PYTHON-PALETTE

@yeseuly.park – 24.03.17 (2nd)

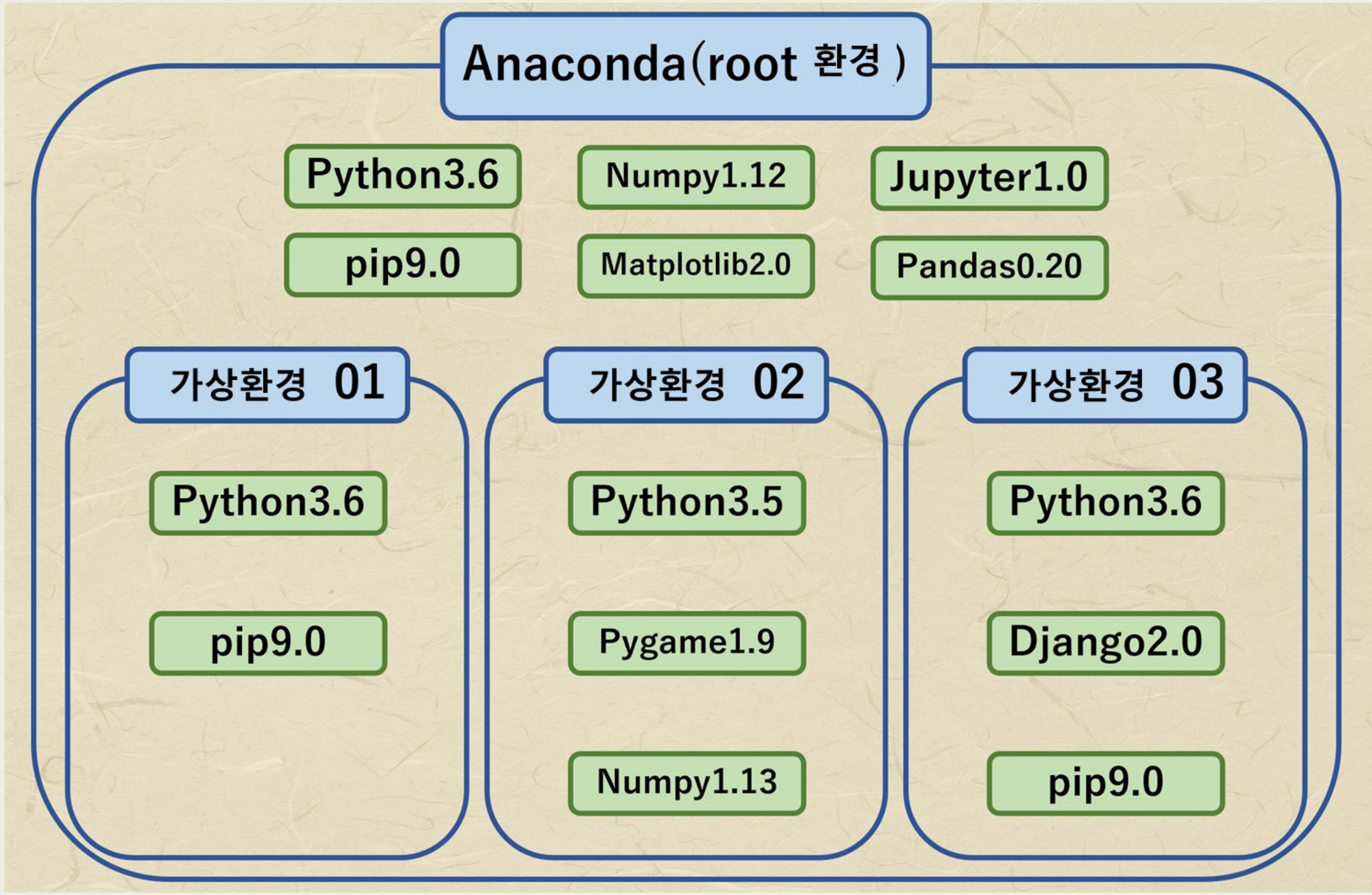
To Do List

- Environment
  - *Conda*
- Development
  - *variable*
  - *function*
  - *If*
  - *for / while*
  - *assignment*

Environment

- Conda
  - *conda create env*
  - *conda info*

Conda – definition



Anaconda(root 환경)

Python3.6 | Numpy1.12 | Jupyter1.0

pip9.0 | Matplotlib2.0 | Pandas0.20

가상환경 01

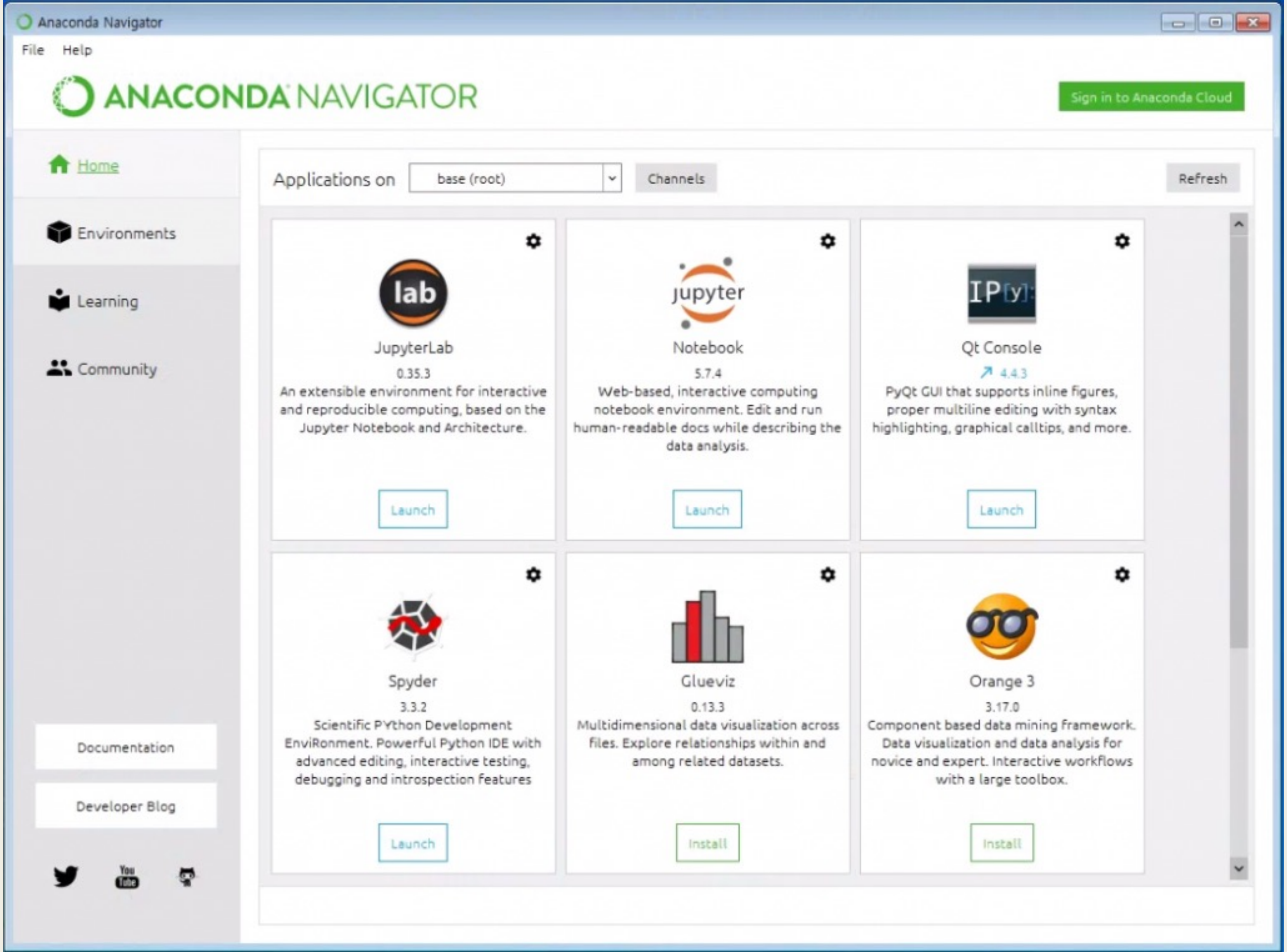
- Python3.6
- pip9.0

가상환경 02

- Python3.5
- Pygame1.9
- Numpy1.13

가상환경 03

- Python3.6
- Django2.0
- pip9.0



JupyterLab
0.35.3
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

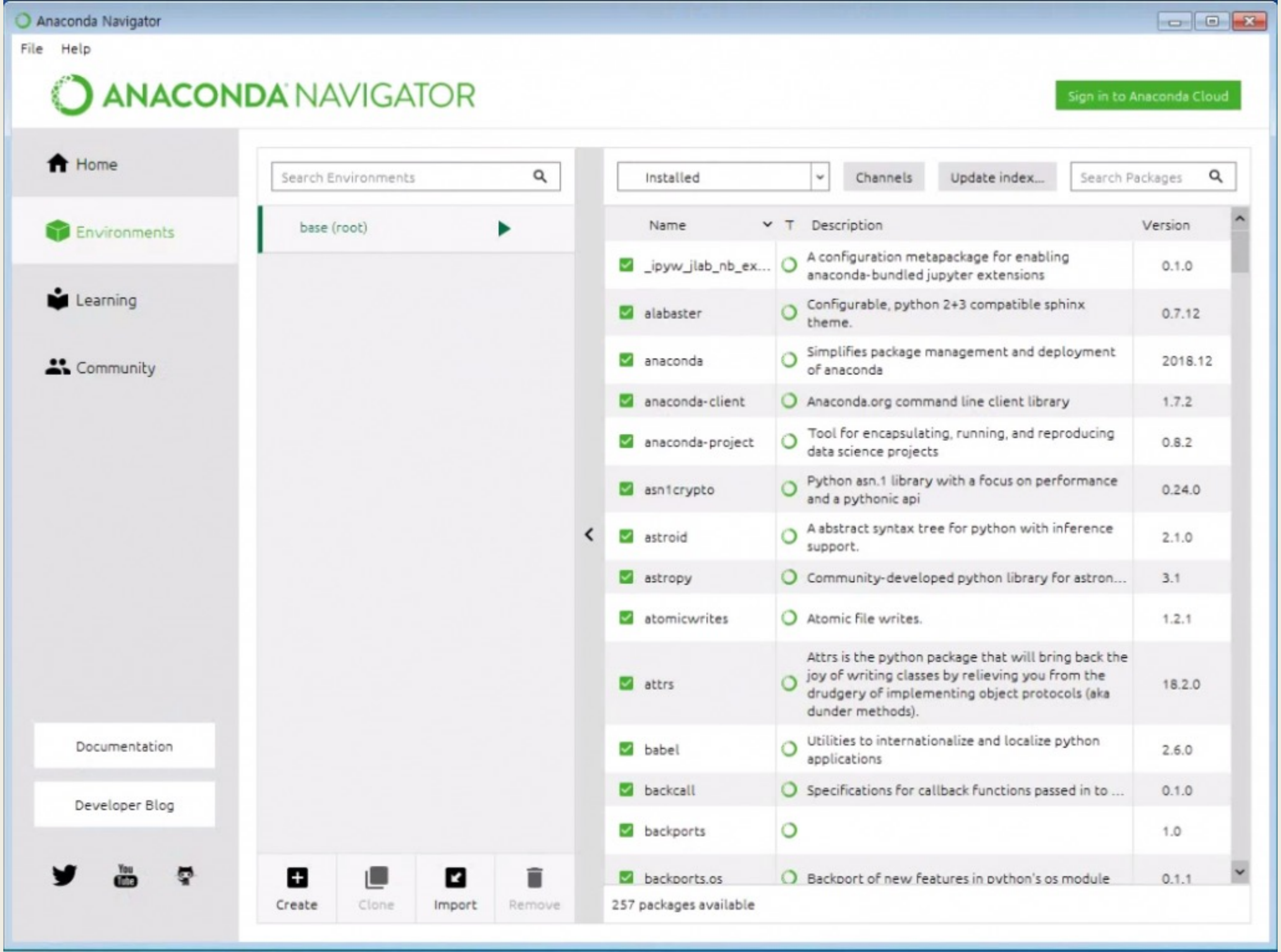
Notebook
5.7.4
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Qt Console
↗ 4.4.3
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Spyder
3.3.2
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Glueviz
0.13.3
Multidimensional data visualization across files. Explore relationships within and among related datasets.

