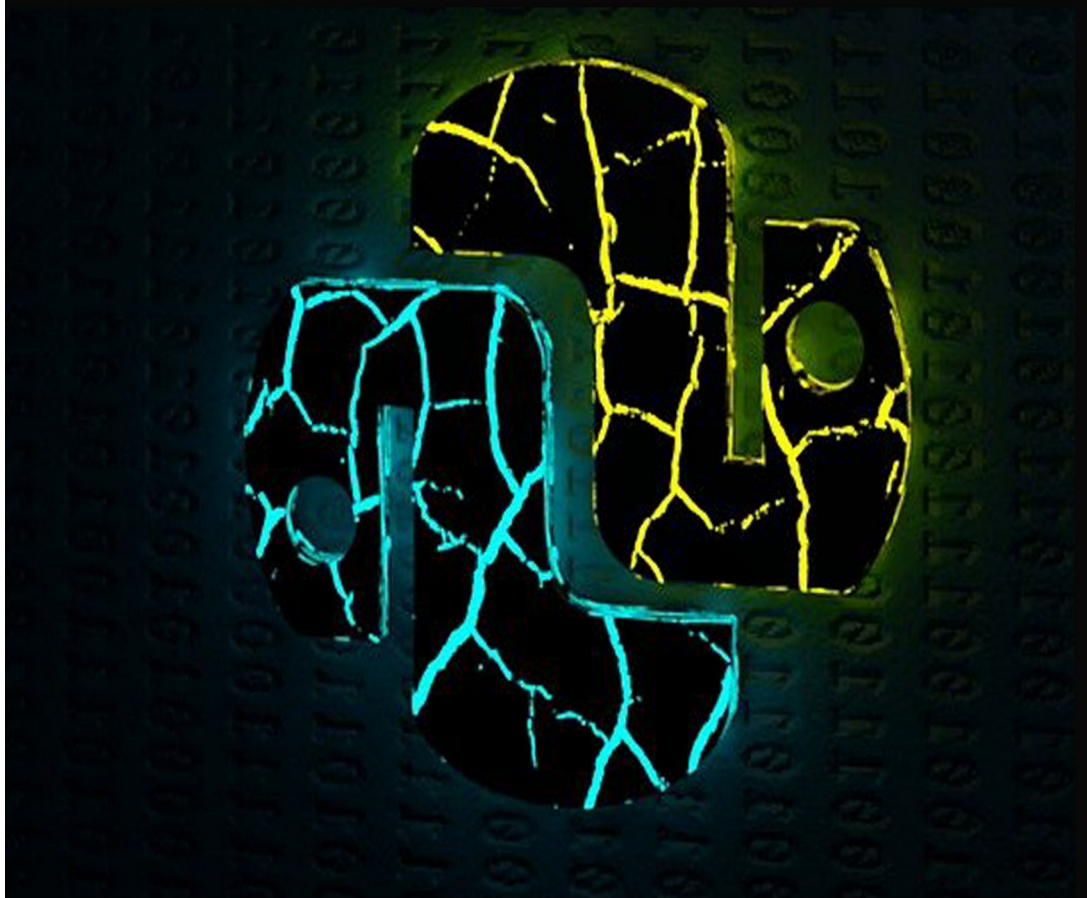


Coding with Python

MYANMAR CYBER YOUTH



Beginner Guide
Written By A coder

Python – 3

Chapter	0
Chapter	1
Chapter	2
Chapter	3
Chapter	4
Chapter	5
Chapter	6
Chapter	7
Chapter	8
Chapter	9
Chapter	10
Chapter	11
Chapter	12
Chapter	13
Chapter	14
Chapter	15
Chapter	16
Chapter	17
Chapter	18

အမှာစာ

ကျနော်ကို ပြုစုစောင့်ရှောက်ပျိုးထောင်ခဲ့သော မိဘနှစ်ပါးကို ပထမဦးစွာ ကျေးဇူးတင်ပါတယ်။
ထို့နောက် ကျနော်ကို သင်ကြားပေးခဲ့သော ဆရာ၊ဆရာမများကို ကျေးဇူးတင်ပါတယ်။ ကျနော်
အားငယ်ချိန်၊ မိသားစု စီးပွားပျက်သွားချိန်တွင် နောက်ကနေ ထောက်ပံ့ပေးသော ကျနော်
ကောင်မလေးကို ကျေးဇူးတင်ပါသည်။ ထို့နောက် လမ်းပျောက်နေချိန် အားလျော့နေချိန် ဘေးကနေ
ဆက်၍ လက်တွဲခေါ်သော ကျနော်ရဲ့နည်းပညာ ညီအကိုမောင်နှမရောင်းရင်းများကို လဲ အထူးပင်
ကျေးဇူးတင်အပ်ပါသည်။ ယခုရေးထားသောစာအုပ် သည် ကျနော် python programming စလေ့လာ
ခါစတွင် လမ်းပျောက်နေတဲ့အချိန် ကျနော်ဖတ်မိလိုက်တဲ့ စာအုပ်တစ်အုပ်ကြောင့် Beginner level
ကနေ တက်မြောက်ခဲ့ပါသည်။ ထို့ကြောင့်ယခု စာအုပ်သည် ထိုစာအုပ်ကို ကိုးကား၍ရေးထား
ပါသည်။ လူငယ်တစ်ယောက်အနေနဲ့ ရေးထားသောကြောင့် အမှားများလဲပါနိုင်ပါသည်။ သို့သော်
အကောင်းဆုံး ရေးဆက်ထားပါသည်။ စာအုပ်ထဲတွင် Theory တင်မကပဲ Challenges များ၊ Example
များနှင့်တွဲဖတ်ရေးထားကြောင့် python ကိုစလေ့မည့်သူများ၊ beginner level ကနေ တစ်ဆင့်တက်
ချင်သူများအတွက် ရည်ရွယ်၍ ရေးထားပါသည်။

စာရေးသူ - A coder (Myanmar Cyber Youth)

CHAPTER-0

Introduction To Python Programming

Computer သည် electronic device တစ်ခုဖြစ်ပြီး Hardware နဲ့ software နှစ်ခု ပေါင်းစပ်ထားသည်။ ကွန်ပျူတာ က software ပိုင်းမရှိရင် electronic box လောက်သာရှိ သည်။ ထို software များကို programming languages များဖြင့်ရေးသားရသည်။ Language ဆိုတာ ဘာသာစကားတစ်ခု တစ်ဦးနှင့်တစ်ဦး၊ တစ်ယောက်နဲ့ တစ်ယောက် စကားပြောစေ ရာတွင် နားလည်စေရန်တီထွင်ထားတဲ့ ဘာသာစကားဆိုလိုသည်။ Computer မှာလဲ ထို သကဲ့သို့ လူနဲ့ စက်ကြားနားလည်စေရန် programming language ကို တီထွင်ခဲ့သည်။ Computer တွေနဲ့ ပထမဆုံး သော programming language သည် binary programming language(1,0) ဖြစ်သည်။ 1 0 10 စသည့်တို့နှင့် software ရေးရာတွင်ရေးရှည်အဆင်မပြေ သ ဖြင့် Assembly language(second generation Programming language) ကို တီထွင်ခဲ့ သည်။ ထို့နောက် C,C++,C#,Java,Python,Php,Javascript စသည့်ဖြင့် အမျိုးမျိုး တီထွင်လာခဲ့ သည်။ထို့ programming များသည် စက်များ မည်သို့လုပ်ဆောင်ရမည်ကို ခိုင်းစေသည်။ဥပမာ- play ဆိုတဲ့ ခလုတ်ကို နိမ့်လျှင် သီချင်းကို ဖွင့်ပါ။ Python programming သည်လည်း ထိုကဲ့သို့ language တစ်ခုဖြစ်သည်။ High-level Programming တစ်ခုဖြစ်သည်။

