

내 의식의 흐름대로 가는

Python

1강

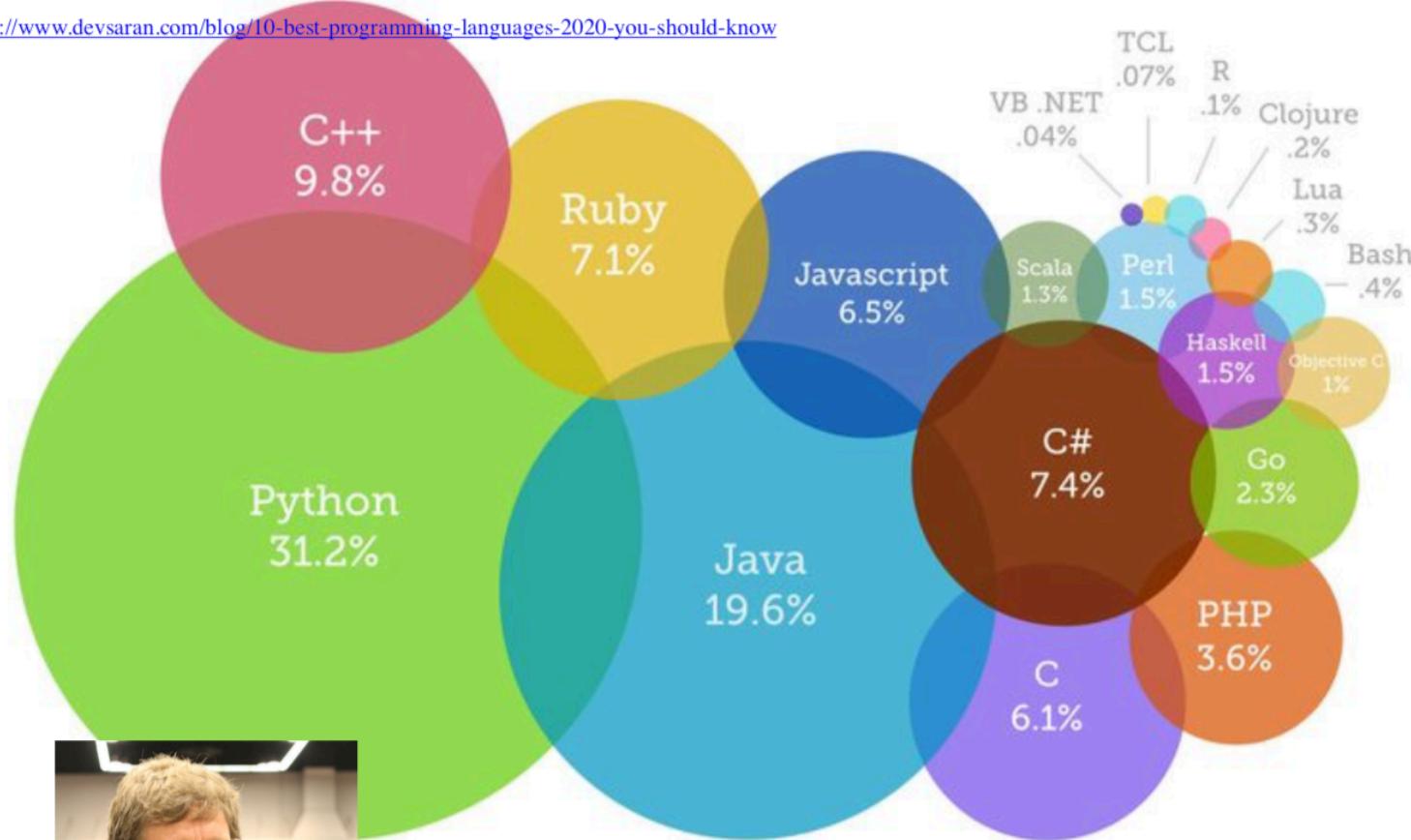
어떤 지구인 (2022)





프로그래밍 입문
조인석- 한빛 아카데미

<https://www.devsaran.com/blog/10-best-programming-languages-2020-you-should-know>



Guido van Rossum



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

The screenshot shows the Python.org website with a dark blue header. The Python logo is on the left. A navigation bar at the top includes links for Python, PSF, Docs, PyPI, Jobs, and other categories. Below the header, a large "python™" logo is centered. A "Download" button is highlighted in yellow. To its right is a search bar with a magnifying glass icon and a "Go" button. A "Donate" button is also visible. The main content area features a sidebar titled "Download" with links for "All releases", "Source code", "Windows", "macOS", "Other Platforms", "License", and "Alternative Implementations". To the right of the sidebar, a section titled "Download for macOS" shows a button for "Python 3.10.6". Below it, text states: "Not the OS you are looking for? Python can be used on many operating systems and environments." and a link to "View the full list of downloads". The background features a stylized illustration of a hot air balloon.

python™

Download

All releases

Source code

Windows

macOS

Other Platforms

License

Alternative Implementations

Download for macOS

Python 3.10.6

Not the OS you are looking for? Python can be used on many operating systems and environments.

[View the full list of downloads.](#)

Active Python Releases

The screenshot shows a web browser window with the title "Python Releases for Windows". The URL in the address bar is "python.org/downloads/windows/". The page content includes a navigation bar with links for About, Downloads, Documentation, Community, Success Stories, and News. Below the navigation bar, the breadcrumb trail shows "Python > Downloads > Windows". The main section is titled "Python Releases for Windows" and contains two lists of releases: "Stable Releases" and "Pre-releases".

Python Releases for Windows

- [Latest Python 3 Release - Python 3.10.6](#)
- [Latest Python 2 Release - Python 2.7.18](#)

Stable Releases

- [Python 3.10.6 - Aug. 2, 2022](#)

Note that Python 3.10.6 cannot be used on Windows 7 or earlier.

 - Download [Windows embeddable package \(32-bit\)](#)
 - Download [Windows embeddable package \(64-bit\)](#)
 - Download [Windows help file](#)
 - Download [Windows installer \(32-bit\)](#)
 - Download [Windows installer \(64-bit\)](#)
- [Python 3.10.5 - June 6, 2022](#)

Note that Python 3.10.5 cannot be used on Windows 7 or earlier.

