation and needs.

Cloud IDE vs Local IDE



클라우드 방식

프로그램과 데이터가 서버에 저장되고 관리

서버에 환경이 대부분 갖추어져 있어 로그인하면 바로 프로그래밍 가능

인터넷 연결만 있으면 어느 곳에서도 개발 가능. 협업 가능

구글의 Colab, 아마존의 SageMaker, 마이크로소프트의 Azure

내 프로젝트에 최적의 환경을 갖추 수 없는 한계

간편한 실습에서는 colab

특히 단기 프로그래밍 과정에서는 설치 과정에 대한 시간 절약

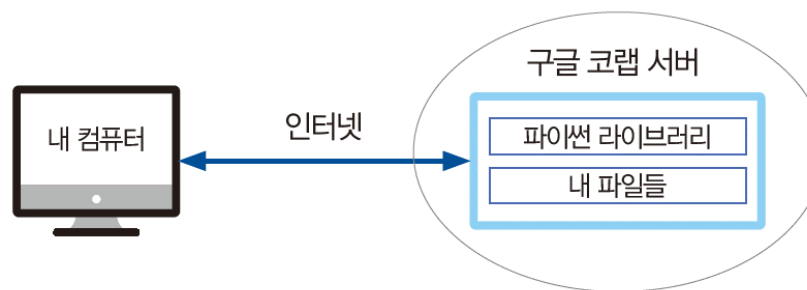
본격적인 개발에서는 스탠드얼론 방식

특히 인공지능 공학자로 성장하려 하는 경우에는 스스로 최적 환경을 구축할 수 있는 스탠드얼론 방식이 적절함

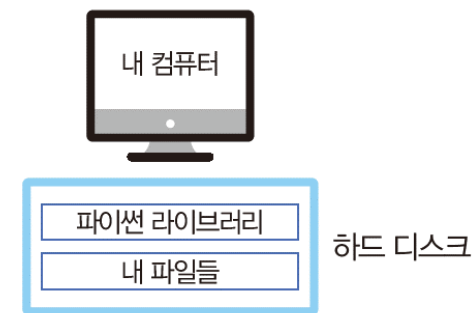
스탠드얼론 방식

자신에 최적의 환경 구축 가능. 프로그램과 데이터가 자신의 컴퓨터에 저장

소프트웨어를 설치하고 환경을 스스로 구축해야 함



(a) 클라우드 방식



(b) 스탠드얼론 방식

Cloud IDE



0. Colab

<https://colab.research.google.com/?hl=ko>

1. Replit

<https://replit.com/languages/python3>

2. Programiz

<https://www.programiz.com/python-programming/online-compiler/>

3. codingground

https://www.tutorialspoint.com/execute_python_online.php

4. online-python

<https://www.online-python.com/>

5. And so on (search : "online python")

Colab



<https://colab.research.google.com/?hl=ko>

예

최근 사용

Google Drive

GitHub

업로드

노트 필터링

제목	마지막 연 시간	처음 연 시간	
Colaboratory에 오신 것을 환영합니다	오후 9:33	6월 20일	
	오후 9:33	오후 5:03	
	오후 8:48	오후 8:48	
	9월 2일	9월 2일	
	9월 2일	9월 2일	

새 노트

Untitled0.ipynb - Colaboratory

colab.research.google.com/drive/14U5dPCPQvxpV_-Z-rQ9lnLg60suawy1?hl=ko

Untitled0.ipynb

파일 수정 보기 삽입 런타임 도구 도움말 모든 변경사항이 저장됨

+ 코드 + 텍스트

```
print('Hello python world!!!')
```

```
Hello python world!!!
```