Python Programming ကို ဘာကြောင့်လေ့လာသင့်လဲ။

Python Programming ရဲ့ syntax(ဖွဲ့စည်းပုံ) သည် နားလည်ရလွယ်ကူပါသည်။ လေ့လာရာတွင် တခြား programming တွေထက် ပိုပြီး ရိုးရှင်းသောကြောင့် beginner တွေ အတွက်အသင့်တော်ဆုံးပါ။ Python Programming ကို Software၊ web၊ Network Automation၊ hacking နှင့် robotic နည်းပညာ စသည့်ဖြင့်နယ်ပယ်မျိုးစုံတွင် အသုံးဝင်လျက် ရှိပါသည်။

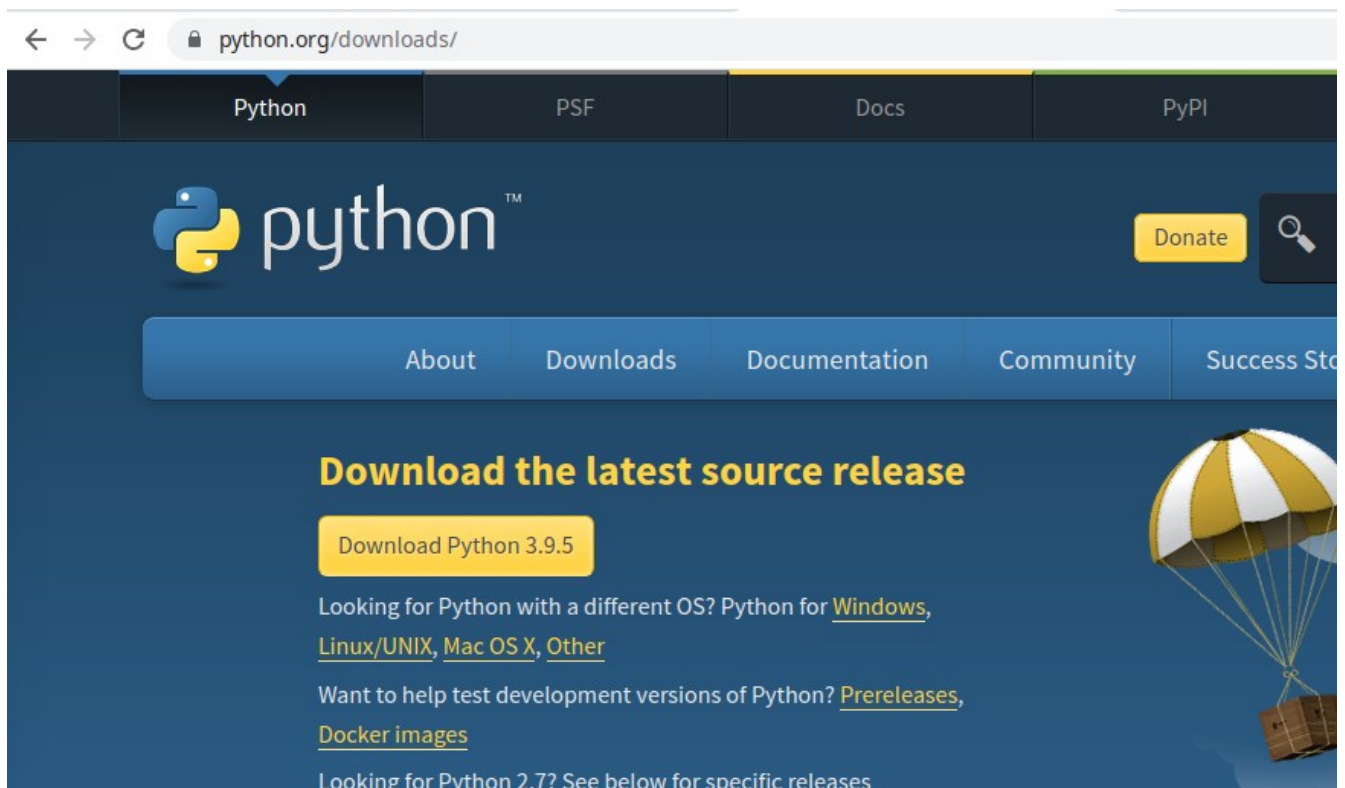
ဆိုကျိုးများ -

ကျနော်အပေါ်မှာ ပြောခဲ့သလိုပဲ နားလည်ရလွယ်ကူသည်။ နယ်ပယ်မျိုးစုံတွင် အသုံးဝင် လျက်ရှိသည်။ ဟိုဟာလေ့လာတော့လဲ လွယ်သယောင်၊ ဒီဟာလေ့လာတော့လဲ လွယ်ကူ စသည့်တို့ကြောင့် စိတ်တွေထွေပြားနေတက်ပါသည်။အကြံပေးလို့တာက ကိုလေ့လာချင်တဲ့ အပိုင်း ကို တွဲဖတ်လေ့လာပါလို့ ပဲ အကြံပေးပါရစေ။

(Note: Second Edition တွင် programming အကြောင်းပိုပြီး ပြည်စုံတင်ပြသွားပါမယ်)

