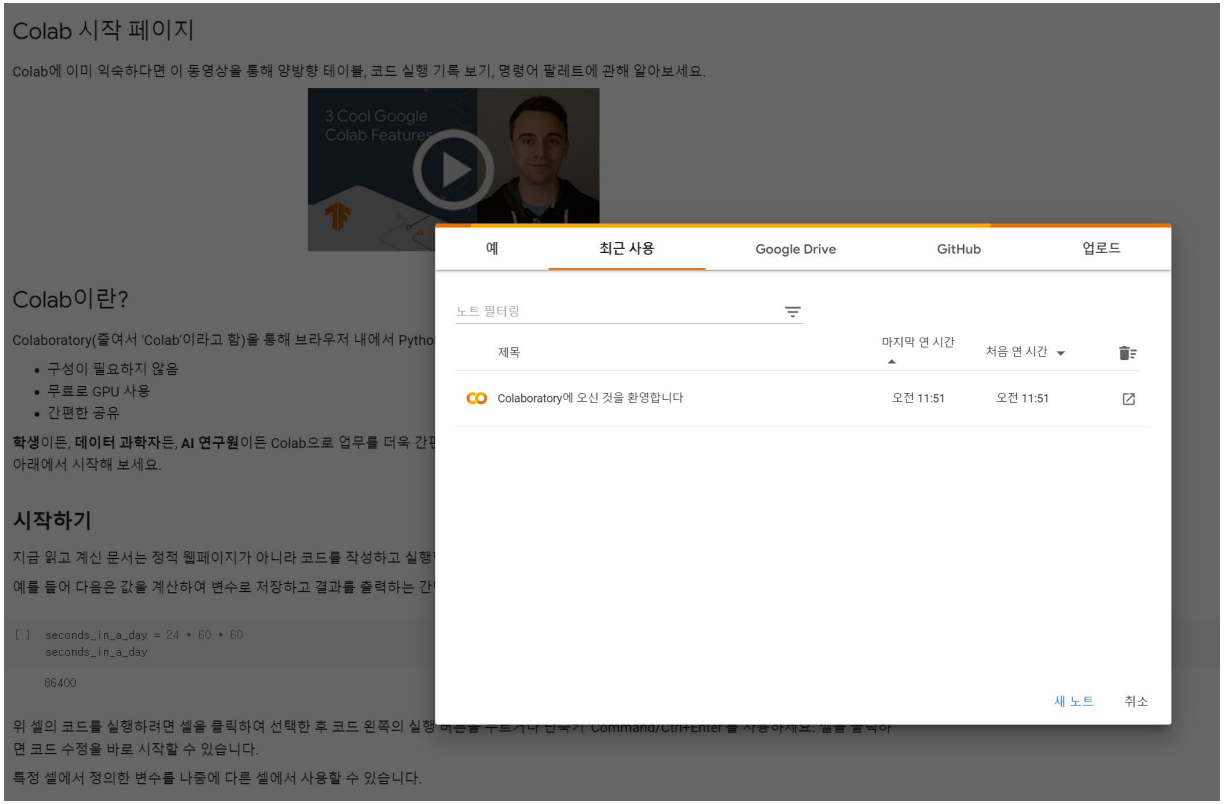
Lab1: Conditionals and loops

Prof. Jonghee Yoon (jyoon48@ajou.ac.kr)

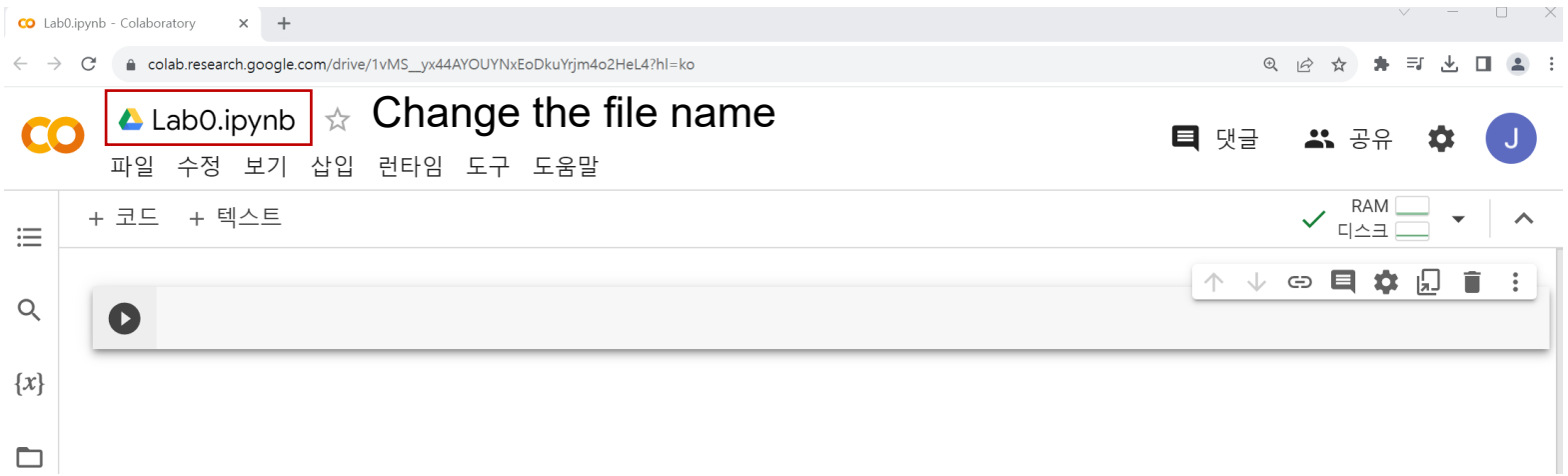
Ajou University, Department of Physics

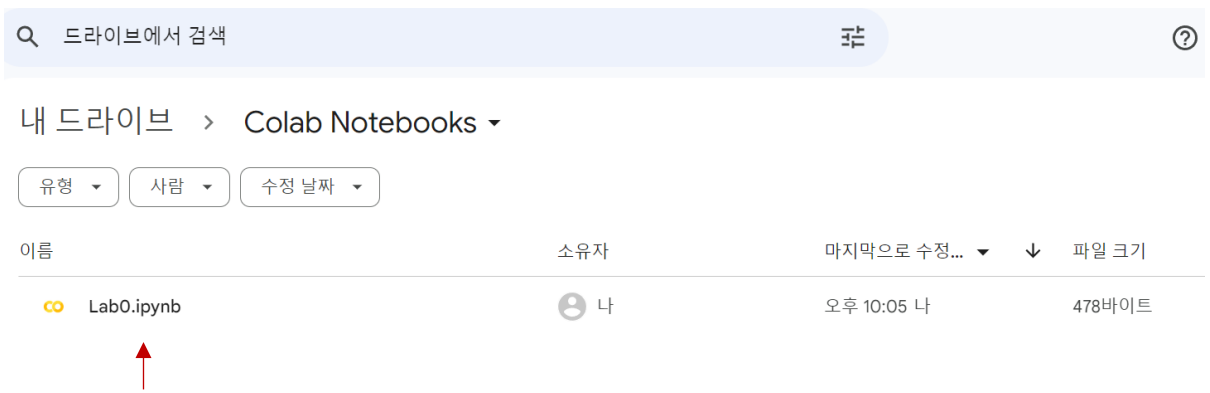
Using Google Colab for Python programming

- Google Colaboratory, or "Colab" as most people call it, is a cloud-based Jupyter notebook environment.
- It runs in your web browser.
- Colab is free of charge to use.

