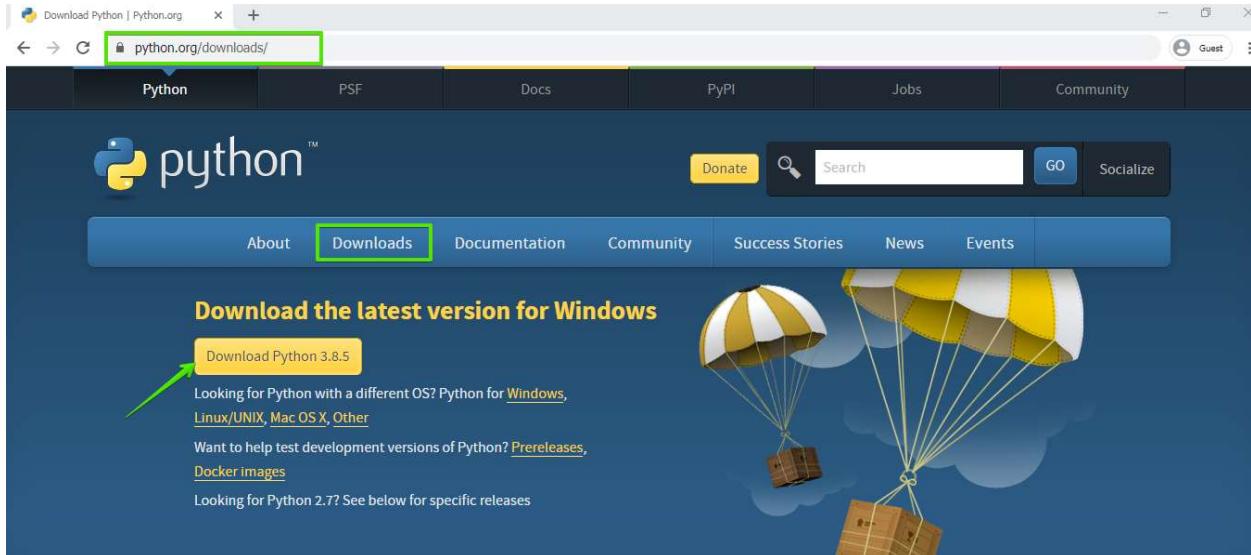
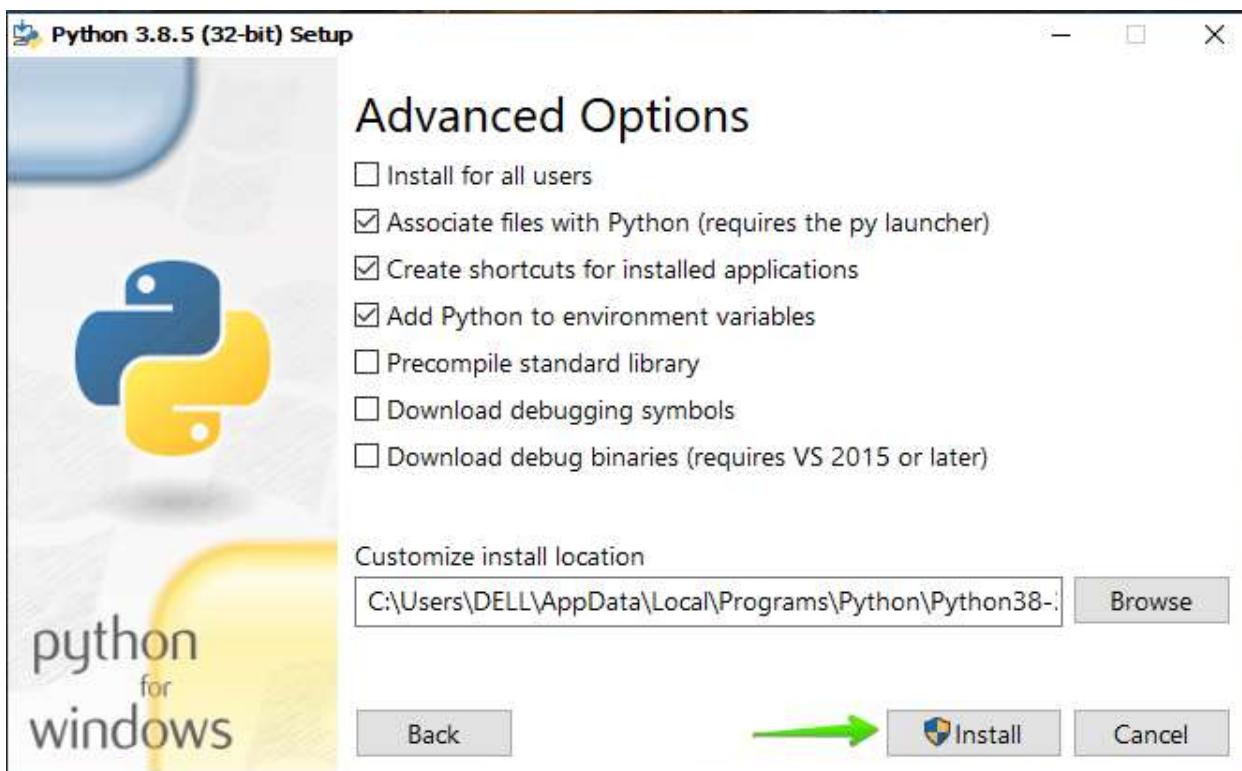
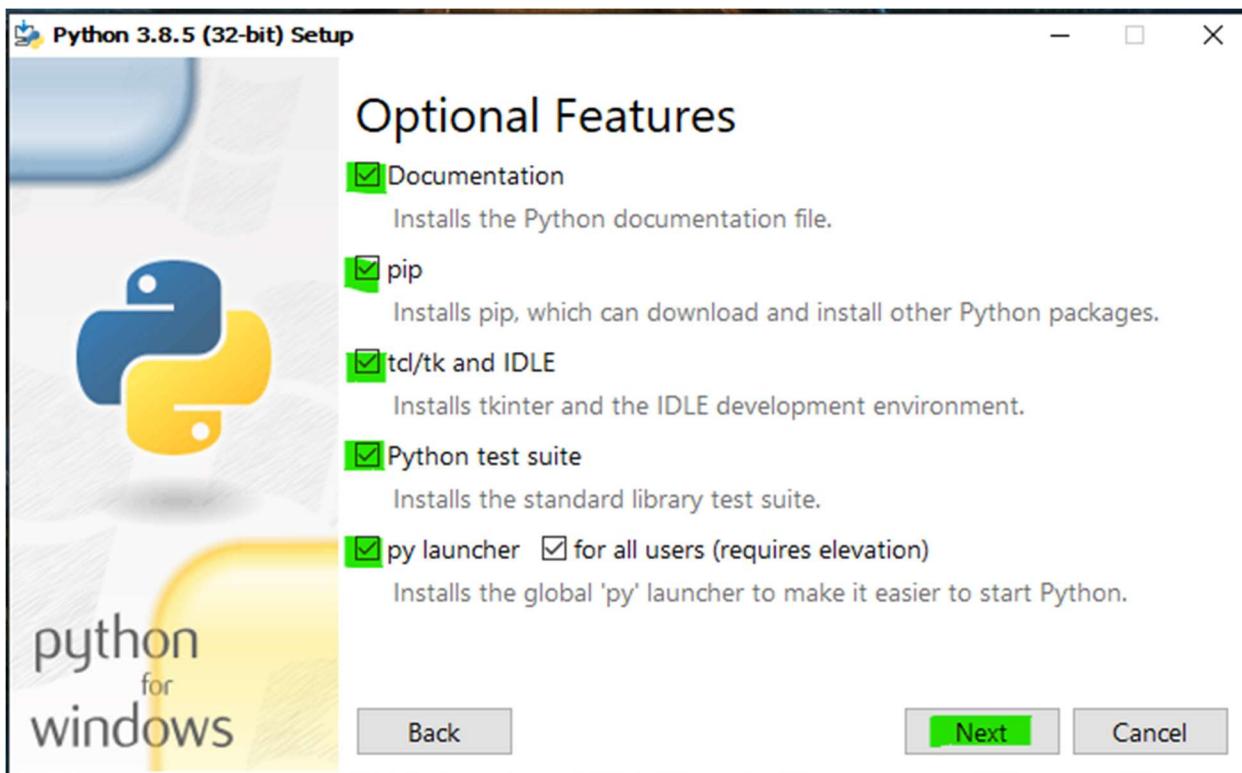
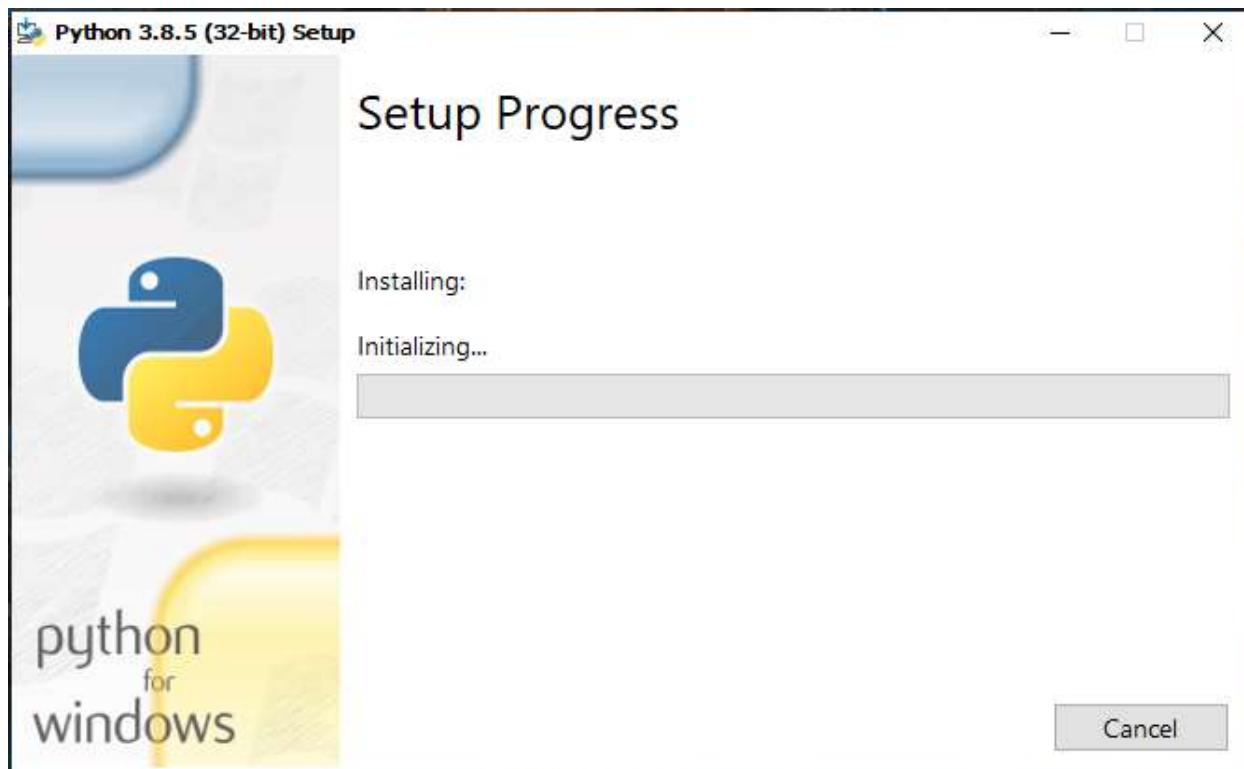