CHAPTER-1

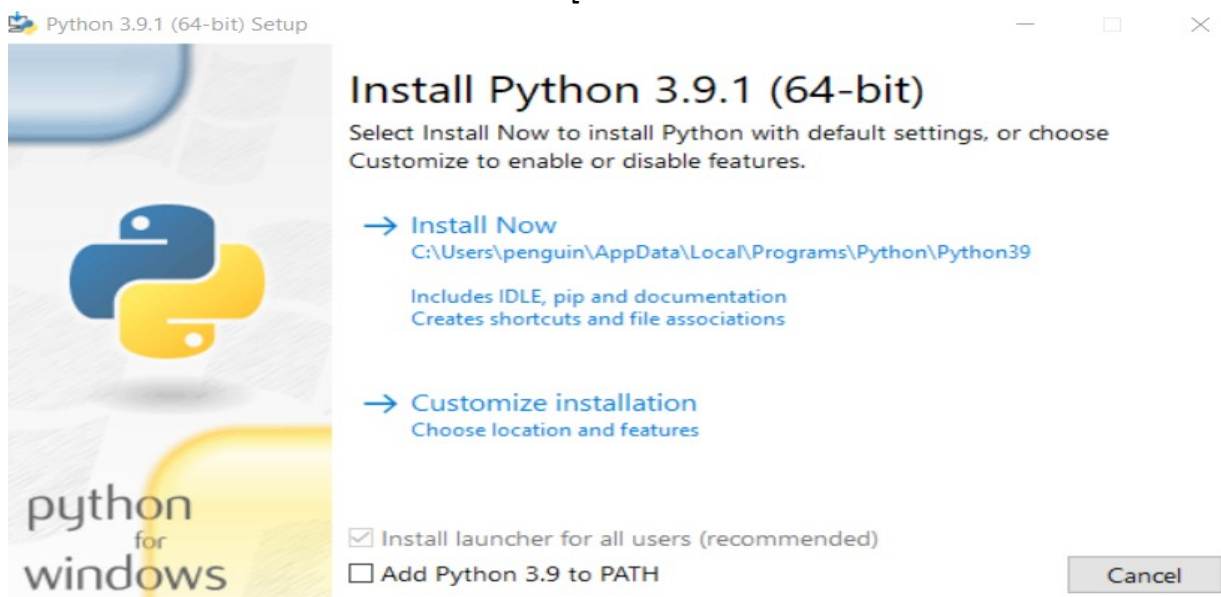
Python3 Software Installation



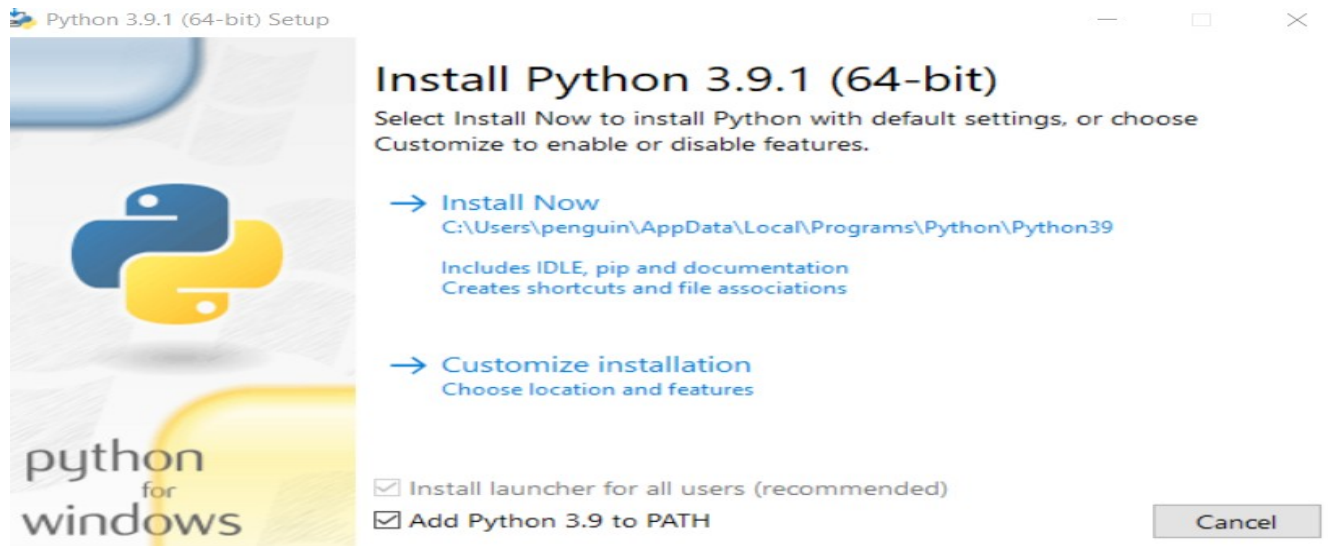
<https://www.python.org/downloads> link ထဲကိုသွားပြီး python idle ကို Download ဆွဲပါ။



.exe file ကို run ပါ။



ယခုကဲ့သို့ပေါ်လာပါလိမ့်မည်။



add python မှာ အမှန် ခြစ်လေးနိမ့်ပြီး Install Now ကိုနှိမ့်ပါ။

Setup Progress

Installing:

Python 3.9.1 Core Interpreter (64-bit)



python
for
windows

Cancel

ဒီလိုပေါ်လာရင် ခနစောင့်ပါ။

Setup was successful

New to Python? Start with the [online tutorial](#) and [documentation](#). At your terminal, type "py" to launch Python, or search for Python in your Start menu.

See [what's new](#) in this release, or find more info about [using Python on Windows](#).



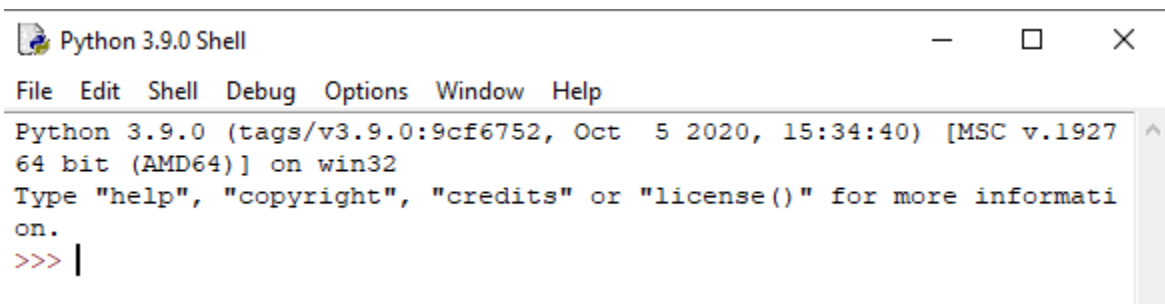
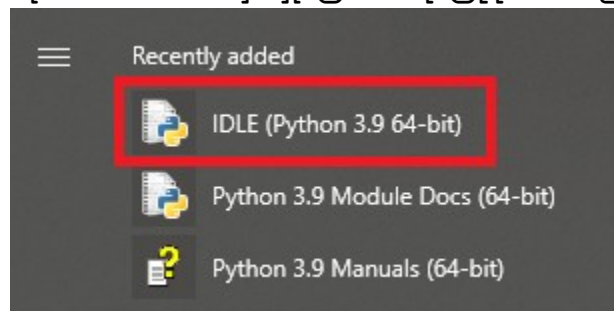
Disable path length limit

Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX_PATH" limitation.

Close

close ကိုနှိပ်ပါ။ python idle ကို install လုပ်ပြီးပါပြီ။

IDLE ကို window ရဲ့ start menu မှာရှာပြီး အသုံးပြုနိုင်ပါသည်။



Main software ပုံစံဖြစ်ပါတယ်။

(Note: Python ရဲ့နောက်ဆုံး version သည် window 7,8,8.1(pro) ဘာညာတွင် install တင် လို့မရတော့ပါ။အကယ်၍ windows 10 မတင်ရသေးရင်လဲ တခြား version များကို ရှာ down နိုင်ပါသည်။)

window ရဲ့ Command Prompt မှာလဲ python လိုရိုက်ပြီးအသုံးပြုနိုင်ပါသေးတယ်။

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.