0초 오후 9:38에 완료됨

드라이브

드라이브에서 검색

+ 새로 만들기

내 드라이브

컴퓨터

공유 문서함

최근 문서함

중요 문서함

내 드라이브 > Colab Notebooks

1개 선택됨

이름

Hello.ipynb의 사본

Untitled0.ipynb

Ctrl + F9

Standalone



파이썬 설치

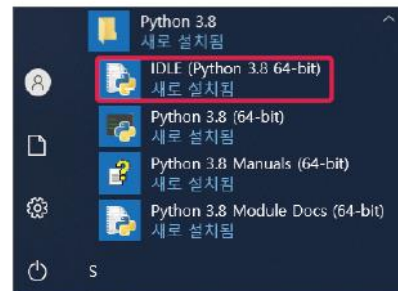
<http://www.python.org>에서 설치 파일 다운로드



(a) 다운로드 화면



(b) 설치 시작 화면



(c) 설치 확인

Standalone



IDLE

IDLE은 파이썬의 통합 개발 및 학습 환경



```
IDLE Shell 3.9.13
File Edit Shell Debug Options Window Help
Python 3.9.13 (tags/v3.9.13:6de2ca5, May 17 2022, 16:36:42) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 12+37
49
>>> print(12+37)
49
>>> print('Python'+ 'is exciting.')
Pythonis exciting.
>>> |
```

```
IDLE Shell 3.9.13
File Edit Shell Debug Options Window Help
Python 3.9.13 (tags/v3.9.13:6de2ca5, May 17 2022, 16:36:42) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 12+37
49
>>> print(12+37)
49
>>> print('Python'+ 'is exciting.')
Pythonis exciting.
>>>
===== RESTART: C:/0python/test.py =====
49
Pythonis exciting.
a = 11 , b = 10 , c = 21
>>> |
```

```
*untitled*
File Edit Format Run Options Window Help

#연산자 오버로딩 예시
print(12+37)                # +는 두 정수를 더하는 연산자
print('Python'+ 'is exciting.') # +는 두 문자열을 결합하는 연산자

#라이브러리 불러오기
import random               # random이라는 라이브러리를 불러옴

#정수 난수 생성
a=random.randint(10, 20)    # 10 ~ 20 사이의 난수를 생성하고 변수 a에 대입
b=random.randint(10, 20)    # 10 ~ 20 사이의 난수를 생성하고 변수 b에 대입

#덧셈을 하고 결과를 출력
c=a+b                       # a와 b를 더하여 변수 c에 대입
print('a = ', a, ', b = ', b, ', c = ',c) # 변수 a, b, c를 출력
```

Anaconda



1. 아나콘다 설치

<https://www.anaconda.com/>

Meet Anaconda Assistant
AI-powered Code Generation, Insights, and Debugging in Your Anaconda Notebook.
[Discover Anaconda Assistant](#)

Free Download
Everything you need to get started in data science on your workstation.

- ✓ Free distribution install
- ✓ Thousands of the most fundamental DS, AI, and ML packages
- ✓ Manage packages and environments from desktop application
- ✓ Deploy across hardware and software platforms

[Code in the Cloud](#) [Download](#)

Get Additional Installers
Windows | Apple | Linux

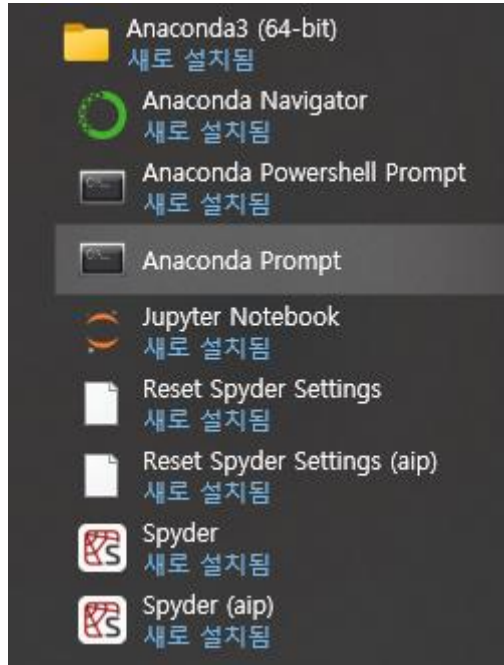
Installation Log:

- Anaconda3 (64-bit) 새로 설치됨
- Anaconda Navigator 새로 설치됨
- Anaconda Powershell Prompt 새로 설치됨
- Anaconda Prompt 새로 설치됨
- Jupyter Notebook 새로 설치됨
- Reset Spyder Settings 새로 설치됨
- Spyder 새로 설치됨

Anaconda



2. 가상 환경 생성(아래 1번 명령)



```
(base> C:\> conda create -n aip python=3.7
(base> C:\> conda activate aip
(aip) C:\> conda install spyder
(aip) C:\> conda install tensorflow
```

```
# 1. aip 가상 환경 생성
# 2. aip 가상 환경으로 이동
# 3. aip 가상 환경에 스파이더 설치
# 4. aip 가상 환경에 텐서플로 설치
```


Anaconda



3. 가상 환경에 Spyder와 tensorflow 모듈 설치 (2, 3, 4번 명령)

4. 스파이더에서 작업 디렉토리 설정

- 1) 바탕 화면에 aipSources 폴더 만들기
- 2) 스파이더에서 [Tools]-[Preferences]-[Current working directory]-[Startup]-[The following directory]를 선택하고 aipSources를 브라우징하여 설정