 - Download [Windows embeddable package \(32-bit\)](#)
 - Download [Windows embeddable package \(64-bit\)](#)

Pre-releases

- [Python 3.11.0rc1 - Aug. 8, 2022](#)
 - Download [Windows embeddable package \(32-bit\)](#)
 - Download [Windows embeddable package \(64-bit\)](#)
 - Download [Windows embeddable package \(ARM64\)](#)
 - Download [Windows installer \(32-bit\)](#)
 - Download [Windows installer \(64-bit\)](#)
 - Download [Windows installer \(ARM64\)](#)
- [Python 3.11.0b5 - July 26, 2022](#)
 - Download [Windows embeddable package \(32-bit\)](#)
 - Download [Windows embeddable package \(64-bit\)](#)
 - Download [Windows embeddable package \(ARM64\)](#)
 - Download [Windows installer \(32-bit\)](#)

Python Releases for Windows | x python 설치 - Google Search x +

← → C ↻ 🔒 google.co.kr/search?q=python+설치&hl=en&sxsrf=ALiCzsYSqjvkLUu02hTPZK0PADPmoIU9UA:1662198469

Google python 설치 x | 🔍

All Images Videos News Maps More Tools

About 82,900 results (0.31 seconds)

dojang.io, 강좌, 프로그래밍, 파이썬 코딩 도장
2.0 파이썬 설치하기 - 코딩 도장

 다운로드한 **python-3.6.0.exe** 파일을 실행하면 설치 화면이 표시됩니다. Add Python 3.6 to PATH에 체크하고 **Install Now**를 클릭합니다. ▽...
코딩 도장 · 길벗IT · Nov 1, 2018
8:50

opentutorials.org › course ▾
Python 설치 - 생활코딩

 8.21 설치 완료 -downloading python installer -installing python (version 3.9.6) -setting en path -execute cmd > input python.
오픈튜토리얼스 · 생활코딩 · May 15, 2015
7:04

www.youtube.com › watch
파이썬 설치 방법: 윈도우에 파이썬 설치하기! - YouTube

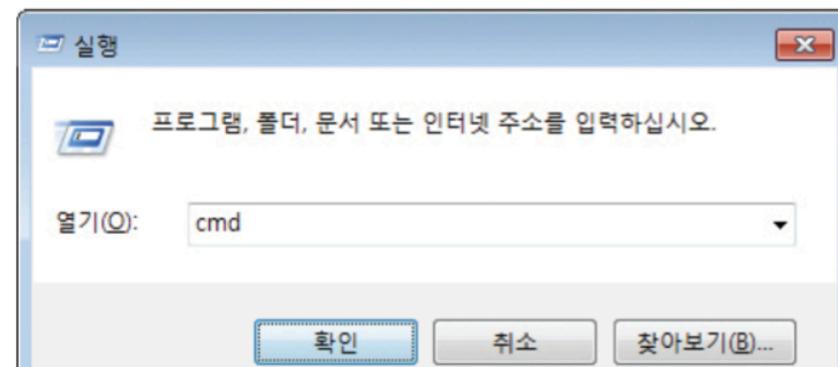
 너무나도 핫한 언어 파이썬!파이썬 프로그래밍을 시작하려면 파이썬을 컴퓨터에 설치해야겠죠!윈도우 운영체제에서 파이썬 설치 방법을 알아봅시...
YouTube · 코드잇 · Mar 30, 2020
6:36

dojang.io, 강좌, 프로그래밍, 파이썬 코딩 도장
44.3 파이썬 패키지 인덱스에서 패키지 설치하기 - 코딩 도장

 44.3.1 pip 설치하기. pip는 파이썬 패키지 인덱스의 패키지 관리 명령어이며 Windows용 파이썬에는 기본으로 내장되어 있습니다. 리눅스와...
코딩 도장 · 길벗 IT 전문서(업로드 중단, 구경정) · Nov 2, 2018
3:54

Lib	2017-09-04 오후...	파일 폴더
libs	2017-09-04 오후...	파일 폴더
Scripts	2017-09-04 오후...	파일 폴더
tcl	2017-09-04 오후...	파일 폴더
Tools	2017-09-04 오후...	파일 폴더
LICENSE	2017-07-08 오전...	텍스트 문서
NEWS	2017-07-08 오전...	30KB
python	2017-07-08 오전...	
python3.dll	2017-07-08 오전...	
python36.dll	2017-07-08 오전...	
pythonw	2017-07-08 오전...	
vcruntime140.dll	2016-07-08 오전...	