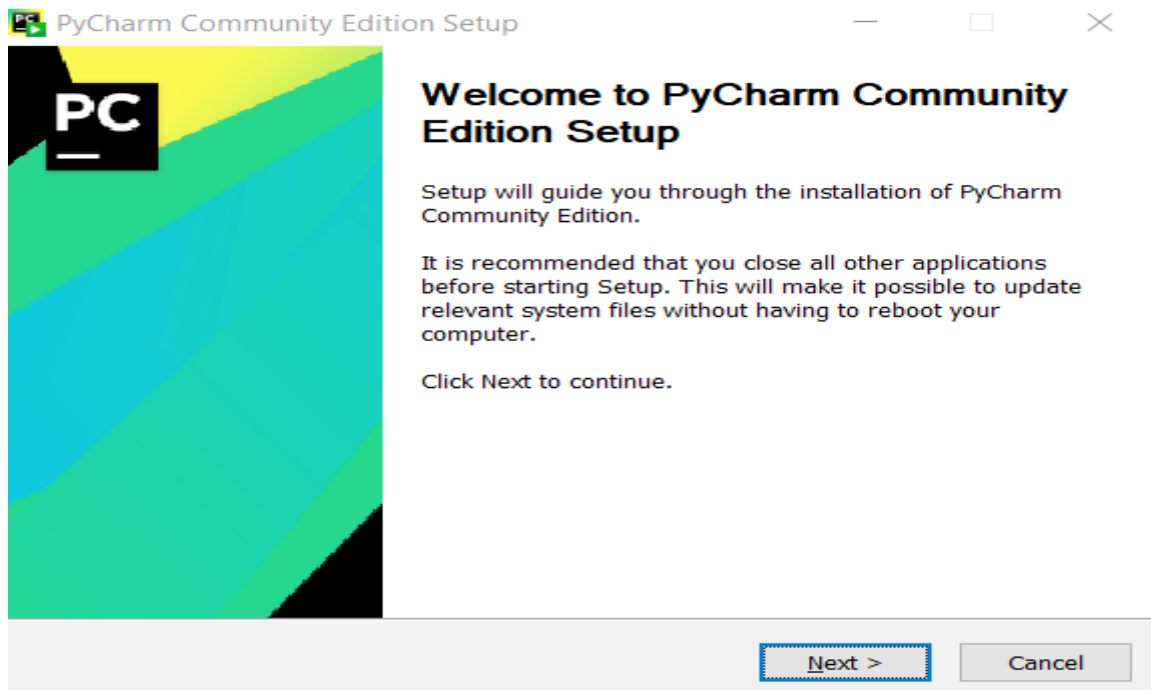
C:\Users\penguin>python
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec  7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print ("Hello World")
Hello World
>>> exit()

C:\Users\penguin>notepad hello.py

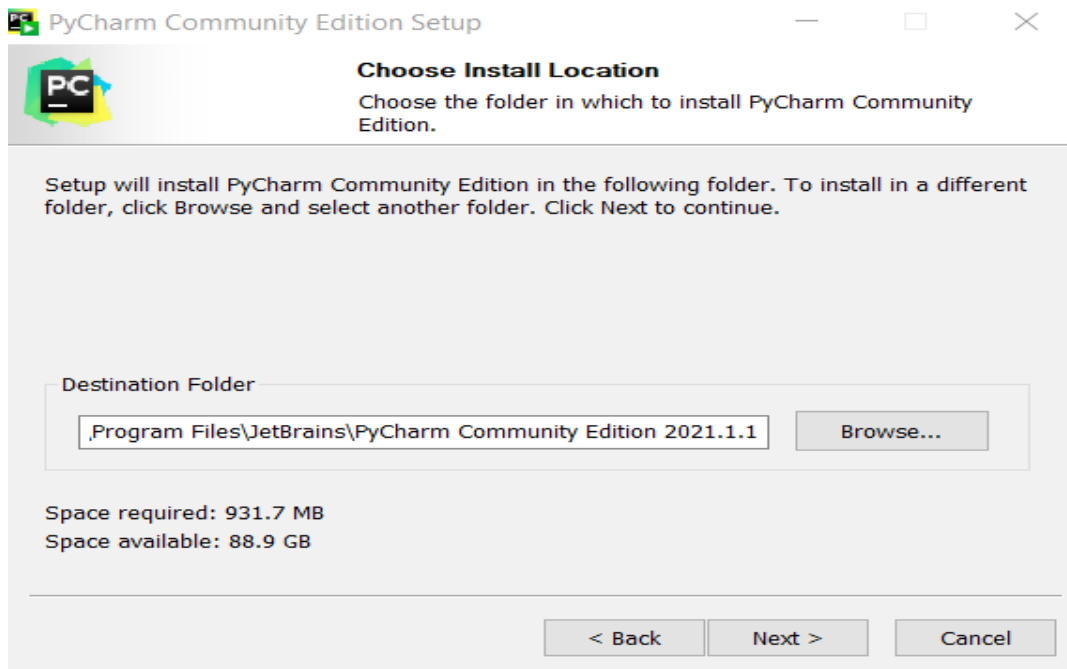
C:\Users\penguin>python hello.py
Hello World