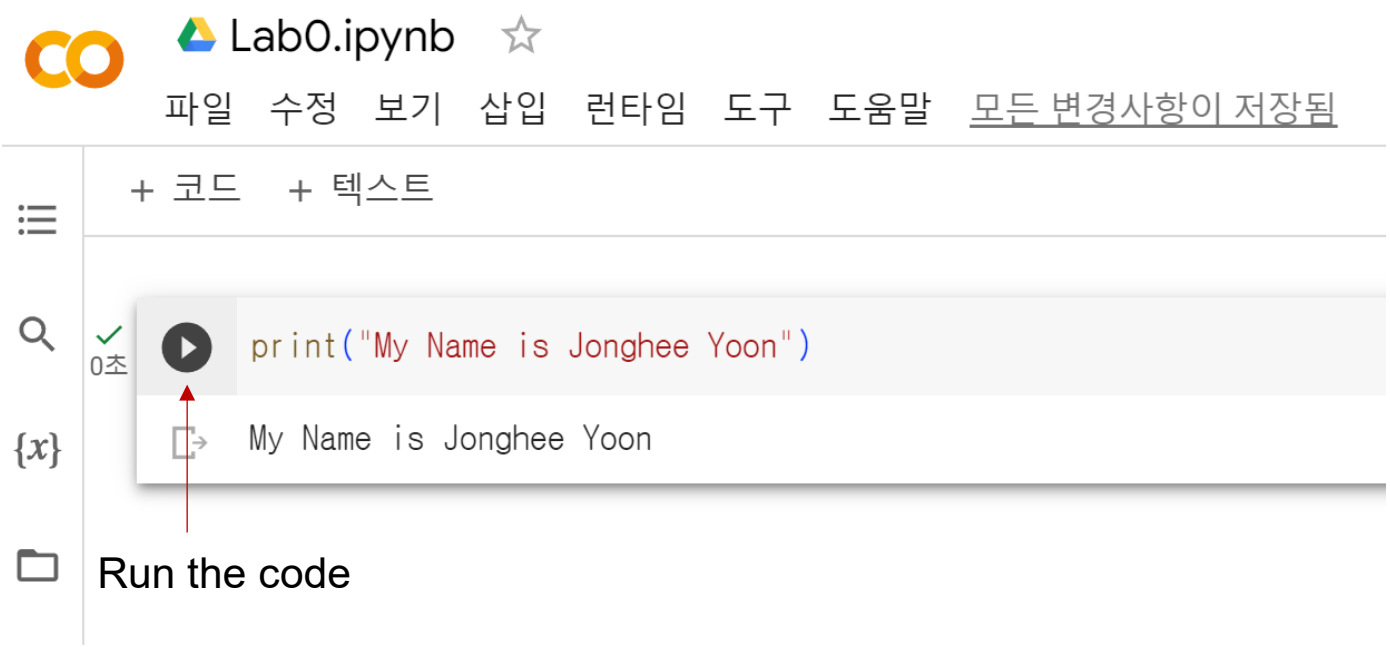


Using Google Colab for Python programming





File you can submit


Lab0.ipynb
파일 수정 보기 삽입 런타임 도구 도움말 모든 변경사항이 저장됨
+ 코드 + 텍스트
0초
print("My Name is Jonghee Yoon")
My Name is Jonghee Yoon

Run the code

Using Google Colab for Python programming

→ comment (not executable)

Check points

- Implement codes to get the following results

Result1

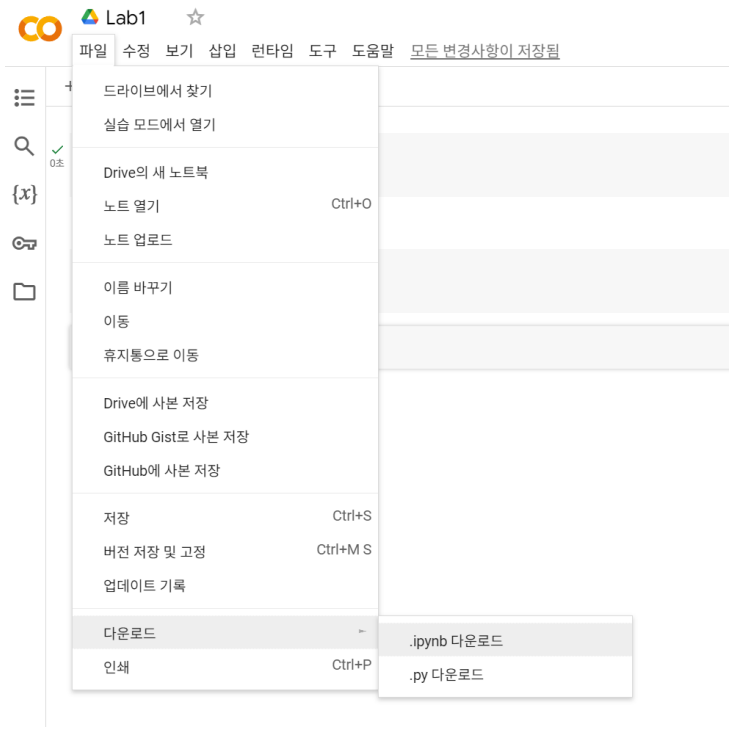
```
My Name is Your name
My student number is Your student number
```

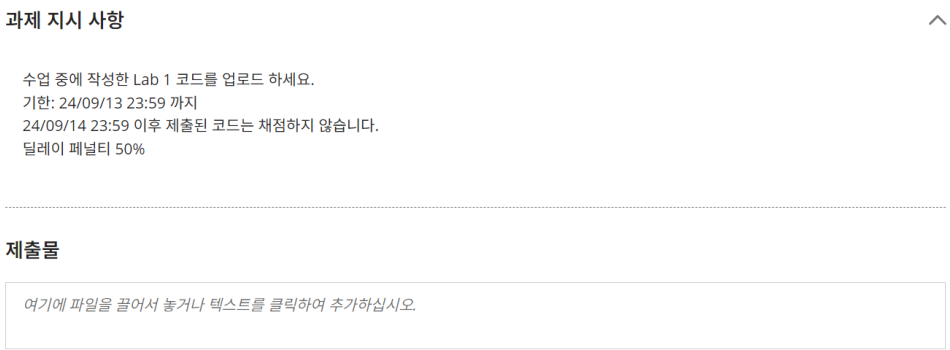
Result2

```
0.5 * 9.8 * 3^2 =
44.1
```

Submit the code

- Download your code at Colab
- Upload your code to AjouBB





What are variables?