Python ന്റെ അവധിയിലുള്ള installer file ന് <https://www.python.org/> നിന്ന് download ചെയ്യാം. install ലുംപോലി





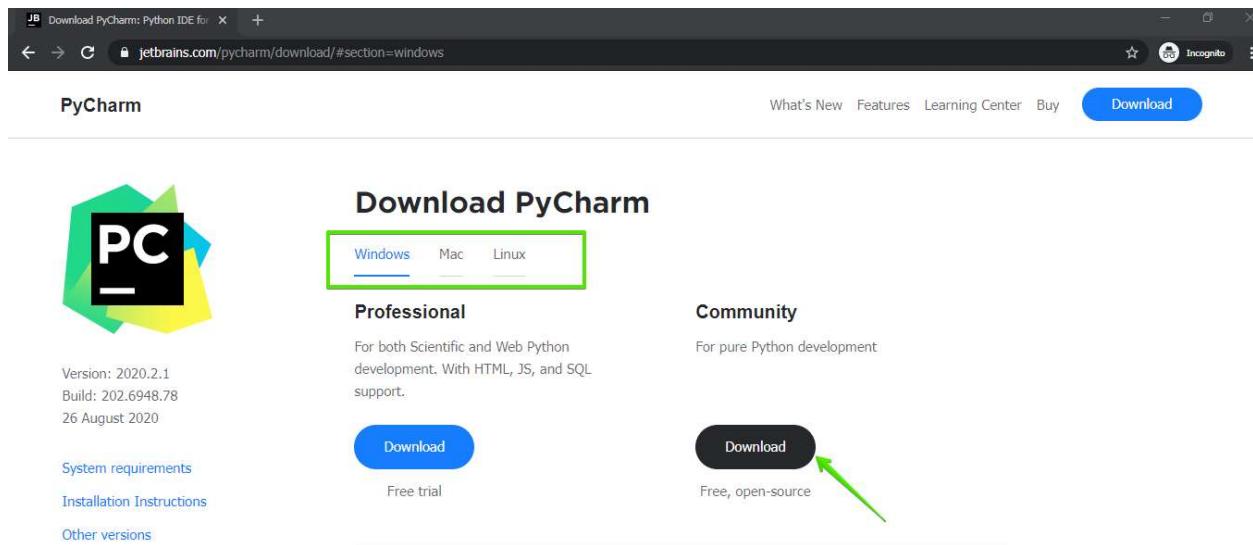


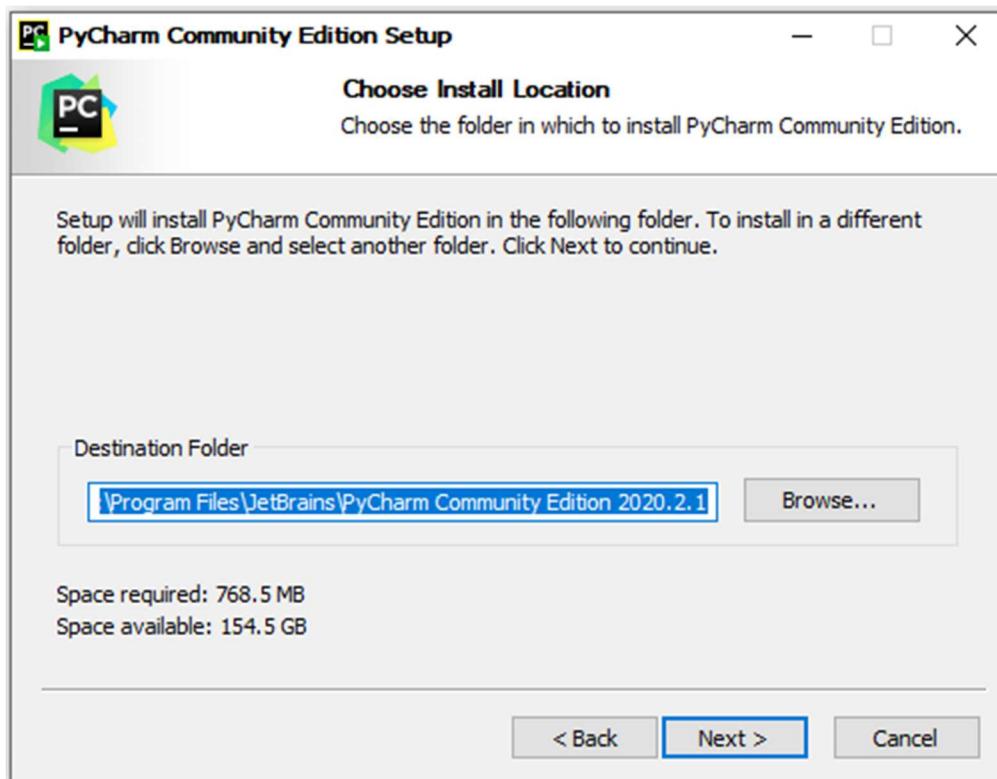
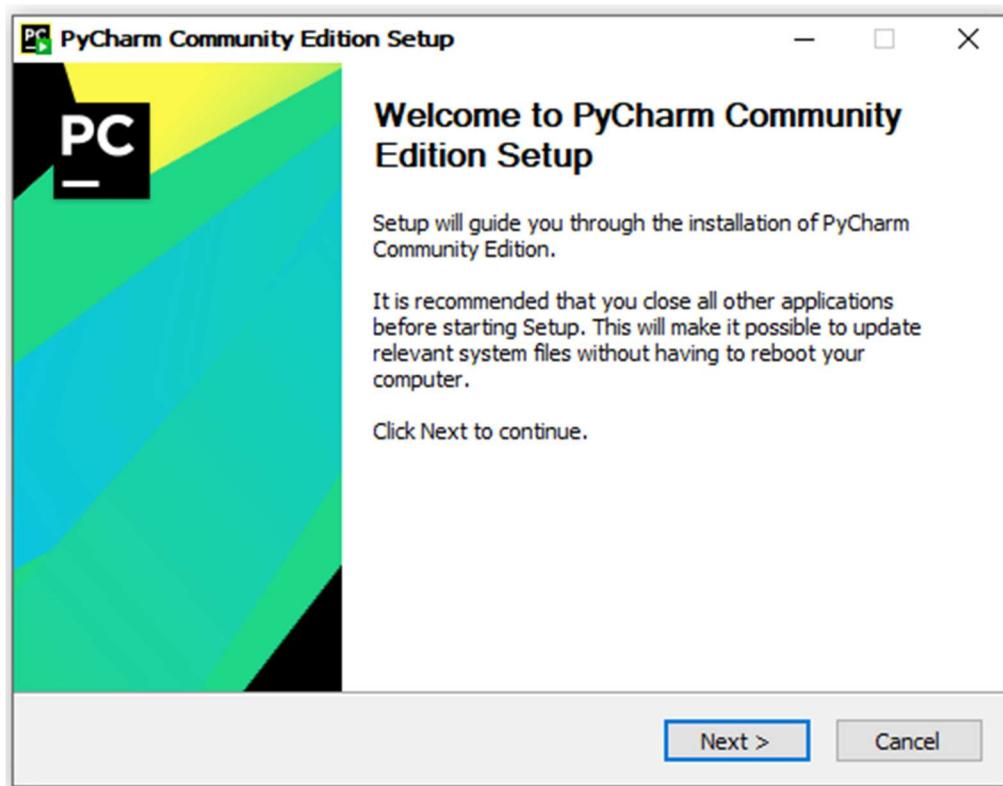
```
C:\Windows\system32\cmd.exe
C:\Users\DELL>python --version
Python 3.8.5
C:\Users\DELL>
```