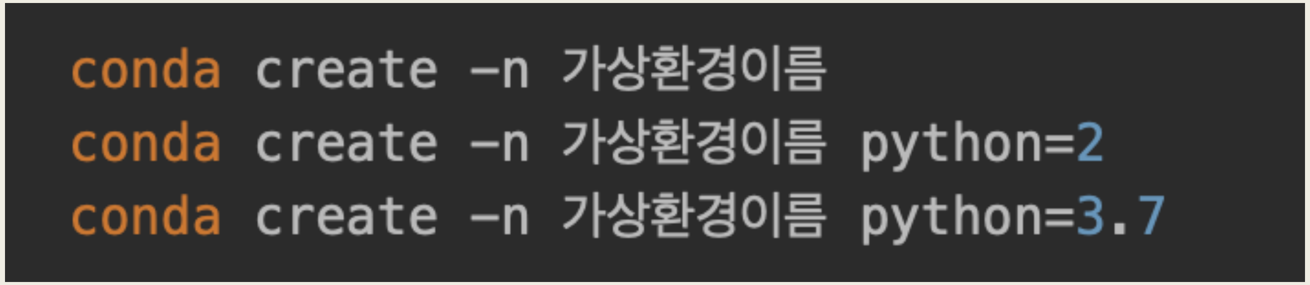
Orange 3
3.17.0
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.



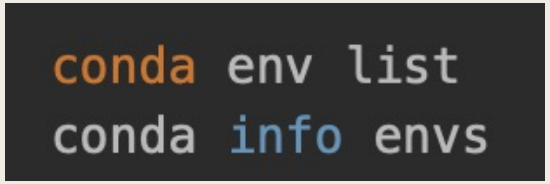
	Name	T	Description	Version
☑	_ipyw_jlab_nb_ex...	○	A configuration metapackage for enabling anaconda-bundled jupyter extensions	0.1.0
☑	alabaster	○	Configurable, python 2+3 compatible sphinx theme.	0.7.12
☑	anaconda	○	Simplifies package management and deployment of anaconda	2018.12
☑	anaconda-client	○	Anaconda.org command line client library	1.7.2
☑	anaconda-project	○	Tool for encapsulating, running, and reproducing data science projects	0.8.2
☑	asn1crypto	○	Python asn.1 library with a focus on performance and a pythonic api	0.24.0
☑	astroid	○	A abstract syntax tree for python with inference support.	2.1.0
☑	astropy	○	Community-developed python library for astron...	3.1
☑	atomicwrites	○	Atomic file writes.	1.2.1
☑	attrs	○	Attrs is the python package that will bring back the joy of writing classes by relieving you from the drudgery of implementing object protocols (aka dunder methods).	18.2.0
☑	babel	○	Utilities to internationalize and localize python applications	2.6.0
☑	backcall	○	Specifications for callback functions passed in to ...	0.1.0
☑	backports	○		1.0
☑	backports.os	○	Backport of new features in python's os module	0.1.1

257 packages available

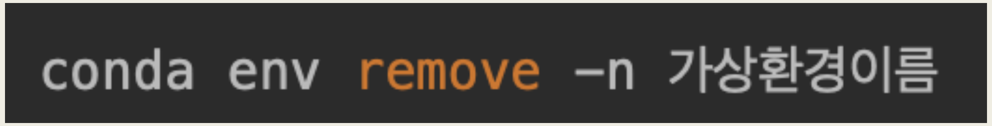
Conda – how to use?



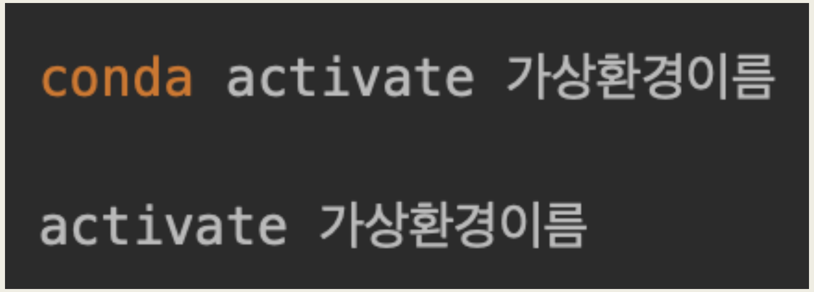
```
conda create -n 가상환경이름
conda create -n 가상환경이름 python=2
conda create -n 가상환경이름 python=3.7
```



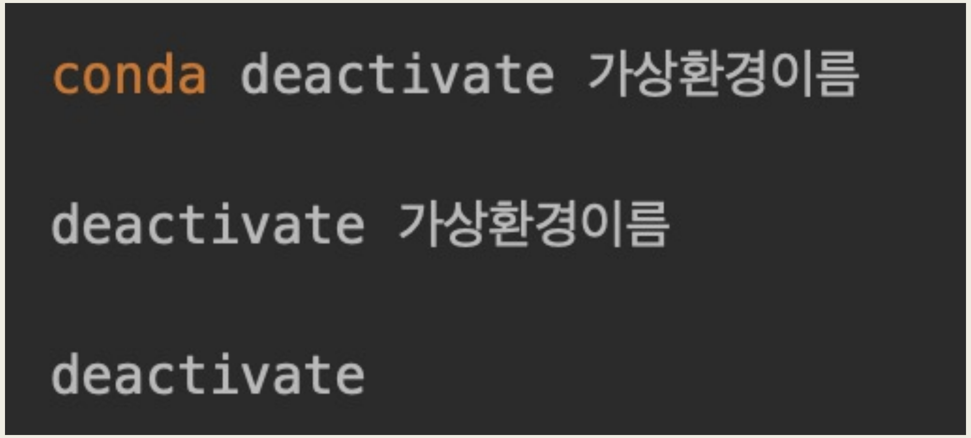
```
conda env list
conda info envs
```



```
conda env remove -n 가상환경이름
```



```
conda activate 가상환경이름

activate 가상환경이름
```

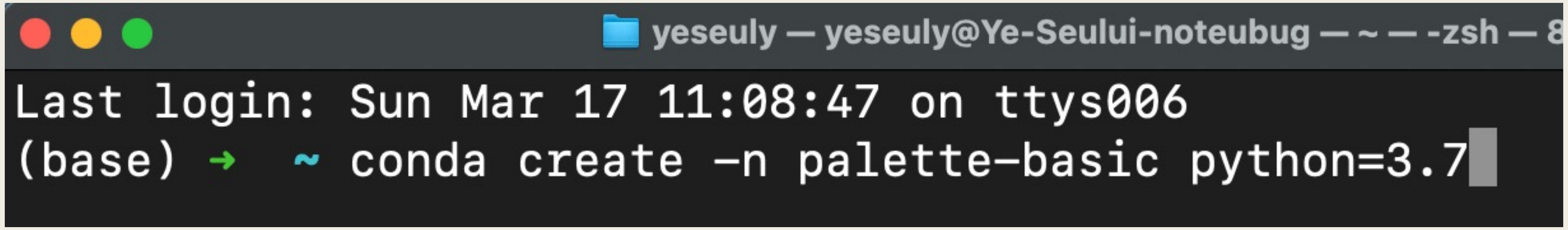


```
conda deactivate 가상환경이름

deactivate 가상환경이름

deactivate
```

Conda –palette-basic

(Terminal Window)