Windows + R을 눌러 [실행]을 열고

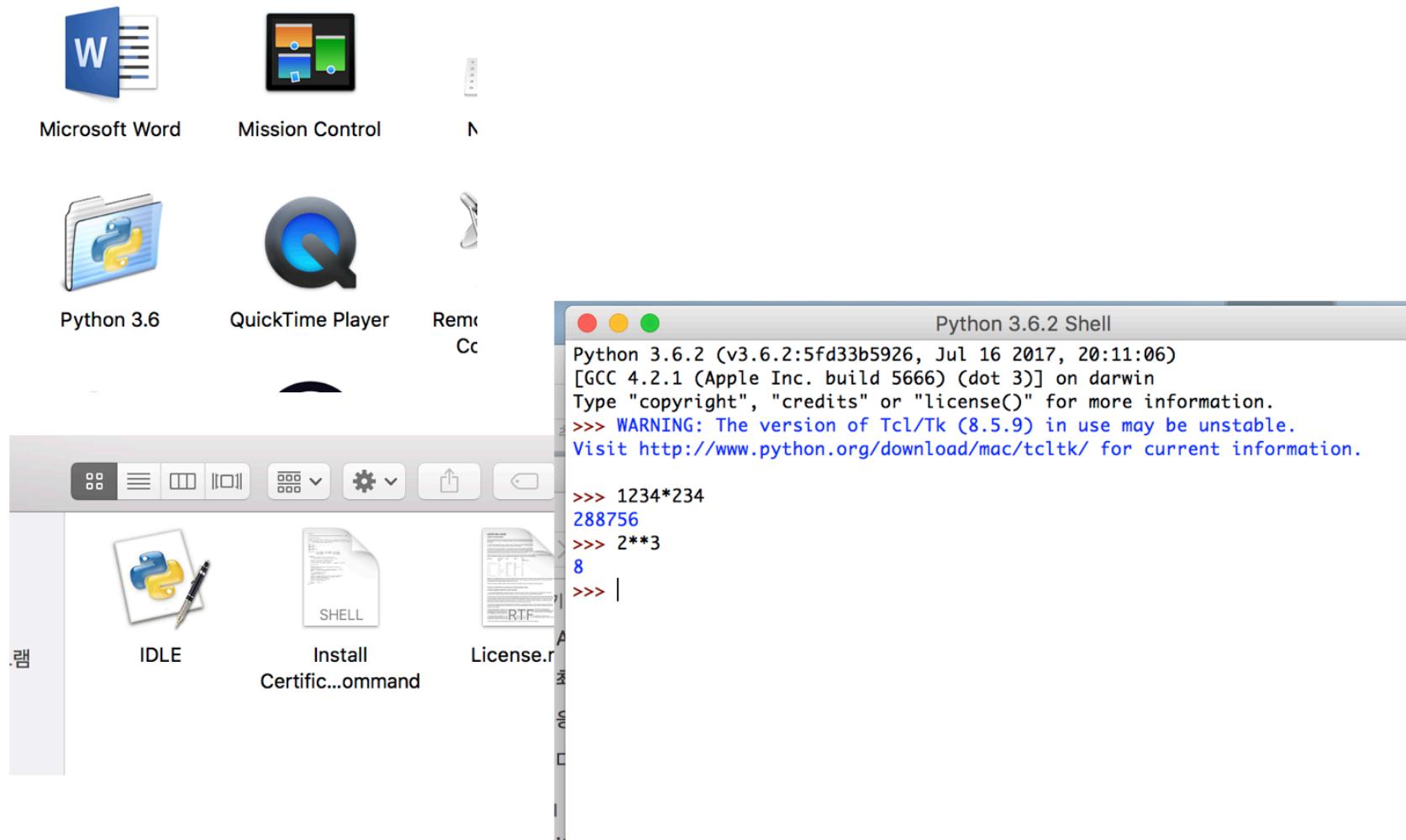
A screenshot of a Windows Command Prompt window titled 'C:\Windows\system32\cmd.exe - python'. The window displays the following text:

```
Microsoft Windows [Version 6.1.7601]
Copyright <c> 2009 Microsoft Corporation. All rights reserved.

C:\Users\John>python
Python 3.6.2 (v3.6.2:5fd33b5, Jul  8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)]
on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```



Integrated Development Environment (IDLE): 통합개발환경



C:\>python -m pip install --upgrade pip

C:\>pip install scipy

C:\>pip list

C:\>pip install numpy

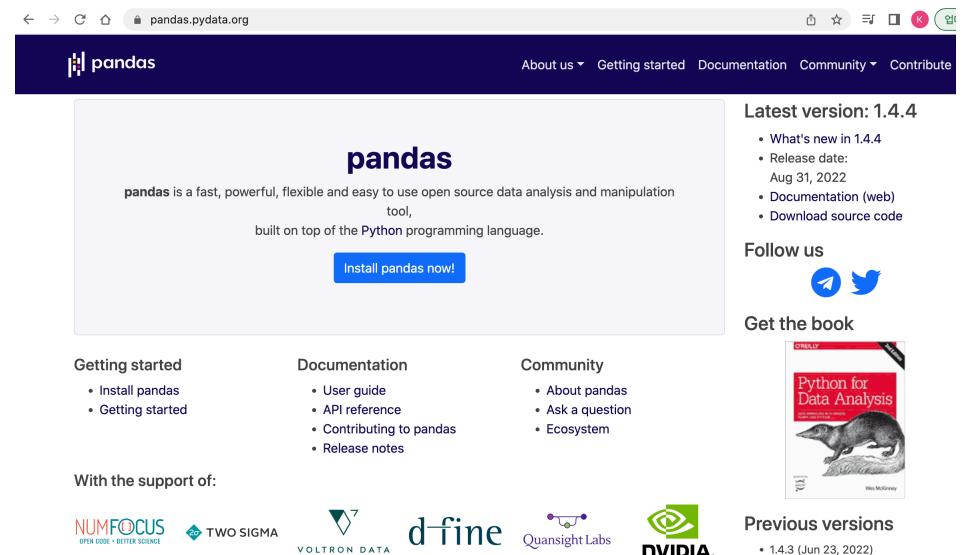
C:\>pip install matplotlib

C:\>pip install tensorflow==2.0.0

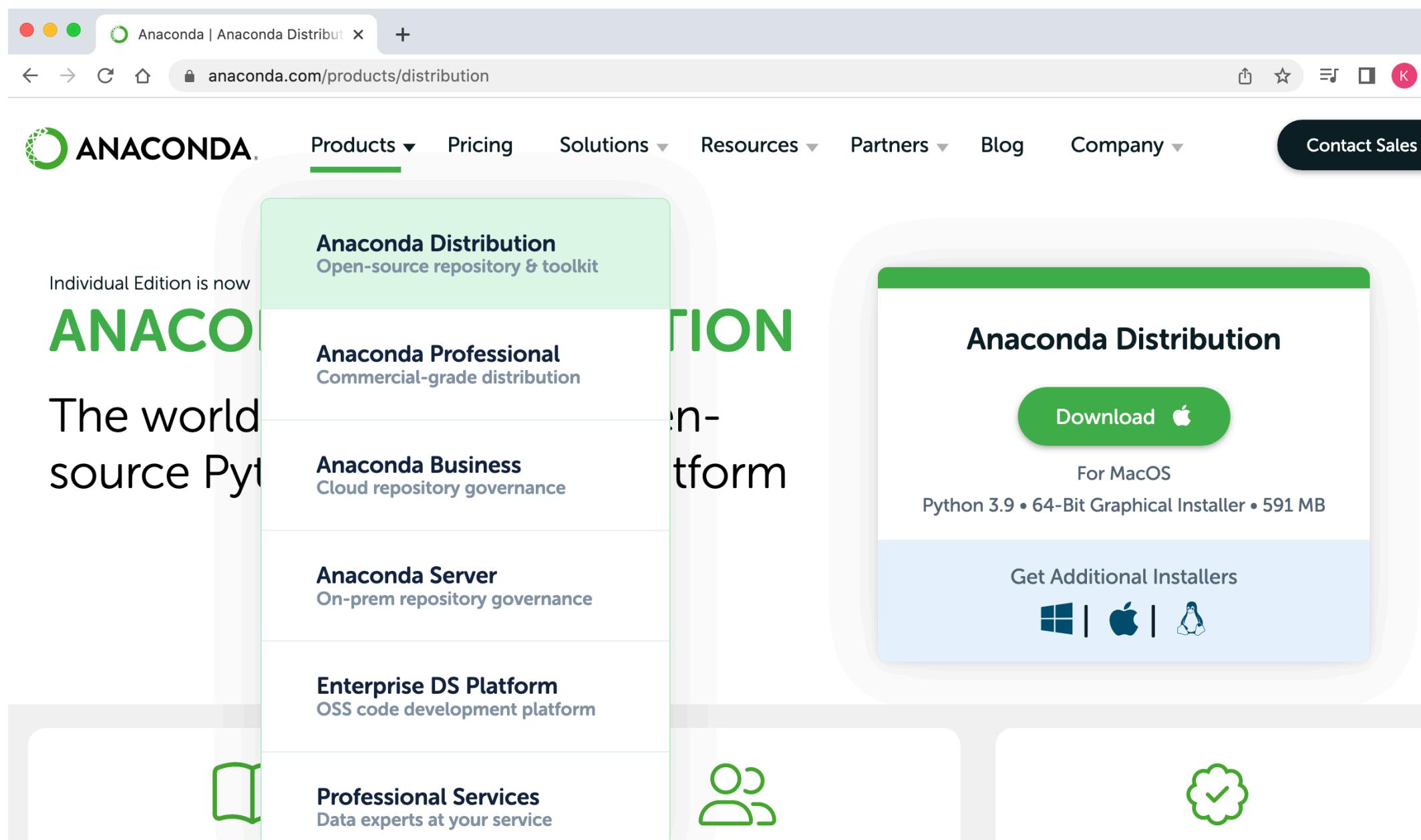
C:\>pip install keras

C:\>pip install jupyter

C:\>jupyter notebook



Previous versions
• 1.4.3 (Jun 23, 2022)



Jupyter notebook