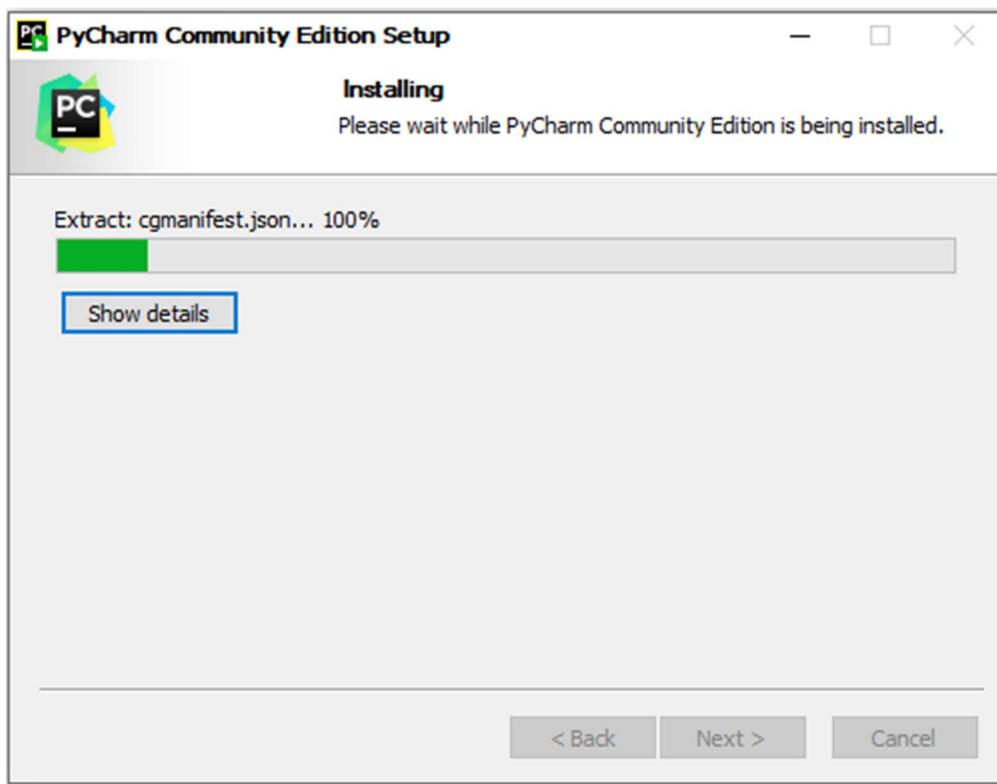
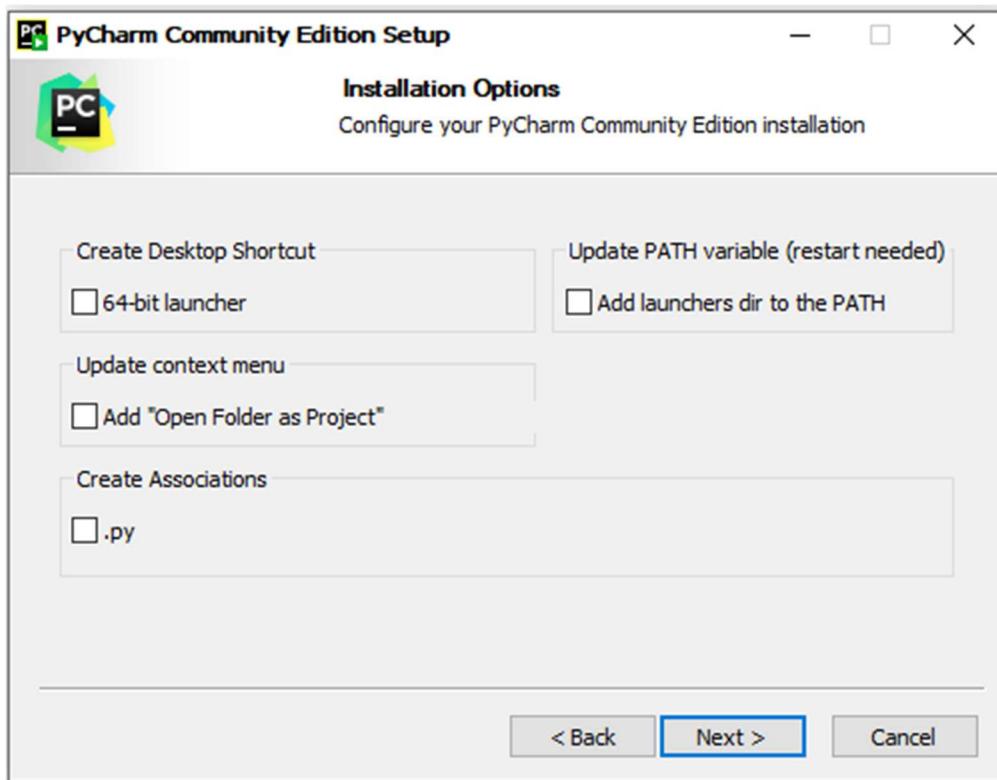
Python ရေးသားရန်အတွက် IDE ကို <https://www.jetbrains.com/pycharm/> တွင် download လုပ်ပြီး install လုပ်ပေးပါ (တော်းကြိုက်နှစ်သက်ရာ IDE ကိုအသုံးပြန်ပါတယ်)

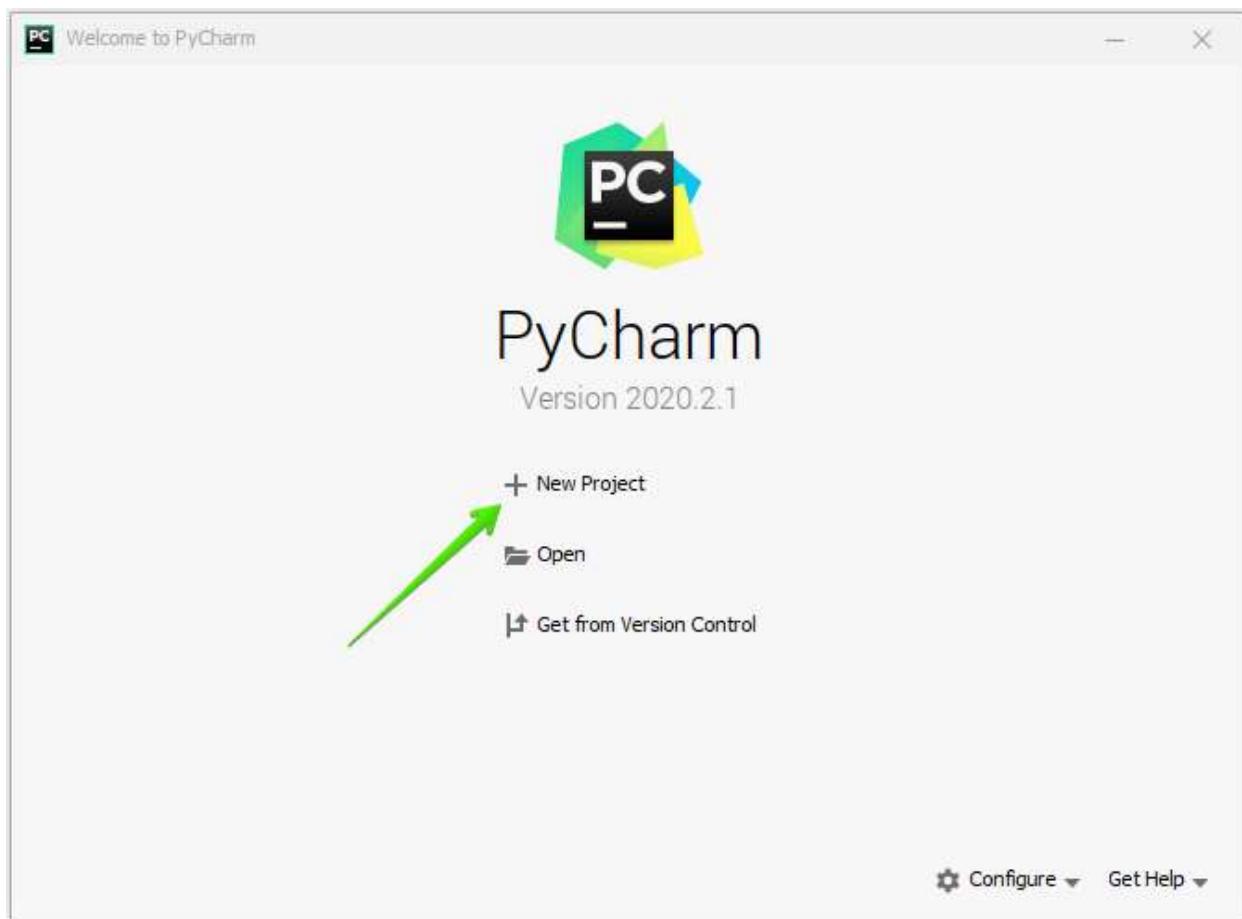


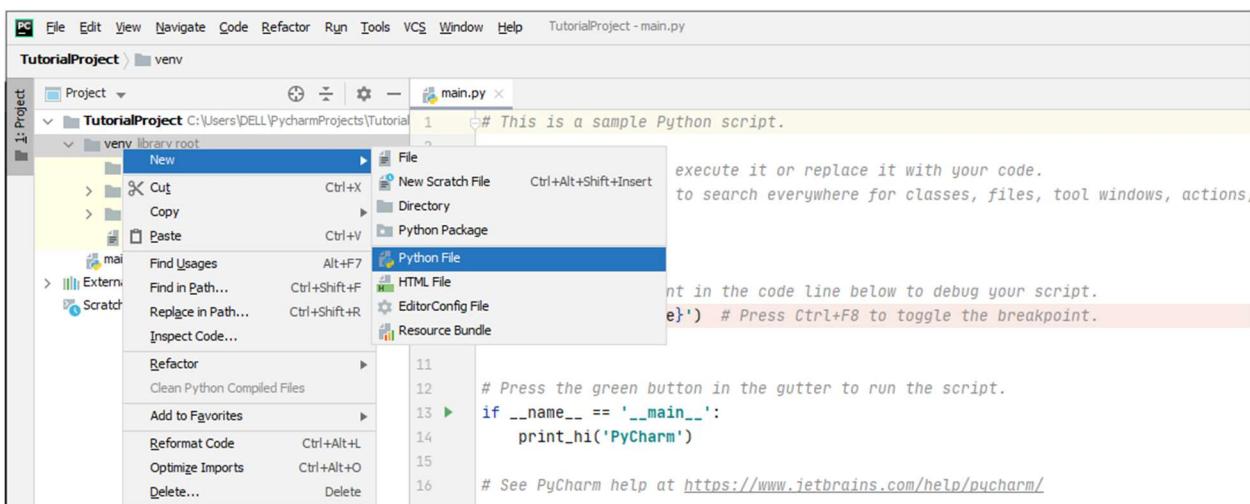
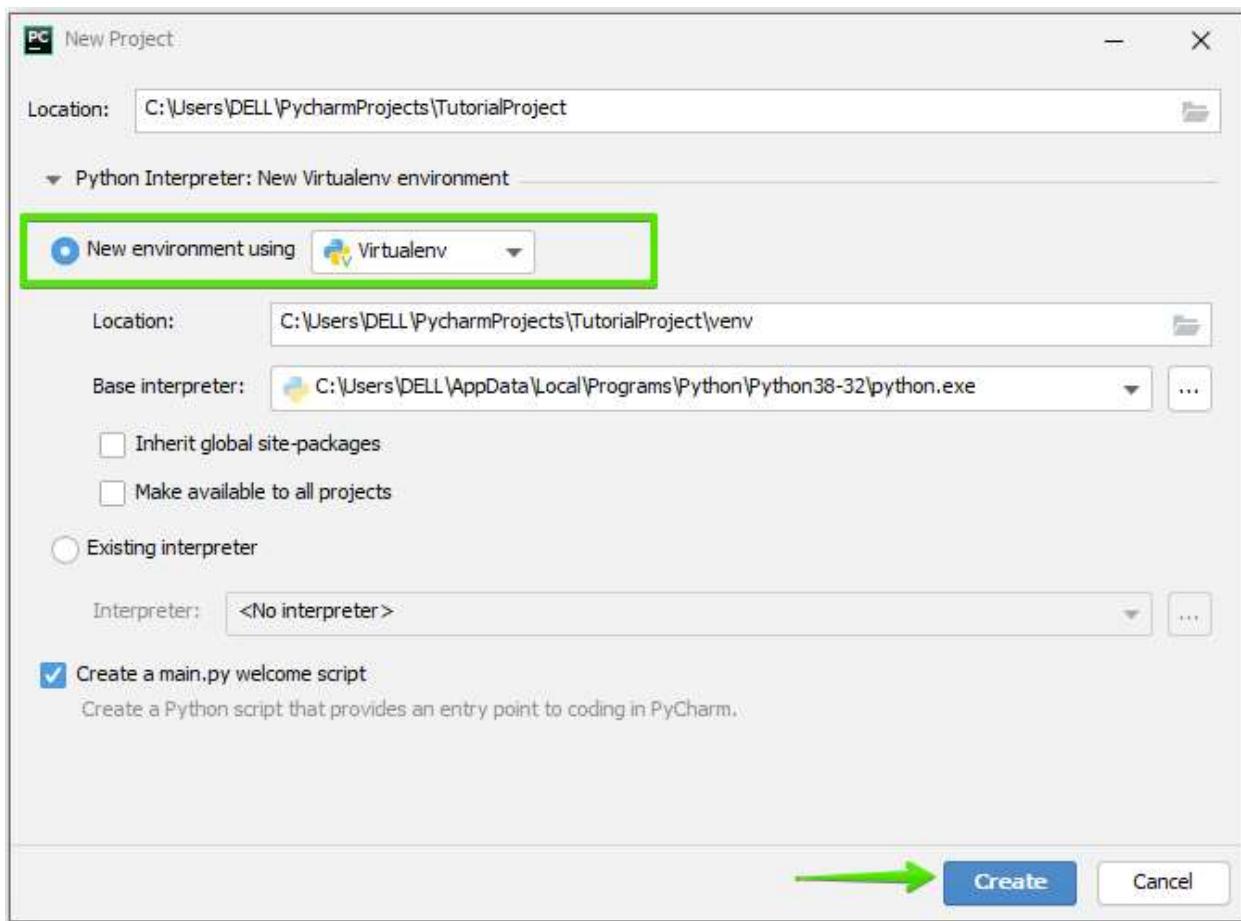
မိမိအသုံးပြုမယ့် OS အလိုက် Community version ကို download ပြုလုပ်ပါ