- conda create –n palette-basic python=3.7



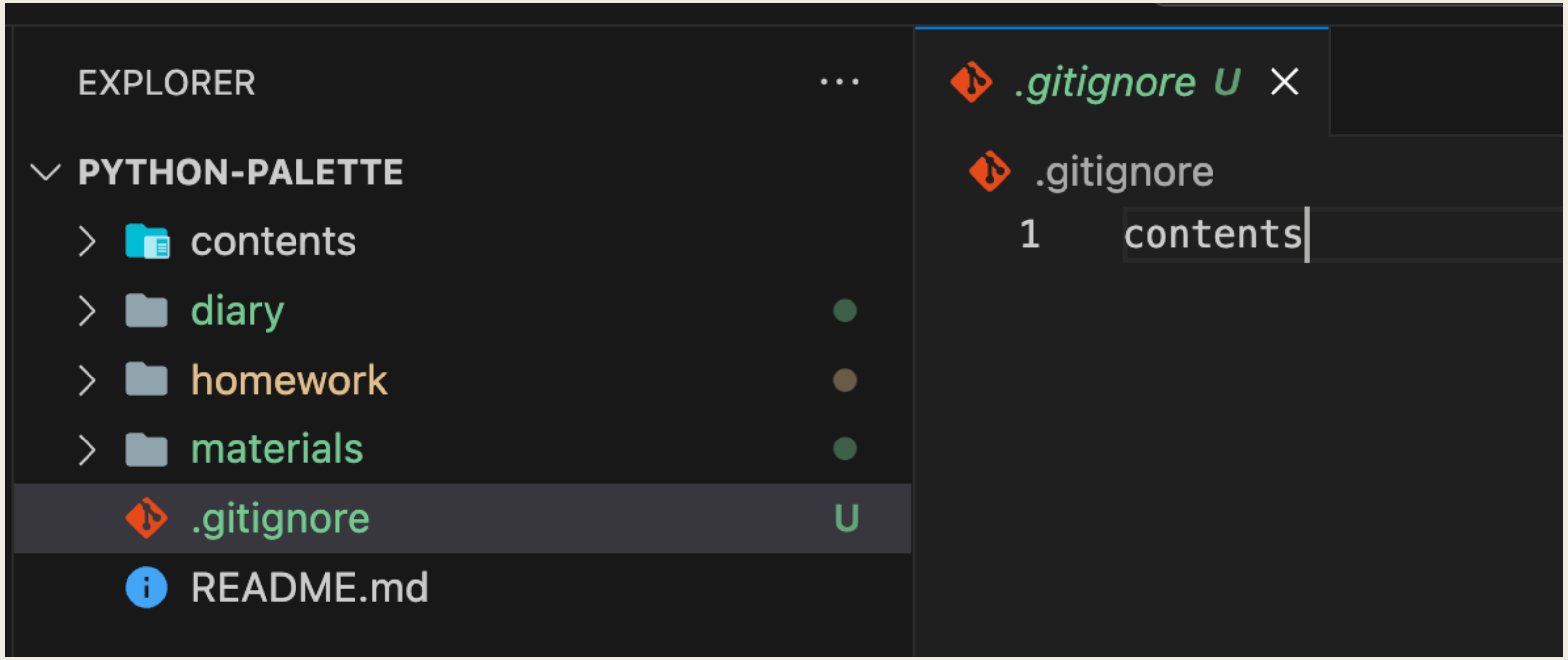
```
Last login: Sun Mar 17 11:08:47 on ttys006
(base) ➜  ~ conda create -n palette-basic python=3.7
```

Conda –palette-basic

(Terminal Window)

- (palette-basic) conda install numpy
- (palette-basic) conda install pandas
- (palette-basic) conda install matplotlib
- (palette-basic) conda install jupyter

Gitignore



```
contents
```

Development

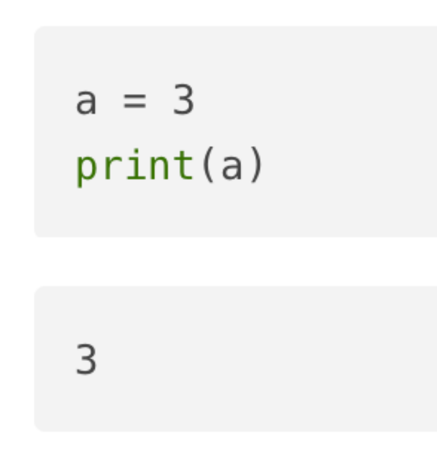
- Basic Python

- variable
- function
- If
- for / while
- assignment

Variables – definition

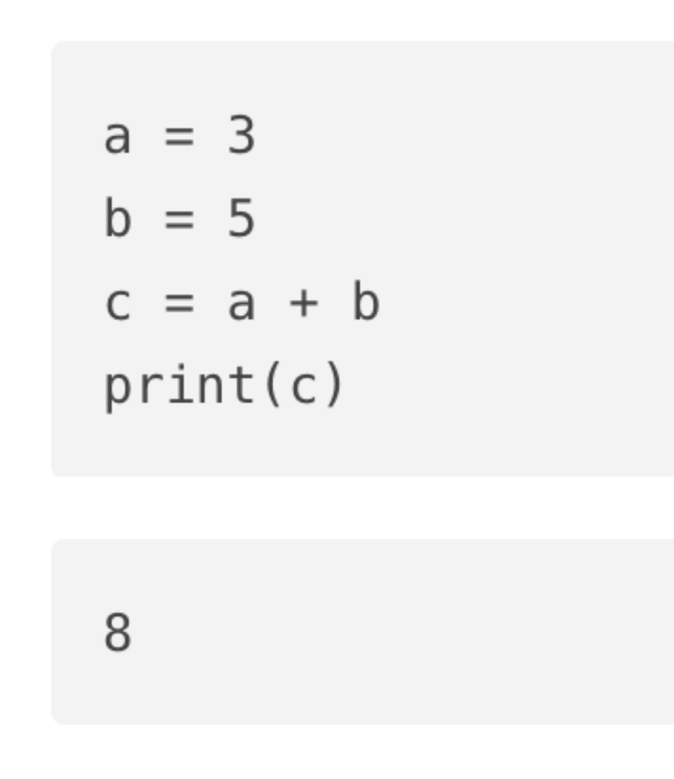
Grammar

→ “Variable Name = Variable Value”



```
a = 3
print(a)
```

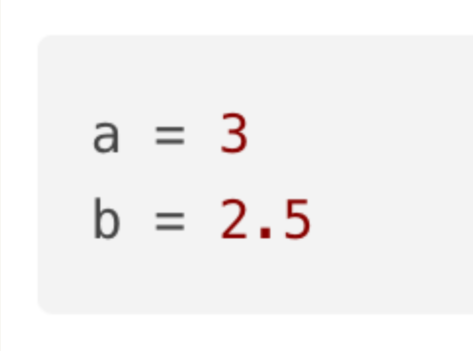
```
3
```



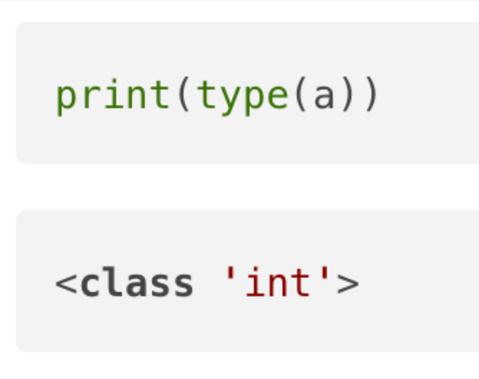
```
a = 3
b = 5
c = a + b
print(c)
```

```
8
```

Variables – int/float

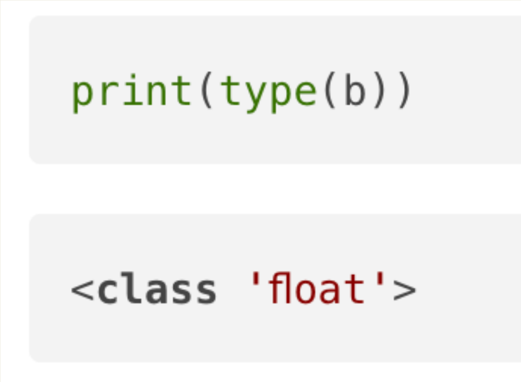


```
a = 3
b = 2.5
```



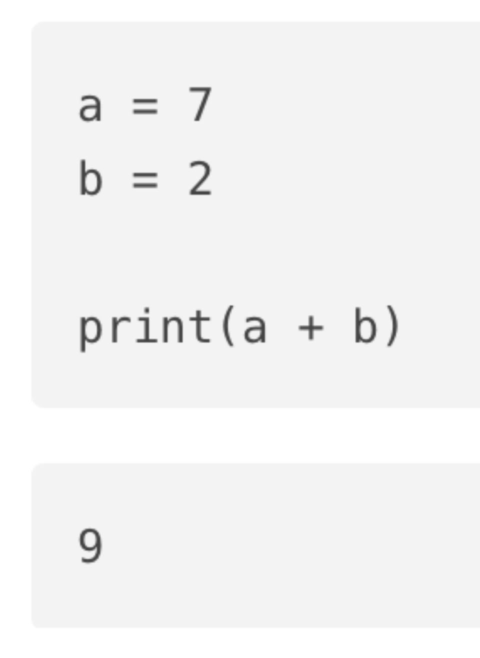
```
print(type(a))
```

```
<class 'int'>
```



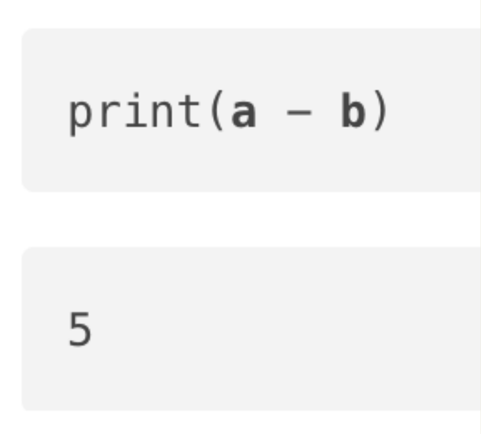
```
print(type(b))
```

```
<class 'float'>
```



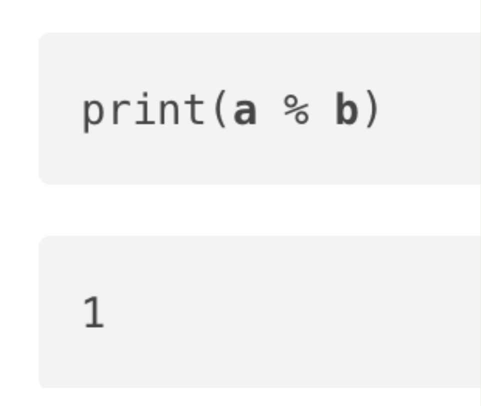
```
a = 7
b = 2

print(a + b)
```

```
9
```



```
print(a - b)
```