5. conda list 명령어로 가상 환경에 설치된 모듈 목록 확인

```
Anaconda Prompt
#
# Name                Version                Build  Channel
_anaconda_depends     2023.07                py311_1
abseil-cpp             202111102.0            hd77b12b_0
aiobotocore            2.4.2                  py311haa95532_0
aiofiles               22.1.0                 py311haa95532_0
aiohttp                3.8.3                  py311h2bbff1b_0
aioitertools           0.7.1                  pyhd3eb1b0_0
aiosignal              1.2.0                  pyhd3eb1b0_0
aiosqlite              0.18.0                 py311haa95532_0
alabaster              0.7.12                 pyhd3eb1b0_0
anaconda-catalogs      0.2.0                  py311haa95532_0
anaconda-client         1.12.0                 py311haa95532_0
anaconda-navigator      2.4.2                  py311haa95532_0
anaconda-project        0.11.1                 py311haa95532_0
anyio                  3.5.0                  py311haa95532_0
appdirs                1.4.4                  pyhd3eb1b0_0
```


Anaconda navigator



Anaconda Navigator

File Help

ANAconda.NAVIGATOR

Connect

Home

Environments

Learning

Community

Anaconda Notebooks

A Full Python IDE directly from the browser

Documentation

Anaconda Blog

All applications on base (root) Channels

DataSpell

DataSpell is an IDE for exploratory data analysis and prototyping machine learning models. It combines the interactivity of Jupyter notebooks with the intelligent Python and R coding assistance of PyCharm in one user-friendly environment.

Install

Anaconda Notebooks

Cloud-hosted notebook service from Anaconda. Launch a preconfigured environment with hundreds of packages and store project files with persistent cloud storage.

Launch

CMD.exe Prompt

0.1.1

Run a cmd.exe terminal with your current environment from Navigator activated

Launch

JupyterLab

3.6.3

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Launch

jupyter Notebook

6.5.4

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Launch

Powershell Prompt

0.0.1

Run a Powershell terminal with your current environment from Navigator activated

Launch

PyCharm Professional

2023.2.1

A Full-Fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.