C:\Users\penguin>
```

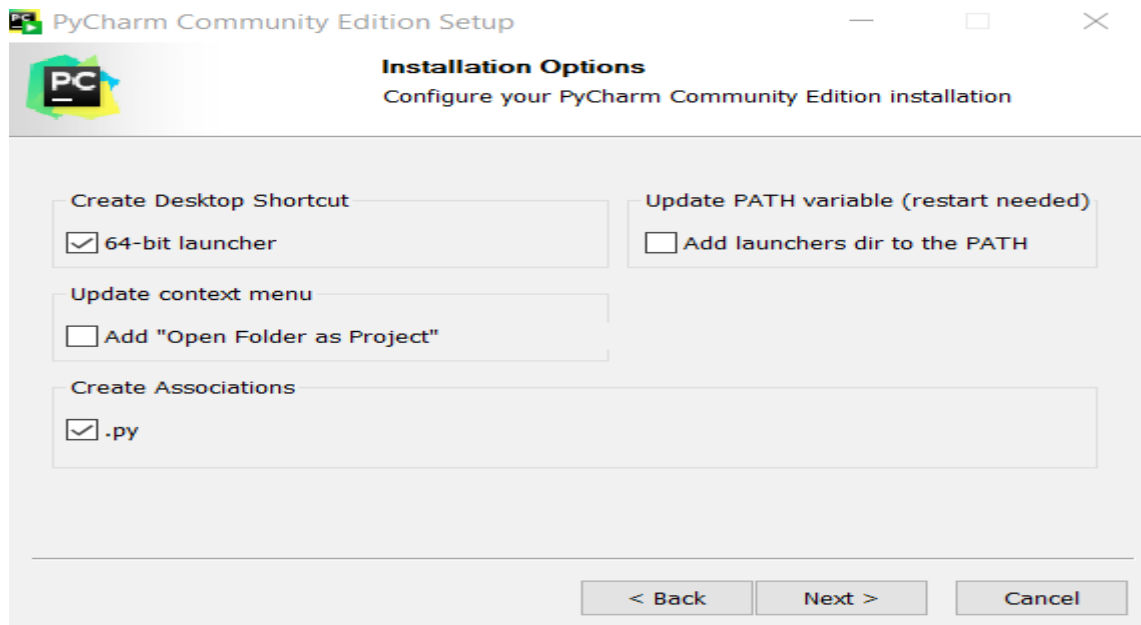
cmd မှာ python လို့ရိုက်လိုက်တာနဲ့ သူရဲ့ version နဲ့အတူ ယခုကဲ့ပေါ်ပါလိမ့်မယ်။ ထို့နောက် print(“Hello Wrld”) ဆိုတာက hello world ဆိုတဲ့ စာသားကို output ထုတ်ပြလိုက်တာဖြစ်ပါတယ်။ exit ဆိုတာတော့ idle က ပြန်ထွက်လိုက်တာဖြစ်ပါတယ်။



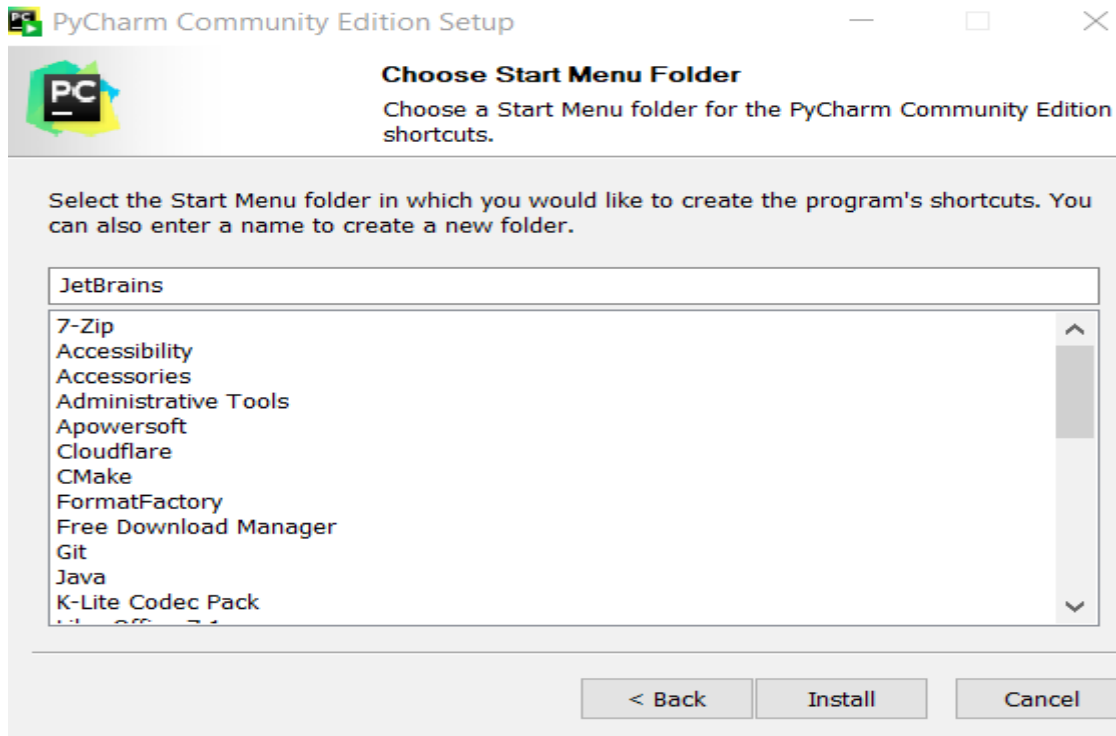
Pycharm exe file ကို နိမ့်လိုက်တဲ့အခါ ယခုကဲ့သို့ပေါ်ရင် next ကိုနိမ့်ပါ။



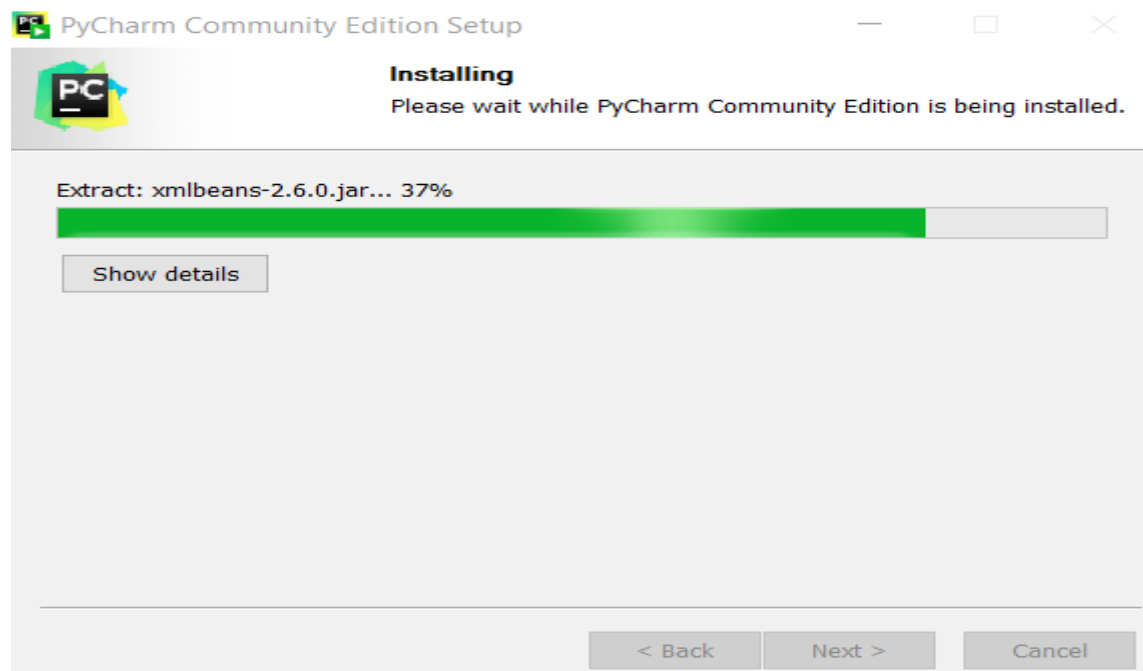
Next ကို ဆက်နိမ့်ပါ