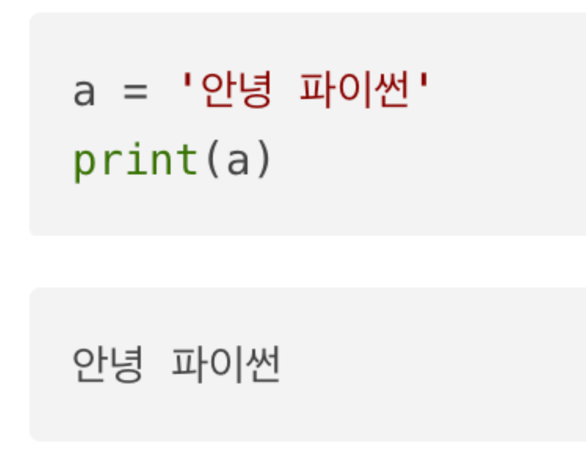
```
5
```



```
print(a % b)
```

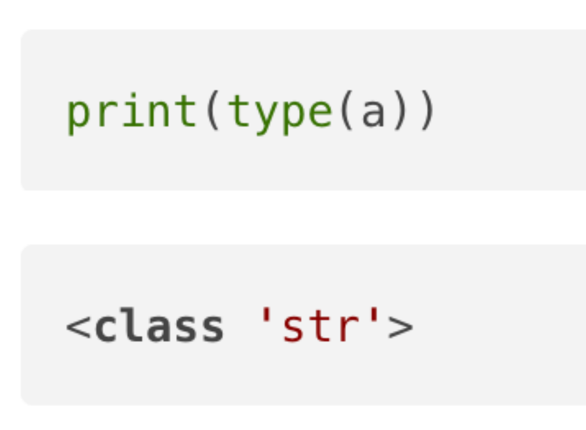
```
1
```

Variables – string



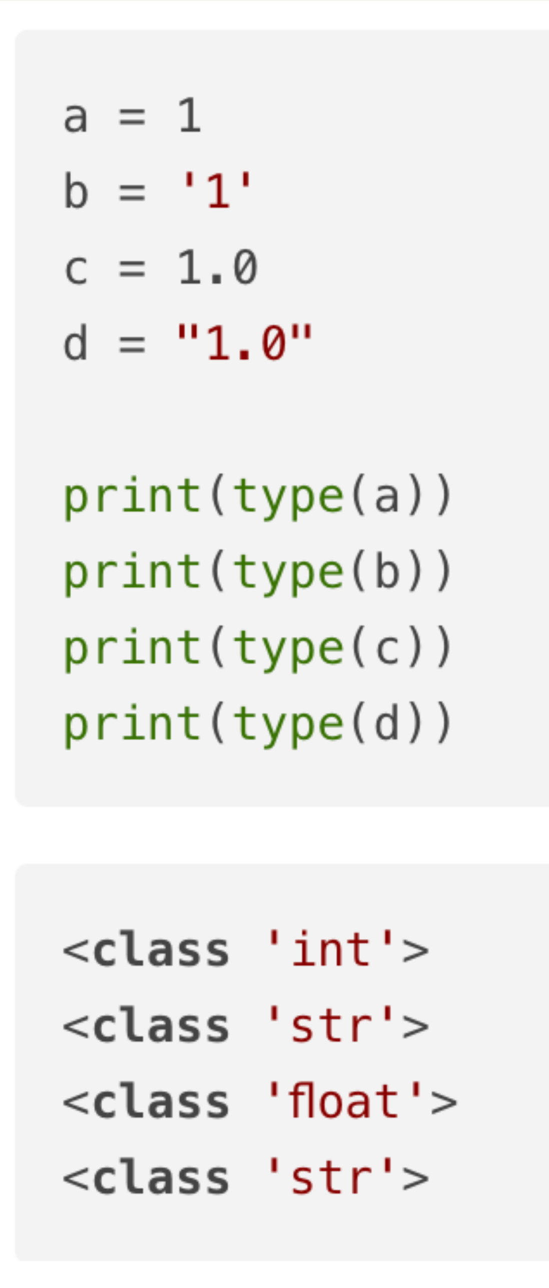
```
a = '안녕 파이썬'
print(a)
```

```
안녕 파이썬
```



```
print(type(a))
```

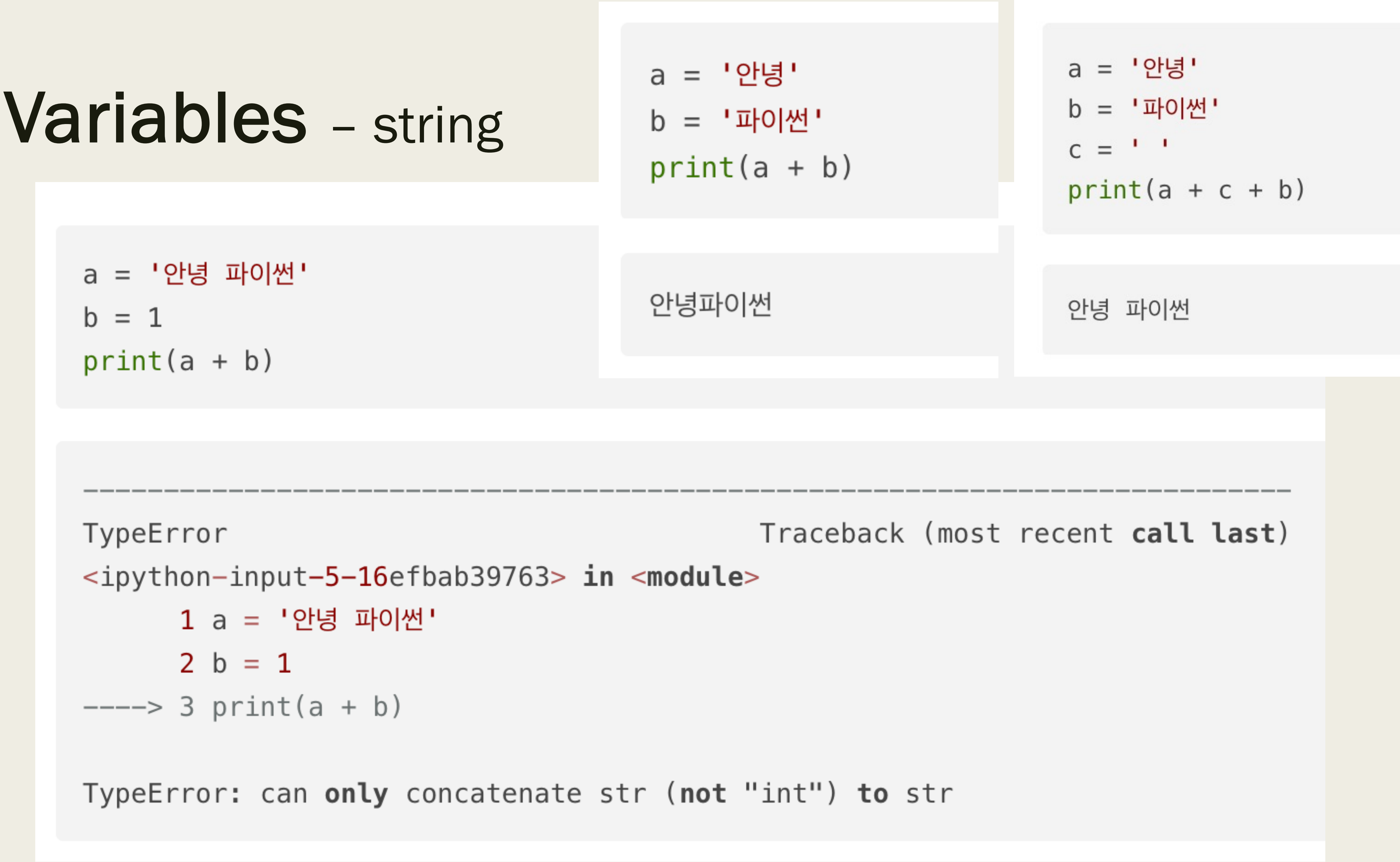
```
<class 'str'>
```



```
a = 1
b = '1'
c = 1.0
d = "1.0"

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

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

Variables – string



```
a = '안녕 파이썬'
b = 1
print(a + b)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-5-16efbab39763> in <module>
      1 a = '안녕 파이썬'
      2 b = 1
----> 3 print(a + b)

TypeError: can only concatenate str (not "int") to str
```

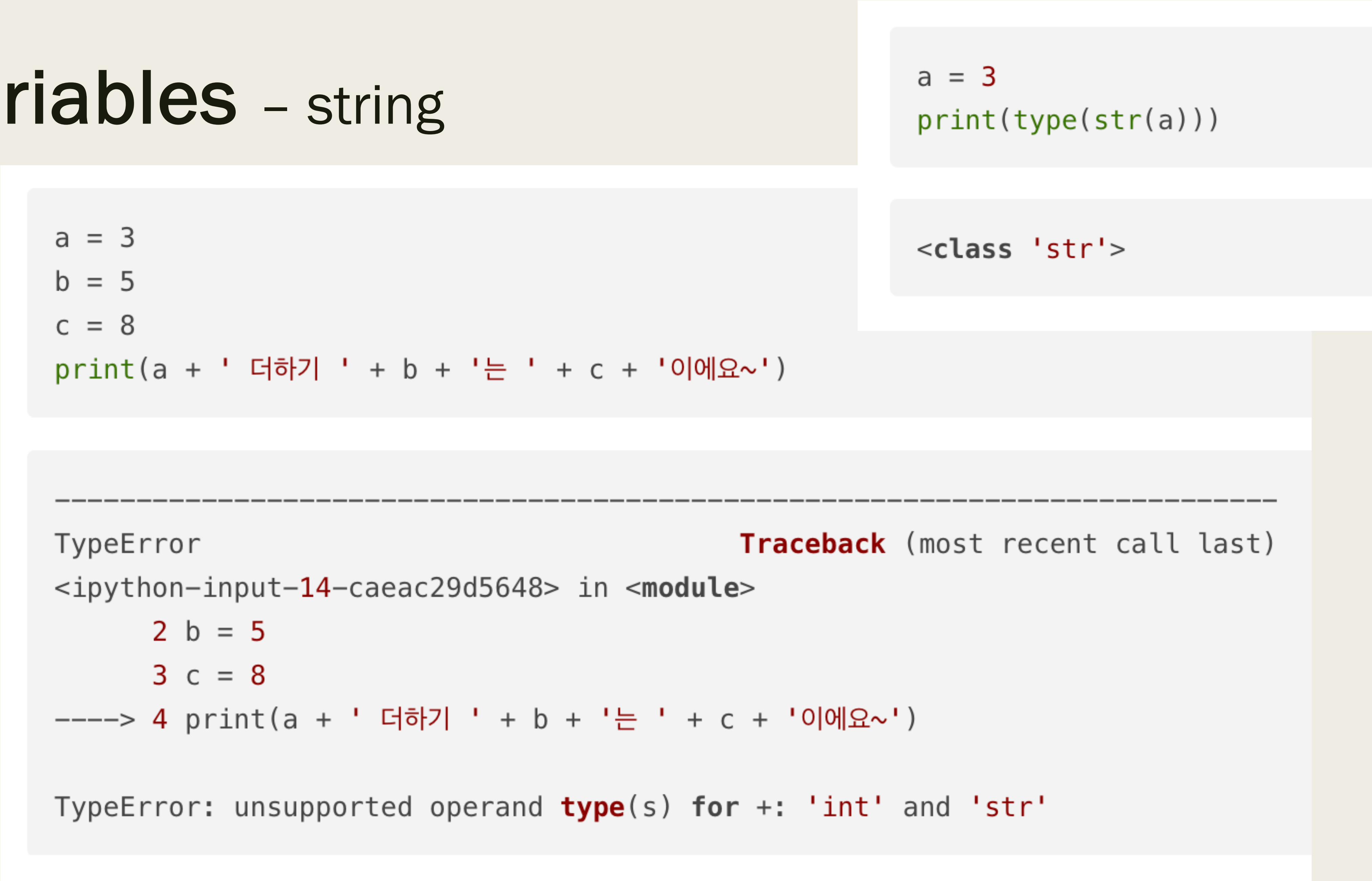
```
a = '안녕'
b = '파이썬'
print(a + b)
```