```
kihyunyun — zsh — 80x24
Last login: Sat Sep  3 19:01:13 on ttys000
(base) kihyunyun@yungihyeon-ui-MacBookPro ~ % jupyter notebook
```

```
kihyunyun — jupyter-notebook — 80x24
(base) kihyunyun@yungihyeon-ui-MacBookPro ~ % jupyter notebook
[I 2022-09-03 19:03:43.534 LabApp] JupyterLab extension loaded from
nyun/opt/anaconda3/lib/python3.9/site-packages/jupyterlab
[I 2022-09-03 19:03:43.534 LabApp] JupyterLab application directory
hyunyun/opt/anaconda3/share/jupyter/lab
[I 19:03:43.540 NotebookApp] Serving notebooks from local directory
unyun
[I 19:03:43.540 NotebookApp] Jupyter Notebook 6.4.8 is running at:
[I 19:03:43.540 NotebookApp] http://localhost:8888/?token=76e723eea
81c5369034a4fb29c2db5318d7
[I 19:03:43.540 NotebookApp] or http://127.0.0.1:8888/?token=76e72
e1f881c5369034a4fb29c2db5318d7
[I 19:03:43.540 NotebookApp] Use Control-C to stop this server and
kernels (twice to skip confirmation).
[C 19:03:43.547 NotebookApp]
```

To access the notebook, open this file in a browser:
file:///Users/kihyunyun/Library/Jupyter/runtime/nbserver-53
Or copy and paste one of these URLs:

http://localhost:8888/?token=76e723eeadd791e5e1e1f881c53690
318d7
or http://127.0.0.1:8888/?token=76e723eeadd791e5e1e1f881c53690
318d7

The screenshot shows a Jupyter Notebook interface running in a web browser. The title bar says "Home Page - Select or create a". The address bar shows "localhost:8888/tree". The main content area displays a file tree with the root folder containing "Applications", "Desktop", "Documents", "Downloads", "Movies", "Music", "opt", "Pictures", and "Public". Two files are listed in the "Running" tab: "Untitled.ipynb" and "Untitled1.ipynb". A sidebar on the right contains a "Notebook" dropdown set to "Python 3 (ipykernel)", an "Upload" button, and a "New" dropdown. Other options in the sidebar include "Text File", "Folder", and "Terminal". The status bar at the bottom indicates "Running seconds ago" and "72 B".

Name	Created	Type
/	an hour ago	Folder
Applications	a month ago	Folder
Desktop	2 years ago	Folder
Documents	a month ago	Folder
Downloads	2 years ago	Folder
Movies	a month ago	Folder
Music	2 years ago	Folder
opt	a year ago	Folder
Pictures	2 years ago	Folder
Public	2 years ago	Folder
Untitled.ipynb	Running seconds ago	72 B
Untitled1.ipynb	Running seconds ago	72 B



localhost:8888/notebooks/Untitled6.ipynb?kernel_name=python3

jupyter Untitled6 Last Checkpoint: a minute ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Trusted



In [2]: `1+2+3`

Out[2]: 6

In [3]: `a=3
b=7
print(a+b)
print(a*b)`

10

21

A screenshot of a web browser window. The address bar shows a Google search for "colab" at google.co.kr/search?q=colab&bt.... The search results page is displayed, featuring the Google logo and a search bar with the query "colab". Below the search bar are filters for "All", "Images", "News", "Maps", "Videos", and "More". A "Tools" button is also present. The main content area shows search results for "Google Colab", "Colaboratory", "Colab", "Filter notebooks", and "Cells".

About 64,000,000 results (0.44 seconds)

<https://colab.research.google.com> ::

Google Colab

With **Colab** you can import an image dataset, train an image classifier on it, and evaluate the model, all in just a few lines of code. **Colab** notebooks execute ...

Google Colab

Colaboratory(줄여서 'Colab'이라고 함)을 통해 브라우저 내에서 ...

Colab

Sign in. Loading... Loading...

Filter notebooks

[Overview of Colaboratory](#) - [Markdown Guide](#) - [Forms](#) - ...

Cells

Code cells. Below is a code cell. Once the toolbar button ...

clik하면 다음 페이지로 가고 누르고 있으면 방문 기록이 나타납니다.

Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

- New notebook
- Open notebook ⌘/Ctrl+O
- Upload notebook
- Rename
- Save a copy in Drive
- Save a copy as a GitHub Gist
- Save a copy in GitHub
- Save ⌘/Ctrl+S
- Revision history
- Download
- Print ⌘/Ctrl+P

Code + Text Copy to Drive

Welcome to Colab!

If you're already familiar with Colab, check out this video to learn about interacting with notebooks and the command palette.



What is Colab?

Colab, or "Colaboratory", allows you to write and execute Python in your browser.

<https://colab.research.google.com/>

The screenshot shows the Google Colab interface. At the top, there are two tabs: "Untitled2.ipynb - Colaboratory" and "Untitled3.ipynb - Colaboratory". The "Untitled3.ipynb" tab is active, indicated by a blue border around its title bar. Below the tabs is a toolbar with icons for back, forward, refresh, and home, followed by the URL "colab.research.google.com/drive/106jQc7mJpS8g3Qwx9fOLHz7-M_gSaWbm". The main workspace is titled "Untitled3.ipynb" and shows a menu bar with File, Edit, View, Insert, Runtime, Tools, Help, and "All changes saved". On the left, there's a sidebar with icons for file navigation and a code/text switcher. A code cell is visible, containing the following Python code:

```
## jupyter와 매우 비슷한 colab
a=2
b=3
print(a+b)
```

The code cell has a play button icon and a count of 5. The output area below the code cell is currently empty.

The screenshot shows a Google Colab interface with the following details:

- Title Bar:** Untitled6.ipynb - Colaboratory
- Address Bar:** colab.research.google.com/drive/1hxajXxPoUe6TIXVf6rGxllgnv...
- File Menu:** File Edit View Insert Runtime Tools Help
- Status Bar:** All changes saved
- Runtime Selection:** RAM Disk (green checkmark)
- Code Cell:** print("주성치 좋아")
Output: 주성치 좋아
- Timeline:** 0s completed at 7:12 PM



Untitled2.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
▶ from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

from google.colab import drive
drive.mount('/content/drive')

import numpy as np
import tensorflow as tf

np.random.seed(3)
tf.random.set_seed(3)

Data_set = np.loadtxt("/content/drive/My Drive/ThoracicSurgery.csv", delimiter=',')

X = Data_set[:,0:17]
Y = Data_set[:,17]

model = Sequential()
model.add(Dense(30, input_dim=17, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

model.compile(loss='mean_squared_error', optimizer = 'adam', metrics=['accuracy'])
model.fit(X, Y, epochs=100, batch_size = 10)
```

Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4

Enter your authorization code:

Untitled2.ipynb ★

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text ✓

```
▶ from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

from google.colab import drive
drive.mount('/content/drive')

import numpy as np
import tensorflow as tf

np.random.seed(3)
tf.random.set_seed(3)

Data_set = np.loadtxt("/content/drive/My Drive/ThoracicSurgery.csv", delimiter=",")

X = Data_set[:,0:17]
Y = Data_set[:,17]

model = Sequential()
model.add(Dense(30, input_dim=17, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

model.compile(loss='mean_squared_error', optimizer = 'adam', metrics=['accuracy'])
model.fit(X, Y, epochs=100, batch_size = 10)
```

Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect_uri=https://colab.research.google.com/drive/&response_type=code&scope=https://www.googleapis.com/auth/drive

Enter your authorization code:
.....

Mounted at /content/drive
Epoch 1/100
47/47 [=====] - 0s 953us/step - loss: 0.1485 - accuracy: 0.8426
Epoch 2/100
47/47 [=====] - 0s 822us/step - loss: 0.1402 - accuracy: 0.8511
Epoch 3/100
47/47 [=====] - 0s 926us/step - loss: 0.1433 - accuracy: 0.8489
Epoch 4/100
47/47 [=====] - 0s 892us/step - loss: 0.1395 - accuracy: 0.8511



Untitled2.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
Epoch 88/100
47/47 [=====] - 0s 886us/step - loss: 0.1184 - accuracy: 0.8511
Epoch 89/100
47/47 [=====] - 0s 817us/step - loss: 0.1236 - accuracy: 0.8574
Epoch 90/100
47/47 [=====] - 0s 904us/step - loss: 0.1315 - accuracy: 0.8532
Epoch 91/100
47/47 [=====] - 0s 972us/step - loss: 0.1352 - accuracy: 0.8489
Epoch 92/100
47/47 [=====] - 0s 892us/step - loss: 0.1264 - accuracy: 0.8574
Epoch 93/100
47/47 [=====] - 0s 819us/step - loss: 0.1180 - accuracy: 0.8574
Epoch 94/100
47/47 [=====] - 0s 1ms/step - loss: 0.1228 - accuracy: 0.8596
Epoch 95/100
47/47 [=====] - 0s 878us/step - loss: 0.1216 - accuracy: 0.8596
Epoch 96/100
47/47 [=====] - 0s 855us/step - loss: 0.1290 - accuracy: 0.8532
Epoch 97/100
47/47 [=====] - 0s 799us/step - loss: 0.1221 - accuracy: 0.8553
Epoch 98/100
47/47 [=====] - 0s 859us/step - loss: 0.1318 - accuracy: 0.8468
Epoch 99/100
47/47 [=====] - 0s 865us/step - loss: 0.1332 - accuracy: 0.8574
Epoch 100/100
47/47 [=====] - 0s 861us/step - loss: 0.1185 - accuracy: 0.8660
<tensorflow.python.keras.callbacks.History at 0x7ff513b266d8>
```



Untitled2.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
Epoch 88/100
47/47 [=====] - 0s 886us/step - loss: 0.1184 - accuracy: 0.8511
↳ Epoch 89/100
47/47 [=====] - 0s 817us/step - loss: 0.1236 - accuracy: 0.8574
Epoch 90/100
47/47 [=====] - 0s 904us/step - loss: 0.1315 - accuracy: 0.8532
Epoch 91/100
47/47 [=====] - 0s 972us/step - loss: 0.1352 - accuracy: 0.8489
Epoch 92/100
47/47 [=====] - 0s 892us/step - loss: 0.1264 - accuracy: 0.8574
Epoch 93/100
47/47 [=====] - 0s 819us/step - loss: 0.1180 - accuracy: 0.8574
Epoch 94/100
47/47 [=====] - 0s 1ms/step - loss: 0.1228 - accuracy: 0.8596
Epoch 95/100
47/47 [=====] - 0s 878us/step - loss: 0.1216 - accuracy: 0.8596
Epoch 96/100
47/47 [=====] - 0s 855us/step - loss: 0.1290 - accuracy: 0.8532
Epoch 97/100
47/47 [=====] - 0s 799us/step - loss: 0.1221 - accuracy: 0.8553
Epoch 98/100
47/47 [=====] - 0s 859us/step - loss: 0.1318 - accuracy: 0.8468
Epoch 99/100
47/47 [=====] - 0s 865us/step - loss: 0.1332 - accuracy: 0.8574
Epoch 100/100
47/47 [=====] - 0s 861us/step - loss: 0.1185 - accuracy: 0.8660
<tensorflow.python.keras.callbacks.History at 0x7ff513b266d8>
```



kihyunyun — jupyter-notebook — 80x24

```
Last login: Mon Aug 24 17:44:13 on ttys000
[yungihyeon-ui-MacBook:~ kihyunyun$ jupyter notebook
[I 14:11:07.154 NotebookApp] Serving notebooks from local directory: /Us
unyun
[I 14:11:07.154 NotebookApp] The Jupyter Notebook is running at:
[I 14:11:07.154 NotebookApp] http://localhost:8888/?token=7a997d0dce53ea
a0a580bf5eb4e05abf96c23a6b
[I 14:11:07.154 NotebookApp] Use Control-C to stop this server and shut
kernels (twice to skip confirmation).
[C 14:11:07.155 NotebookApp]
```

