내 의식의 흐름대로 가는

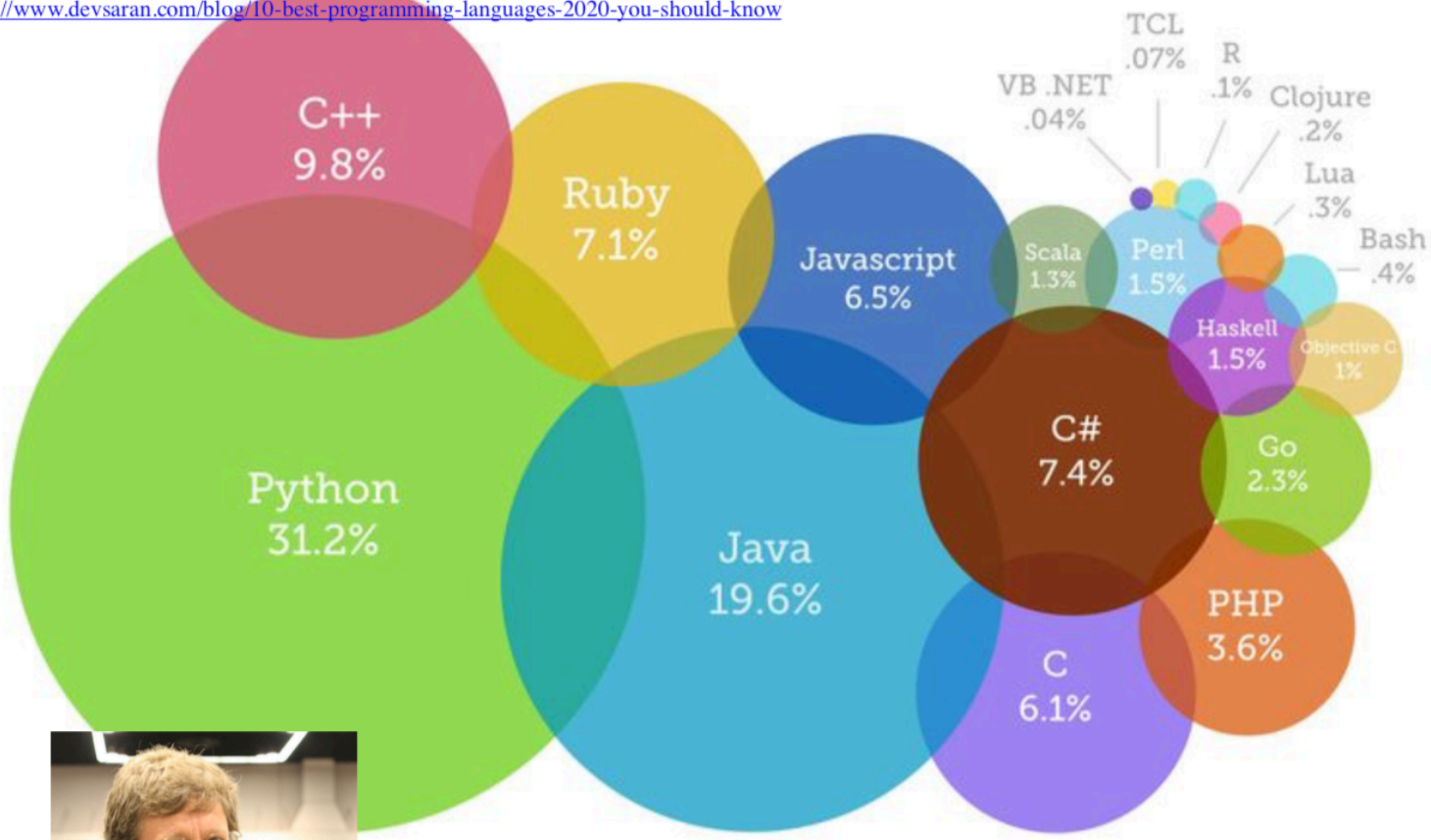
Python

1강

어떤 지구인 (2022)