Launch

Qt Console

5.4.2

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Launch

Spyder

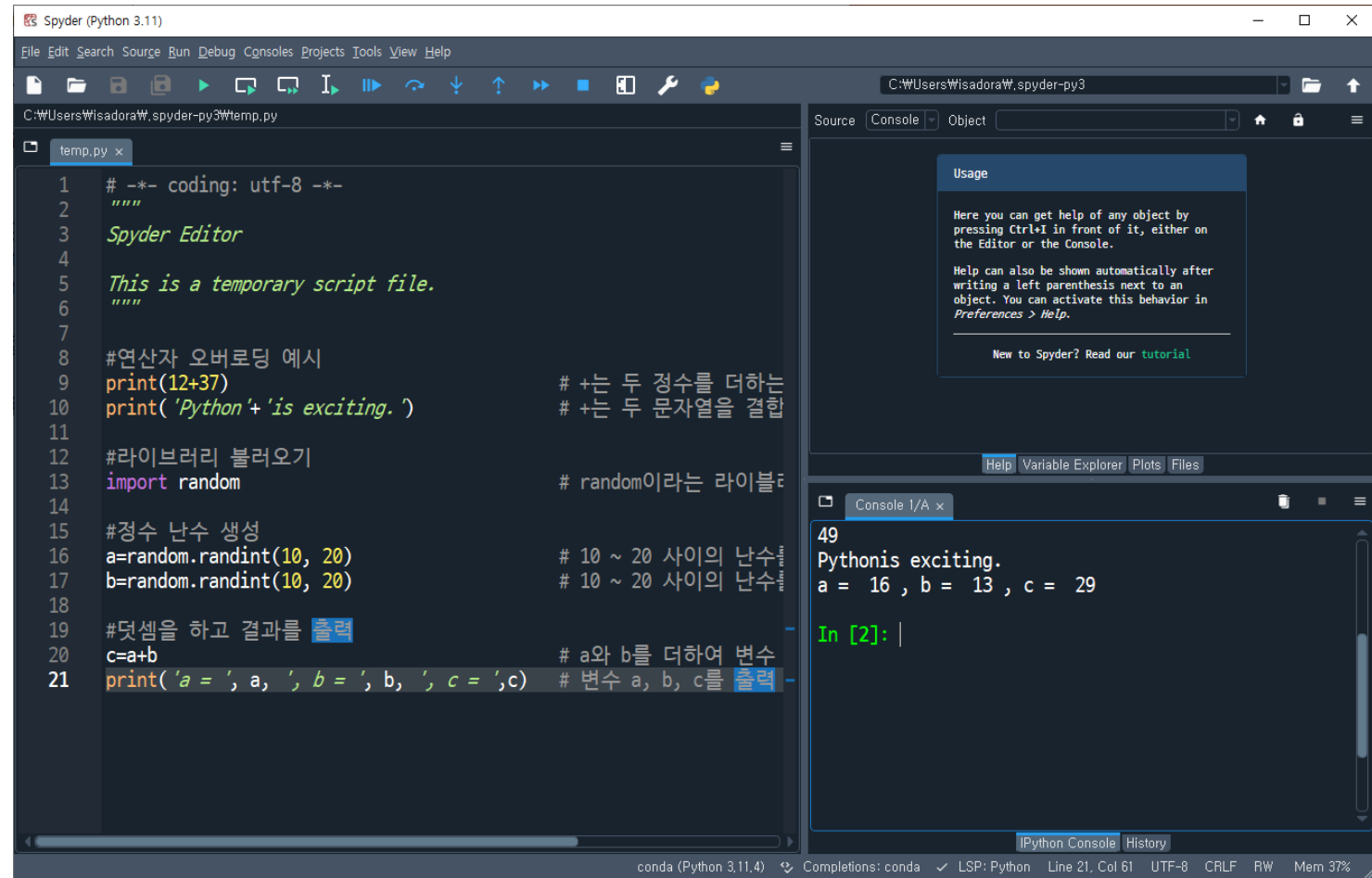


Spyder

5.4.3

Scientific PYthon Development
EnviRonment. Powerful Python IDE with
advanced editing, interactive testing,
debugging and introspection Features

Launch



Jupyter notebook



Notebook

6.5.4

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Launch



localhost:8888/tree

jupyter

Quit Logout

Files Running Clusters

jupyter Untitled1 Last Checkpoint: 몇 초 전 (unsaved changes)

Logout

Upload New

Notebook:

Python 3 (ipykernel)

Other:

Text File

Folder

Terminal

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3 (ipykernel)

In [1]:

```
#연산자 오버로딩 예시
print(12+37)
print('Python'+ 'is exciting.')

#라이브러리 불러오기
import random

#정수 난수 생성
a=random.randint(10, 20)
b=random.randint(10, 20)

#덧셈을 하고 결과를 출력
c=a+b
print('a = ', a, ', b = ', b, ', c = ',c)
```

49

Pythonis exciting.

a = 10 , b = 11 , c = 21

In []:

PyCharm

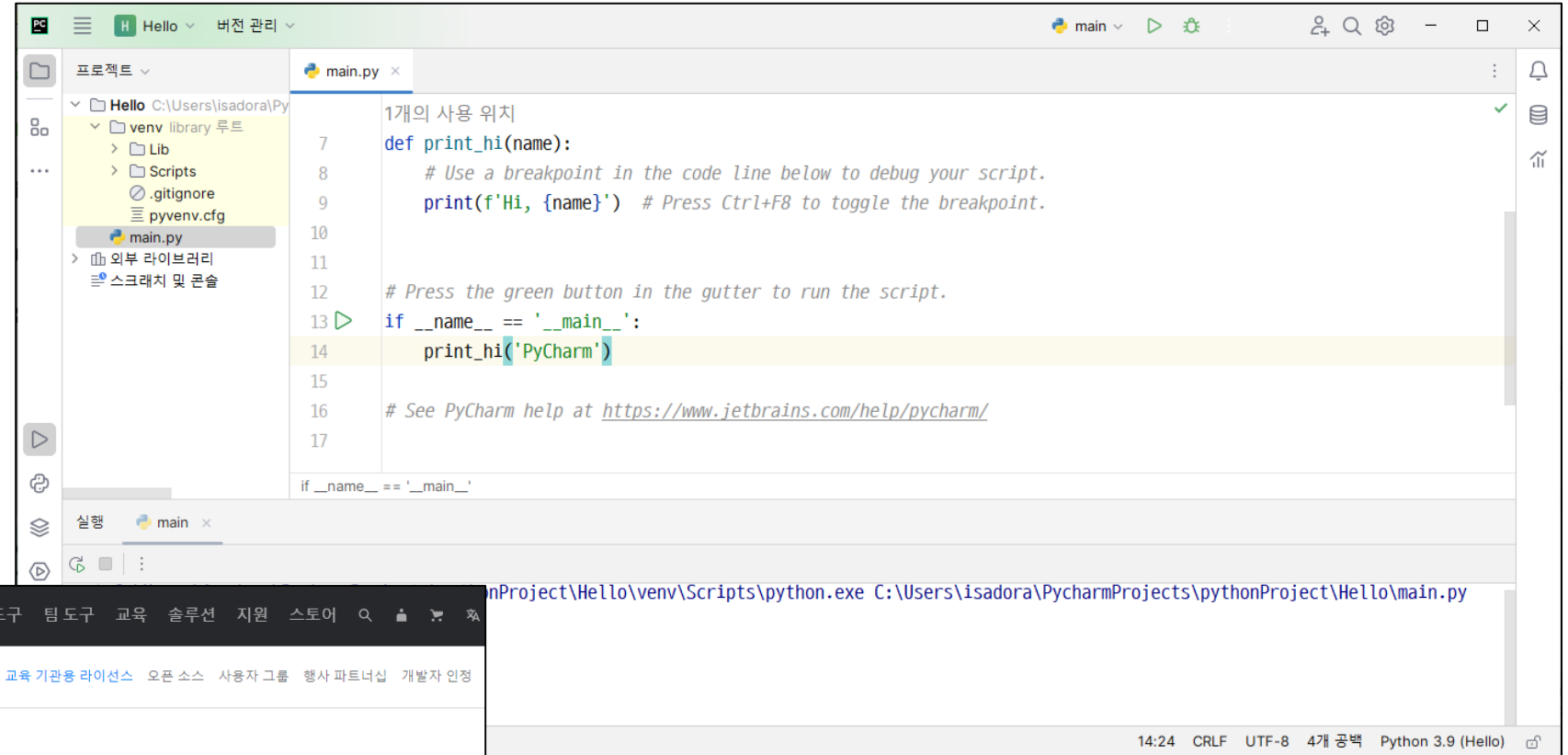


PyCharm Professional

2023.2.1

A full-fledged IDE by JetBrains for both
Scientific and Web Python development.
Supports HTML, JS, and SQL.

Launch



**JET
BRAINS**

개발자 도구 팀 도구 교육 솔루션 지원 스토어

무료 라이선스 프로그램

교육 기관용 라이선스 오픈 소스 사용자 그룹 행사 파트너십 개발자 인정

무료 교육용 라이선스

업계 최고의 JetBrains 개발 도구를 사용해 코딩을 학습하고 교육하세요!

Visual studio code



VS Code

1.80.1

Streamlined code editor with support for development operations like debugging, task running and version control.

Launch

```
File Edit Selection View Go Run Terminal Help • test.py - Python - Visual Studio Code

EXPLORER
  PYTHON
    test.py

test.py
1  #연산자 오버로딩 예시
2  print(12+37)
3  print('Python'+ 'is exciting.')
4
5  #라이브러리 불러오기
6  import random
7
8  #정수 난수 생성
9  a=random.randint(10, 20)
10 b=random.randint(10, 20)
11
12 #덧셈을 하고 결과를 출력
13 c=a+b
14 print('a = ', a, ', b = ', b, ', c = ',c)
15
16
17

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS Python Debug Console
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
49
Pythonis exciting.
a = 15 , b = 15 , c = 30
PS C:\python>
```

예제 소스 다운로드



자료 다운로드시 유의사항

▶ [부록/자료 관련 FAQ](#)

Q. 다운로드한 파일이 압축해제가 되지 않습니다.

Q. 구글 드라이브(GDrive)로 제공하는 부록/예제소스가 다운로드되지 않습니다.







IT CookBook, 파이썬으로 만드는 인공지능

by 오일석 외 1명

브랜드 : 한빛아카데미

발행일 : 2021-01-05

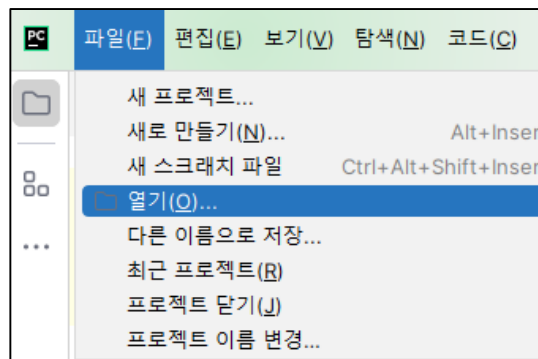
[오탈자 보기](#)

-  예제소스3 2021-01-21
-  예제소스2 2021-01-21
-  예제소스1 2021-01-21
-  [구글 드라이브\(GDrive\)에서 부록/예제소스 다운로드하는 방법](#)