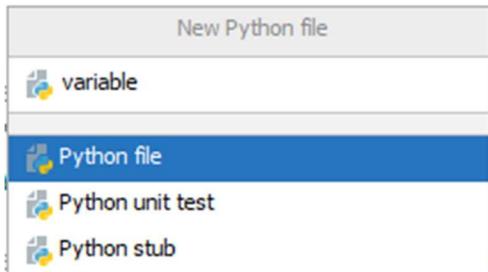












variable name သည် case sensitive ဖြစ်တဲ့အတွက် variableName နဲ့ VARIABLENAME သည် မတူညီပါ။ Output ထုတ်ရန်အတွက် print() function ကိုသုံးနိုင်ပါတယ်။

print(variableName) ဖြင့်ရေးလျှင် output သည် variableName = 1 (Integer type) ဖြစ်သည်။
print(str(variableName)) ဖြင့်ရေးလျှင် output သည် variableName = 1 (String type) ဖြစ်သည်။ Integer -> String type convert လုပ်ချင်တဲ့အတွက် str() function ကိုသုံးထားခြင်း ဖြစ်ပါတယ်။

```

variableName = 1
_variablename = 2
VARIABLENAME = 3
variable_name = 4

print("variableName : " + str(variableName))
print("_variablename : " + str(_variablename))
print("VARIABLENAME : " + str(VARIABLENAME))
print("variable_name : " + str(variable_name))
    
```

variable names are case sensitive.

convert Interger to String

String Data Type ကိုရေးလျှင် single(' ') or double quotes (" ") အတွင်းတွင် ထည့်သွင်းရေးသားရပါတယ်။ https://www.w3schools.com/python/python_datatypes.asp အသေးစိတ်ကိုလေ့လာနိုင်ပါတယ်။ variable တစ်ခုရဲ့ data type ကို သိလိုပါက type(variablename) ဖြင့်ရှာနိုင်ပါတယ်။

Single line comment အတွက် `#comment` ဖြင့်ရေးသားနိုင်ပါတယ်။

Multiple line comment အတွက် `""" multiple line comment """` ဖြင့်ရေးနိုင်ပါတယ်။

The screenshot shows the PyCharm IDE interface. The project structure on the left includes files like main.py, variable.py, and dataType.py. The dataType.py file is open in the editor, showing code that defines various variable types and prints them. A callout box highlights the line `print(type(bool_variable))` with the text "single line comment". The run output on the right shows the printed values: "This is an example string", 2020, 37.5, True, 3j, and <class 'bool'>. The output for the type of bool_variable is highlighted with a green box.

```

string_variable = "This is an example string"
int_variable = 2020
float_variable = 37.5
bool_variable = True
complex_variable = 3j

print(string_variable) #Strings Type
print(int_variable) #Integers Type
print(float_variable) #Float Type
print(bool_variable) #bool Type
print(complex_variable) #complex type
print(type(bool_variable))

```

String formatting የቻውን ማሸጊያዊዎች: {} በኋላ ዘመን ጥሩ ተስተካክልበትን ይገልጻል။

```

print("Add new numbers {} + {} = {}".format(int_variable, float_variable, int_variable+float_variable))
    1      2      3

```

Add new numbers 2020 + 37.5 = 2057.5

```

string_variable = "This is an example string"
new_variable = 'learning python'

print("{} for {}".format(string_variable, new_variable))

```

This is an example string for learning python

```

TutorialProject > venv > dataType.py
Project: main.py variable.py dataType.py
1: string_variable = "This is an example string"
2: int_variable = 2020
3: float_variable = 37.5
4: bool_variable = True
5: complex_variable = 3j
6: new_variable = 'learning python'
7:
8: print("Add new numbers {} + {} = {}".format(int_variable, float_variable, int_variable+float_variable))
9: print("{} for {}".format(string_variable, new_variable))
10:
11:
12:
2020 + 37.5
This is an example string for learning python
Process finished with exit code 0

```

The screenshot shows the PyCharm IDE interface. The top bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help, and TutorialProject - C:\Users\DELL\PycharmProjects\TutorialProject\venv\ dataType.py. The Project tool window on the left lists main.py, variable.py, and dataType.py. The Editor tab for dataType.py contains the provided Python code. The Run tool window at the bottom shows the output of the script: 'Add new numbers 2020 + 37.5 = 2057.5' and 'This is an example string for learning python'. The status bar at the bottom right indicates the time as 10:14, encoding as CRLF, encoding as UTF-8, 4 spaces, and Python 3.8 (TutorialProject).

Function

Python function အတွက် def keyword နဲ့ စရေးပါတယ် colon : နဲ့ အဆုံးသတ်ပါတယ်

```
def FunctionName ( ):
```

function အတွင်းမှာရေးသားမည့် statement တွေကို 1 indent (4spaces) ခုနှင့်ပြီးစတင်ရေးသားရပါတယ်။ PyCharm မှာတော့ tab auto ခုနှင့်ထားပေးပါတယ်။ Function တစ်ခုခုကို အလုပ်လုပ်စေချင်ရင် function call ပေးရပါတယ်။

The screenshot shows the PyCharm IDE interface with the following details:

- Project:** TutorialProject > venv > function.py
- Code Editor:** The code defines a function `functionName()` which adds two numbers and prints the result. A blue arrow points to the opening brace of the function definition, with the annotation "1 indent = 4 spaces". A green arrow points to the function call `functionName()`, with the annotation "function declaration => def Name (parameter) :".
- Run Tab:** Shows the command `C:\Users\DELL\PycharmProjects\TutorialProject\venv\Scripts\python.exe C:/Users/DELL/PycharmProjects/TutorialProject/venv/function.py` and the output `113.893`.
- Bottom Status Bar:** Shows "5:1 CRLF UTF-8 4 spaces Python 3.8 (TutorialProject)".

Runtime ကျမှု user input ကို လက်ခံရန်အတွက် `input()` function ကိုသုံးစိုင်ပါတယ်။ `input()` function ကောင် data pass ပေးလိုက်တာက Strings type အဖြစ်ဖြစ်ပါ။

The screenshot shows the PyCharm IDE interface with the following details:

- Project:** TutorialProject > venv > function.py
- Code Editor:** The code defines a function `calculateDOB()` which prompts for age, calculates birth year, and prints the result. A red arrow points to the `input()` statement with the annotation "To get String data from User Input". Another red arrow points to the `int(age)` conversion with the annotation "convert Strings to Integers".
- Run Tab:** Shows the command `C:\Users\DELL\PycharmProjects\TutorialProject\venv\Scripts\python.exe C:/Users/DELL/PycharmProjects/TutorialProject/venv/function.py`. The user input "24" is highlighted in yellow, with the annotation "User Inputted value". The output shows "Your birth year is 1996".
- Bottom Status Bar:** Shows "5:1 CRLF UTF-8 4 spaces Python 3.8 (TutorialProject)".

Position Arguments

Position argument ဆိတာက function declaration မှာ ကြော်ထားတဲ့ arguments order အလိုက် data passing လုပ်တာဖြစ်ပါတယ်။ default argument တွေကို နောက်ဆုံးမှာ ကြော်ပေးရပါမယ်။

```
def signUpForm( name,age,address ="Yangon",phone ):
```

^

SyntaxError: non-default argument follows default argument

The screenshot shows the PyCharm IDE interface. The project navigation bar at the top indicates 'TutorialProject' and 'function.py'. Below it, the 'Project' tool window lists files: main.py, variable.py, dataType.py, and function.py. The code editor window displays the following Python code:

```

8
9 def signUpForm(name,age,phone,address ="Yangon"):
10     print("Registration information : {}, {}, {}, {}.".format(name,age,phone,address))
11
12 signUpForm("SuPhyu",30,"091234567")
13
14 signUpForm("Aye Aye")
15
16
17

```

The line 'signUpForm("Aye Aye")' is highlighted in yellow, indicating a syntax error. The 'Run' tool window below shows the output of the script:

```

C:\Users\DELL\PycharmProjects\TutorialProject\venv\Scripts\python.exe C:/Users/DELL/PycharmProjects/TutorialProject/venv/fu
Registration information : SuPhyu, 30, 091234567, Yangon
Traceback (most recent call last):
  File "C:/Users/DELL/PycharmProjects/TutorialProject/venv/function.py", line 14, in <module>
    signUpForm("Aye Aye")
TypeError: signUpForm() missing 2 required positional arguments: 'age' and 'phone'

Process finished with exit code 1

```

The status bar at the bottom right shows the file path 'C:\Users\DELL\PycharmProjects\TutorialProject\venv\function.py' and the Python version 'Python 3.8 (TutorialProject)'.