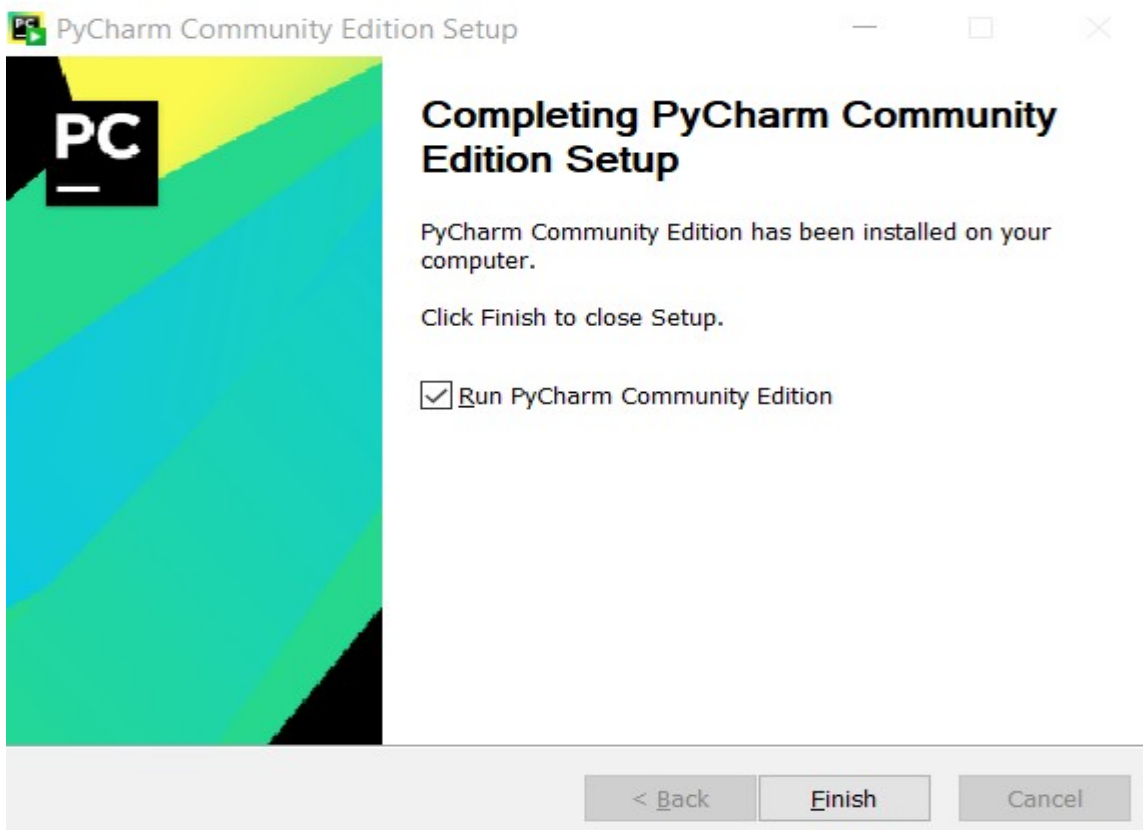
(အကုန်အမှန်ခြစ်လဲရတယ်နော်) Next ကိုနိမ့်ပါ



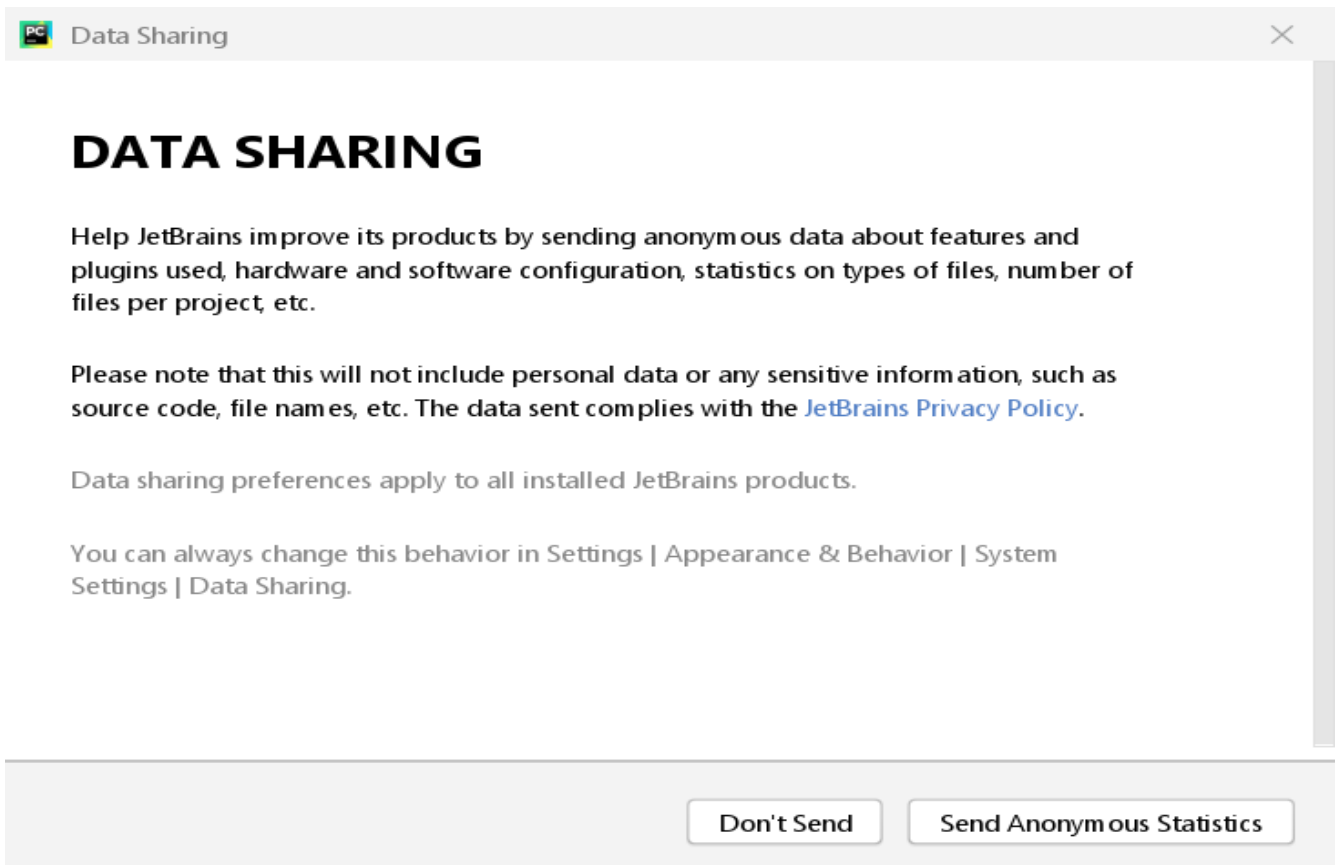
Install ကိုနှိပ်ပါ။



ခနစောင့်ပါ



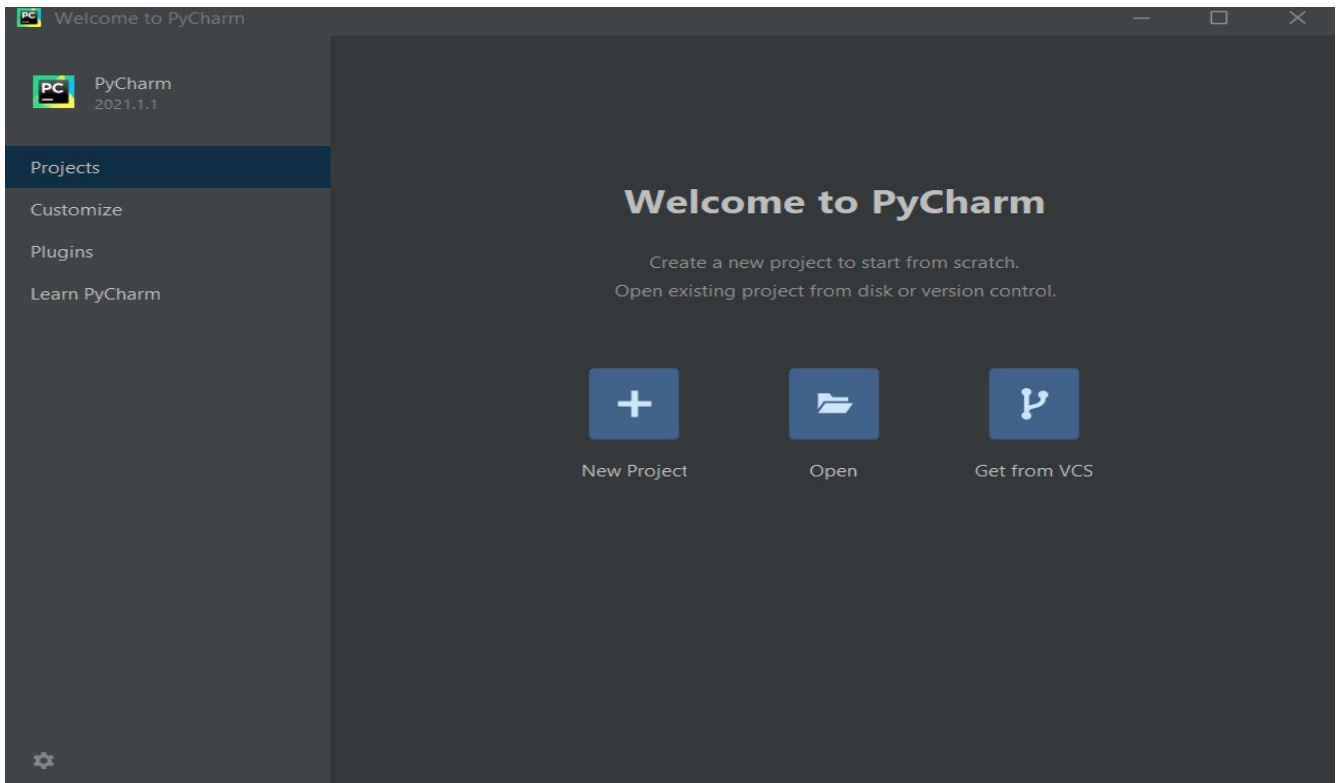
Finish ကိုနှိပ်ပါ



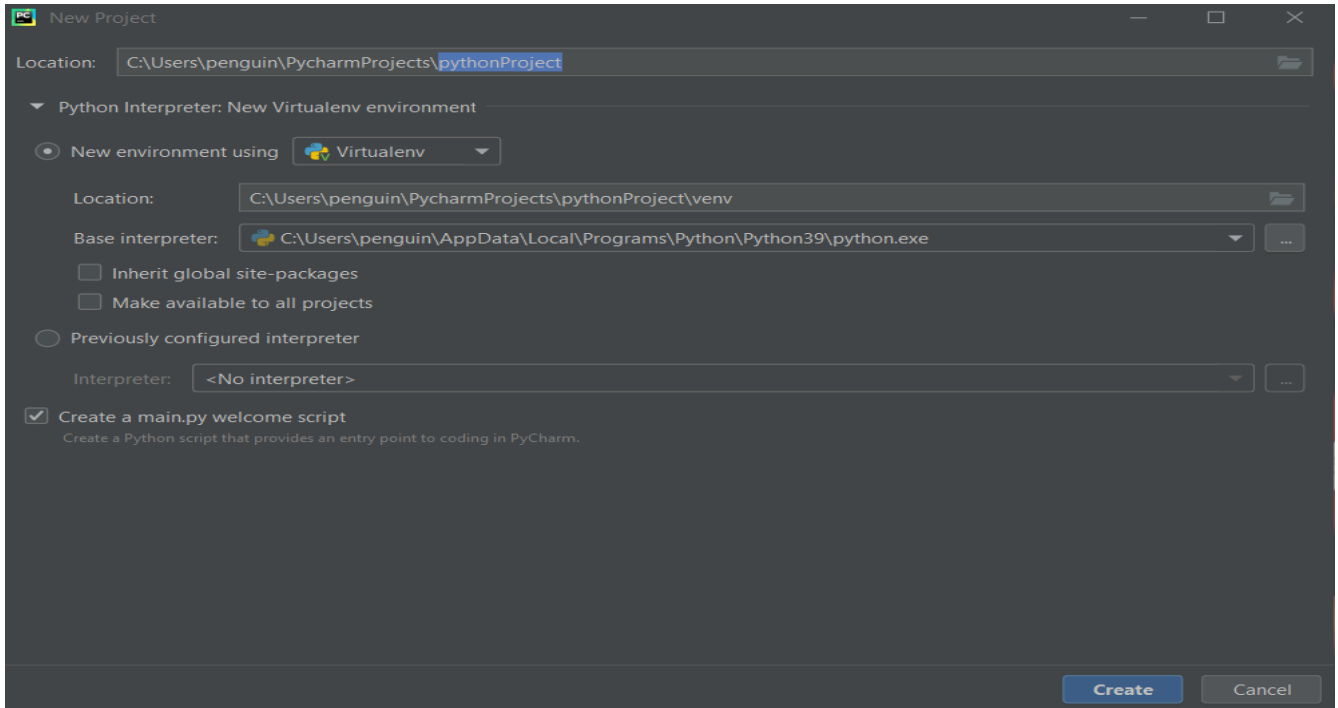
Don't send ကိုနှိပ်ပါ။



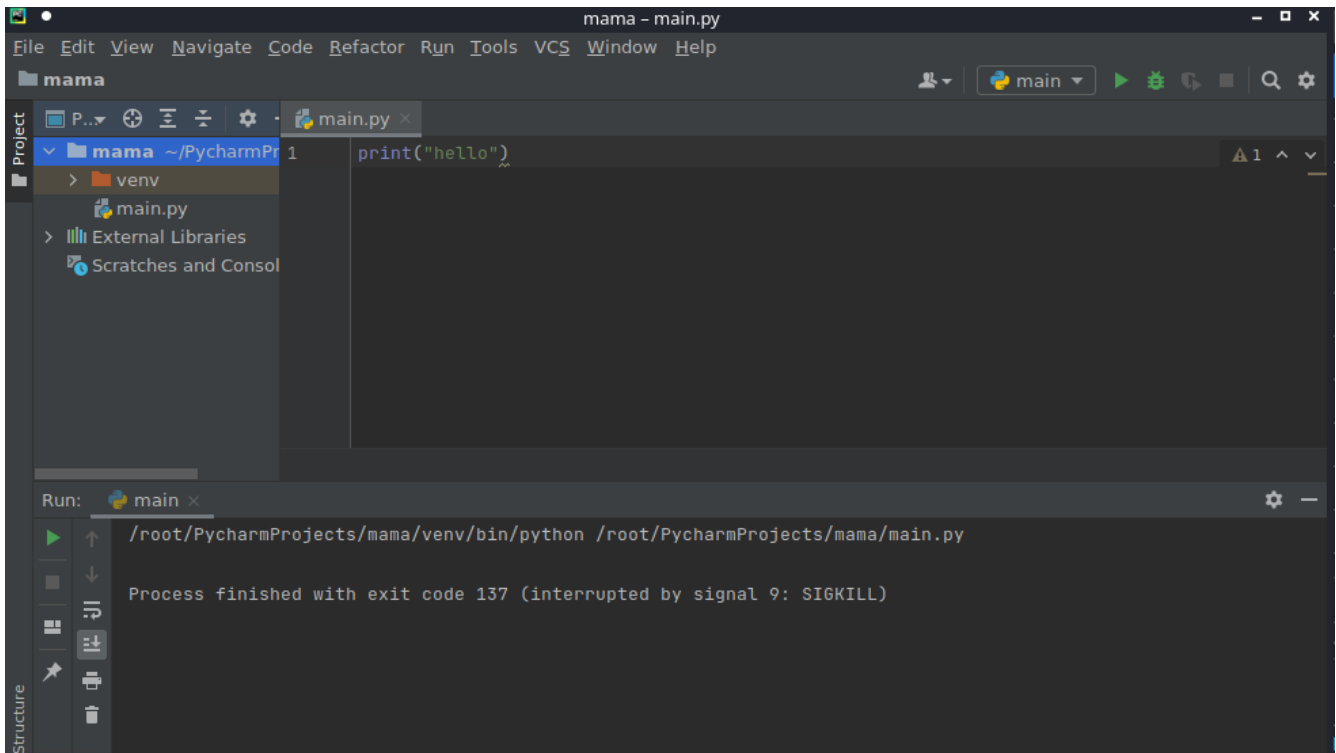
ယခုကွဲသို့ပေါ်လာရင် ခနစောင့်ပါ



new project ကိုနိမ့်ပါ



ယခုကဲ့သို့ပေါ်လာရင် အပြောရောင်စာနေရာမှာကြိုက်တာရေးခွဲပါ(project name ကိုပြောင်းချင်းဖြစ်ပါတယ်)



ဒါကတော့ pycharm ရဲ့ main software ပုံဖြစ်ပါတယ်။ run မယ်ဆိုရင် ဘေးနား အစိမ်းရောင် triangle လေးနိမ့်ရပါမယ်။အောက်က youtube channel မှာ လဲကြည့်ရှုနိုင်ပါတယ်။

Python IDLE installation

<https://youtu.be/29e0E-RUWj4>

Pycharm installation

<https://youtu.be/SZUNUB6nz3g>

(Note: Photo တွေနဲ့ရှုပ်နေမှာဆိုလို့ ယခုလို့ပေးလိုက်တာဖြစ်ပါတယ်။)

CHAPTER-2

2.0 Print Function

Python ရဲ့ print function ကို output ပြန်ထုတ်တဲ့နေရာမှာ အသုံးပြုပါတယ်။ အောက်မှာကြည့်ပါ။