인공지능 프로그래밍 예제 1



셈 지능

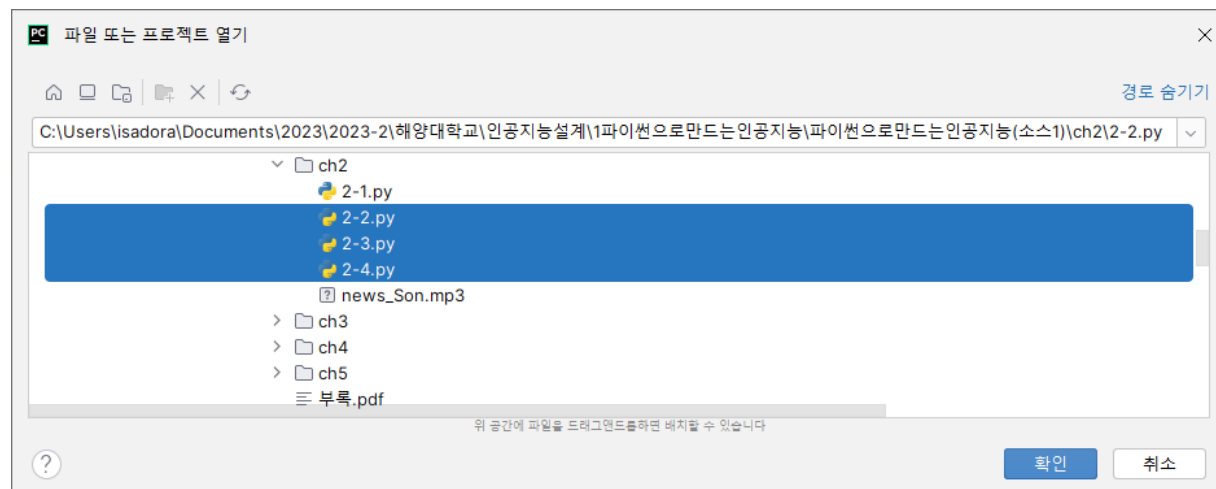


import time # 시간 측정 라이브러리

```
start=time.time() # 시작 직전 시각을 기록
sum=0
for i in range(1,1000000001): # 1억 번 반복
    sum=sum+i
```

end=time.time() # 끝난 직후 시각을 기록

```
print('1+2+...+1000000000=', sum)
print('소요 시간은 ', end-start, '초입니다.') # 시간 차이를 계산하여 출력
```



```
C:\Users\isadora\PycharmProjects\pythonProject\
1+2+...+1000000000= 500000000500000000
소요 시간은  4.570966005325317 초입니다.
```