Copy/paste this URL into your browser when you connect for the first
time to login with a token:

<http://localhost:8888/?token=7a997d0dce53ea9fc38fe2a0a580bf5eb4e>

23a6b

```
[I 14:11:07.964 NotebookApp] Accepting one-time-token-authenticated conn
rom ::1
```



Quit

Logout

Files Running Clusters

Select items to perform actions on them.

0

Name

Upload

New



Applications

Desktop

Documents

Downloads

Movies

2 days ago

Music

2 years ago

Pictures

2 years ago

Notebook:

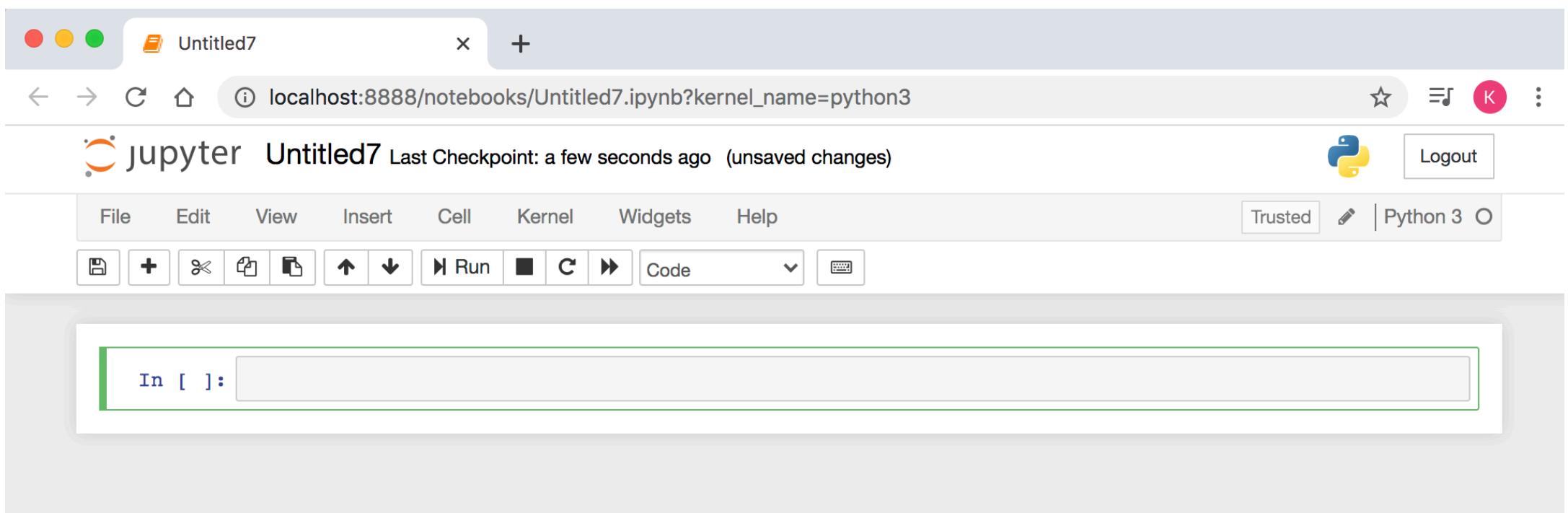
Python 3

Other:

Text File

Folder

Terminal



In [2]:

```
a=1  
b=2  
c=1.5  
d="Hello World"
```

```
print(a+b)  
print(b+c)  
print(d)
```

3

3.5

Hello World

In [2]:

```
a=1  
b=2  
c=1.5  
d="Hello World"
```

```
print(a+b)  
print(b+c)  
print(d)
```

3
3.5
Hello World

In [6]:

```
print(type(a))  
print(type(b))  
print(type(c))  
print(type(d))
```

<class 'int'>
<class 'int'>
<class 'float'>
<class 'str'>

In [7]:

```
type(d)
```

Out[7]: str

예약어

Reserved words

Keyword	Keyword	Keyword
and	exec	not
assert	finally	or
break	for	pass
class	from	print
continue	global	raise
def	if	return
del	import	try
elif	in	while
else	is	with
except	lambda	yield

```
In [12]: import numpy as np
```

```
x=3
```

```
y=2
```

```
z=np.exp(x+y)
```

```
print(z)
```

```
148.4131591025766
```

▶ In [*]: `Your_1st_name = input ('이름이 뭔가요? ')
print("당신의 이름은", Your_1st_name)`

이름이 뭔가요?

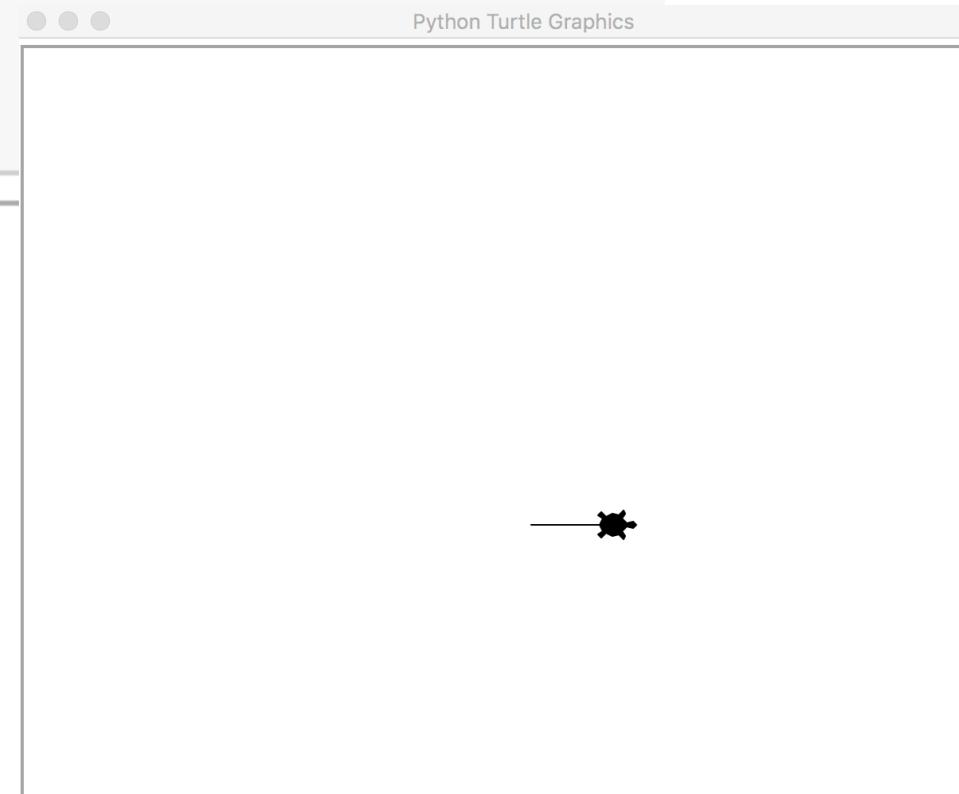
▶ In [17]: `Your_1st_name = input ('이름이 뭔가요? ')
print("당신의 이름은", Your_1st_name)`