Keyword Arguments

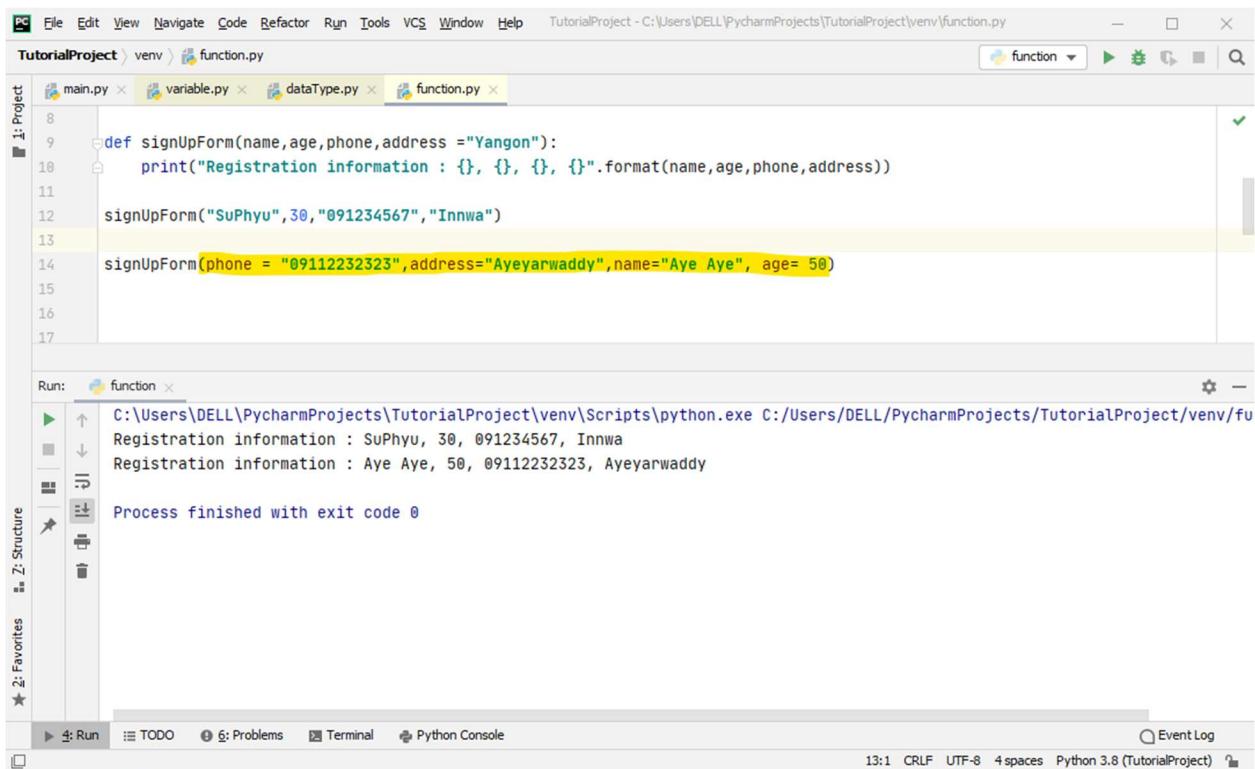
Keyword argument မှာ function declaration မှာ ကြော်ထားတဲ့ arguments တွေ၏ position အလိုက် data passing လုပ်စရာမလိုတော့ပဲ argument name နဲ့တွဲပြီးသုံးတာမျိုးဖြစ်ပါတယ်။ function call တဲ့အခါ Position argument ကိုအရင်ရေးရပါမယ်။ ပြီးမှ keyword arguments တွေကိုရေးသားရပါမယ်။

```
signUpForm(phone = "09112232323", address="Ayeyarwaddy", "Aye Aye", 50)
```

```
signUpForm(phone = "09112232323", address="Ayeyarwaddy", "Aye Aye", 50)
```

^

SyntaxError: positional argument follows keyword argument



```
def signUpForm(name,age,phone,address ="Yangon"):
    print("Registration information : {}, {}, {}, {}".format(name,age,phone,address))

signUpForm("SuPhyu",30,"091234567", "Innwa")

signUpForm(phone = "09112232323",address="Ayeyarwaddy",name="Aye Aye", age= 50)
```

Run: function

```
C:\Users\DELL\PycharmProjects\TutorialProject\venv\Scripts\python.exe C:/Users/DELL/PycharmProjects/TutorialProject/venv/fu
Registration information : SuPhyu, 30, 091234567, Innwa
Registration information : Aye Aye, 50, 09112232323, Ayeyarwaddy

Process finished with exit code 0
```

*args

unlimited positional arguments တွက် ကြိုတင်ခြေခြားစရာမလိုပါ data passing လုပ်နည်း

*arg (function should be able to handle any number of positional arguments) ကို အသုံးပြုနိုင်ပါတယ်။

The screenshot shows a PyCharm interface with the file `arguments.py` open. The code defines a function `addition` that prints the sum of its arguments. A call to `addition(1,2,3,4,5)` is highlighted in yellow. The run output shows a `TypeError` indicating that `addition()` takes 1 positional argument but 5 were given.

```

1: Project 1: TutorialProject > venv > arguments.py
1 def addition(args):
2     print(sum(args))
3
4 addition(1,2,3,4,5)
5
6
7
8
9

Run: arguments
C:\Users\DELL\PycharmProjects\TutorialProject\venv\Scripts\python.exe C:/Users/DELL/PycharmProjects/TutorialProject/venv/arguments.py
Traceback (most recent call last):
  File "C:/Users/DELL/PycharmProjects/TutorialProject/venv/arguments.py", line 4, in <module>
    addition(1,2,3,4,5)
TypeError: addition() takes 1 positional argument but 5 were given

Process finished with exit code 1
  
```

The screenshot shows a PyCharm interface with the file `arguments.py` open. The code has been modified to use `*args` instead of `args`. The run output shows the sum of all arguments, which is 28.8. Red arrows point from the text "Unlimited positional arguments" to the `*args` placeholder in the function definition and to the list of arguments in the call statement.

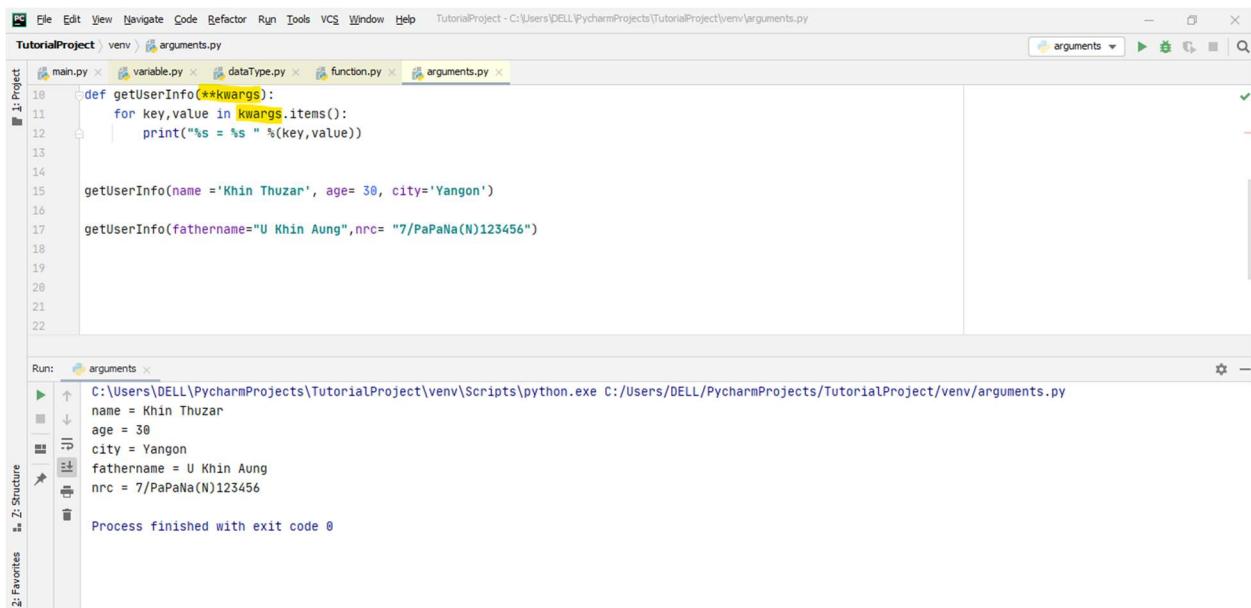
```

1: Project 1: TutorialProject > venv > arguments.py
1 def addition(*args):
2     print(sum(args))
3
4 addition(1,2,3,4,5,6,8,7,0)
5
6
7
8
9

Run: arguments
C:\Users\DELL\PycharmProjects\TutorialProject\venv\Scripts\python.exe C:/Users/DELL/PycharmProjects/TutorialProject/venv/arguments.py
28.8
Process finished with exit code 0
  
```

**kwargs

unlimited keyword arguments තොග් තිශ්චායා තුළ නොමැදුම් දත්ත සාක්ෂියෙන් **kwargs (function should be able to handle any number of keyword arguments) තුළ ප්‍රාග්ධන තියාවා



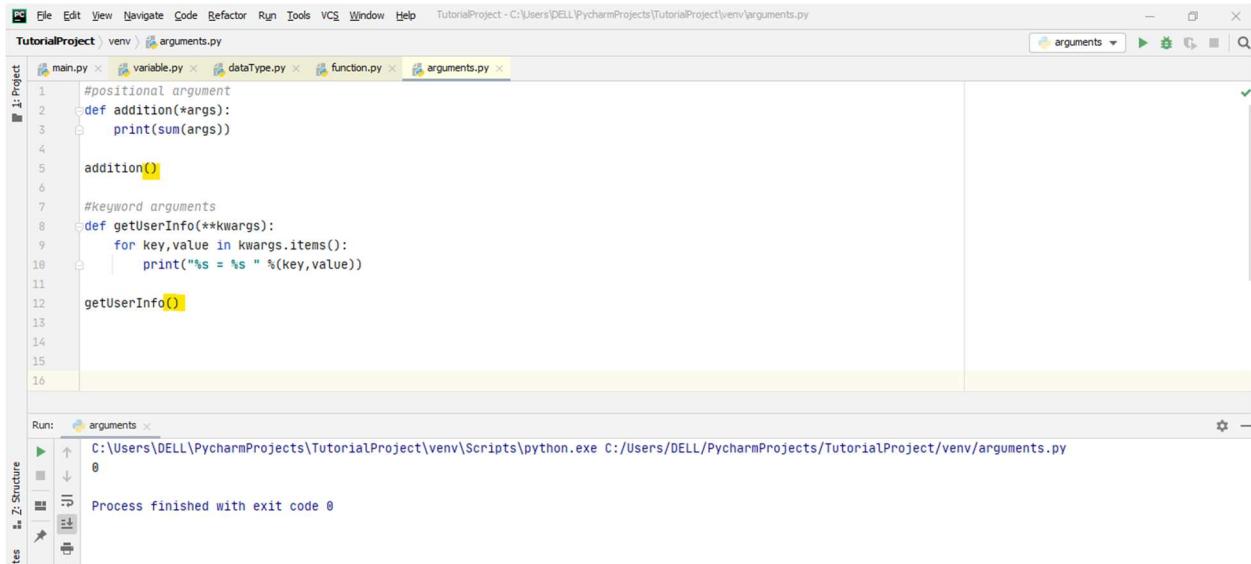
The screenshot shows the PyCharm IDE interface with the following details:

- Project:** TutorialProject > venv > arguments.py
- Code:**

```

10 def getUserInfo(**kwargs):
11     for key,value in kwargs.items():
12         print("%s = %s " %(key,value))
13
14
15(userInfo = 'Khin Thuzar', age= 30, city='Yangon')
16
17(userInfo=fathername="U Khin Aung",nrc= "7/PaPaNa(N)123456")
18
19
20
21
22
    
```
- Run:** arguments
 - Process finished with exit code 0

unlimited သုံးနိုင်သလို zero argument အနေဖြင့်လည်း သုံးနိုင်ပါတယ်



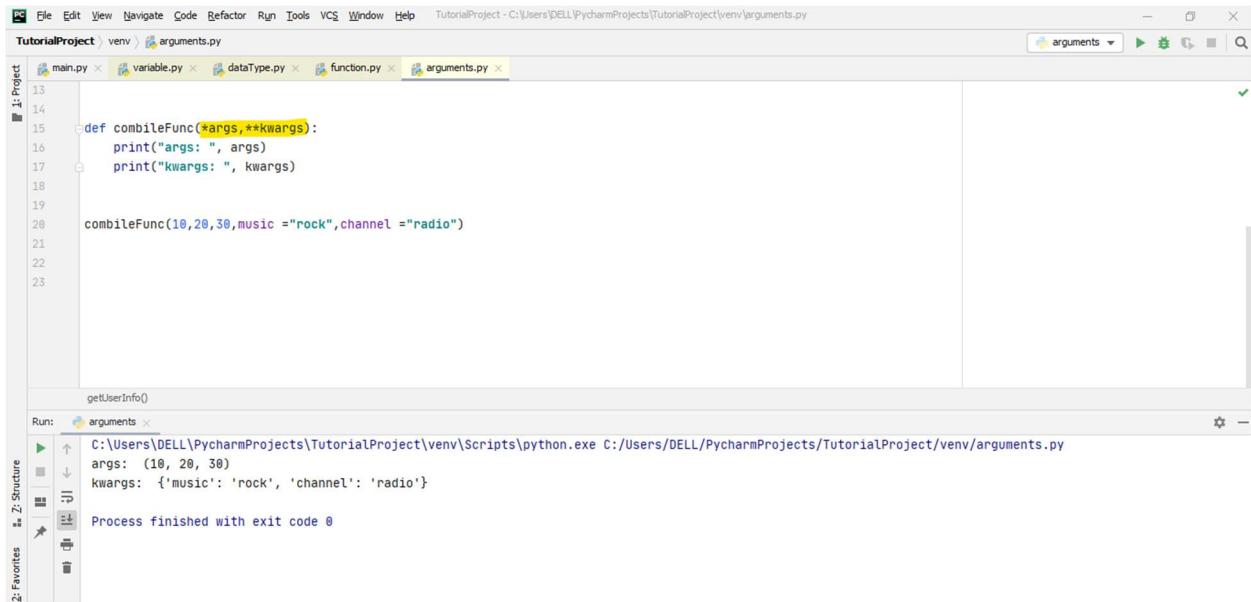
The screenshot shows the PyCharm IDE interface with the following details:

- Project:** TutorialProject > venv > arguments.py
- Code:**

```

1 #positional argument
2 def addition(*args):
3     print(sum(args))
4
5 addition(1)
6
7 #keyword arguments
8 def getUserInfo(**kwargs):
9     for key,value in kwargs.items():
10        print("%s = %s " %(key,value))
11
12 getUserInfo()
13
14
15
16
    
```
- Run:** arguments
 - Process finished with exit code 0

Combining function အတွက်ဆိုရင် *args ကိုအရင်ရေးရပါမယ်။ ပြီးမှ **kwargs တွေကိုရေးသားရပါမယ်

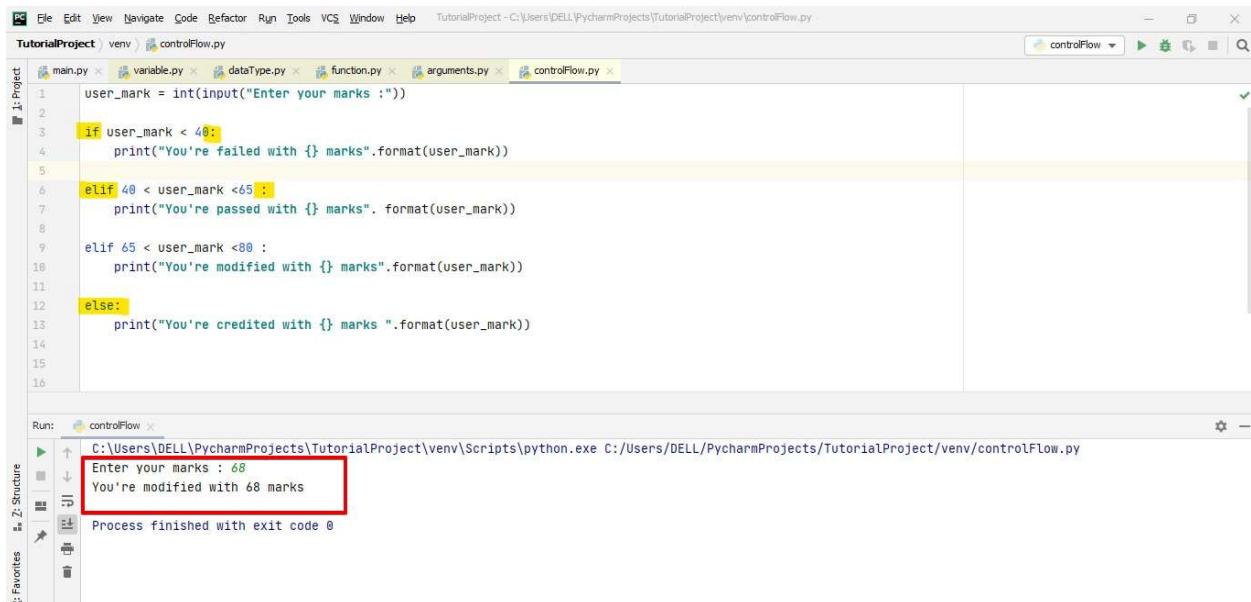


```

File Edit View Navigate Code Refactor Run Tools VCS Window Help TutorialProject - C:\Users\DELL\PycharmProjects\TutorialProject\venv\arguments.py
TutorialProject > venv arguments.py
Project: main.py variable.py dataType.py function.py arguments.py
13
14
15 def combineFunc(*args,**kwargs):
16     print("args: ", args)
17     print("kwargs: ", kwargs)
18
19
20 combineFunc(10,20,30,music ="rock",channel ="radio")
21
22
23
getUserInfo()
Run: arguments
C:\Users\DELL\PycharmProjects\TutorialProject\venv\Scripts\python.exe C:/Users/DELL/PycharmProjects/TutorialProject/venv/arguments.py
args: (10, 20, 30)
kwargs: {'music': 'rock', 'channel': 'radio'}
Process finished with exit code 0

```

Control Structure

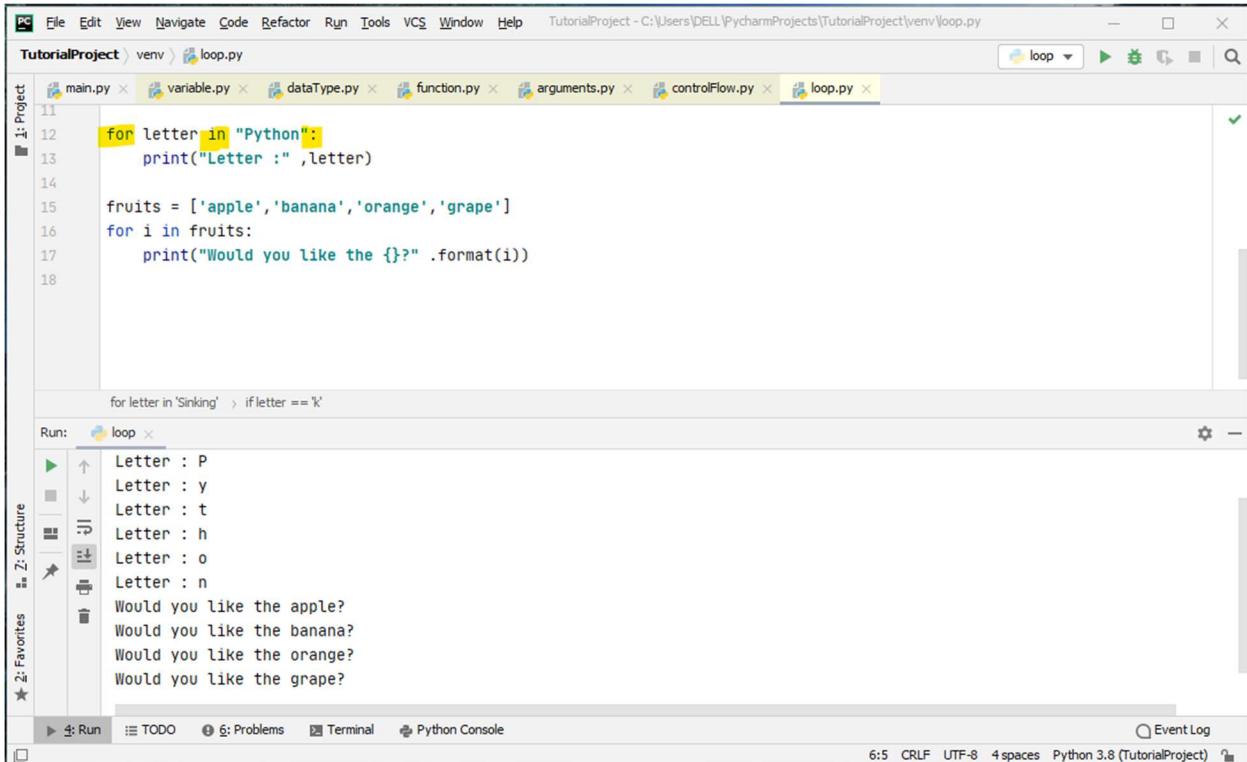


```

File Edit View Navigate Code Refactor Run Tools VCS Window Help TutorialProject - C:\Users\DELL\PycharmProjects\TutorialProject\venv\controlFlow.py
TutorialProject > venv controlFlow.py
Project: main.py variable.py dataType.py function.py arguments.py controlFlow.py
1 user_mark = int(input("Enter your marks :"))
2
3 if user_mark < 40:
4     print("You're failed with {} marks".format(user_mark))
5
6 elif 40 < user_mark <65 :
7     print("You're passed with {} marks".format(user_mark))
8
9 elif 65 < user_mark <80 :
10    print("You're modified with {} marks".format(user_mark))
11
12 else:
13     print("You're credited with {} marks ".format(user_mark))
14
15
16
Run: controlFlow
C:\Users\DELL\PycharmProjects\TutorialProject\venv\Scripts\python.exe C:/Users/DELL/PycharmProjects/TutorialProject/venv/controlFlow.py
Enter your marks : 68
You're modified with 68 marks
Process finished with exit code 0