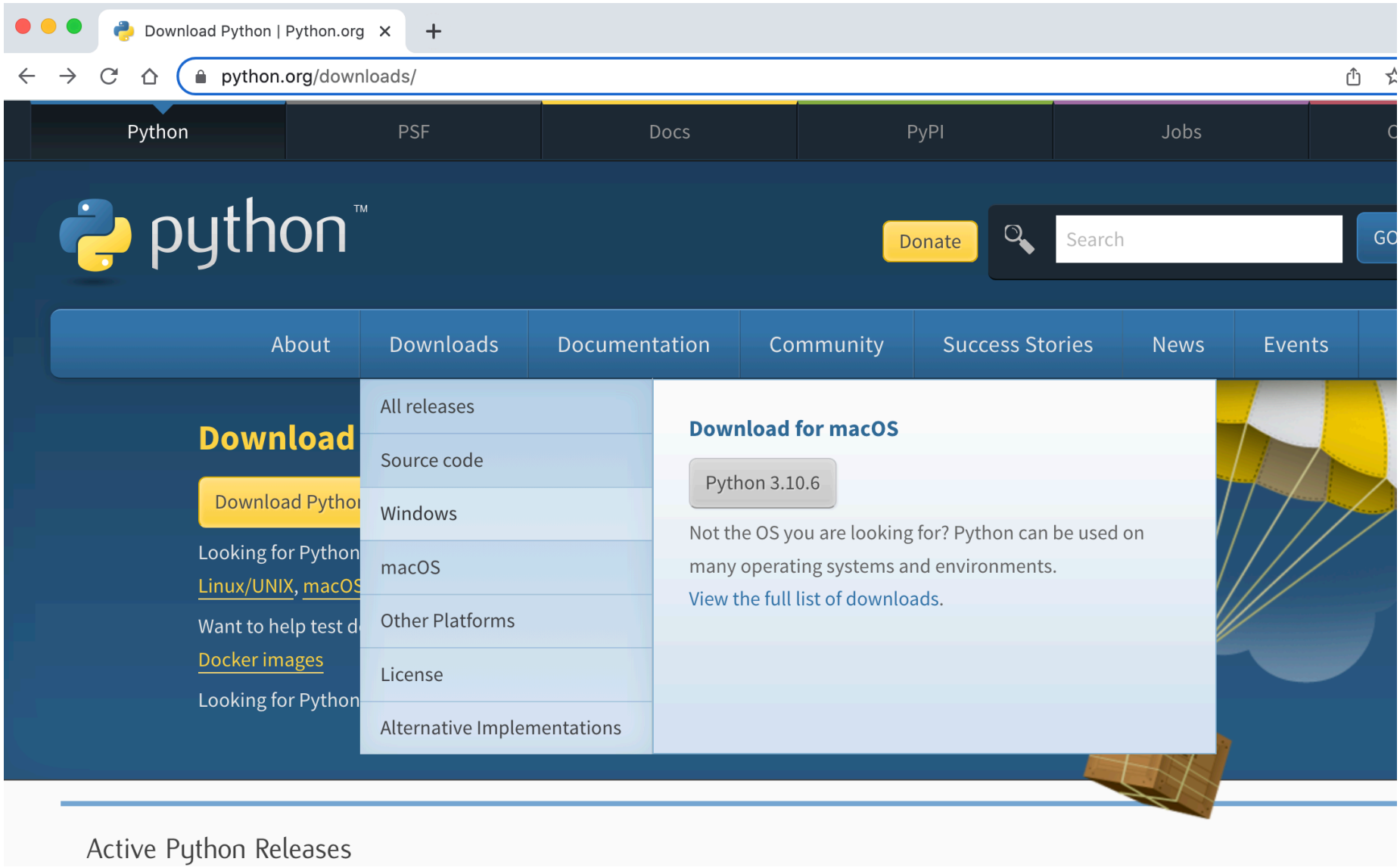
프로그래밍 입문
조인석- 한빛 아카데미

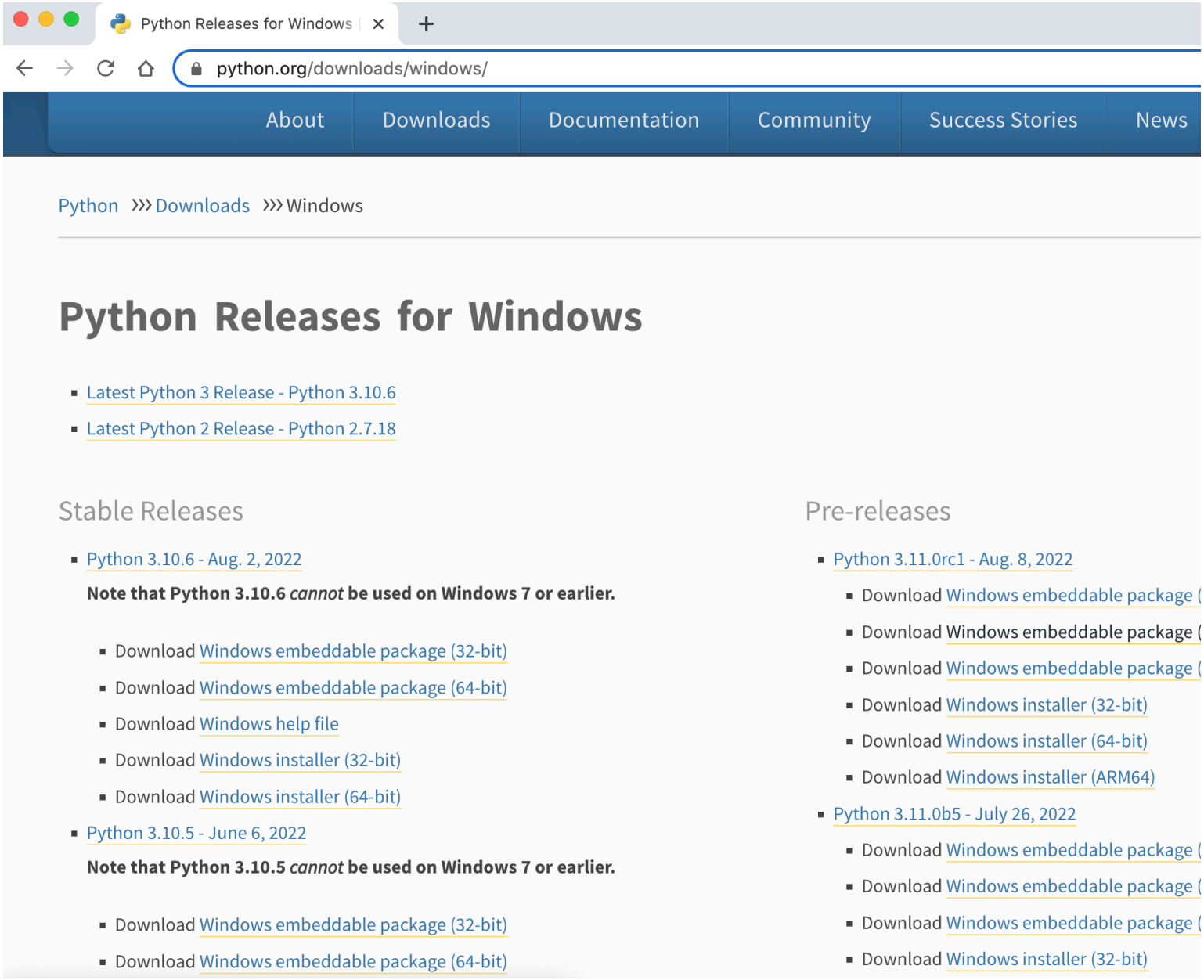
https://www.devsaran.com/blog/10-best-programming-languages-2020-you-should-know



Guido van Rossum

https://www.python.org/




Python Releases for Windows
python.org/downloads/windows/
About
Downloads
Documentation
Community
Success Stories
News
Python ››› Downloads ››› Windows
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 (
Download Windows embeddable package (
Download Windows embeddable package (
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 (
Download Windows embeddable package (
Download Windows embeddable package (
Download Windows installer (32-bit)

Python Releases for Windows | python 설치 - Google Search

google.co.kr/search?q=python+설치&hl=en&sxsrf=ALiCzsYSqjvkLUu02hTPZK0PADPmoIU9UA:166219846

Google

python 설치

All Images Videos News Maps More Tools

About 82,900 results (0.31 seconds)

dojang.io › 강좌 › 프로그래밍 › 파이썬 코딩 도장

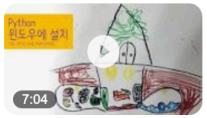
2.0 파이썬 설치하기 - 코딩 도장

다운로드한 **python**-3.6.0.exe 파일을 실행하면 **설치** 화면이 표시됩니다. Add **Python** 3.6 to PATH에 체크하고 **Install** Now를 클릭합니다. ▽…

코딩 도장 · 길벗IT · Nov 1, 2018

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

www.youtube.com › watch

파이썬 설치 방법: 윈도우에 파이썬 설치하기! - YouTube

너무나도 핫한 언어 **파이썬!파이썬** 프로그래밍을 시작하려면 **파이썬**을 컴퓨터에 설치해야겠죠!윈도우 운영체제에서 **파이썬 설치** 방법을 알아봅시…