- Variables are containers for data values.

  f(x)=3x+5, y=1/x, F=ma

- A variable is created at the moment you first assign a value to it (memory address).

```
78    print(x)

Exception has occurred: NameError
name 'x' is not defined
```

```
x=3
print("A value of X is",x)
x=7
print("A value of X is",x)
```

```
A value of X is 3
A value of X is 7
```

- Arithmetic operators take numerical values (either literals or variables) as their operands and return a single numerical value.

```
x=2
y=5
print("x is",x,"y is",y)
print("x+y=",x+y,"x-y=",x-y,"x*y=",x*y,"x/y=",x/y)
```

```
x is 2 y is 5
x+y= 7 x-y= -3 x*y= 10 x/y= 0.4
```

What are data types?

- There are many data types in Python

- Text type (letters): str
- Numeric type (numbers): int, float, complex
- Boolean type (true, false) : bool

```python
x = "Ajou"
print(x)
print(x+" University")
```

```
Ajou
Ajou University
```

```python
x = 3
print(x)
print(x+5)
```

```
3
8
```

Task0: Print your name, date, lab number, and data types

- Print your name, date, lab number, and data types using variables

```
# Task 0
name =
class_name =
lab_num = 1
print("My name is ",      )
print("This class is ",          )
print("Lab number is",        )
print(    (name))
print(    (class_name))
print(    (lab_num))
```

```
My name is  Jonghee Yoon
This class is  AI Physics
Lab number is 1
<class 'str'>
<class 'str'>
<class 'int'>
```

Task1: Motion under Constant Acceleration

- Calculate the velocity and distance at a certain time *t*

표 2.2 등가속도 운동을 하는 입자의 운동학 방정식

식	방정식	방정식에 표시된 정보
2.13	$v_{xf} = v_{xi} + a_x t$	시간의 함수로 나타낸 속도
2.15	$x_f = x_i + \frac{1}{2}(v_{xi} + v_{xf})t$	속도와 시간의 함수로 나타낸 위치
2.16	$x_f = x_i + v_{xi}t + \frac{1}{2}a_x t^2$	시간의 함수로 나타낸 위치
2.17	$v_{xf}^2 = v_{xi}^2 + 2a_x(x_f - x_i)$	위치의 함수로 나타낸 속도

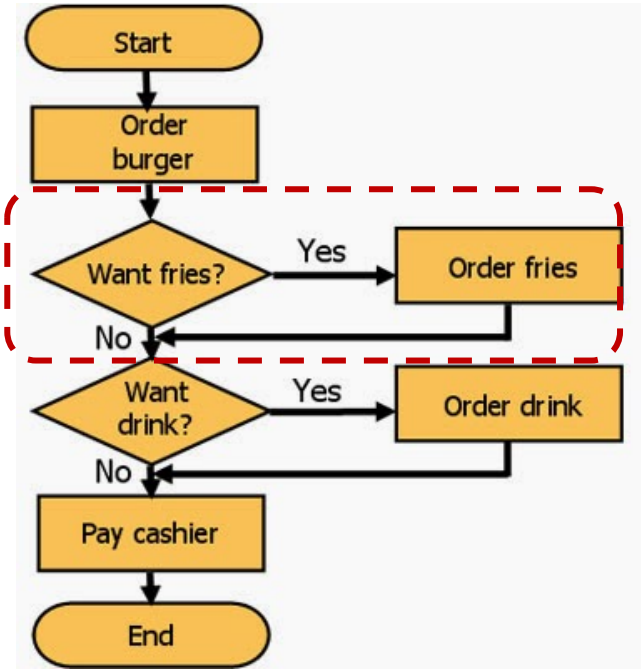
운동은 x 방향임.

```
# Task 1
v0 = 10 # m/s
d0 = -35 # m
a = 2 #m/s^2
t = 10 #s

v =
d =

print(                    )
print(                    )
```

```
Velocity at  10 s is  30 m/s
Distance at  10 s is  165.0 m/s
```

What are conditionals?

- What makes programming so much more powerful are conditional statements.
- A **condition** is either **True** or **False** (***Boolean***)



if (a condition evaluates to Ture):

then do these things only for '**True**'

e.g. Oder fries

else:

otherwise do these things only for '**False**'

e.g. Ask the next question

Conditional Examples

```
if True:
    print("Today is Wednesday")
if False:
    print("Every student will get A+!")
```

Run

```
Today is Wednesday
```

```
if 5>3:
    print("5 is larger than 3")
else:
    print("5 is smaller than 3")
```

Run

```
5 is larger than 3
```

not True is ***False***

```
print(5>3)
```

```
True
```

```
print(not 5>3)
```

```
False
```

```
if not 5>3:
    print("5 is larger than 3")
else:
    print("5 is smaller than 3")
```

Run

```
5 is smaller than 3
```

Comparison Operators

Operator	Description	Example
==	If values of two operands are equal, the condition becomes true	(3==3): True, (3==5): False
!=	If values of two operands are not equal, then the condition becomes true	(3!=3): False, (3!=5): True
>	If the value of left operand is greater than the value of right operand, then condition becomes true.	(3>5): False, (5>3): True
<	If the value of right operand is greater than the value of left operand, then condition becomes true	(3<5): True, (5>3): False
>=	If the value of left operand is greater than or equal to the value of right operand, then condition becomes true	(3>=3): True, (3>=5): False
<=	If the value of right operand is greater than or equal to the value of left operand, then condition becomes true	(3<=3): True, (5<=3): False