```

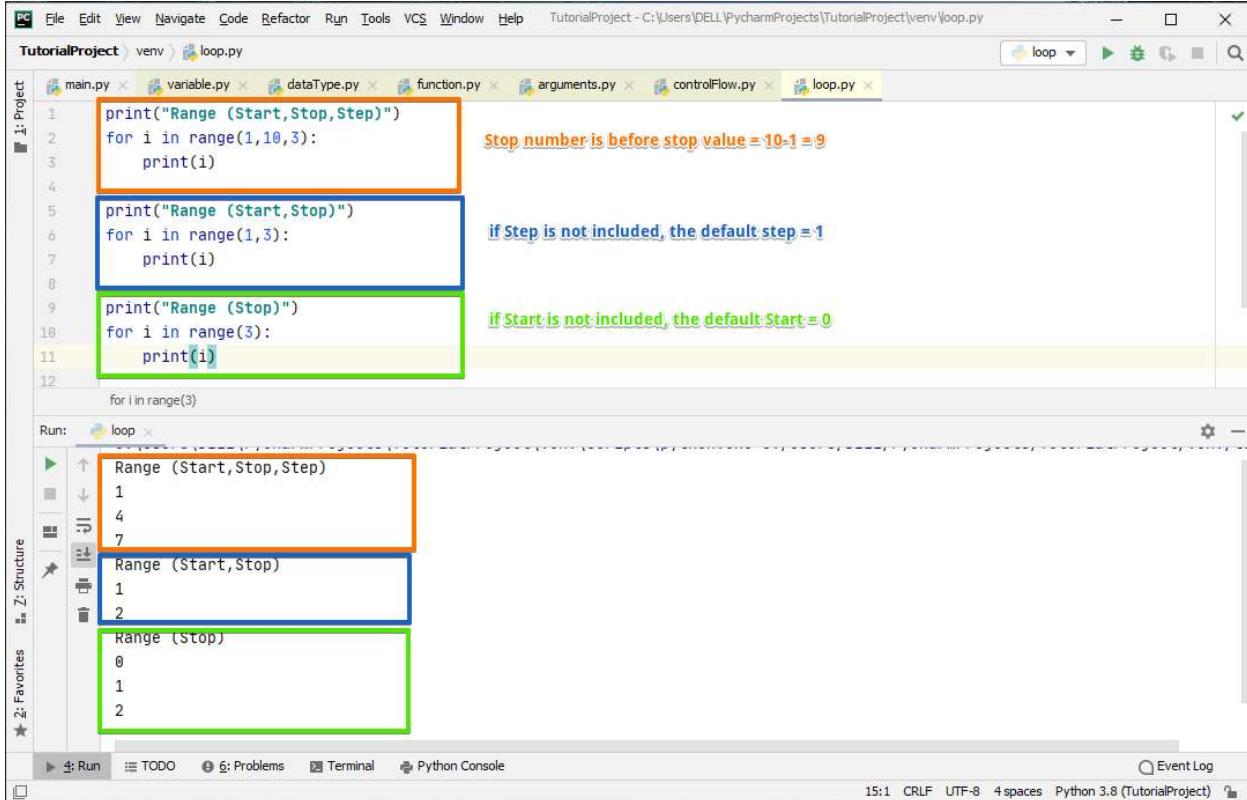
Loop



```

11
12     for letter in "Python":
13         print("Letter :" ,letter)
14
15     fruits = ['apple','banana','orange','grape']
16     for i in fruits:
17         print("Would you like the {}?".format(i))
18
19
20     for letter in 'Sinking' > if letter == 'k'
21
22 Run: loop
23
24 Letter : P
25 Letter : y
26 Letter : t
27 Letter : h
28 Letter : o
29 Letter : n
30 Would you like the apple?
31 Would you like the banana?
32 Would you like the orange?
33 Would you like the grape?

```



```

1:1 print("Range (Start,Stop,Step)")
1:2 for i in range(1,10,3):
1:3     print(i)
1:4
1:5 print("Range (Start,Stop)")
1:6 for i in range(1,3):
1:7     print(i)
1:8
1:9 print("Range (Stop)")
1:10 for i in range(3):
1:11     print(i)
1:12
1:13 for i in range(3)
1:14
1:15     Range (Start,Stop,Step)
1:16     1
1:17     4
1:18     7
1:19     Range (Start,Stop)
1:20     1
1:21     2
1:22     Range (Stop)
1:23     0
1:24     1
1:25     2

```

Stop number is before stop value = 10-1 = 9

If Step is not included, the default step = 1

If Start is not included, the default Start = 0

The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The title bar indicates the project is "TutorialProject" and the file is "loop.py". The code editor displays the following Python code:

```

12
13     count = 0
14     while count<4:
15         print(count)
16         count += 1
17
18
19
20
21
22
23

```

The run configuration "loop" is selected in the bottom-left panel. The output window shows the execution results:

```

Run: loop x
C:\Users\DELL\PycharmProjects\TutorialProject\venv\Scripts\python.exe C:/Users/DELL/PycharmProjects/TutorialProject/venv/loop.py
0
1
2
3

Process finished with exit code 0

```

The bottom status bar shows the time as 11:15, encoding as CRLF, character set as UTF-8, and indentation as 4 spaces. The Python version is listed as Python 3.8 (TutorialProject).

The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The title bar indicates the project is "TutorialProject" and the file is "loop.py". The code editor displays the following Python code:

```

30
31     for number in range(1,5):
32         if number == 3:
33             break
34         else:
35             print("The number is {}.".format(number))
36
37
38
39

```

The run configuration "loop" is selected in the bottom-left panel. The output window shows the execution results:

```

Run: loop x
C:\Users\DELL\PycharmProjects\TutorialProject\venv\Scripts\python.exe C:/Users/DELL/PycharmProjects/TutorialProject/venv/loop.py
The number is 1.
The number is 2.

Process finished with exit code 0

```

The image shows two screenshots of the PyCharm IDE interface, each displaying a Python script named `loop.py` and its run output.

Screenshot 1 (Top):

- Code:**

```

for number in range(1,5):
    if number == 3:
        continue
    else:
        print("The number is {}.".format(number))
    
```
- Run Output:**

```

for number in range(1,5) > if number == 3
C:\Users\DELL\PycharmProjects\TutorialProject\venv\Scripts\python.exe C:/Users/DELL/PycharmProjects/TutorialProject/venv/loop.py
The number is 1.
The number is 2.
The number is 4.

Process finished with exit code 0
    
```

Screenshot 2 (Bottom):

- Code:**

```

for number in range(1,5):
    if number == 3:
        pass
    else:
        print("The number is {}.".format(number))
    
```
- Run Output:**

```

for number in range(1,5) > if number == 3
C:\Users\DELL\PycharmProjects\TutorialProject\venv\Scripts\python.exe C:/Users/DELL/PycharmProjects/TutorialProject/venv/loop.py
The number is 1.
The number is 2.
The number is 4.

Process finished with exit code 0
    
```

In both screenshots, the line containing the `continue` or `pass` statement is highlighted in yellow. The run output shows that the loop iterates from 1 to 4, skipping the value 3 in the first example and doing nothing in the second example.

တိကျသေချာတဲ့ တန်ဖိုးတစ်ခုပေးထားတဲ့အခါမျိုးမှာ for loop, condition တစ်ခုချပေါ်မှာ ဆိုရင် while loop ဆိုပြီး သုံးကြပါတယ်။ Python မှာ do-while loop မရှိပါဘူးသူ့အစား while ကိုပဲ သုံးနိုင်ပါတယ်။ loop control အတွက်ကတော့ Continue ဆိုရင် လက်ရှိ section ကနေထွက်ပြီး ဆက်လက်လုပ်ဆောင်ပါတယ်။ break

ကတေသာ existing loop ကနေကိုထွက်လိုက်တာပါ။ pass ကတေသာ testing code အတွက် အသုံးများတာဖြစ်ပြီးတော့ continue လိုပဲ အသုံးပြုခိုင်ပါတယ်။

Lists

မတူညီတဲ့ data type တွေ items အမျိုးမျိုးကို list `listexample= [, , ,]` မှာ အသုံးပြုခိုင်ပါတယ်။

`append()` - သည် list ထဲကို item တစ်ခုထပ်ပေါင်းတဲ့အခါမှာ သုံးပါတယ်။ [exactly one argument]

`remove()` - သည် list ထဲကို item တစ်ခုဖယ်ထဲတဲ့ အခါမှာ သုံးပါတယ်။ [exactly one argument]

`pop()` သည် index နဲ့ ဘွဲ့နဲ့ပြီး `remove` လုပ်တာမျိုးပါ `pop(0)` သည် first index ဖြစ်ပြီး Last index ကိုတော့ `pop(-1)` နဲ့ရေးလိုပါတယ်။

list တစ်ခုနဲ့ တစ်ခု ပေါင်းချင်ရင်တော့ `extend()` ကိုသုံးပါတယ်။

```

1: list_example = [12, 89.443, "Apples", "1998", 40.3, 2]
2: print(list_example)
3:
4: list_example.append("orange")
5: print(list_example)
6:
7: another_list = ["python", "java", "php"]
8: list_example.extend(another_list)
9: print(list_example)
10: print(another_list)
11:
12: another_list.remove("java")
13: print(another_list)
14: print(list_example)
15:
16: list_example.pop(0) The first index
17: print(list_example)
18:
19: list_example.pop(-1) The last index
20: print(list_example)
21:
22: print(list_example.index("orange"))
23:
24: print("python" in list_example)
25: print("swift" in list_example)

```

Tuple

Tuple နဲ့ list ကွားတောင်တော့ Tuple မှတ်တော့ tuple = ('Peter', 'Josep', 123, 69.50) နှုန်းပါတယ်

list = mutable type (object creation လုပ်ပြီးရင် modified လုပ်ခွို့ရှိပါတယ်)

tuple = immutable type (object creation ပြီးတာနဲ့ modified လုပ်ခွင့်မရှိတော့ပါ)

The screenshot shows the PyCharm IDE interface with the following details:

- File Structure:** The left sidebar shows files: ts.py, controlFlow.py, loop.py, list.py, tuple.py.
- Code Editor:** The main editor contains Python code in tuple.py:

```
list = ['Peter', 'Josep', 123 ,69.50]
print(type(list))
print(list)

tuple = ('Peter', 'Josep', 123 ,69.50)
print(type(tuple))
print(tuple)

list[0] = "Mathew"
print(list)

tuple[0] = "Mathew" ←
print(tuple)
```
- Run Tab:** The Run tab shows the command: C:\Users\DELL\PycharmProjects\TutorialProject\venv\Scripts\python. The output pane displays the traceback and the error message: `TypeError: 'tuple' object does not support item assignment`. The line `tuple[0] = "Mathew"` is highlighted in yellow, and the error message is also highlighted.
- Bottom Navigation:** The bottom navigation bar includes TODO, Problems, Terminal, Python Console, and Event Log tabs.