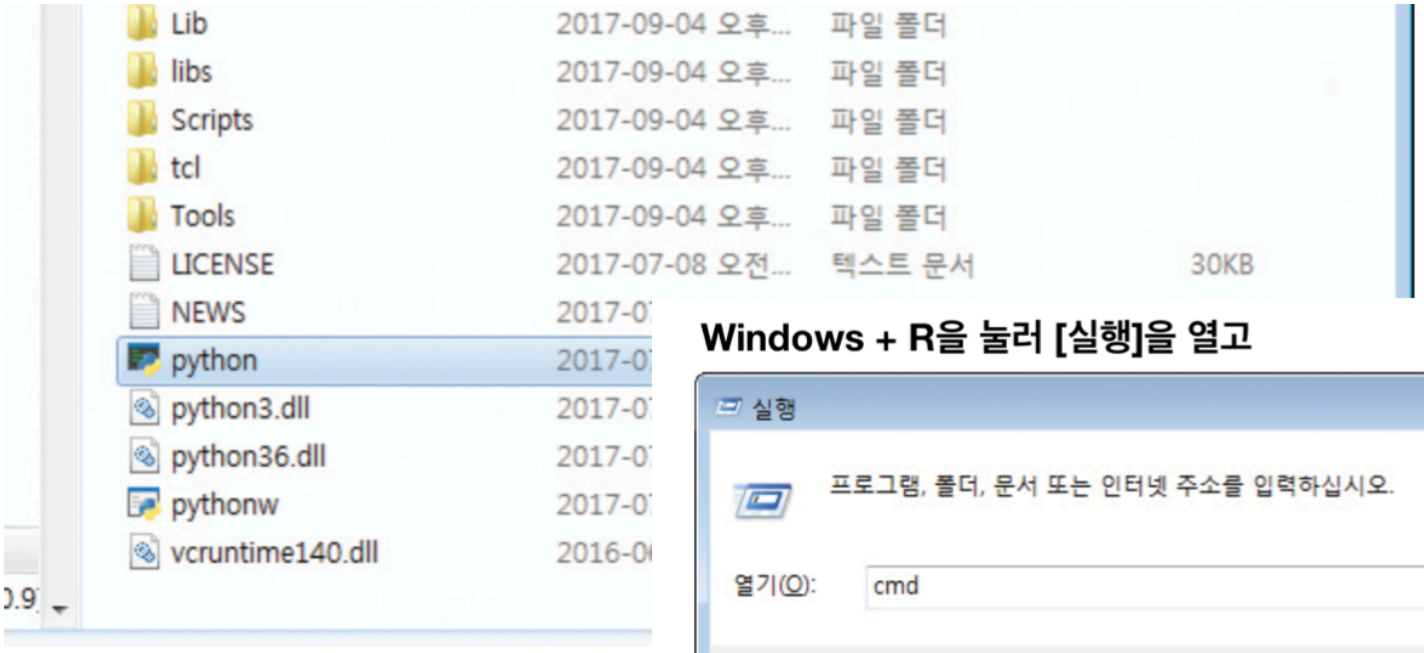
YouTube · 코드잇 · Mar 30, 2020

dojang.io › 강좌 › 프로그래밍 › 파이썬 코딩 도장

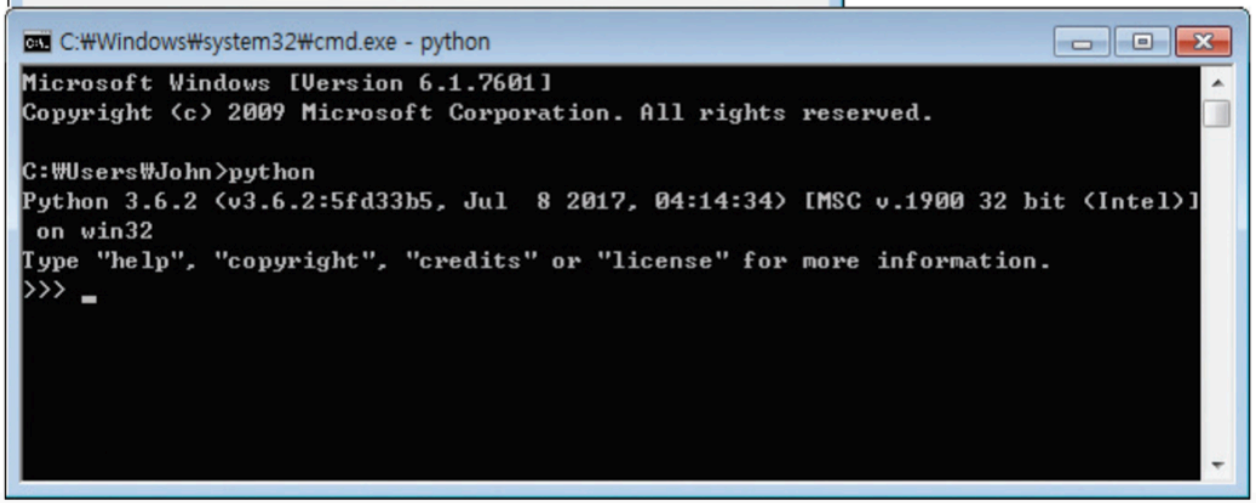
44.3 파이썬 패키지 인덱스에서 패키지 설치하기 - 코딩 도장

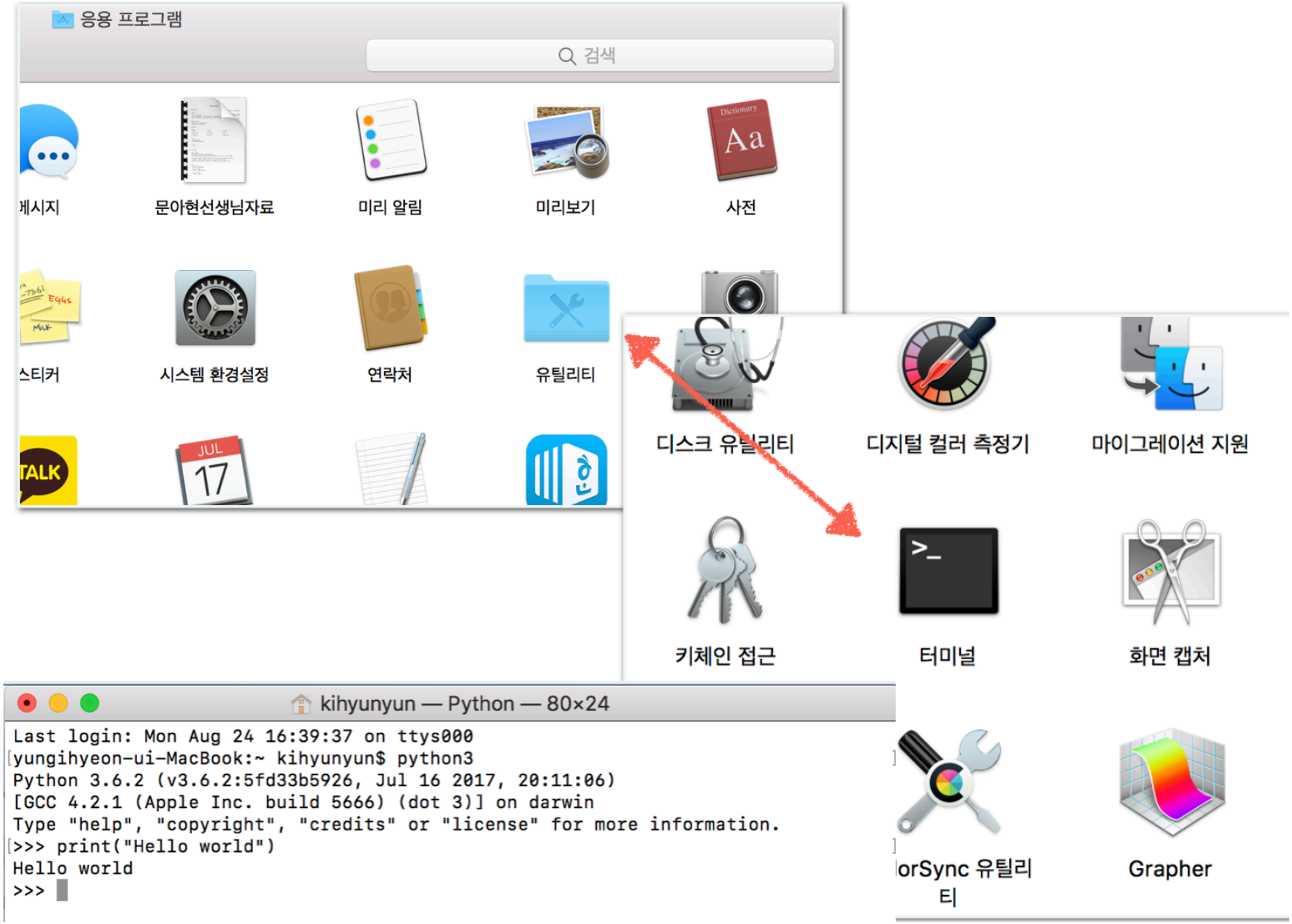
44.3.1 pip **설치**하기. pip는 **파이썬** 패키지 인덱스의 패키지 관리 명령어이며 Windows용 **파이썬**에는 기본으로 내장되어 있습니다. 리눅스와…

코딩 도장 · 길벗 IT 전문서(업로드 중단, 구계정) · Nov 2, 2018

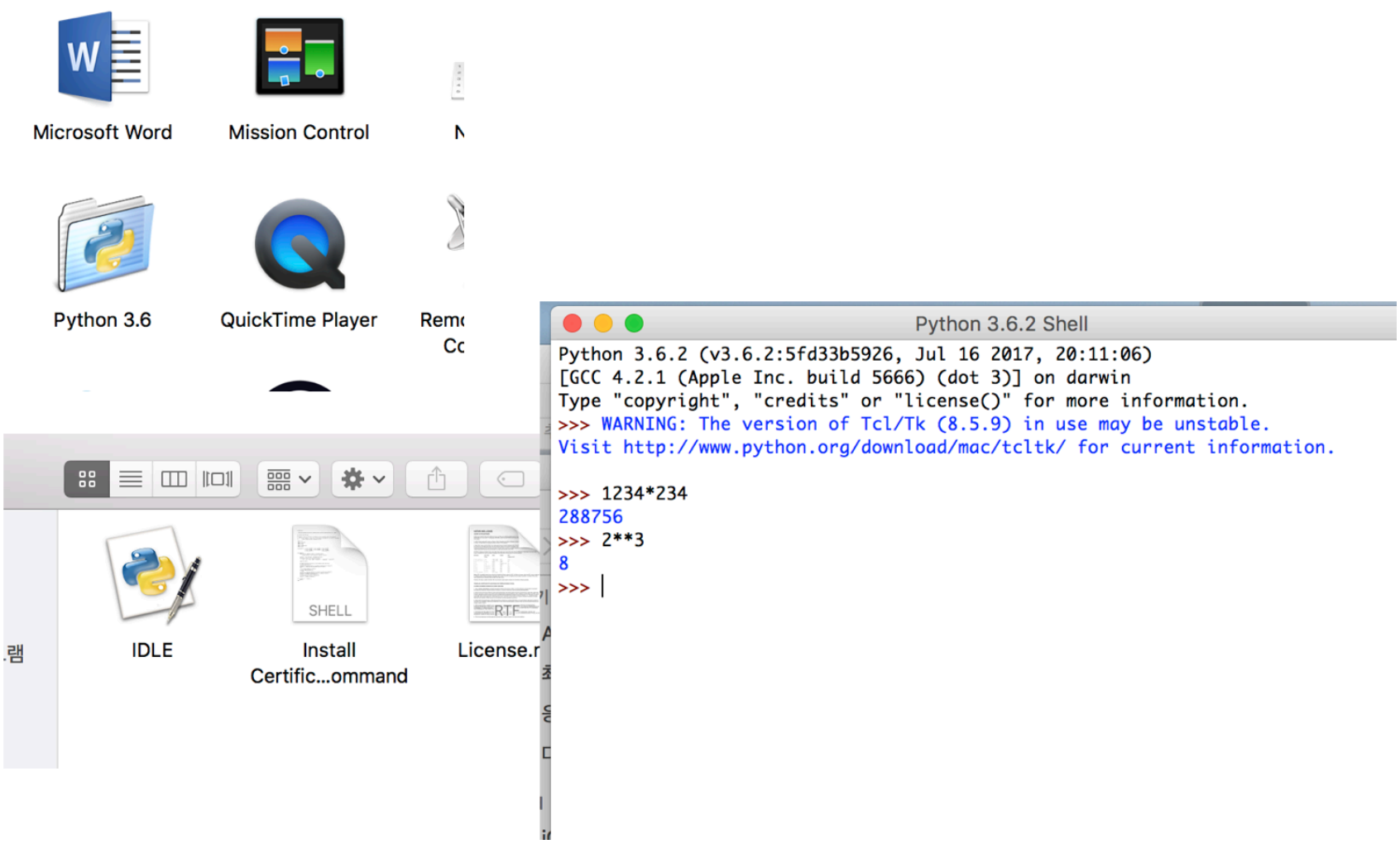


Windows + R을 눌러 [실행]을 열고




응용 프로그램
검색
메시지
문아현선생님자료
미리 알림
미리보기
사전
스티커
시스템 환경설정
연락처
유틸리티
디스크 유틸리티
디지털 컬러 측정기
마이그레이션 지원
키체인 접근
터미널
화면 캡처
orSync 유틸리티
Grapher
kihyunyun — Python — 80×24
Last login: Mon Aug 24 16:39:37 on ttys000
yungihyeon-ui-MacBook:~ kihyunyun$ python3
Python 3.6.2 (v3.6.2:5fd33b5926, Jul 16 2017, 20:11:06)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello world")
Hello world
>>>