```
안녕파이썬
```

```
a = '안녕'
b = '파이썬'
c = ' '
print(a + c + b)
```

```
안녕 파이썬
```

Variables – string



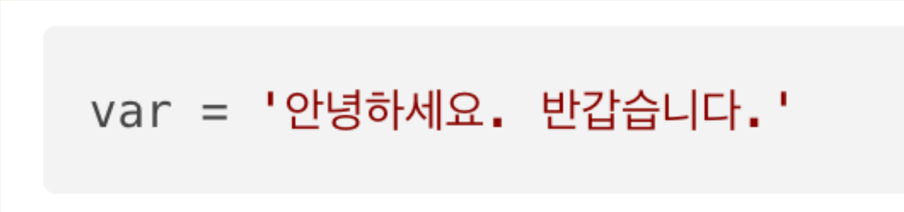
```
a = 3
b = 5
c = 8
print(a + ' 더하기 ' + b + '는 ' + c + '이에요~')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-14-caeac29d5648> in <module>
      2 b = 5
      3 c = 8
----> 4 print(a + ' 더하기 ' + b + '는 ' + c + '이에요~')

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

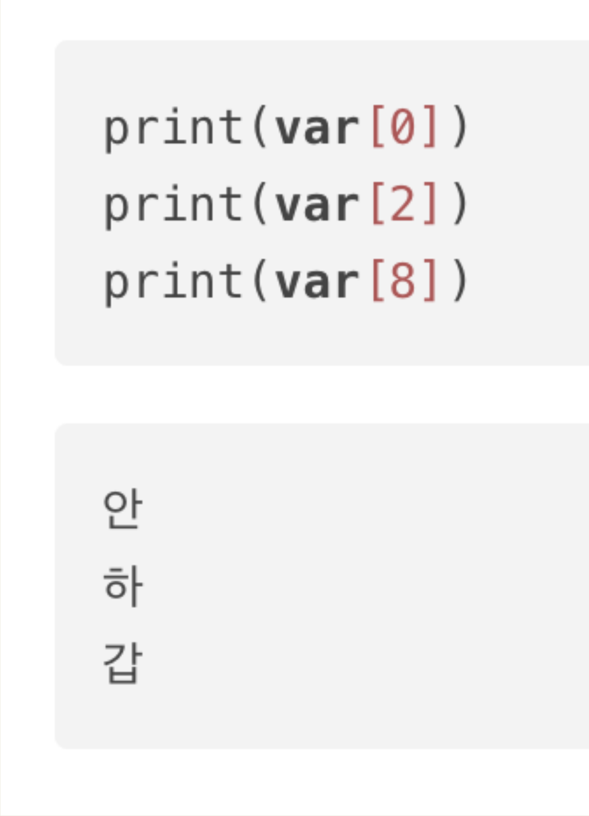
```
a = 3
print(type(str(a)))
```

```
<class 'str'>
```

Variables – string

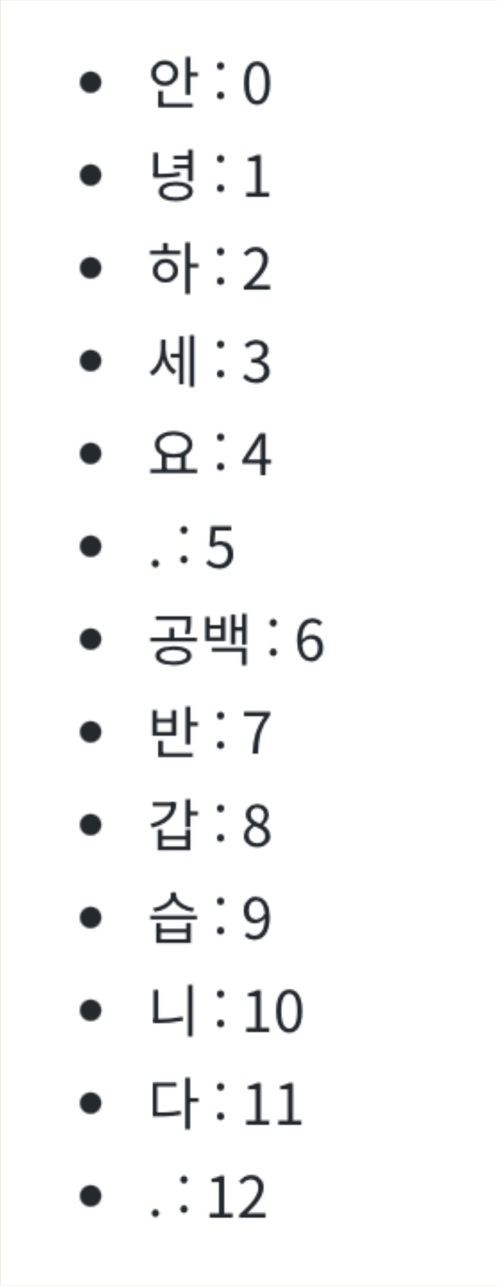


```
var = '안녕하세요. 반갑습니다.'
```



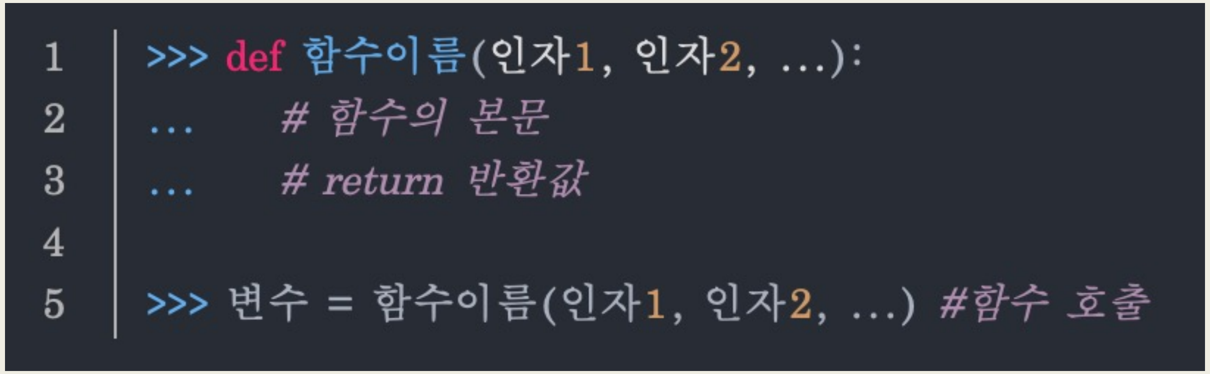
```
print(var[0])
print(var[2])
print(var[8])
```

```
안
하
갑
```

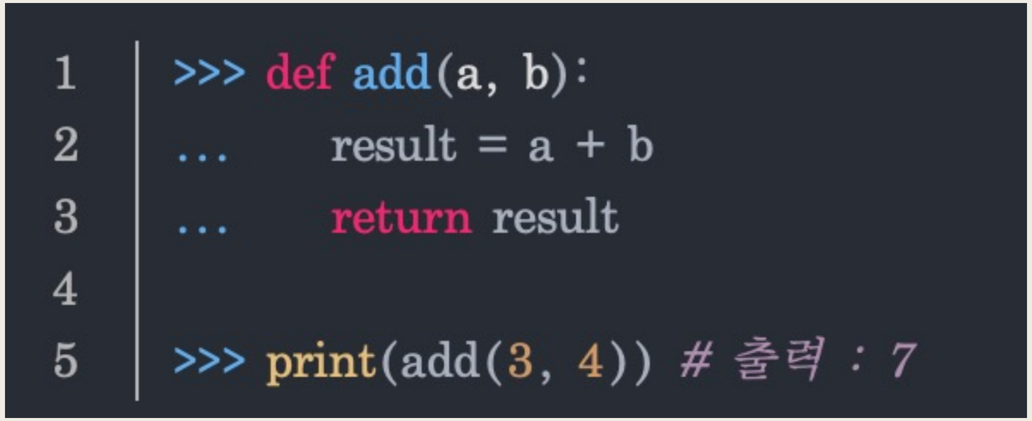


- 안:0
- 녕:1
- 하:2
- 세:3
- 요:4
- .:5
- 공백:6
- 반:7
- 갑:8
- 습:9
- 니:10
- 다:11
- .:12

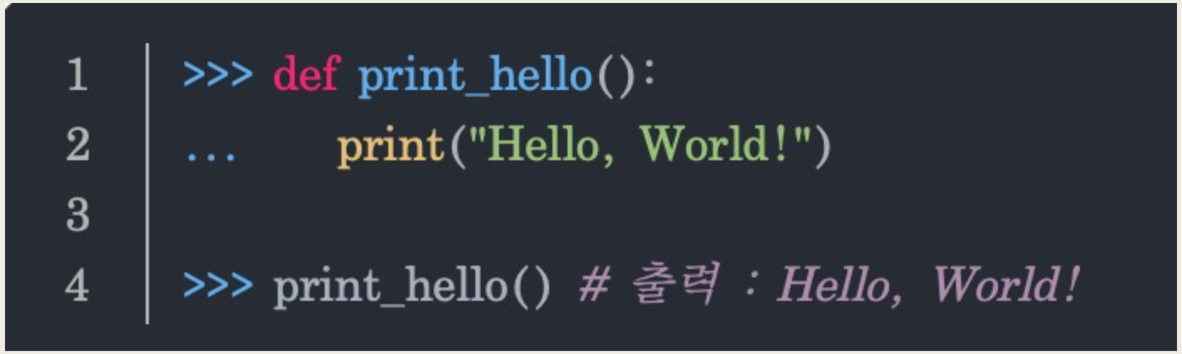
Function – declaration



```
>>> def 함수이름(인자1, 인자2, ...):
...     # 함수의 본문
...     # return 반환값

>>> 변수 = 함수이름(인자1, 인자2, ...) #함수 호출
```