Multiple options (if & elif)

if Option1: *e.g. scores>90*

then do Task1 *e.g. get A*

elif Option2: *e.g. scores>80*

then do Task2 *e.g. get B*

elif Option3: *e.g. scores>70*

then do Task3 *e.g. get C*

elif Option4: *e.g. scores>60*

then do Task4 *e.g. get D*

else: *e.g. scores<=60*

then do Task5 *e.g. get F*

```python
score = 75
if score>90:
    print("Grade is A")
elif score>80:
    print("Grade is B")
elif score>70:
    print("Grade is C")
elif score>60:
    print("Grade is D")
else:
    print("Grade is F")
```

```
Grade is C
```

```python
score = 65
if score>90:
    print("Grade is A")
elif score>80:
    print("Grade is B")
elif score>70:
    print("Grade is C")
elif score>60:
    print("Grade is D")
else:
    print("Grade is F")
```

```
Grade is D
```

Task2: Grading program based on an average score

- Write a code for a letter grade based on an average score
- A (average score >=90), B(average score >=80), C(average score >=70), D

```
math = 70
english = 57
biology = 83
physics = 93

average_score =

print(                    )
if
    print("Grade is A")

    print("Grade is B")

    print("Grade is C")

    print("Gradie is D")
```

```
Average score is 75.75
Grade is C
```

for loop

- The *range()* function enables the iteration of a set of code a specified number of times
- The *range()* function returns a sequence of numbers, starting from 0 (by default), and increments by 1(by default), and ends at a specified number

```
for x in range(3):
    print("Repeat")
```

```
Repeat
Repeat
Repeat
```

```
for x in range(5):
    print("Repeat")
```

```
Repeat
Repeat
Repeat
Repeat
Repeat
```

```
for x in range(3):
    print(x)
```

```
0
1
2
```

```
for x in range(5):
    print(x)
```

```
0
1
2
3
4
```

```
for x in range(2,5):
    print(x)
```

```
2
3
4
```

```
for x in range(2,15,3):
    print(x)
```

```
2
5
8
11
14
```

while loop

- With the ***while*** loop we can execute a set of statements as long as a condition is true.

```
while True:
    print("........")
```

```
........
........
........
........
........
........
........
........
........
........
........
```

```
x=0
while x<10:
    print(x)
    x=x+1
```

```
0
1
2
3
4
5
6
7
8
9
```

- ‘Control + C’ could stop the loop

Task3: Printing power of numbers using for or while loops

- Make a code for printing power (1 – 10) according to the predefined value
- Use pow() function (e.g. pow(2,3)=2^3=8)
- Use for or while loops

```
input_value = 4
for x in [redacted]
    print [redacted]
```

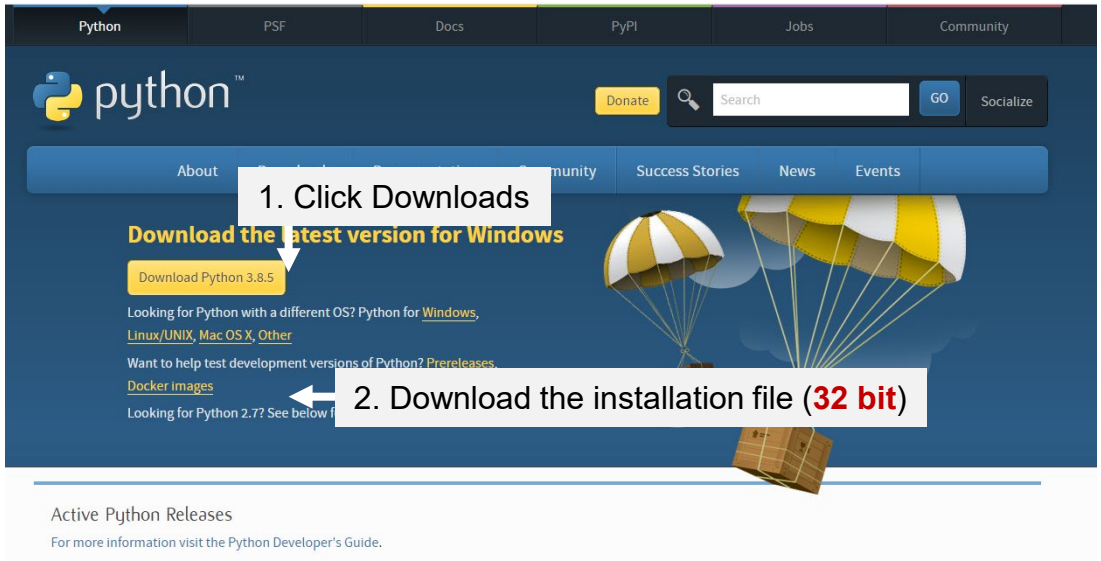
```
1 ^ 4 is 1
2 ^ 4 is 16
3 ^ 4 is 81
4 ^ 4 is 256
5 ^ 4 is 625
6 ^ 4 is 1296
7 ^ 4 is 2401
8 ^ 4 is 4096
9 ^ 4 is 6561
10 ^ 4 is 10000
```

```
input_value = 2
x = 0
while [redacted]
    [redacted]
    print [redacted]
```

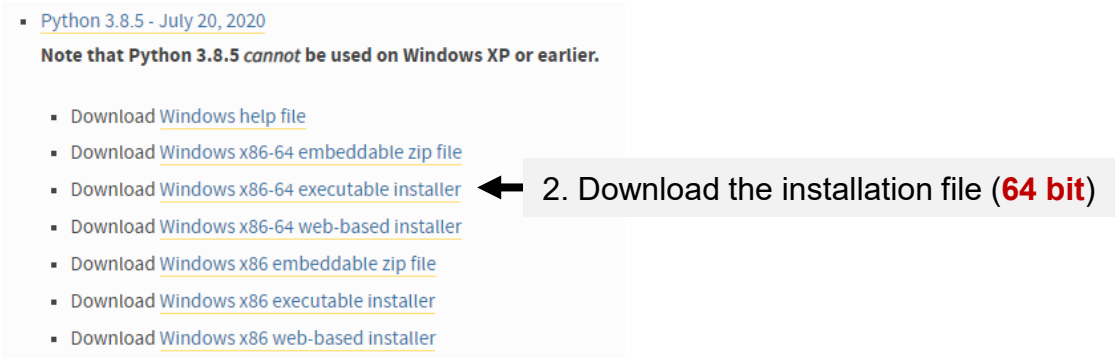
```
1 ^ 2 is 1
2 ^ 2 is 4
3 ^ 2 is 9
4 ^ 2 is 16
5 ^ 2 is 25
6 ^ 2 is 36
7 ^ 2 is 49
8 ^ 2 is 64
9 ^ 2 is 81
10 ^ 2 is 100
```