Integrated DeveLopment Environment (**IDLE**): 통합개발환경



C:₩…>python -m pip install — upgrade pip

C:₩…>pip install scipy

C:₩…>pip list

C:₩…>pip install numpy

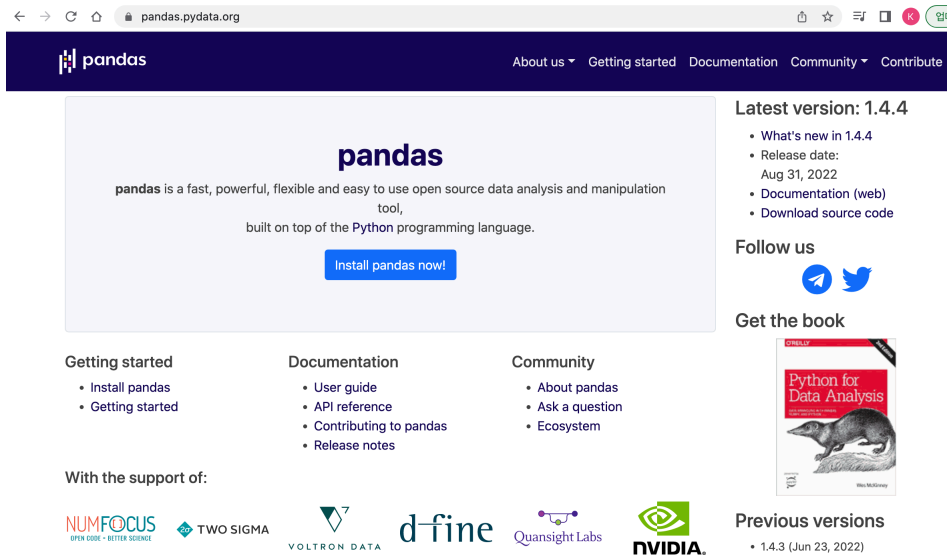
C:₩…>pip install matplotlib

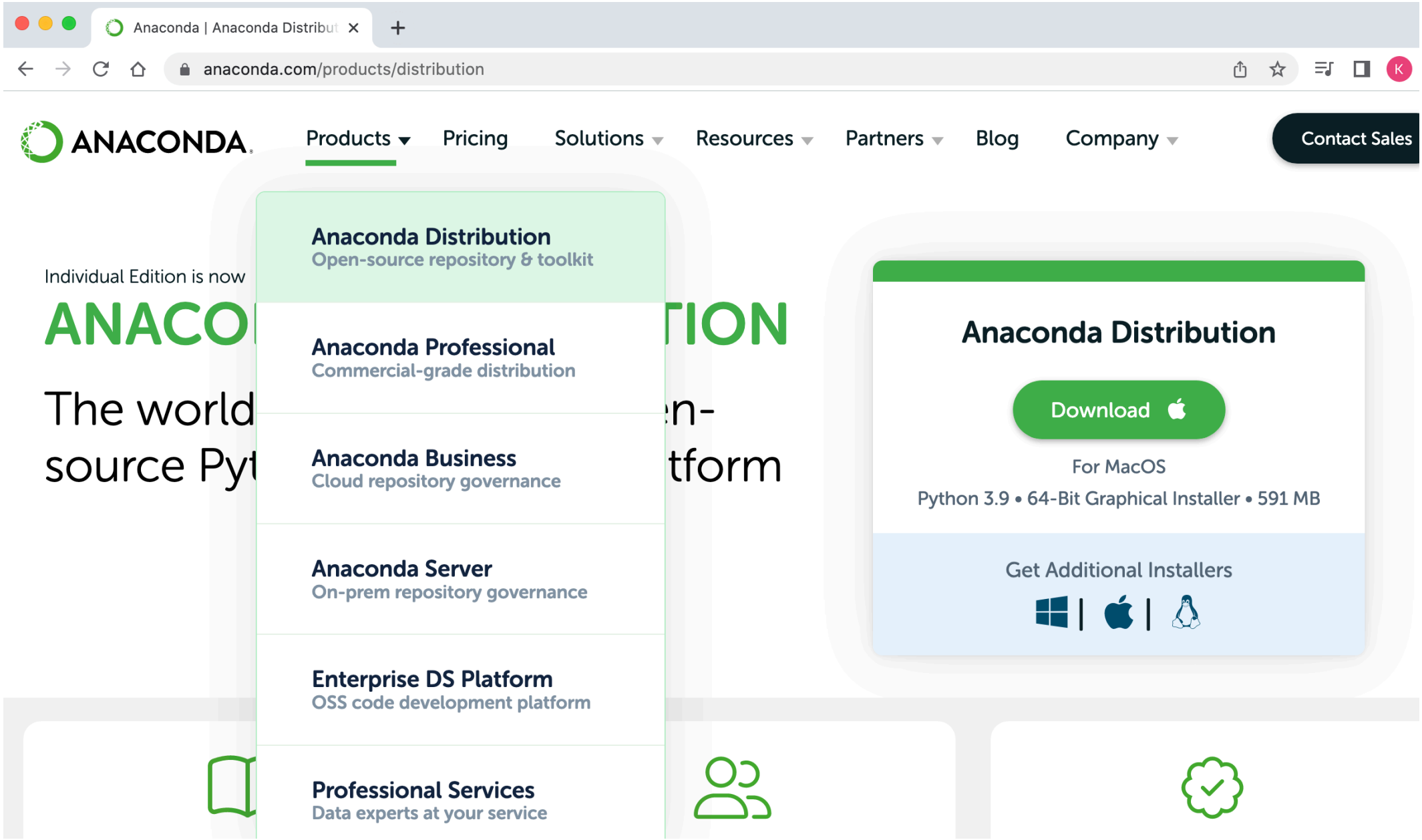
C:₩…>pip install tensorflow==2.0.0

C:₩…>pip install keras

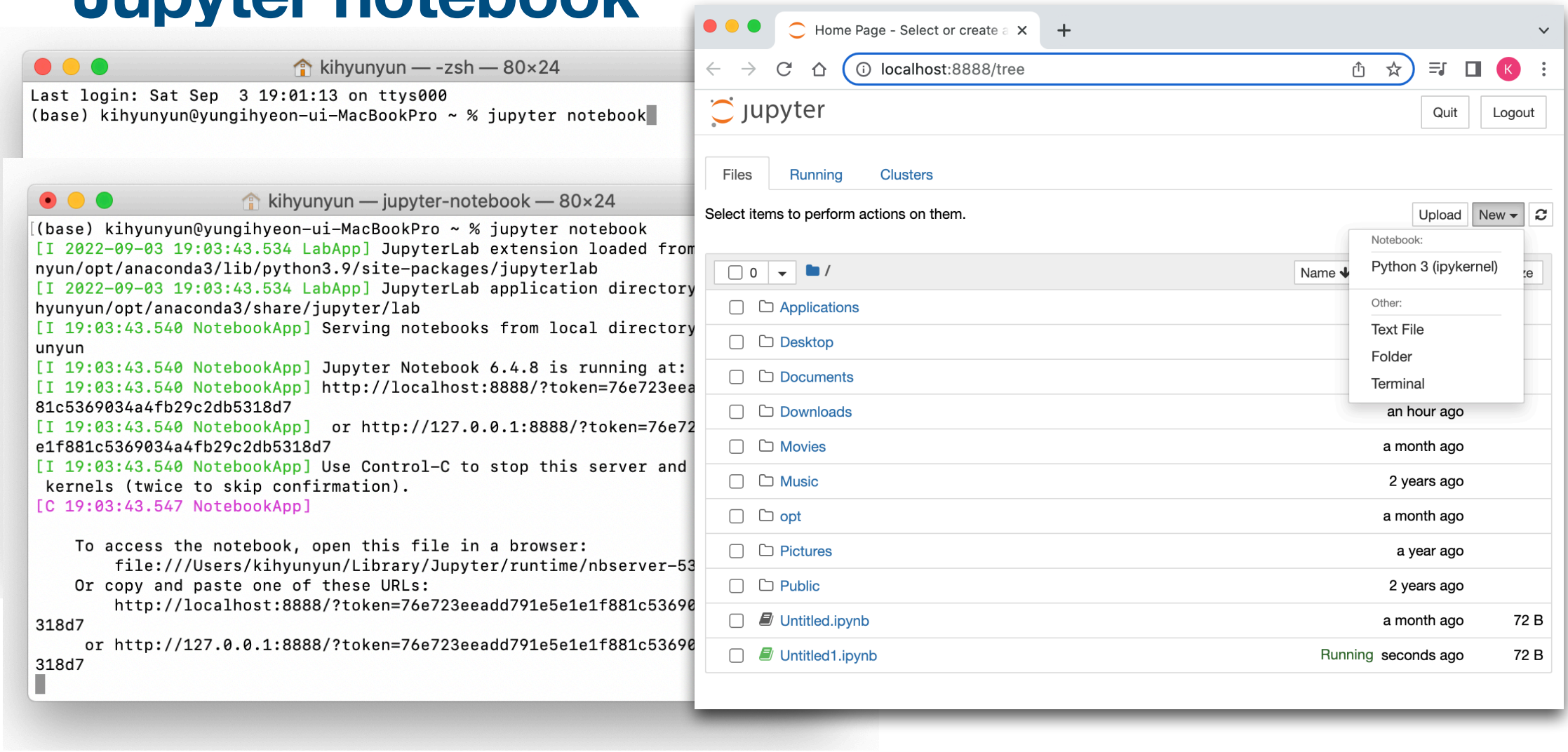
C:₩…>pip install jupyter

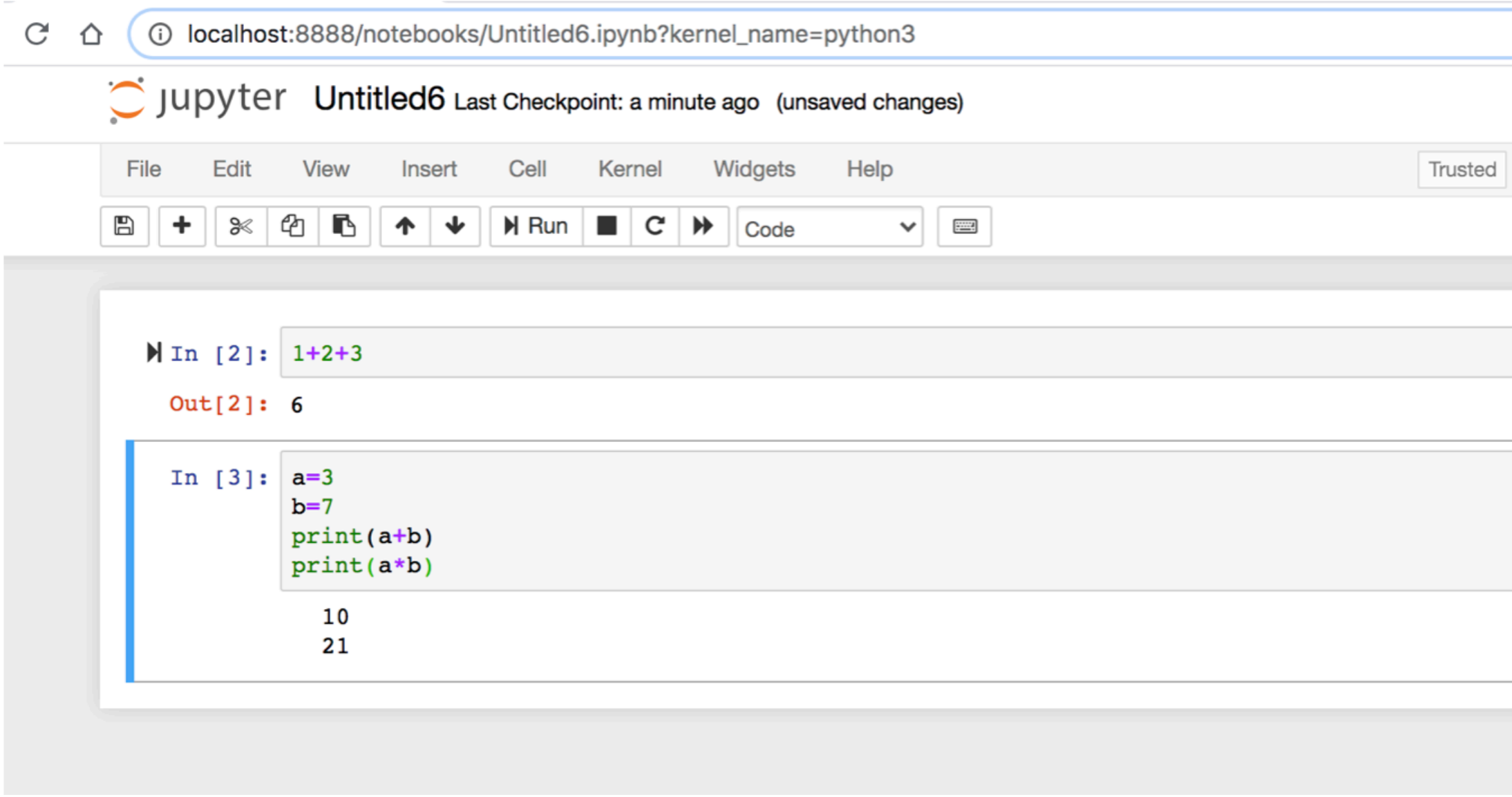
C:₩…> jupyter notebook

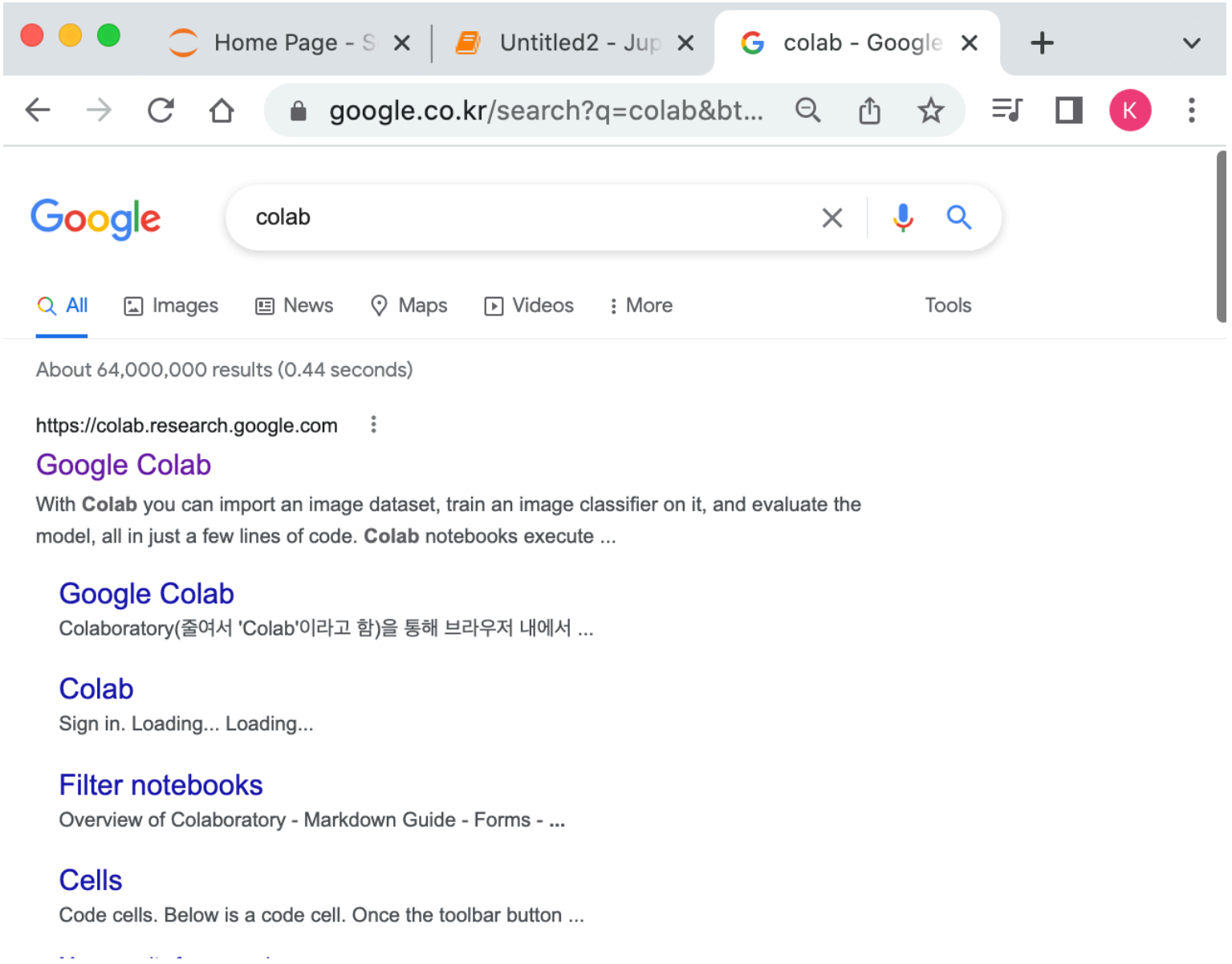



Anaconda | Anaconda Distribut
anaconda.com/products/distribution
ANACONDA.
Products
Pricing
Solutions
Resources
Partners
Blog
Company
Contact Sales
Anaconda Distribution
Open-source repository & toolkit
Anaconda Professional
Commercial-grade distribution
Anaconda Business
Cloud repository governance
Anaconda Server
On-prem repository governance
Enterprise DS Platform
OSS code development platform
Professional Services
Data experts at your service
Individual Edition is now
ANACO
TION
The world
n-
source Pyt
tform
Anaconda Distribution
Download
For MacOS
Python 3.9 • 64-Bit Graphical Installer • 591 MB
Get Additional Installers

Jupyter notebook




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
Run
Code
In [2]:
1+2+3
Out[2]:
6
In [3]:
a=3
b=7
print(a+b)
print(a*b)
10
21


Home Page - S
Untitled2 - Jup
colab - Google
google.co.kr/search?q=colab&bt...
K
Google
colab
All
Images
News
Maps
Videos
More
Tools
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 ...