이름이 뭔가요?
당신의 이름은 주성치

```
In [45]: import turtle as ttl
```

```
ttl.shape("turtle")
```

```
ttl.forward(50)
```



In [15]:

x=2

X=3

print(x)

print(X) # 대소문자 구분

2

3

문자열

▶ In [23]: `Lyrics1 = "쥐가 한마리, 두마리"`

`Lyrics2 = '세마리'`

`#Lyrics3= "네마리,
#다섯마리"`

`print(Lyrics1)`

`print(Lyrics2)`

`#print(Lyrics3)`

쥐가 한마리, 두마리
세마리

▶ In [26]: `Lyrics1 = "쥐가 한마리, 두마리"`

`Lyrics2 = '세마리'`

`Lyrics3= '''네마리,
다섯마리'''`

`print(Lyrics1)
print(Lyrics2)`

`print(Lyrics3)`

쥐가 한마리, 두마리
세마리
네마리,
다섯마리

```
▶ In [29]: Lyrics1 ="쥐가 한마리, "
Lyrics2 ='두마리'
Lyrics3 = Lyrics1 + Lyrics2
print(Lyrics1 + Lyrics2)
```

쥐가 한마리, 두마리

```
▶ In [30]: Lyrics1 ="주가 "
Lyrics2 ='매우 '
Lyrics3 = '많아'
Lyrics4 = Lyrics1 + Lyrics2*3 + Lyrics3
print(Lyrics4)
```

주가 매우 매우 매우 많아

The image displays two side-by-side screenshots of a web application interface for word2vec search, titled "word2vec.kr/search/?query=".

Screenshot 1 (Left):

- Query:** 사랑+이별
- QUERY Buttons:** +사랑/Noun, +이별/Noun
- Result:** 추억/Noun

Screenshot 2 (Right):

- Query:** 스타워즈-광선검
- QUERY Buttons:** +스타워즈/Noun, -광선검/Noun
- Result:** 스타트렉/Noun

```
In [33]: Very_long_named_king="순조선각연덕현도경인순희체성응명흠광석경계천배극융원돈휴의행소윤희화준렬대중지정홍훈철모건시태형창운"
len(Very_long_named_king)
```

```
Out[33]: 78
```

```
In [34]: Very_long_named_king="순조선각연덕현도경인순희체성응명흠광석경계천배극융원돈휴의행소윤희화준렬대중지정홍훈철모건시태형창운"
len(Very_long_named_king)
print(len(Very_long_named_king))
```

78

```
In [36]: word= "Python"  
word.upper()
```

Out[36]: 'PYTHON'

```
In [37]: word.lower()
```

Out[37]: 'python'

```
In [38]: "-".join(word)
```

Out[38]: 'P-y-t-h-o-n'

```
In [39]: "*".join(word)
```

Out[39]: 'P*y*t*h*o*n'

```
In [40]: print(word)
```

Python

숫자 타입

```
In [2]: type(100)
```

```
Out[2]: int
```

```
In [1]: type(100.0)
```

```
Out[1]: float
```

```
In [3]: type("100")
```

```
Out[3]: str
```

```
In [5]: 2*3 + 4*6
```

```
Out[5]: 30
```

```
In [6]: 234/32
```

```
Out[6]: 7.3125
```

► In [7]: `round(234/32,2)`

```
Out[7]: 7.31
```

```
In [8]: 234//32
```

```
Out[8]: 7
```

```
In [9]: 234%32
```

```
Out[9]: 10
```

```
In [10]: 100+'100'
```

```
-----  
TypeError                                 Traceback (most recent call last)  
<ipython-input-10-321071eb308c> in <module>()  
----> 1 100+'100'  
  
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

```
In [11]: 100 + int('100')
```

```
Out[11]: 200
```

복합대입연산자 (Assignment operator)

```
In [13]: x=1  
print(x)  
  
x=x+1  
print(x)  
  
x=x+1  
print(x)  
  
x=x+1  
print(x)|
```

1
2
3
4

```
► In [14]: x=1  
print(x)  
  
x+=1  
print(x)  
  
x+=1  
print(x)  
  
x+=1  
print(x)
```

1
2
3
4

In [15]:

```
x=1  
print(x)  
  
x=x*2  
print(x)  
  
x=x*2  
print(x)  
  
x=x*2  
print(x)
```

1
2
4
8

► In [16]:

```
x=1  
print(x)  
  
x*=2  
print(x)  
  
x*=2  
print(x)  
  
x*=2  
print(x)
```

1
2
4
8

논리타입(Boolean type)

```
In [32]: int(True)
```

```
Out[32]: 1
```

```
In [33]: int(False)
```

```
Out[33]: 0
```

```
In [34]: bool(1)
```

```
Out[34]: True
```

```
In [35]: bool(0)
```

```
Out[35]: False
```

In [20]: `100 < 200`

Out[20]: `True`

In [21]: `100 > 200`

Out[21]: `False`

In [22]: `100 == 200`

Out[22]: `False`

In [23]: `100 != 200`

Out[23]: `True`

In [26]: `a= 10`
`a is 10`

Out[26]: `True`

In [27]: `a is not 10`

Out[27]: `False`