Install Python

Downloads Python: https://www.python.org/

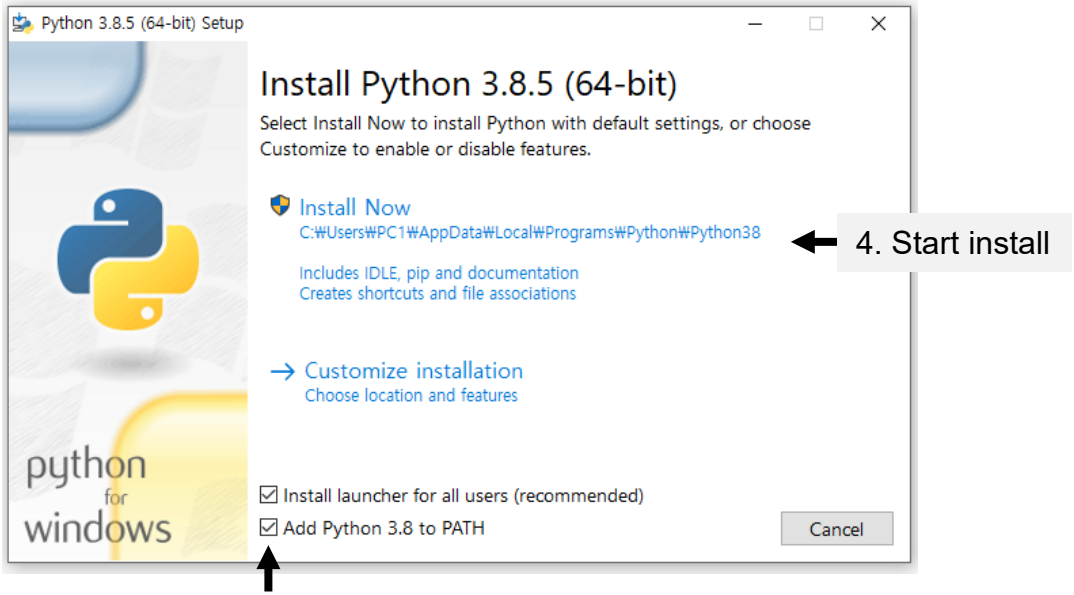


If your computer supports 64 bit, then download *'Windows x86-64 executable installer'*

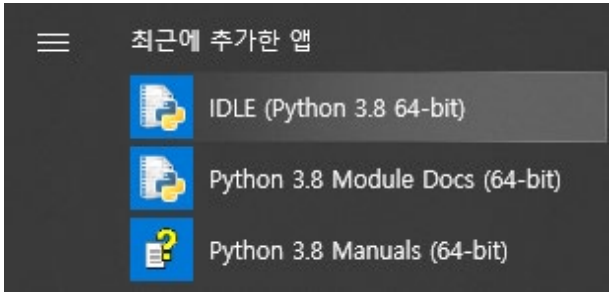


Install Python

Run the installation file

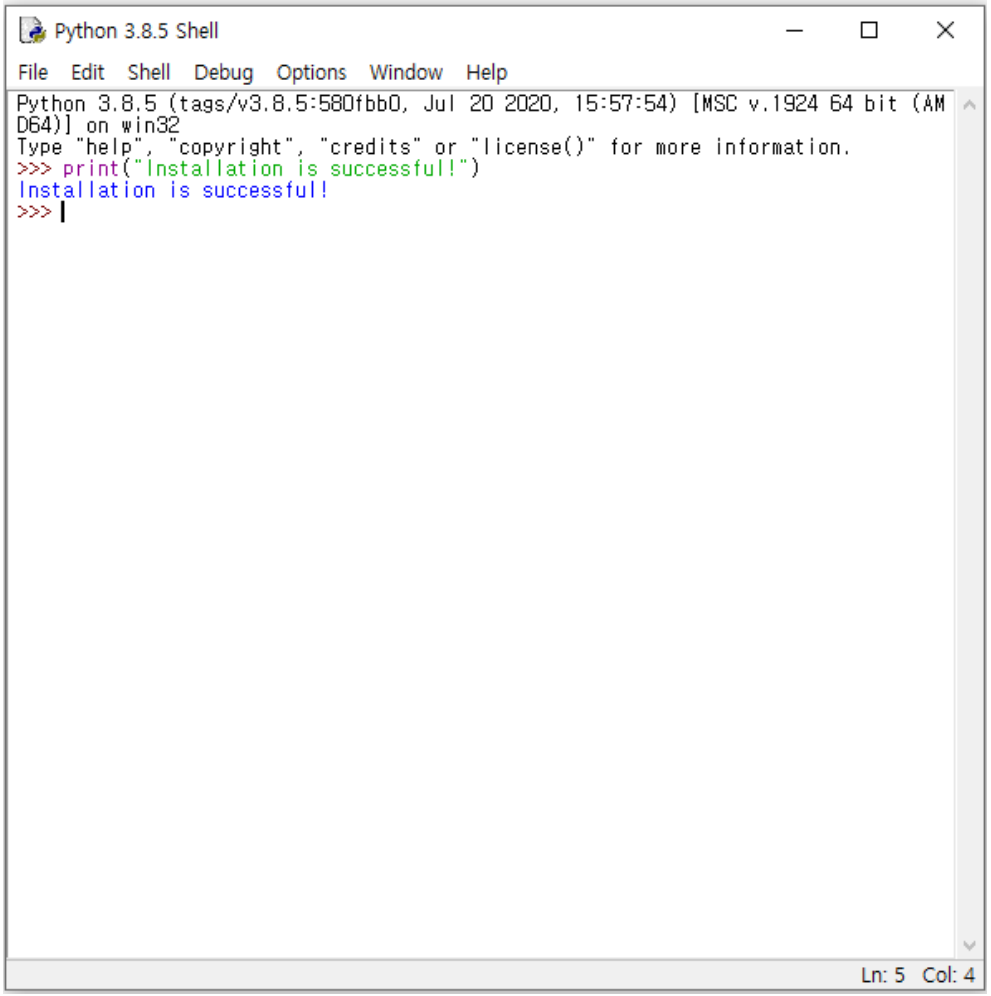


Run IDLE (Python 3.8)