종료 코드 0(으)로 완료된 프로세스

인공지능 프로그래밍 예제 2



뉴스 기사를 자동으로 작성하는 인공지능 기자

```
from datetime import datetime
```

```
# 경기 결과 입력 받는 곳
place=input("경기가 열린 곳은? ")
time=input("경기가 열린 시간은? ")
opponent=input("상대 팀은? ")
goals=input("손흥민은 몇 골을 넣었나요? ")
aids=input("도움은 몇 개인가요? ")
score_me=input("손흥민 팀이 넣은 골 수는? ")
score_you=input("상대 팀이 넣은 골 수는? ")
```

```
# 기사 작성하는 곳
news="[프리미어 리그 속보("+str(datetime.now())+")]\n"
news=news+"손흥민 선수는 "+place+"에서 "+time+"에 열린 경기에 출전하였습니다.\n"
news=news+"상대 팀은 "+opponent+"입니다. "
```

```
if score_me>score_you:
    news=news+"손흥민 선수의 팀이 "+score_me+"골을 넣어 "+score_you+"골을 넣은 상대 팀을 이겼습니다. \n"
elif score_me<score_you:
    news=news+"손흥민 선수의 팀이 "+score_me+"골을 넣어 "+score_you+"골을 넣은 상대 팀에게 졌습니다. \n"
else:
    news=news+"두 팀은 "+score_me+"대"+score_you+"로 비겼습니다. \n"

if int(goals)>0 and int(aids)>0:
    news=news+"손흥민 선수는 "+goals+"골에 도움 "+aids+"개로 승리를 크게 이끌었습니다. \n"
elif int(goals)>0 and int(aids)==0:
    news=news+"손흥민 선수는 "+goals+"골로 승리를 이끌었습니다. \n"
elif int(goals)==0 and int(aids)>0:
    news=news+"손흥민 선수는 골은 없지만 도움 "+aids+"개로 승리하는 데 공헌하였습니다. \n"
else:
    news=news+"아쉽게도 이번 경기에서 손흥민의 발끝은 침묵을 지켰습니다. \n"

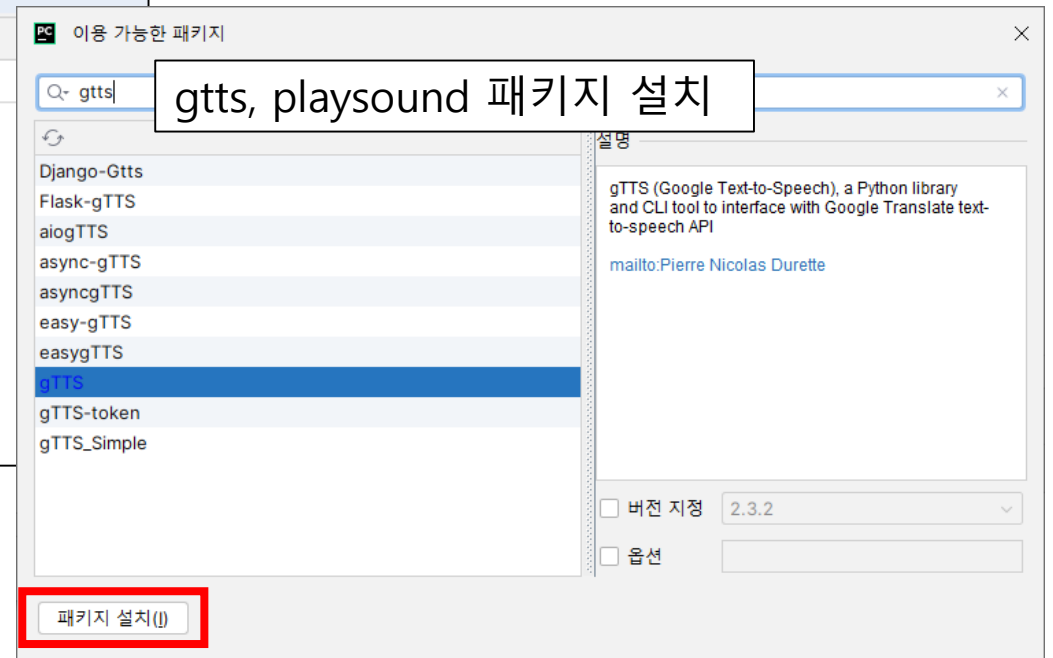
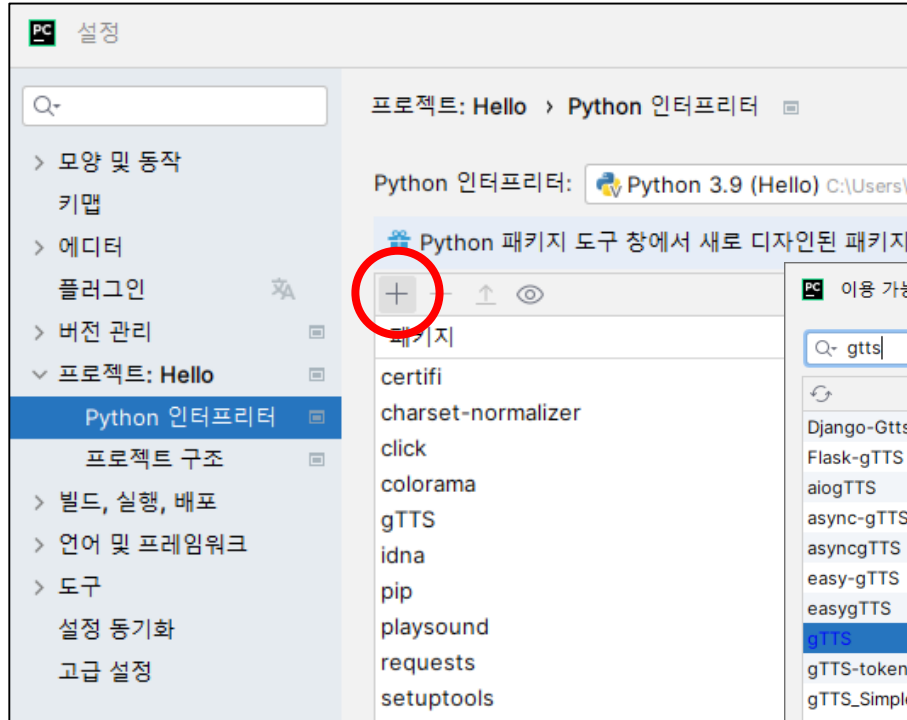
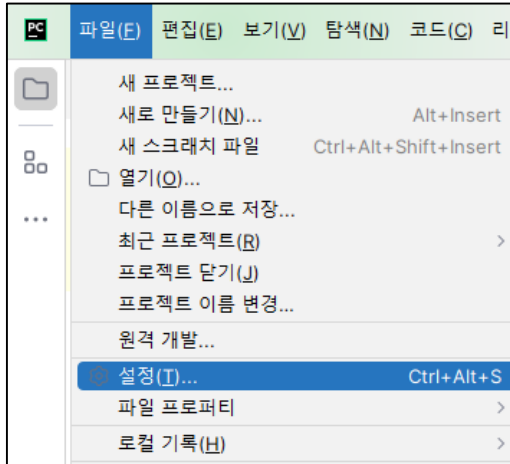
print(news)
```

```
경기가 열린 곳은? 런던 스타디움
경기가 열린 시간은? 9월 22일 오후 7시
상대 팀은? 맨체스터 유나이티드
손흥민은 몇 골을 넣었나요? 2
도움은 몇 개인가요? 1
손흥민 팀이 넣은 골 수는? 4
상대 팀이 넣은 골 수는? 2
[프리미어 리그 속보(2023-09-16 19:14:15.598875)]
손흥민 선수는 런던 스타디움에서 9월 22일 오후 7시에 열린 경기에 출전하였습니다.
상대 팀은 맨체스터 유나이티드입니다. 손흥민 선수의 팀이 4골을 넣어 2골을 넣은 상대 팀을 이겼습니다.
손흥민 선수는 2골에 도움 1개로 승리를 크게 이끌었습니다.
```