Dictionaries

dictionary က key,value အတွဲလိုက်သုံးပါတယ်။ key သည် unique ဖြစ်ရပါမယ်

key နဲ့ value ကို ရှာဖို့အတွက် get("key") ကိုသုံးနိုင်ပါတယ်။ items () ကတော့ dictionary မှာရှိတဲ့ (key,value) အတဲ့လိုက်ကို outputထုတ်ပေးပါတယ်။

`keys()` - dictionary မှာရှိတဲ့ key တွေပဲထုတ်ပေးတာပါ

values () - dictionary မှာရှိတဲ့ values တွေပဲထုတ်ပေးပါတယ်။ popitem () - the last item ကို remove လုပ်ဖို့သုံးပါတယ်. key ပေးပြီး value update လုပ်ဖို့အတွက်update (), dictionary အချင်းချင်း ပေါင်းဖို့အတွက်လည်းသုံးပါတယ်။ item အသစ်ထပ်ထည့်ဖို့ setdefault () တိုကို သုံးနိုင်ပါတယ်။

```

1: dictionary = {"rice": 2500 , "noodle": 300 , "oil" : 1700}
2: print(dictionary)
3:
4: print(dictionary.get("noodle"))
5: print(dictionary.items())
6: print(dictionary.keys())
7: print(dictionary.values())
8:
9: dictionary.popitem()
10: print(dictionary)
11:
12: dictionary.setdefault("egg",1890) ← Add new item
13: print(dictionary)
14:
15: dictionary.setdefault("juice") ← the last item is removed
16: print(dictionary)
17:
18: another_dict = {"vivo" :134009 , "samsung" :450093}
19: dictionary.update(another_dict)
20: print(dictionary)
21:
22: another_dict.update(vivo= 90000)
23: print(another_dict)
24:
25:
26:

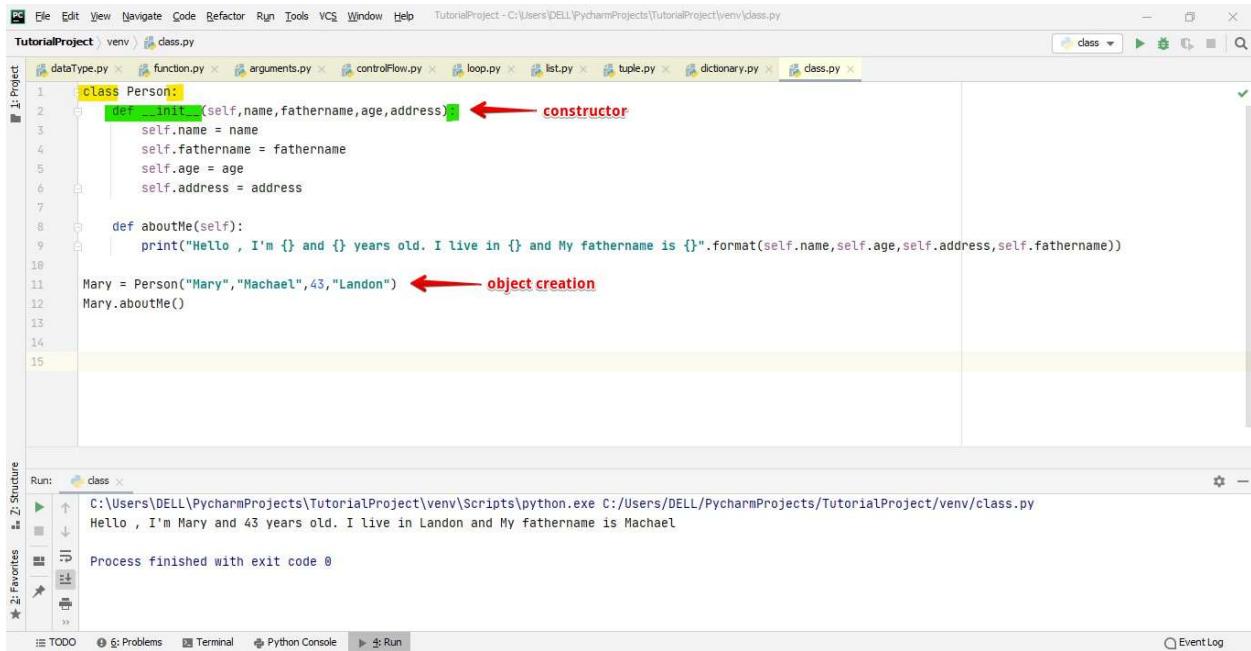
```

Class

Class ဆိုတာ group related functions and variables လိုပြောနိုင်ပါတယ်။ class တစ်ခုမှာရှိတဲ့ function, variable ကို အခြား class တစ်ခုကနေ ယူသုံးစွင့်ဖို့ inheritance လုပ်နိုင်ပါတယ်။ Python မှာ multiple inheritance ကို လက်ခံပါတယ်

def __init__ () ကတော့ constructor ပါ။ instance create လုပ်တဲ့အခါ auto execute လုပ်ပေးပါတယ်။ တကယ်လို့မရေးထားခဲ့ရင် default constructor အနေနဲ့လည်းအလုပ်လုပ်ပေးပါတယ်။

init ရဲအရှေ့နဲ့အနောက်မှာ (2 underscores __) နဲ့သုံးထားပါတယ်။



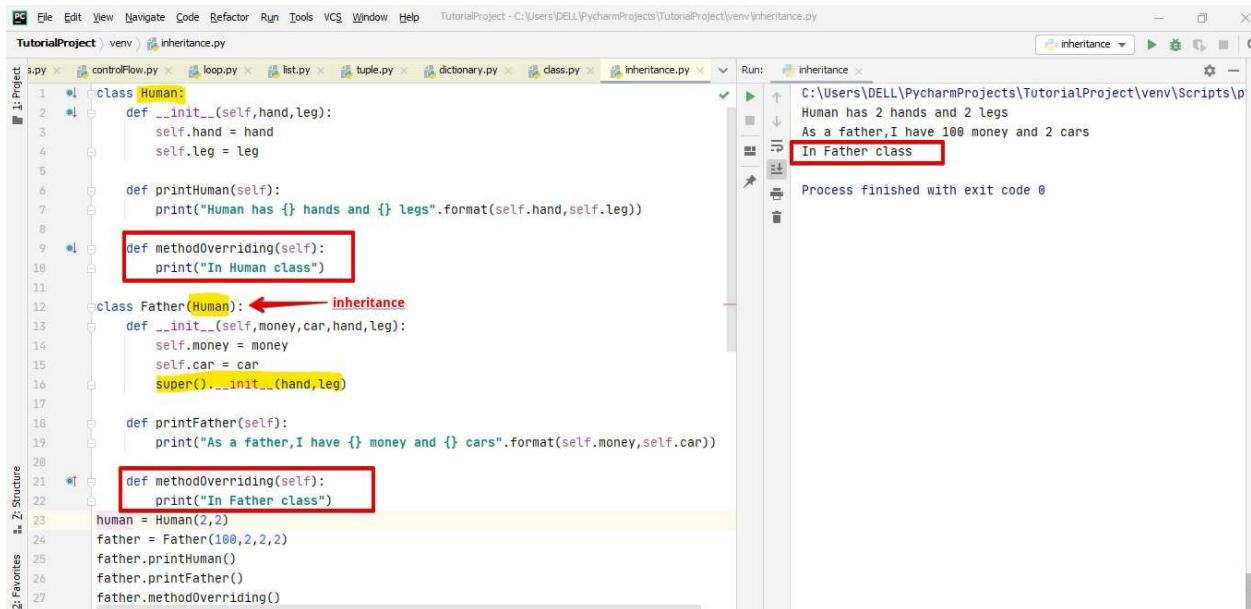
The screenshot shows a PyCharm project window for a file named 'class.py'. The code defines a class 'Person' with an __init__ constructor and a method 'aboutMe'. An object 'Mary' is created with the arguments ('Mary', 'Machael', 43, 'Landon'). The run output shows the printed message: 'Hello , I'm {} and {} years old. I live in {} and My fathername is {}'.format(self.name, self.age, self.address, self.fathername)). Red arrows point from the annotations to the constructor definition and the object creation line.

```

1 class Person:
2     def __init__(self, name, fathername, age, address):
3         self.name = name
4         self.fathername = fathername
5         self.age = age
6         self.address = address
7
8     def aboutMe(self):
9         print("Hello , I'm {} and {} years old. I live in {} and My fathername is {}".format(self.name, self.age, self.address, self.fathername))
10
11 Mary = Person("Mary", "Machael", 43, "Landon") ← object creation
12 Mary.aboutMe()
13
14
15

```

Inheritance



The screenshot shows a PyCharm project window for a file named 'inheritance.py'. It contains two classes: 'Human' and 'Father'. The 'Human' class has an __init__ constructor and a printHuman method. The 'Father' class inherits from 'Human' and adds a __init__ constructor with additional parameters (money, car) and overrides the printHuman method. An object 'father' is created with the arguments (100, 2, 2, 2). The run output shows the printed messages: 'Human has 2 hands and 2 legs' and 'As a father, I have 100 money and 2 cars'. A red box highlights the 'In Father class' message in the output, and another red box highlights the 'method overriding' code in the 'Father' class definition.

```

1 class Human:
2     def __init__(self, hand, leg):
3         self.hand = hand
4         self.leg = leg
5
6     def printHuman(self):
7         print("Human has {} hands and {} legs".format(self.hand, self.leg))
8
9     def methodOverriding(self):
10        print("In Human class")
11
12 class Father(Human): ← inheritance
13     def __init__(self, money, car, hand, leg):
14         self.money = money
15         self.car = car
16         super().__init__(hand, leg)
17
18     def printFather(self):
19         print("As a father, I have {} money and {} cars".format(self.money, self.car))
20
21     def methodOverriding(self):
22        print("In Father class")
23
24 human = Human(2,2)
25 father = Father(100,2,2,2)
26 father.printHuman()
27 father.printFather()
28 father.methodOverriding()

```

Source: <https://docs.python.org/3/tutorial/index.html>