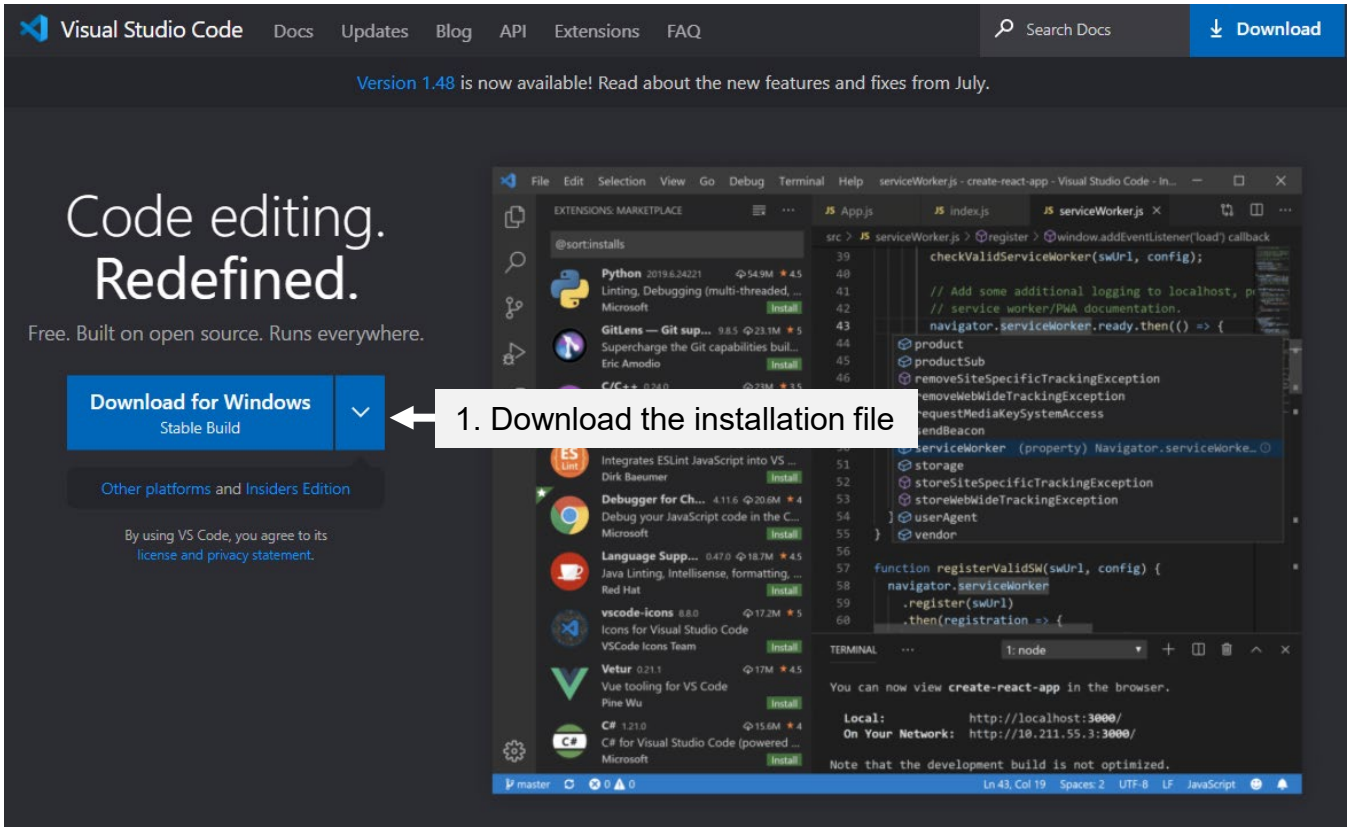
Install Python

If you can see this window, then the installation of Python is successful.
Type *print("Installation is successful!")* in prompt (>>>), then '*Installation is successful!*' will be printed.



Install Text Editor

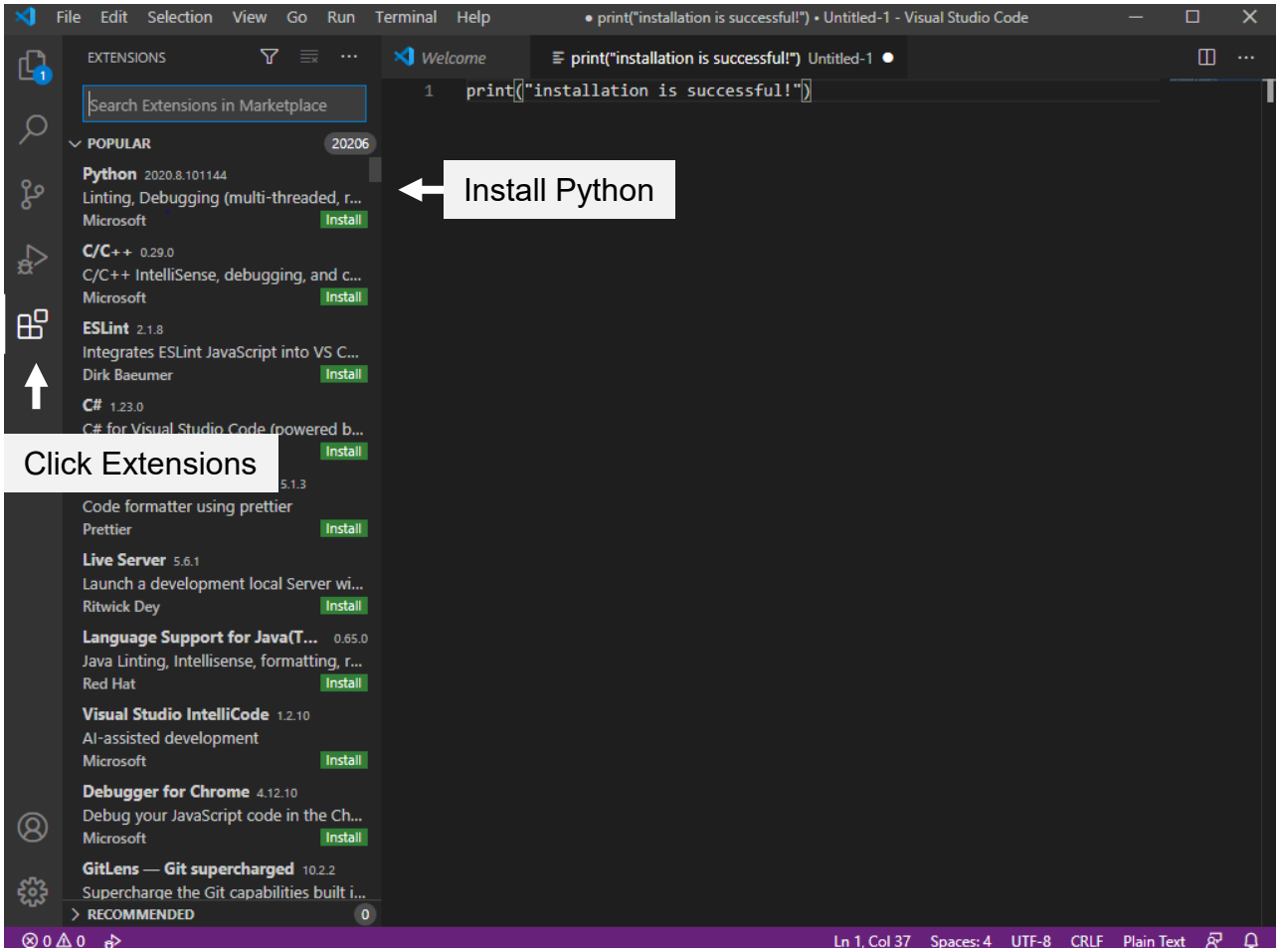
Download Visual Studio Code: https://code.visualstudio.com/



Install Visual Studio Code by running the installation file.