Welcome To Colaboratory - Co

colab.research.google.com

클릭하면 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

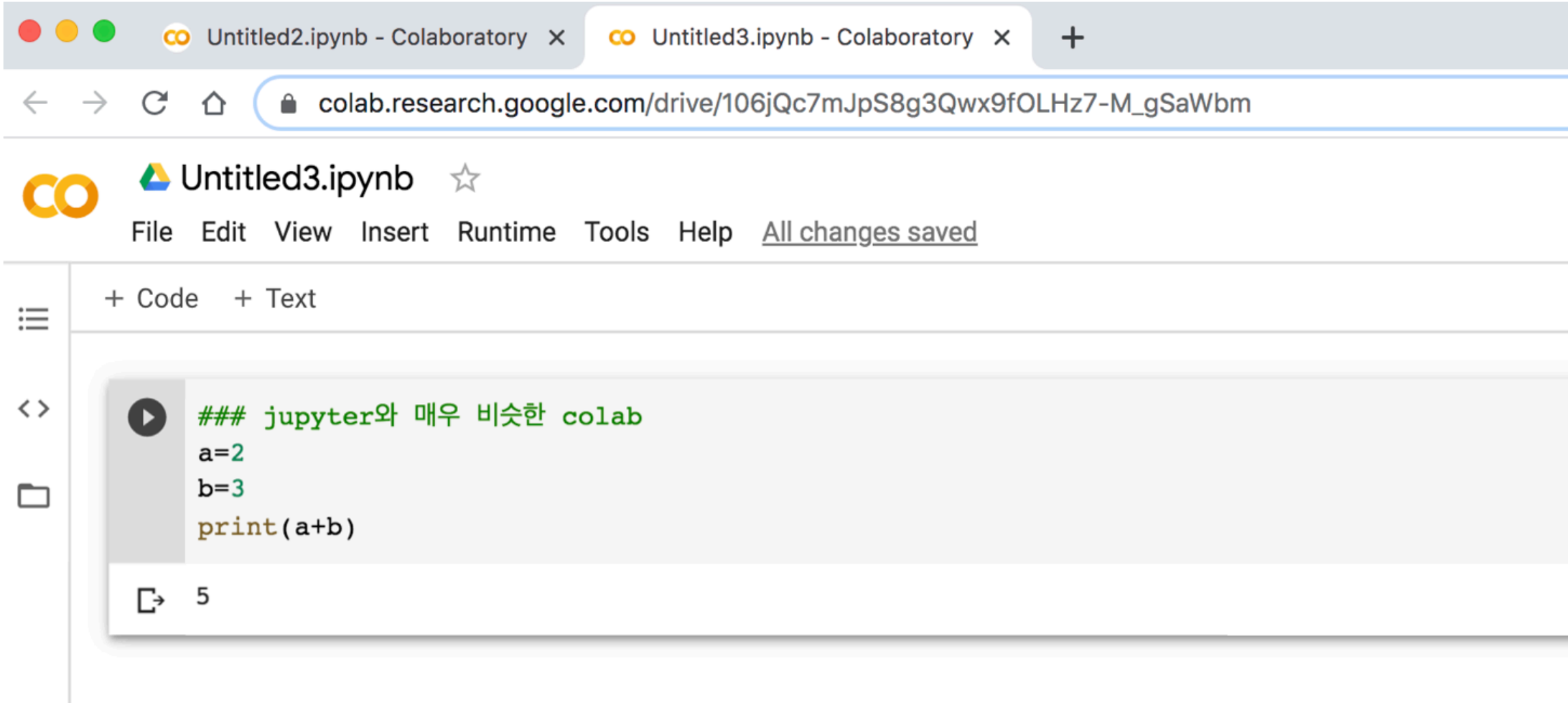
Welcome to Colab!

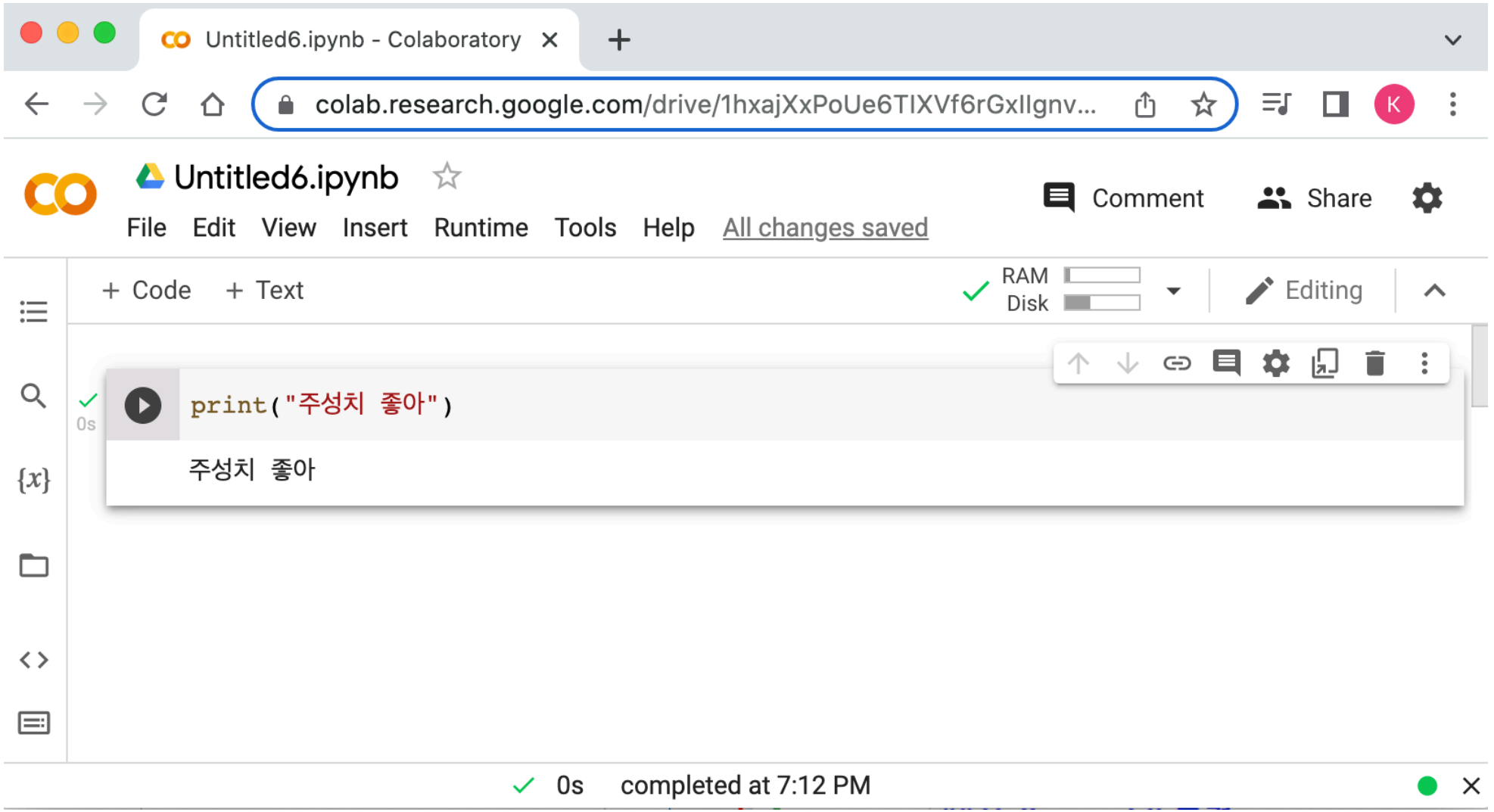
f you're already familiar with Colab, check out this video to learn about interact
nd the command palette.

What is Colab?

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

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




Untitled6.ipynb - Colaboratory
colab.research.google.com/drive/1hxajXxPoUe6TIXVf6rGxIlgnv...
Untitled6.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
print("주성치 좋아")
0s
주성치 좋아
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