```
In [28]: 100> 200 or 200 > 5
```

Out[28]: True

```
In [29]: 100> 200 and 200 > 5
```

Out[29]: False

```
In [30]: not 100> 200
```

Out[30]: True

종류	설명
>	왼쪽 값이 크다
<	왼쪽 값이 작다
\geq	왼쪽 값이 크거나 같다
\leq	왼쪽 값이 작거나 같다
$=$	값이 같다
\neq	값이 같지 않다
is	값과 타입이 같다
is not	값과 타입이 같지 않다

<https://ooyoung.tistory.com/29dptj>에서 가지고 왔어요.

종류

and

or

not

```
>>> print (True and True)  
True
```

```
>>> print (True and False)  
True
```

```
▶ print(False and True)  
False
```

```
>>> print (False and False)  
False
```

```
>>> print(not True)  
False
```

```
>>> print(not False)  
True
```

조건분기문

```
In [*]: Answer = input ("주성치 영화를 본 적이 있나? (예/아니오):")  
  
if Answer == '예':  
    print("진정한 영화를 보셨군요.")
```

주성치 영화를 본 적이 있나? (예/아니오):

```
▶ In [5]: Answer = input ("주성치 영화를 본 적이 있나? (예/아니오):")  
  
if Answer == '예':  
    print("진정한 영화를 보셨군요.")
```

주성치 영화를 본 적이 있나? (예/아니오): 예
진정한 영화를 보셨군요.

```
▶ In [46]: Answer = input ("주성치 영화를 본 적이 있나? (예/아니오):")  
  
if Answer == '예':  
    print("진정한 영화를 보셨군요.")
```

주성치 영화를 본 적이 있나? (예/아니오): 아니오

```
In [41]: Answer = input ("주성치 영화를 본 적이 있나? (예/아니오) : ")

if Answer == '예':
    print("진정한 영화를 보셨군요.")
else:
    print("아쉽군요.")
```

주성치 영화를 본 적이 있나? (예/아니오) : 봤어요.
아쉽군요.

```
▶ In [44]: Answer = input ("주성치 영화를 본 적이 있나? (예/아니오) : ")

if Answer == '예':
    print("진정한 영화를 보셨군요.")
elif Answer == '아니오':
    print("아쉽군요.")
else:
    print("'예' 혹은 '아니오'로만 대답하세요.")
```

주성치 영화를 본 적이 있나? (예/아니오) : 예
진정한 영화를 보셨군요.

```
# 조건문 1
```

```
hungry = True
```

```
if hungry:  
    print("배가 고프다.")
```

배가 고프다.

```
# 조건문 2
```

```
hungry = False
```

```
if hungry:  
    print("배가 고프다.")
```

```
# 조건문 3 : else 추가
```

```
hungry = False
```

```
if hungry:  
    print("배가 고프다.")  
else:  
    print("배가 안고파요")
```

배가 안고파요

: # 질문1: 마음에 드는 물건이 있는가?

```
answer = input('마음에 드는 물건을 찾았는가? (예/아니오) : ')
```

조건 분기문

```
if answer == '예':
    print('좋구나!')
elif answer == '아니오':
    print('더 열심히 찾으라.')
else:
    print("예' 혹은 '아니오'라고 대답하라고 했다.")
```

마음에 드는 물건을 찾았는가? (예/아니오) :글쎄요

예' 혹은 '아니오'라고 대답하라고 했다.

```
n [4]: # 질문1: 마음에 드는 물건이 있는가?
```

```
answer = input('마음에 드는 물건을 찾았는가? (예/아니오): ')
```

```
# 조건 분기문1
```

```
if answer == '예':
```

```
    print('좋구나!')
```

```
    # 질문2: 가격입력
```

```
    price = input("얼마인가? ")
```

```
# 조건 분기문2
```

```
if price <= 1000:
```

```
    print("다음에 사줄게.")
```

```
else:
```

```
    print("절제가 사람을 만드는 법이다.")
```

```
elif answer == '아니오':
```

```
    print('더 열심히 찾으라.')
```

```
else:
```

```
    print("예 혹은 '아니오'라고 대답하라고 했다.")
```

```
마음에 드는 물건을 찾았는가? (예/아니오): 예
```

```
좋구나!
```

```
얼마인가? 1000
```

```
-----
```

```
TypeError
```

```
Traceback (most recent call last)
```

```
<ipython-input-4-486852ef8e36> in <module>()
```

```
    11
```

```
    12     # 조건 분기문2
```

```
---> 13     if price <= 1000:
```

```
    14         print("다음에 사줄게.")
```

```
    15     else:
```

```
TypeError: '<=' not supported between instances of 'str' and 'int'
```

```
마음에 드는 물건을 찾았는가? (예/아니오): 예
```

```
좋구나!
```

```
얼마인가? 1000
```

```
# 질문1: 마음에 드는 물건이 있는가?
```

```
answer = input('마음에 드는 물건을 찾았는가? (예/아니오) : ')
```

```
# 조건 분기문1
```

```
if answer == '예':  
    print('좋구나!')  
    # 질문2: 가격입력  
    price = input("얼마인가? ")
```

```
price= int(price) ←
```

```
# 조건 분기문2  
if price <= 1000:  
    print("다음에 사줄게.")  
else:  
    print("절제가 사람을 만드는 법이다.")
```

```
elif answer == '아니오':  
    print('더 열심히 찾으라.')  
else:  
    print("예 혹은 '아니오'라고 대답하라고 했다.")
```

마음에 드는 물건을 찾았는가? (예/아니오) : 예
좋구나!
얼마인가? 2000
절제가 사람을 만드는 법이다.

While

```
▶ In [1]: # 밥을 많이 먹는 집

repeat = True

while repeat:

    answer = input('밥 한 그릇 더 먹을래? (예/아니오) : ')

    if answer == '예':
        print("더 먹어라.")
    else:
        print("요즘 식사량이 줄었구나. 어디 아프니?")
    repeat = False
```

```
밥 한 그릇 더 먹을래? (예/아니오) : 예
더 먹어라.
밥 한 그릇 더 먹을래? (예/아니오) : 아니오
요즘 식사량이 줄었구나. 어디 아프니?
```

```
In [2]: aaa=1
```

```
while aaa <= 10:  
    aaa+=1  
    print(aaa)  
  
print("끝났다.")
```

```
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
끝났다.
```

▶ In [5]: aaa=1

```
while aaa <= 10:  
    aaa+=1  
    print(aaa)  
  
print("끝났다.")
```

11
끝났다.