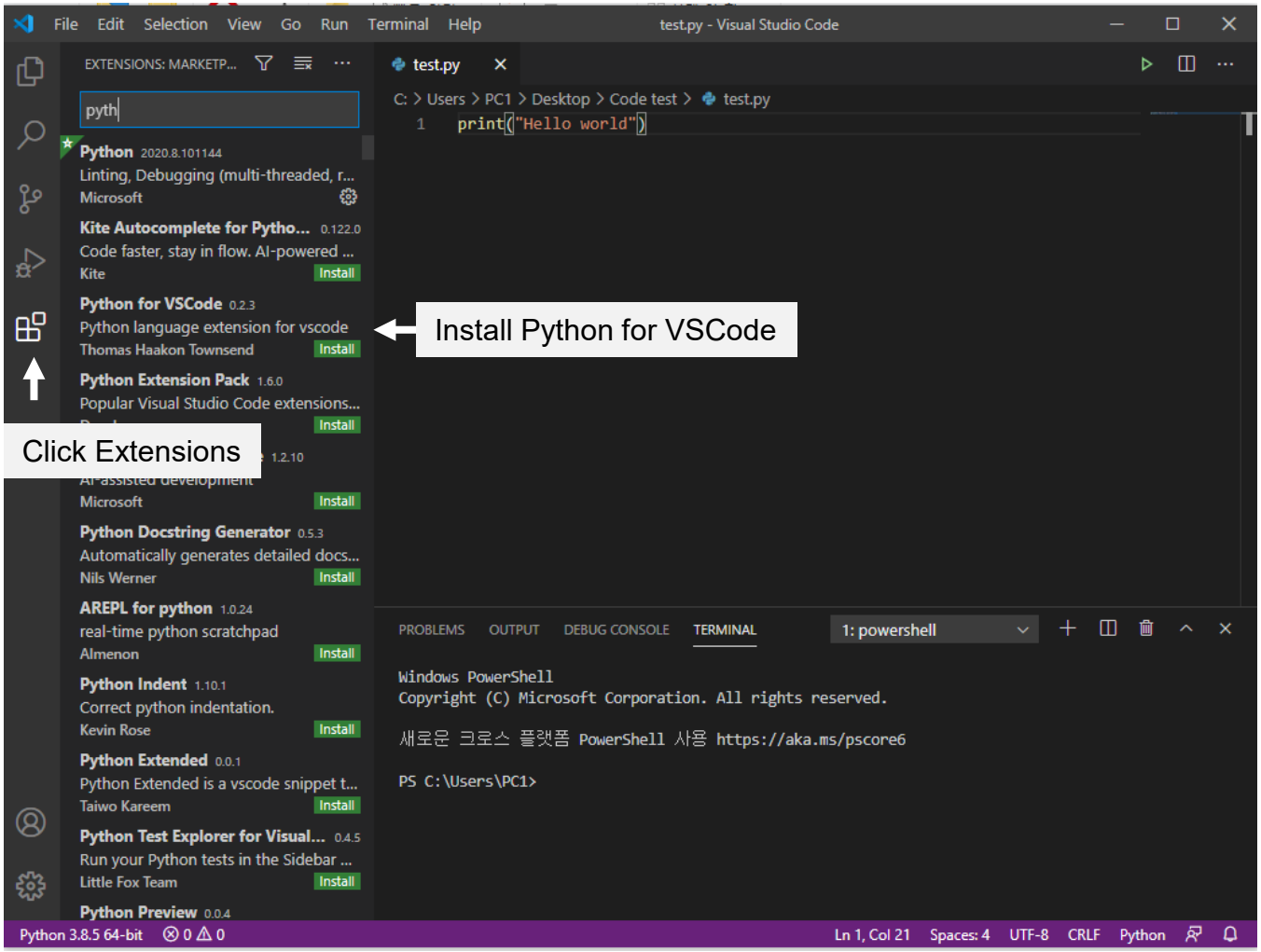
Install Text Editor

Download Visual Studio Code: https://code.visualstudio.com/

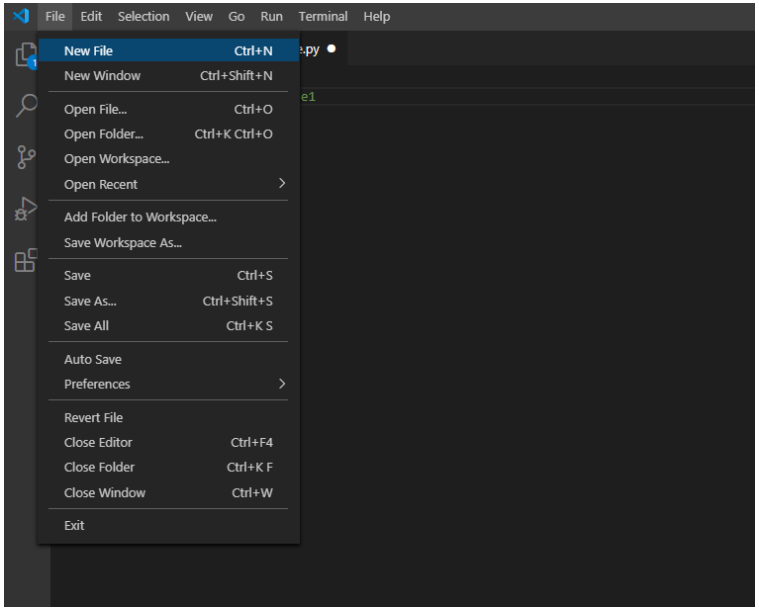


Install Text Editor

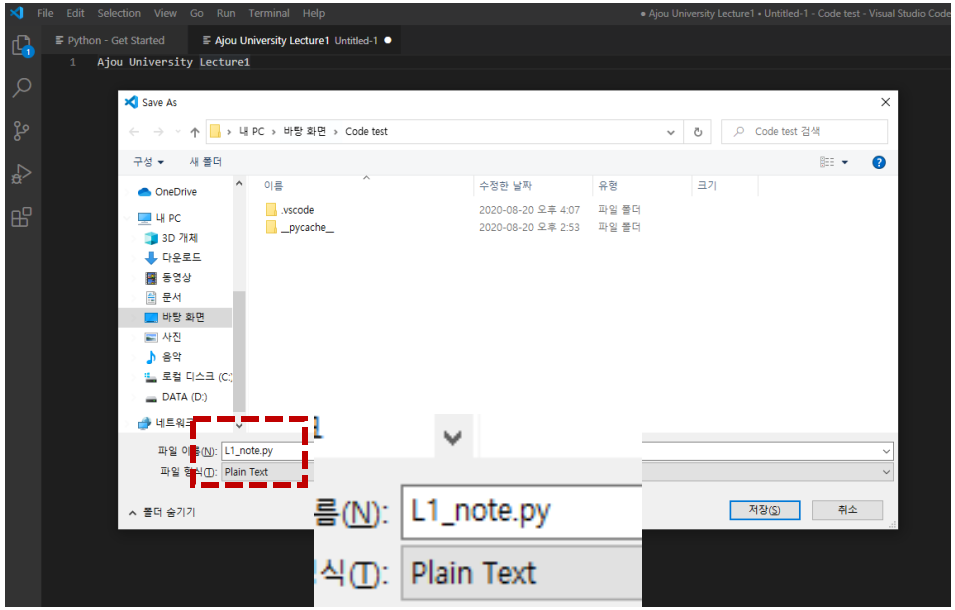
Download Visual Studio Code: https://code.visualstudio.com/



Install Text Editor

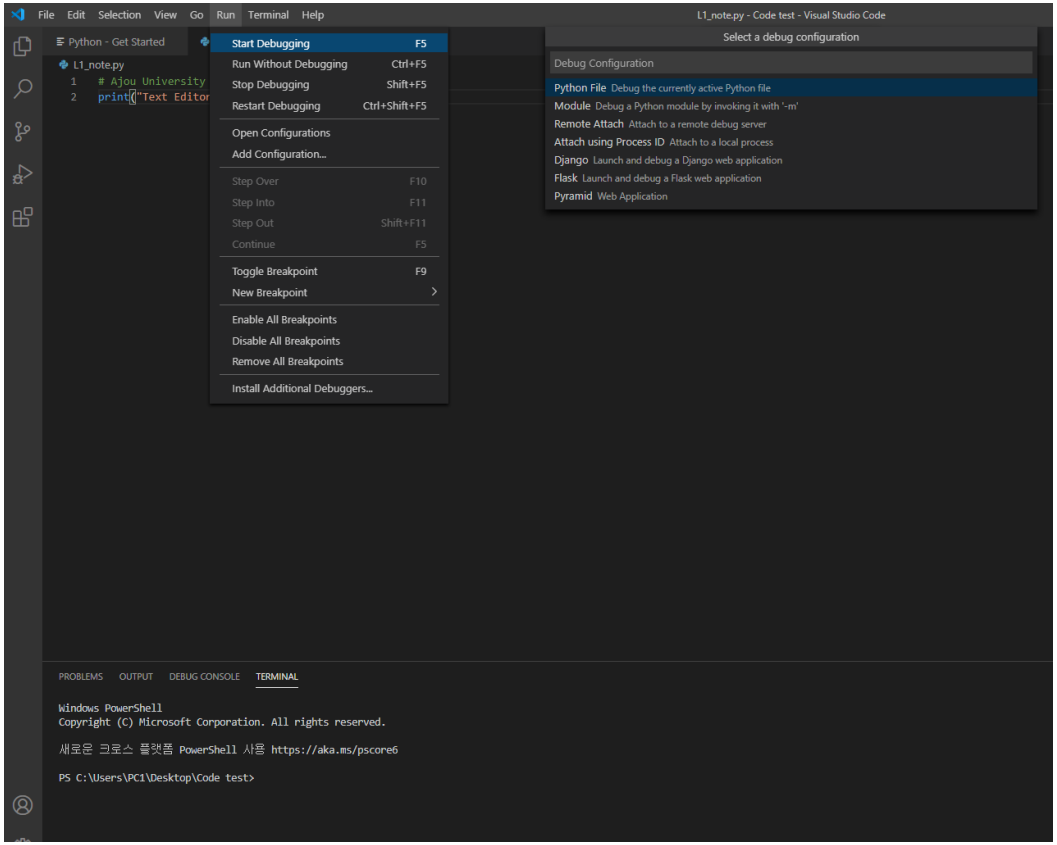


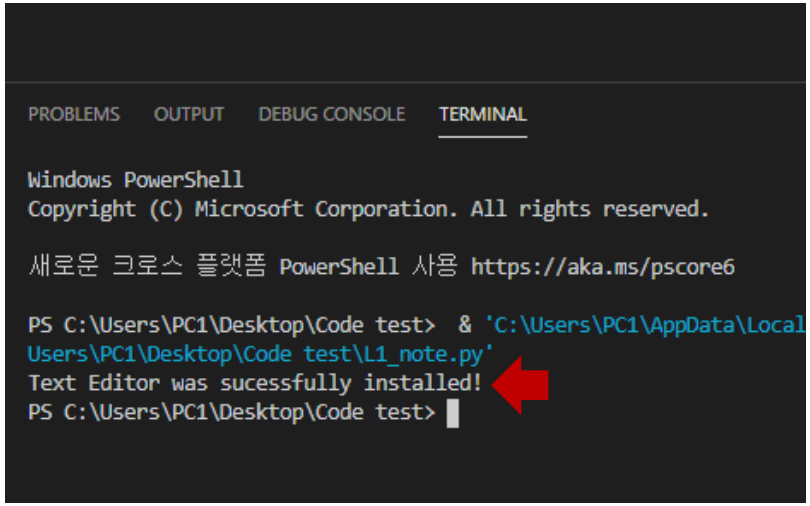
1. Make a new file



2. Save it as {file_name}.py

Install Text Editor





6. If everything goes well, then you can see the sentence you write on Terminal

3. Type this sentence
printf(“Text you want to print”)

4. Press ‘F5’ or Click Run->Start Debugging

5. Select Python File Debug the currently active Python file
(If VScode shows any message that you need to install additional files, then just install it!)