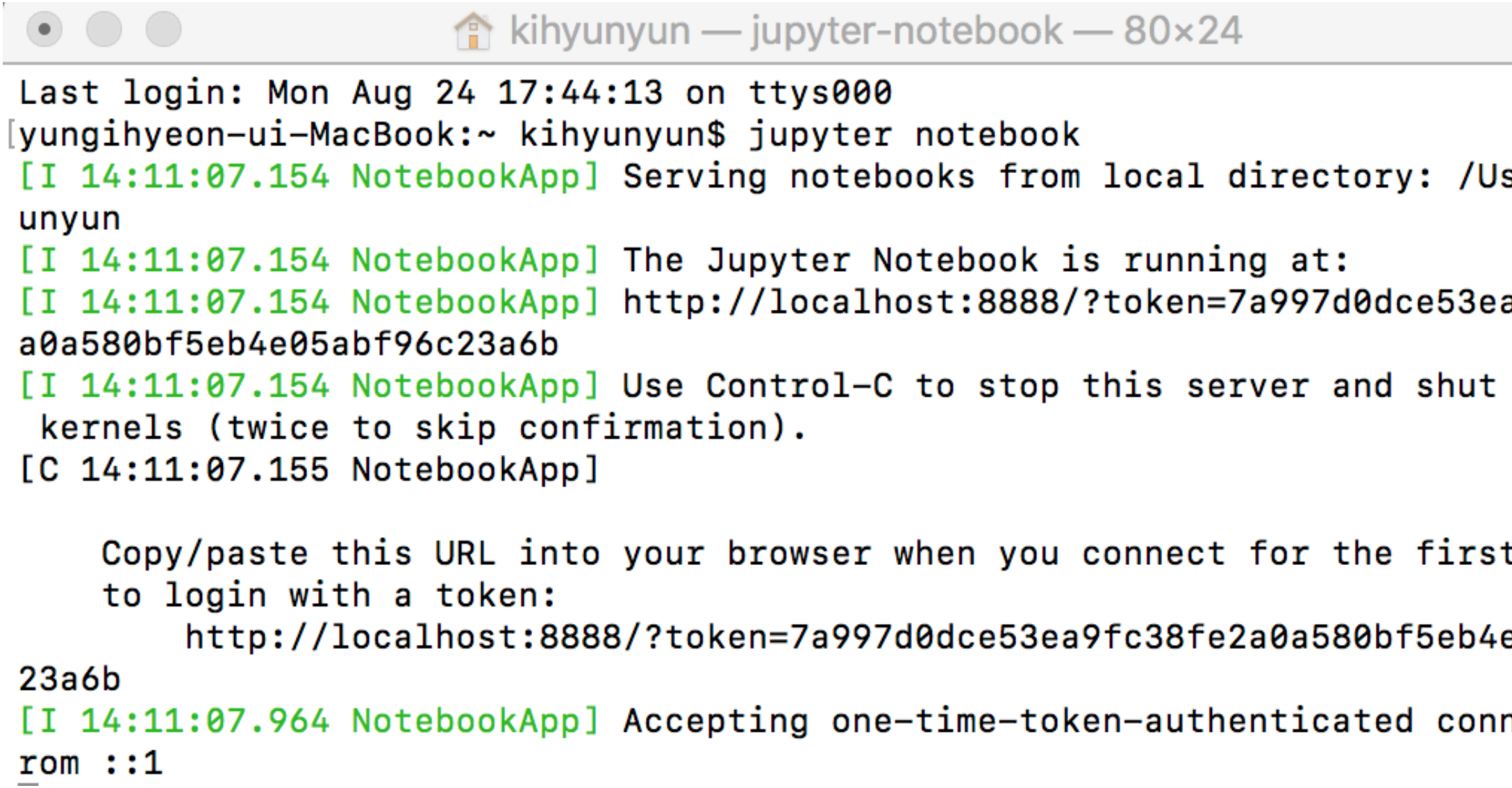
+ Code + Text

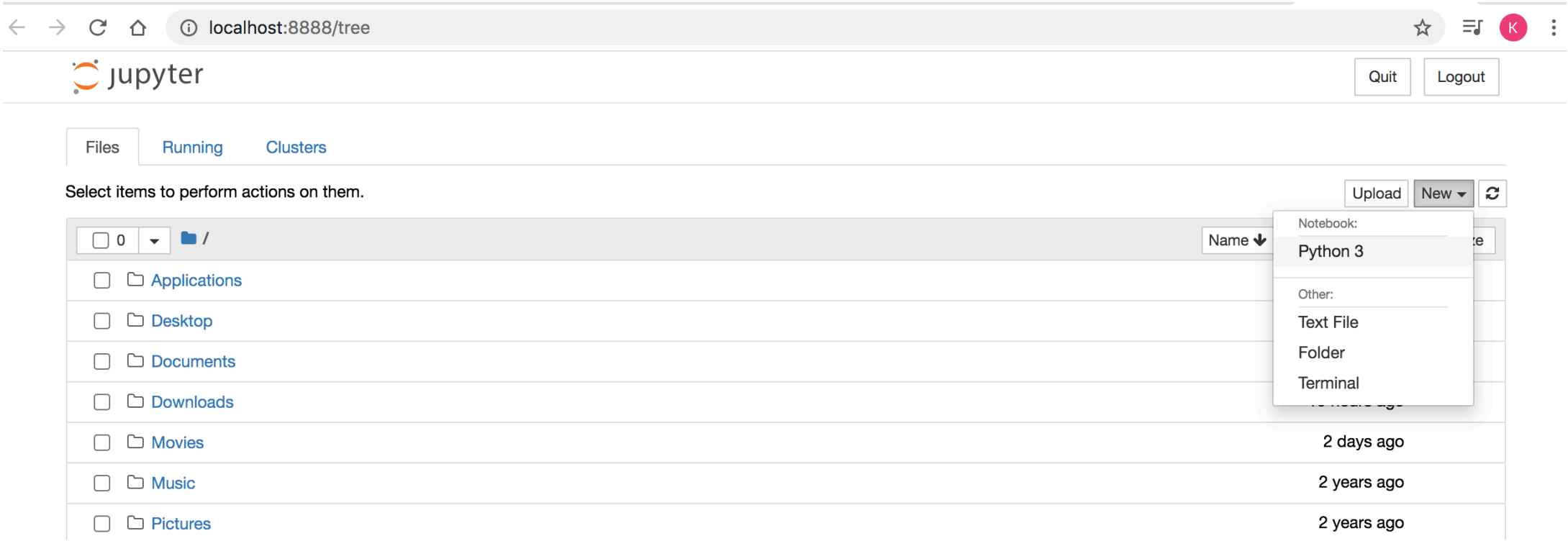
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

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)
```

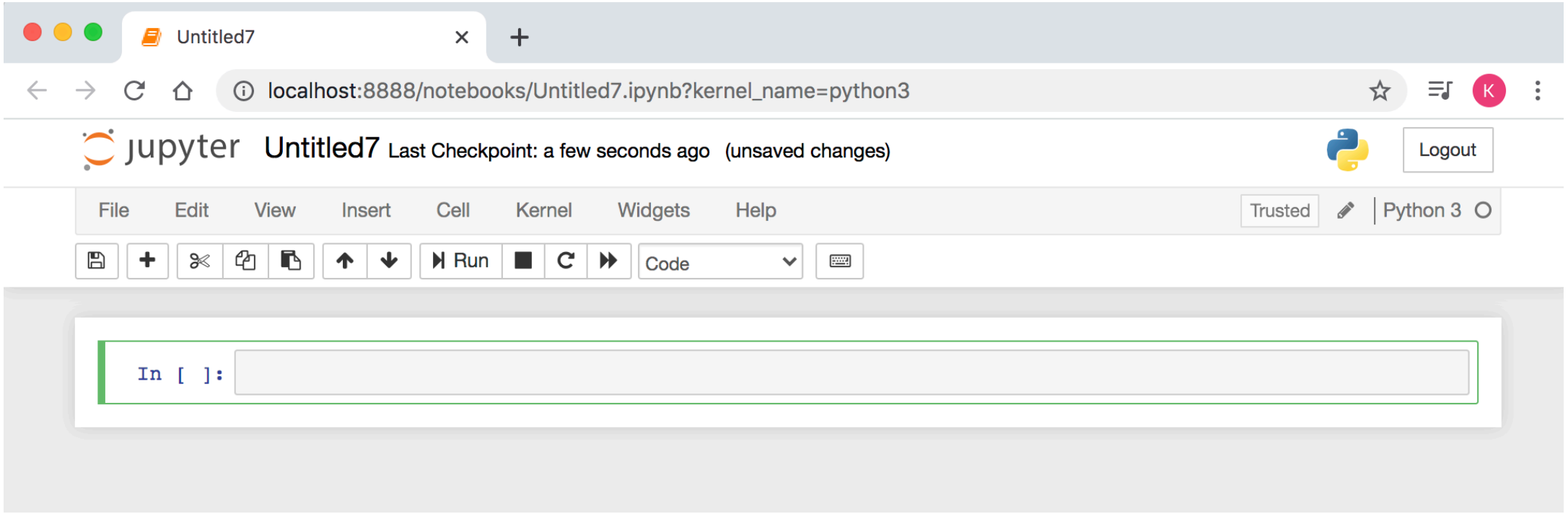
```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&

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 — 80×24
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
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


localhost:8888/tree
jupyter
Quit
Logout
Files
Running
Clusters
Select items to perform actions on them.
Upload
New
Notebook:
Python 3
Other:
Text File
Folder
Terminal
0
/
Name
Applications
Desktop
Documents
Downloads
Movies
2 days ago
Music
2 years ago
Pictures
2 years ago


Untitled7
localhost:8888/notebooks/Untitled7.ipynb?kernel_name=python3
jupyter
Untitled7
Last Checkpoint: a few seconds ago
(unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
In [ ]:

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)
```

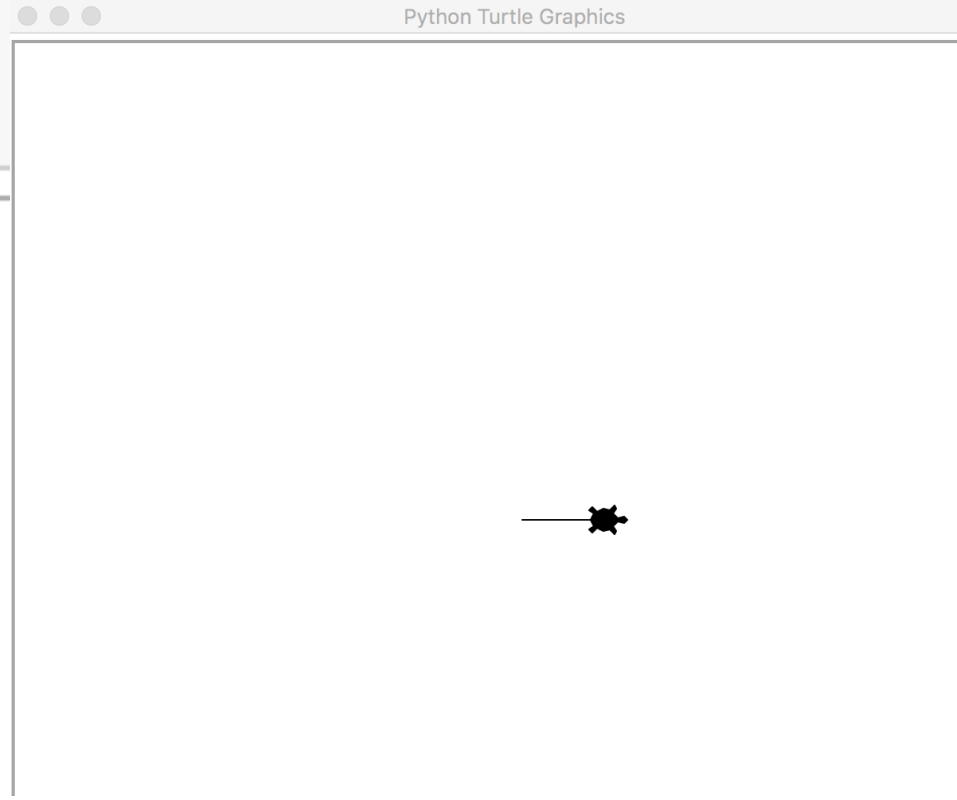
```
이름이 뭔가요?
```

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