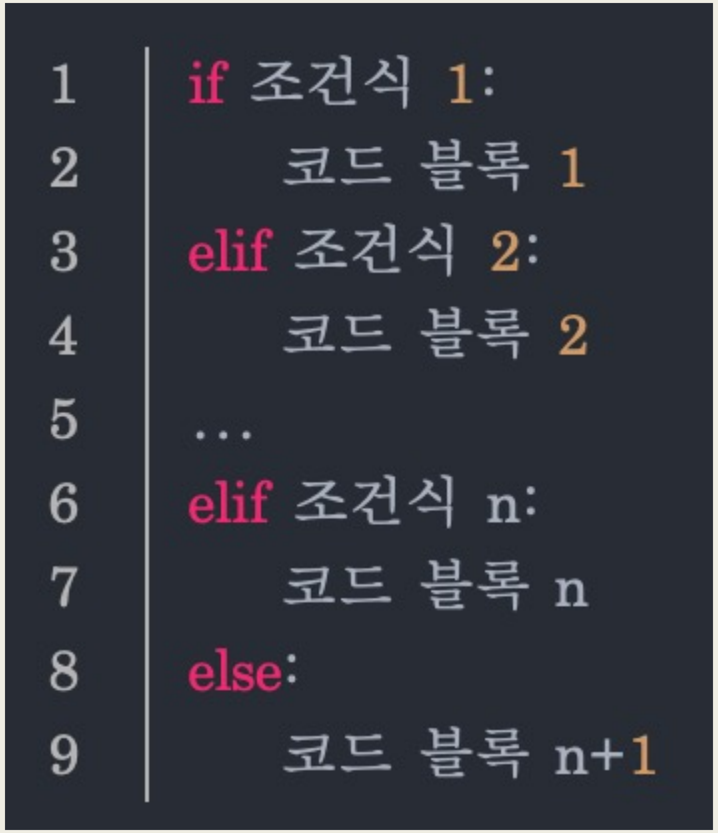


```
>>> def add(a, b):
...     result = a + b
...     return result

>>> print(add(3, 4)) # 출력 : 7
```

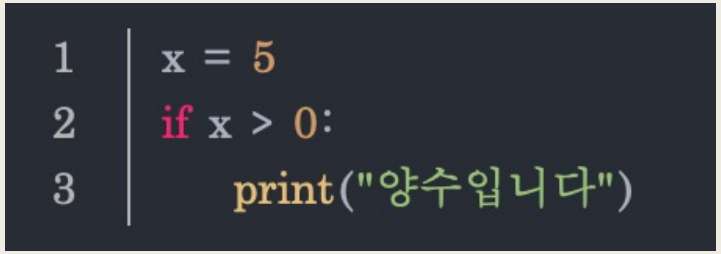


```
>>> def print_hello():
...     print("Hello, World!")

>>> print_hello() # 출력 : Hello, World!
```

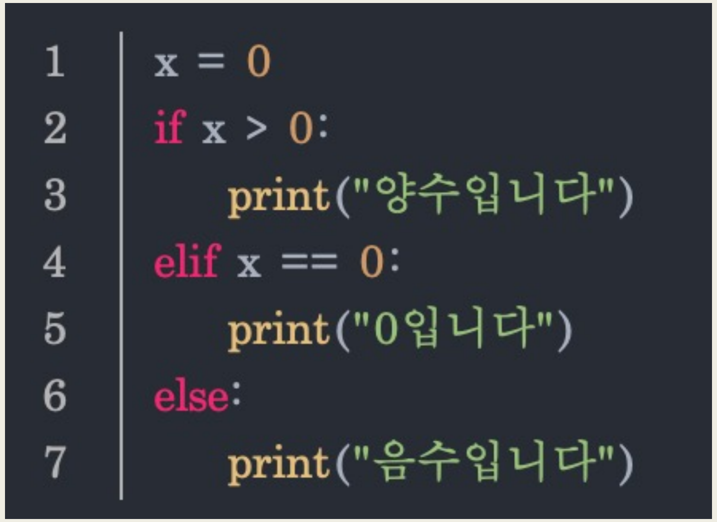
IF – declaration



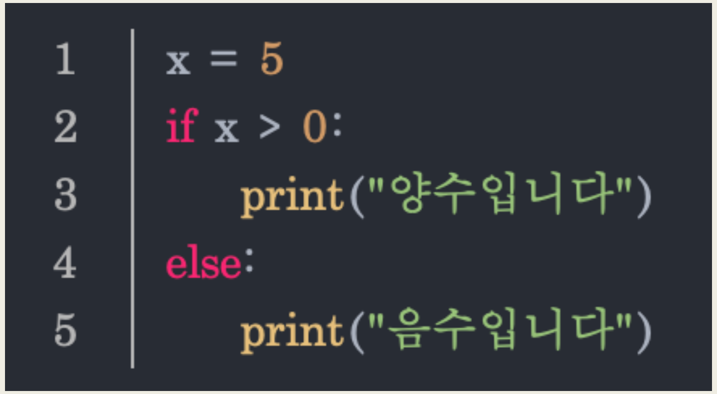
```
if 조건식 1:
    코드 블록 1
elif 조건식 2:
    코드 블록 2
...
elif 조건식 n:
    코드 블록 n
else:
    코드 블록 n+1
```



```
x = 5
if x > 0:
    print("양수입니다")
```

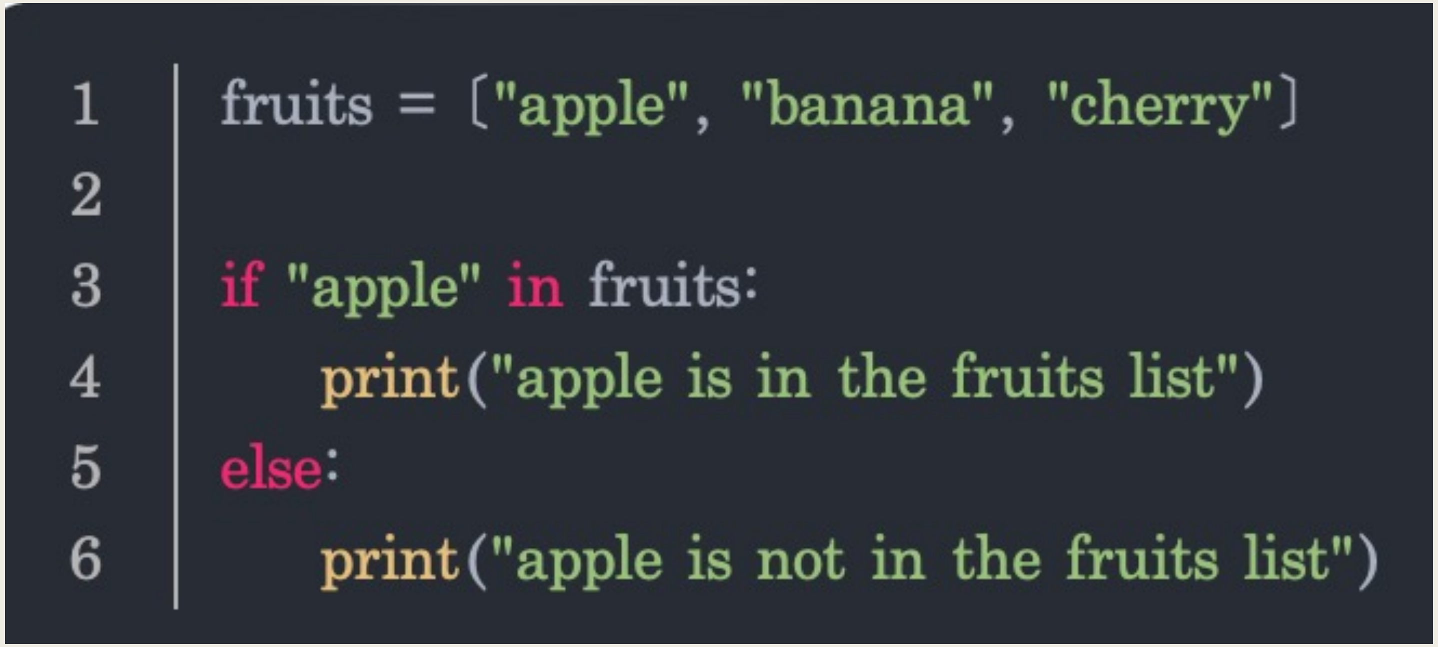


```
x = 0
if x > 0:
    print("양수입니다")
elif x == 0:
    print("0입니다")
else:
    print("음수입니다")
```



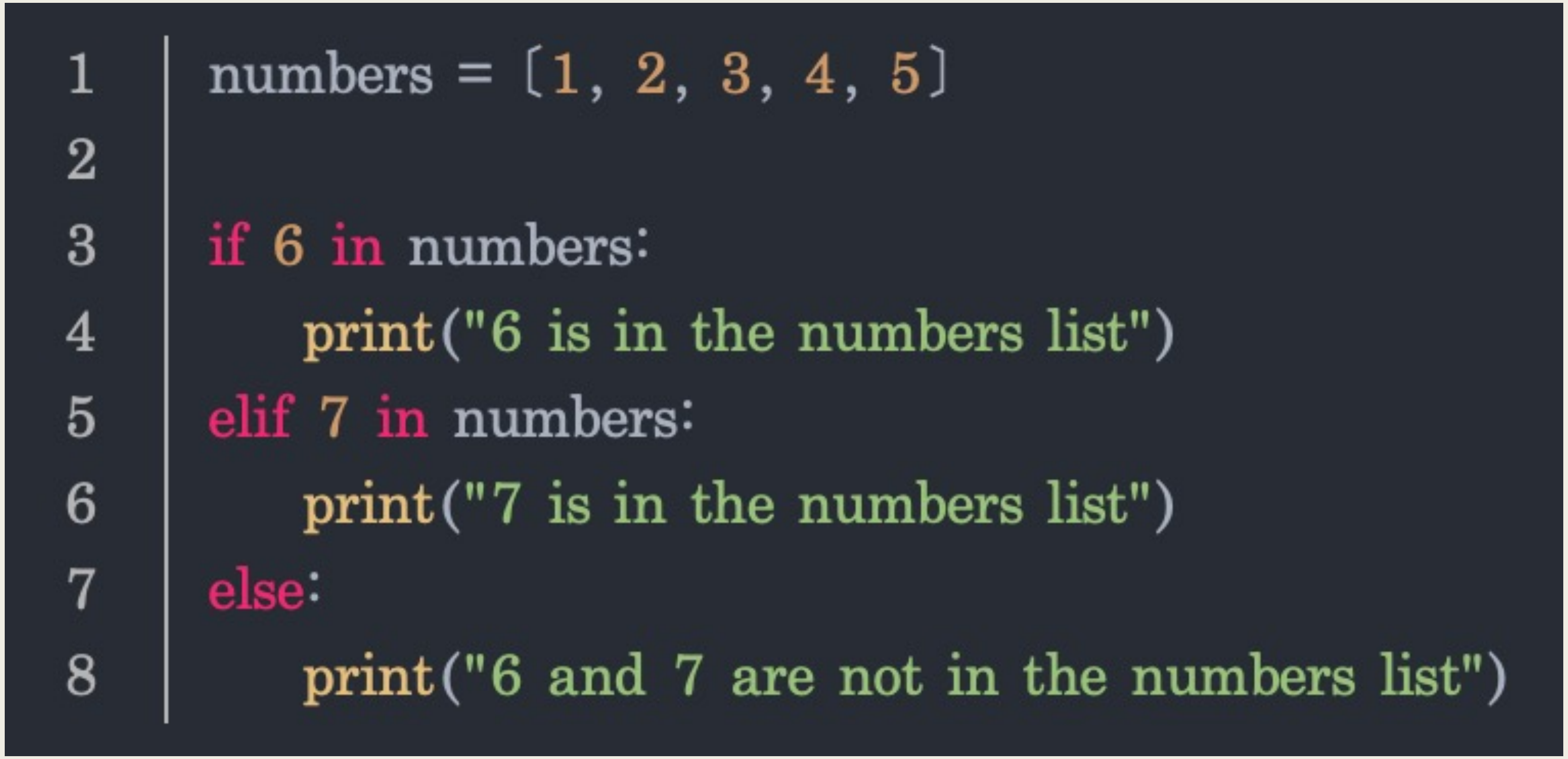
```
x = 5
if x > 0:
    print("양수입니다")
else:
    print("음수입니다")
```