인공지능 프로그래밍 예제 2



뉴스 기사를 자동으로 작성 및 읽어 주는 인공지능 기자2



인공지능 프로그래밍 예제 2



```
from datetime import datetime
```

```
# 경기 결과 입력 받는 곳
place=input("경기가 열린 곳은? ")
time=input("경기가 열린 시간은? ")
opponent=input("상대 팀은? ")
goals=input("손흥민은 몇 골을 넣었나요? ")
aids=input("도움은 몇 개인가요? ")
score_me=input("손흥민 팀이 넣은 골 수는? ")
score_you=input("상대 팀이 넣은 골 수는? ")
```

```
# 기사 작성하는 곳
news="[프리미어 리그 속보("+str(datetime.now())+")]\\n"
news=news+"손흥민 선수는 "+place+"에서 "+time+"에 열린 경기에 출전하였습니다.\\n"
news=news+"상대 팀은 "+opponent+"입니다. "
```

```
if score_me>score_you:
    news=news+"손흥민 선수의 팀이 "+score_me+"골을 넣어 "+score_you+"골을 넣은 상대 팀을 이겼습니다. \\n"
elif score_me<score_you:
    news=news+"손흥민 선수의 팀이 "+score_me+"골을 넣어 "+score_you+"골을 넣은 상대 팀에게 졌습니다. \\n"
else:
    news=news+"두 팀은 "+score_me+"대"+score_you+"로 비겼습니다. \\n"

if int(goals)>0 and int(aids)>0:
    news=news+"손흥민 선수는 "+goals+"골에 도움 "+aids+"개로 승리를 크게 이끌었습니다. \\n"
elif int(goals)>0 and int(aids)==0:
    news=news+"손흥민 선수는 "+goals+"골로 승리를 이끌었습니다. \\n"
elif int(goals)==0 and int(aids)>0:
    news=news+"손흥민 선수는 골은 없지만 도움 "+aids+"개로 승리하는 데 공헌하였습니다. \\n"
else:
    news=news+"아쉽게도 이번 경기에서 손흥민의 발끝은 침묵을 지켰습니다. \\n"

print(news)
```

```
# 음성으로 들려주는 곳
from gtts import gTTS
import playsound

tts=gTTS(text=news,lang='ko') # 문자열 news를 위한 한국어 음성 합성
tts.save("news_Son.mp3")
playsound.playsound("news_Son.mp3",True)
```

인공지능 개발에 많이 쓰는 라이브러리



구분		공식 사이트	튜토리얼 문서
언어	파이썬 (Python)	https://www.python.org	• The Python Tutorial: https://docs.python.org/3/tutorial/
라이브러리 관리	파이파이 (Pypi)	https://pypi.org	• Installing packages: https://packaging.python.org/tutorials/installing-packages • Packaging Python projects: https://packaging.python.org/tutorials/packaging-projects
라이브러리	넘파이 (Numpy)	https://numpy.org	• Numpy Tutorial: https://numpy.org/devdocs/user/quickstart.html
	맷플롯립 (Matplotlib)	https://matplotlib.org	• User's guide: https://matplotlib.org/users/index.html
	사이킷 런 (Scikit-learn)	https://scikit-learn.org	• User guide: https://scikit-learn.org/stable/user_guide.html
	텐서플로 (TensorFlow)	https://www.tensorflow.org	• Tensorflow Tutorials: https://www.tensorflow.org/tutorials
	케라스 (Keras)	https://keras.io	• Keras documentation: https://keras.io
	파이토치 (PyTorch)	https://pytorch.org	• Welcome to Pytorch tutorials: https://pytorch.org/tutorials

Reference



- 모두의 인공지능 with 파이썬, 이영호, 길벗
- 처음 만나는 인공지능, 김대수, 생능출판
- 인공지능, 이건명, 생능출판
- 처음 배우는 인공지능, 다다 사토시, 송교석, 한빛미디어
- <https://www.python.org/>
- <https://docs.python.org>
- <https://www.interviewbit.com/blog/best-free-python-course/>
- <https://www.techrepublic.com/article/best-ide-software/>
- <https://colab.research.google.com/?hl=ko>
- <https://replit.com/languages/python3>
- <https://www.programiz.com/python-programming/online-compiler/>
- https://www.tutorialspoint.com/execute_python_online.php
- <https://www.online-python.com/>
- <https://docs.python.org/ko/3/library/idle.html>
- <https://www.anaconda.com/>