```
이름이 뭔가요?주성치
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
```

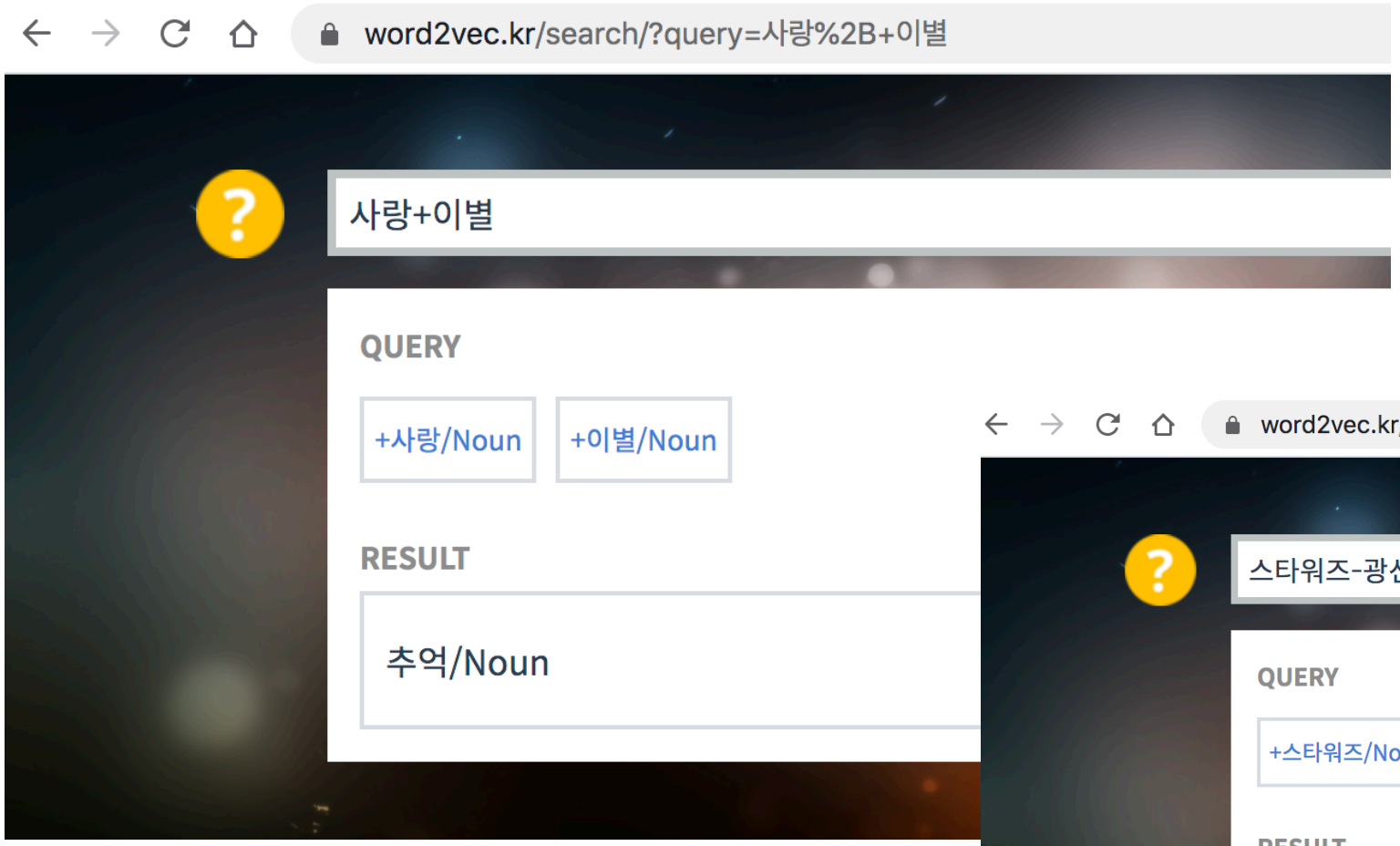
In [29]:
```
Lyrics1 ="쥐가 한마리, "

Lyrics2 ='두마리'

Lyrics3 = Lyrics1 + Lyrics2
print(Lyrics1 + Lyrics2)
```

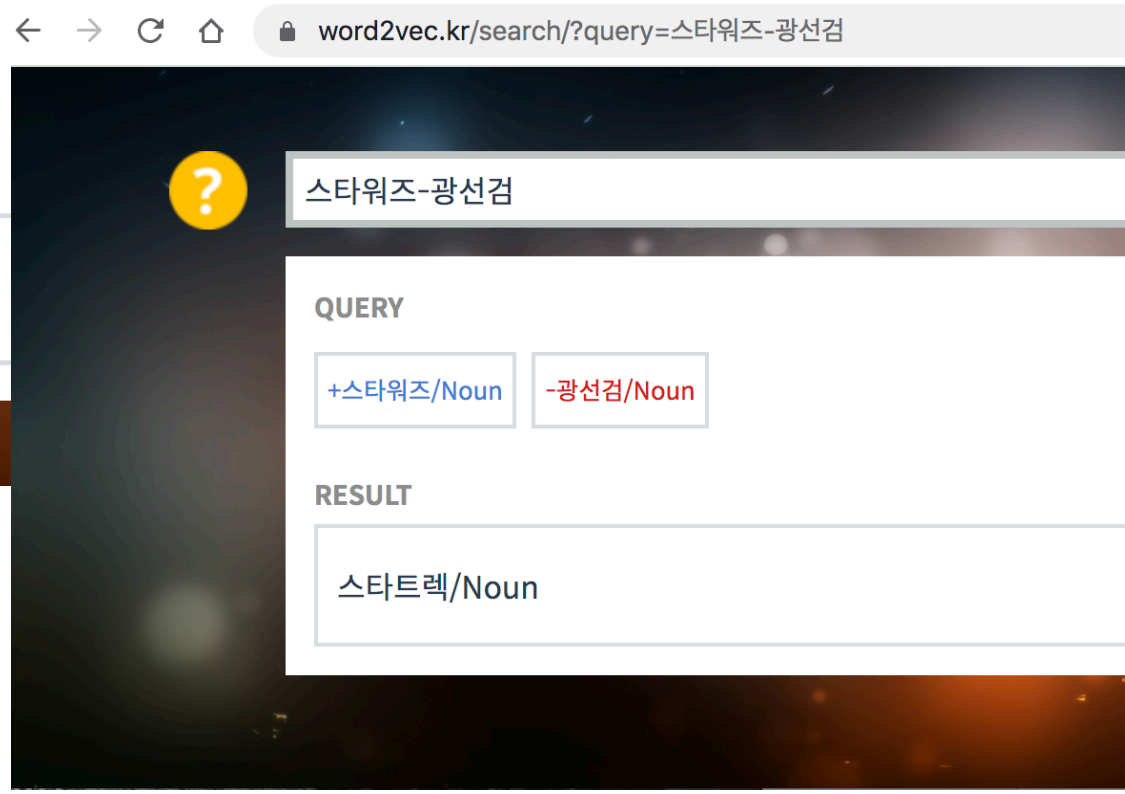
쥐가 한마리, 두마리

In [30]:

```
Lyrics1 ="쥐가 "

Lyrics2 ='매우 '

Lyrics3 = '많아'

Lyrics4 = Lyrics1 + Lyrics2*3 + Lyrics3
print(Lyrics4)
```

쥐가 매우 매우 매우 많아


word2vec.kr/search/?query=사랑%2B+이별
사랑+이별
QUERY
+사랑/Noun
+이별/Noun
RESULT
추억/Noun


word2vec.kr/search/?query=스타워즈-광선검
스타워즈-광선검
QUERY
+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="순조선각연덕현도경인순희체성응명흠광석경계천배극융원돈휴의행소윤희화준렬대중지정홍훈철모건시태형창운홍

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
>=	왼쪽 값이 크거나 같다
<=	왼쪽 값이 작거나 같다
==	값이 같다
!=	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
```

```
ue)

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

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

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

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

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.
```

```python
# 조건문 1

hungry = True

if hungry:
    print("배가 고프다.")
```

```
배가 고프다.
```

```python
# 조건문 2

hungry = False

if hungry:
    print("배가 고프다.")
```

```python
# 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.
```

In [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
```

```
마음에 드는 물건을 찾았는가?(예/아니오):예
좋구나!
얼마인가?1000

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call la
<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'
```

```python
# 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.
```