IF – declaration



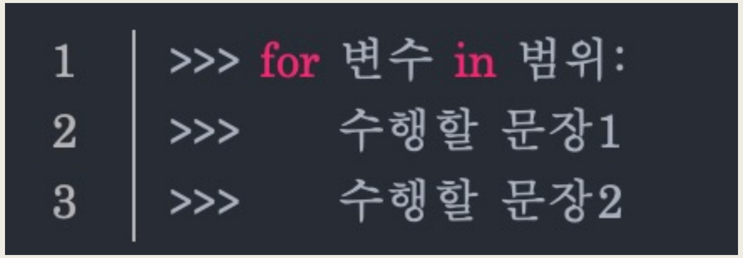
```python
fruits = ["apple", "banana", "cherry"]

if "apple" in fruits:
    print("apple is in the fruits list")
else:
    print("apple is not in the fruits list")
```

IF – declaration

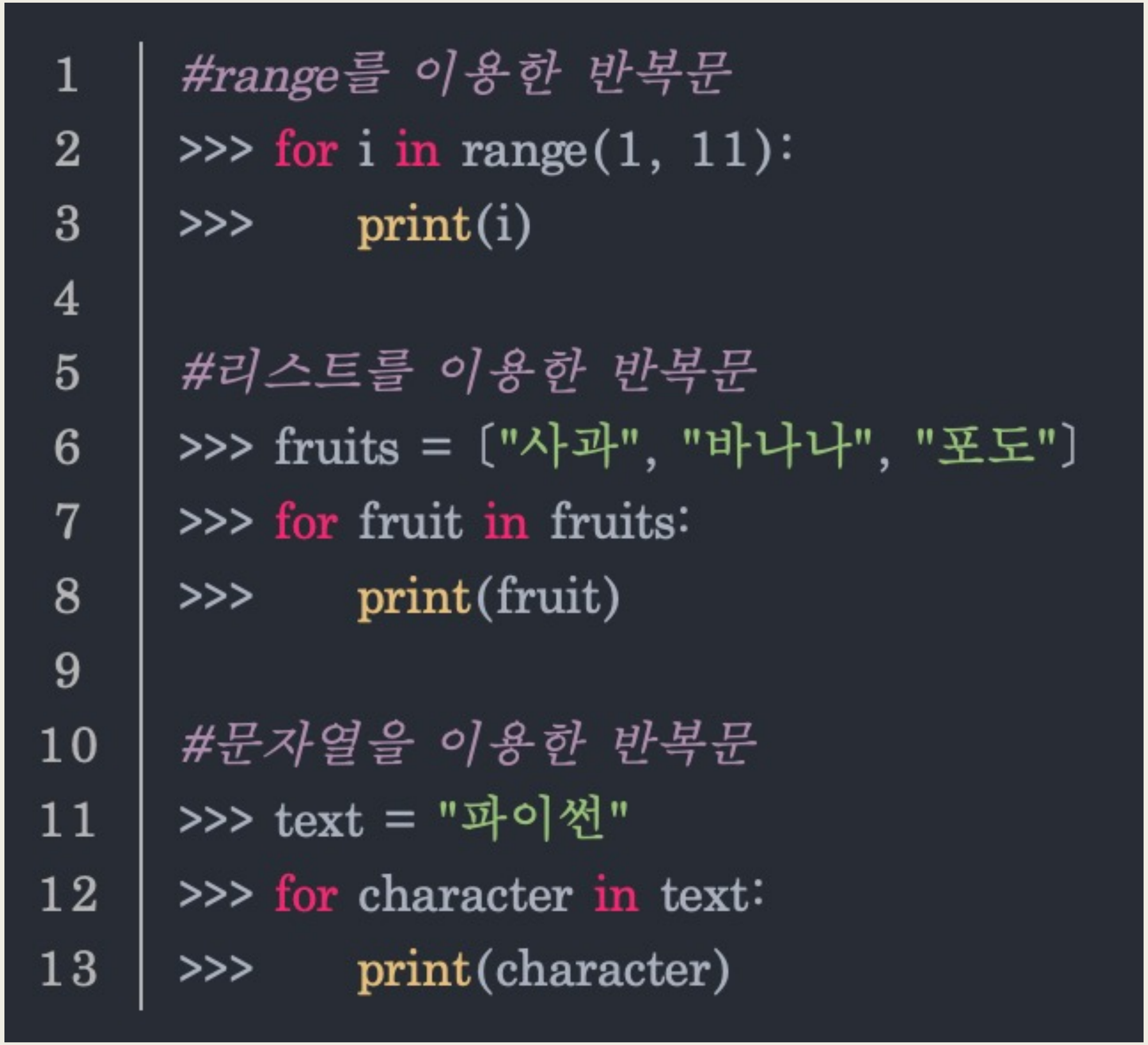


```python
numbers = [1, 2, 3, 4, 5]

if 6 in numbers:
    print("6 is in the numbers list")
elif 7 in numbers:
    print("7 is in the numbers list")
else:
    print("6 and 7 are not in the numbers list")
```

FOR – declaration

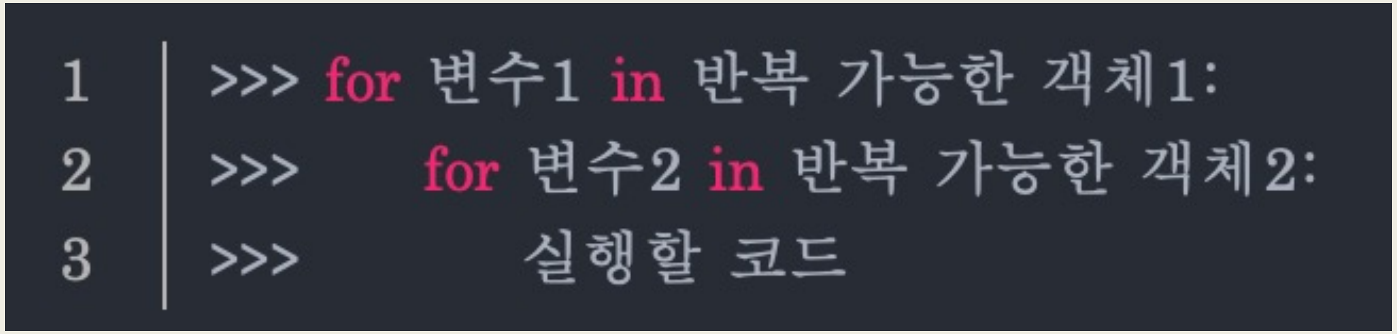


```
>>> for 변수 in 범위:
>>>     수행할 문장1
>>>     수행할 문장2
```

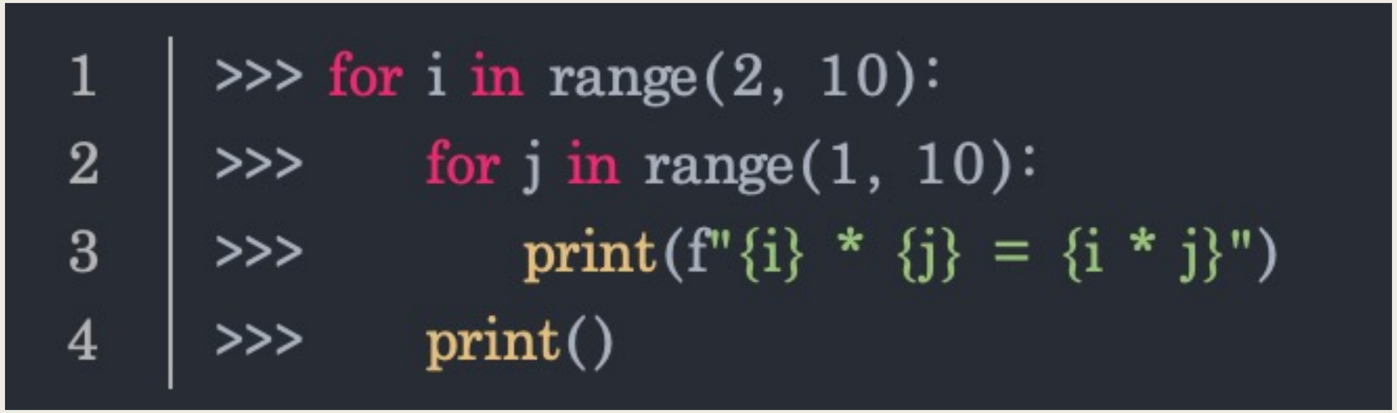


```
#range를 이용한 반복문
>>> for i in range(1, 11):
>>>     print(i)

#리스트를 이용한 반복문
>>> fruits = ["사과", "바나나", "포도"]
>>> for fruit in fruits:
>>>     print(fruit)

#문자열을 이용한 반복문
>>> text = "파이썬"
>>> for character in text:
>>>     print(character)
```

FOR – declaration

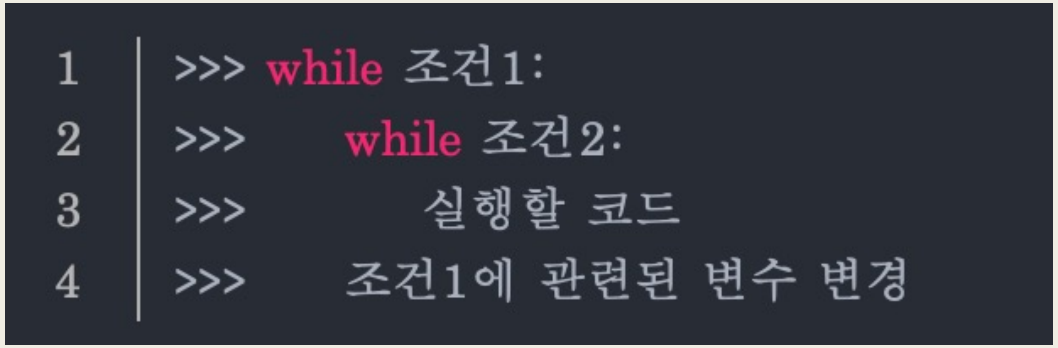


```
>>> for 변수1 in 반복 가능한 객체1:
>>>     for 변수2 in 반복 가능한 객체2:
>>>         실행할 코드
```

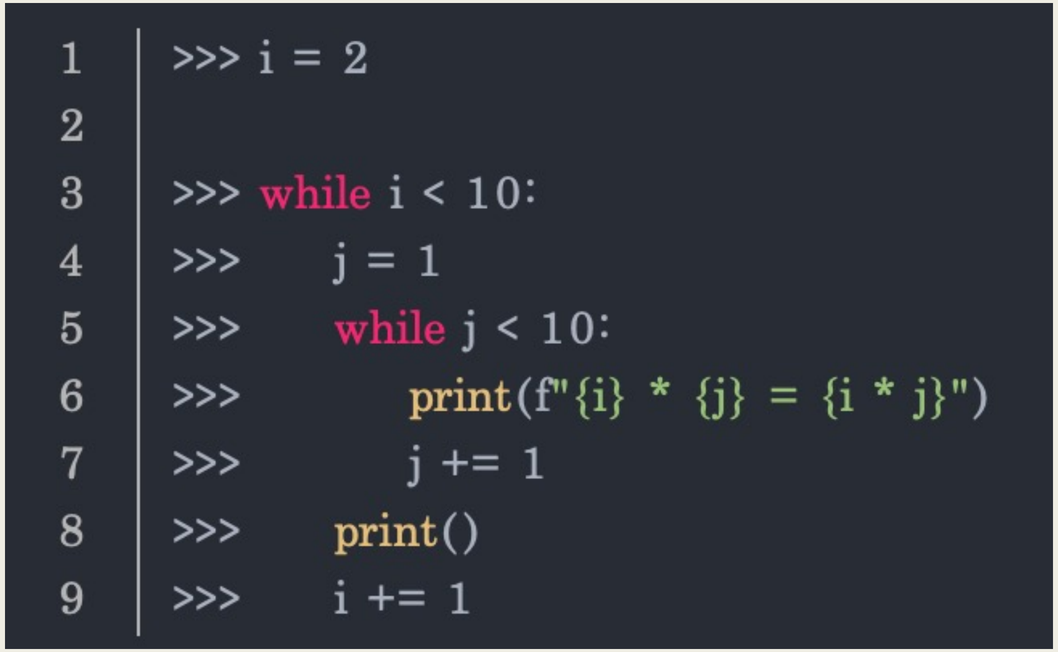


```
>>> for i in range(2, 10):
>>>     for j in range(1, 10):
>>>         print(f"{i} * {j} = {i * j}")
>>>     print()
```

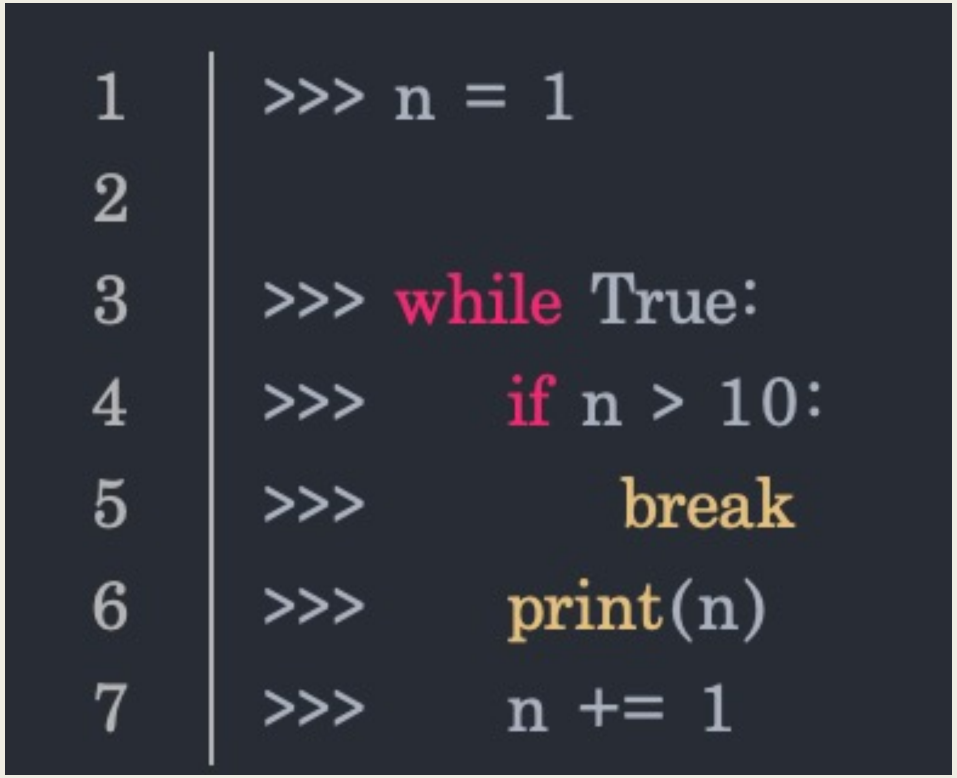
WHILE – declaration



```
>>> while 조건1:
>>>     while 조건2:
>>>         실행할 코드
>>>     조건1에 관련된 변수 변경
```



```
>>> i = 2

>>> while i < 10:
>>>     j = 1
>>>     while j < 10:
>>>         print(f"{i} * {j} = {i * j}")
>>>         j += 1
>>>     print()
>>>     i += 1
```



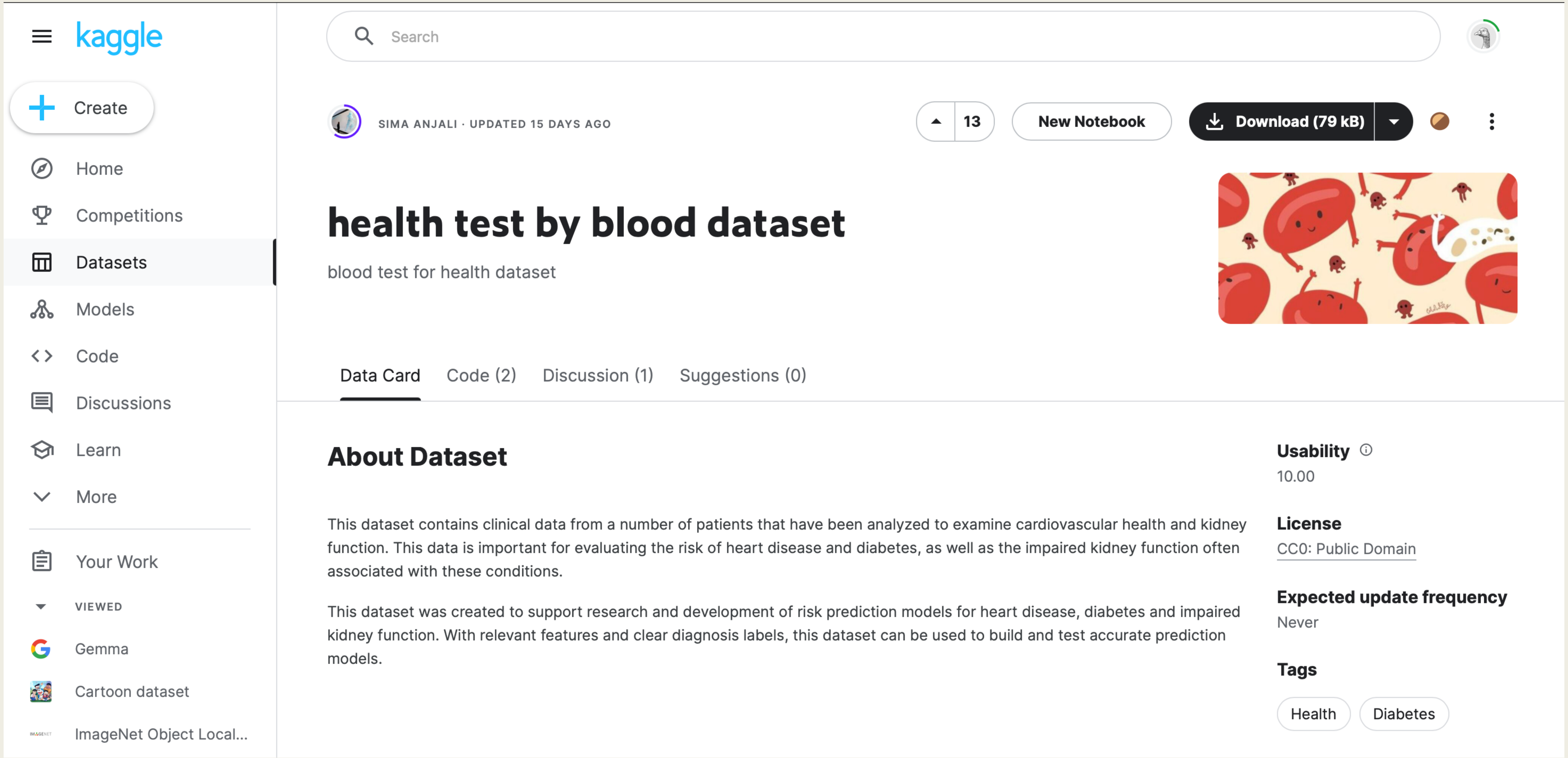
```
>>> n = 1

>>> while True:
>>>     if n > 10:
>>>         break
>>>     print(n)
>>>     n += 1
```

# Assignment 1 – python basic

Warm-up Issues: Find multiple of 3

- Variable
- If-else
- for

# Assignment 2 – blood data analysis



SIMA ANJALI · UPDATED 15 DAYS AGO

health test by blood dataset

blood test for health dataset

About Dataset

This dataset contains clinical data from a number of patients that have been analyzed to examine cardiovascular health and kidney function. This data is important for evaluating the risk of heart disease and diabetes, as well as the impaired kidney function often associated with these conditions.

This dataset was created to support research and development of risk prediction models for heart disease, diabetes and impaired kidney function. With relevant features and clear diagnosis labels, this dataset can be used to build and test accurate prediction models.

Usability
10.00

License
CC0: Public Domain

Expected update frequency
Never

Tags
Health
Diabetes