```
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello")
Hello
>>>
```

Hello ဆိုပြီး output ထွက်လာတာကို မြင်ပါလိမ့်မယ်။

Python comment

Comment ကို output ထုတ်ပေးမည်မဟုတ်ပါ။သူတို့ note အဖြစ်မှတ်ထားစေရန်အတွက်အသုံးပြုသည်။

```
# python3
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> #hello
>>> print("hello")
hello
>>>
```

Challenge one – သင့်ရဲ့ အမည်ကို output ထုတ်ပြပါ။

challenge two – သင့်အချစ်ခင်ရဆုံးသူရဲ့ အမည်ကို output ထုတ်ပြပါ။

My solution

Challenge 1 (C-1)

```
print("Htet Arkar Kyaw")
```

C-2

```
print("Ma Ma")
```

Chapter - 3

3.0 Variable

Variable ဆိုတာ ဘာလဲ။

Variable ဆိုတာ Value တွေသိုလှောင်ထားတဲ့ container တစ်ခုပါ။ Container တွေ(မှားလို့ variable တွေ) ကိုနာမည် ပေးဖို့လိုတယ်ဗျ။ သူက data တွေကိုယာယီပဲ သိုလှောင်တယ်။နောက်ပိုင်းမှာ ပိုမို နားလည်လာပါလိမ့်မယ်။နာမည်တွေကို python keyword တွေကလွဲကျန်တာပေးလို့ရပါတယ်။ သူမှာ rule ရှိတယ်ဗျ။

Example 3.0

```
>>> a=1
>>> print(a)
1
>>> █
```

အပေါ်က code တွေကိုကြည့်ရင် 1 ဆိုတဲ့တန်ဖိုး ကို a ထဲမှာ သိုလှောင်လိုက်တယ်။ ထို့နောက် print Function နဲ့ ပြန်ပြီး output ထုတ်ပေးပါတယ်။

3.1

Variable Name ကိုသတ်မှတ်ခြင်း။ variable name ကို python ရဲ့ Keywords တွေကိုပေးလို့မရပါဖူး။

```
help> keywords

Here is a list of the Python keywords.  Enter any keyword to get more help.

False          break          for            not
None           class         from           or
True           continue      global         pass
__peg_parser__ def           import         raise
and            del           in             return
as             elif          is             try
assert         else          lambda         while
async          except        nonlocal       with
await          finally

help> █
```

Python Keywords တွေဖြစ်ပါတယ်။တစ်နည်းပြောရရင် python ရဲ့ built-in function တွေဖြစ်ပါတယ်။ သူတို့ကို variable name အဖြစ်ပေးလို့မရပါဖူး။

Example

```
>>> True="a"
      File "<stdin>", line 1
        True="a"
        ^
SyntaxError: cannot assign to True
>>>
```

Python keyword ထဲက True ကို variable name အဖြစ်သတ်မှတ်လိုက်ပေမဲ့ သတ်မှတ်လို့မရကြောင်း syntax Error တက်ပြပါတယ်။(syntax error ဆိုတာ ဖွဲ့စည်းပုံမမှန်တာပါ)

နောက်တစ်ချက် က variable name တွေထဲမှာ operator တွေထည့်ပေးလို့မရပါဖူး။ထည့်ခဲ့ရင် လည်း syntax Error တက်ပါတယ်။

```
SyntaxError: cannot assign to operator
>>> a-b="e"
      File "<stdin>", line 1
        a-b="e"
        ^
SyntaxError: cannot assign to operator
>>>
```

နောက်ဆုံးတစ်ချက်ကတော့ underscore နဲ့ variable name ပေးလို့ရပါတယ်။

```
>>> u_name="Htet"
>>> print(u_name)
Htet
>>>
```

ယခုလောက်ဆိုနားလည်ပြီးထင်ပါဗျ။

Example 3.2

```
>>> b="Hello"
>>> print(b)
Hello
>>>
```

code အရဆိုရင် hello ဆိုတဲ့ output ကိုထုတ်ပေးတာကိုမြင်ပါလိမ့်မယ်။

C-3

သင့်အကြိုက်ဆုံးနံပါတ်ကို c ဆိုတဲ့ variable ထဲမှာ သိုလှောင်လိုက်ပါ။ ထို့နောက်ပြန်ပြီး output ထုတ်ပြပါ။

C-4

သင့်ချစ်ရတဲ့ လူတစ်ယောက်ရဲ့ အမည်ကို heart ဆိုတဲ့ variable ထဲမှာ သိုလှောင်လိုက်ပါ။ ထို့နောက် output ပြန်ထုတ်ပြပါ။

C-3

c=9

print(c)

9

C-4

heart="Top Secret"

print(heart)

Top Secret

Chapter- 4

4.0

Python String Data Type

String Data Type ဆိုတာ alphabet တစ်နည်း စကားလုံးတွေကို ဖြစ်ပါသည်။ String ကို double quote or single quote ကြားထဲမှာ ရေးရပါတယ်။

Example

```
Python 3.9.1+ (default, Feb 5 2021, 13:46:56)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("I'm robot")
I'm robot
>>> print('Hello')
Hello
>>> █
```

Challenge ==>

C-5 => သင့်ရဲ့နာမည်ကို double quote နဲ့ output ထုတ်ပြပါ။

C-6 ==> သင့်ရဲ့အကြိုက်ဆုံးအစားစာကို single quote နဲ့ output ထုတ်ပြပါ။

Solution

C-5 ==> print("Htet Arkar Kyaw")

C-6==> print('Banana')

(Note: Single quote နဲ့ Double Quote အသုံးပြုပုံတစ်ချို့နောက်ပိုင်းသင်ခန်းစာမှာဖော်ပြထားပါတယ်။)

4.1 Numeric Data Type

Numeric Data Type ကတော့ ကိန်းဂဏန်းကို သိုလှောင်ဖို့အတွက်အသုံးပြုပါသည်။ Numeric Data Type ကတော့ လေးမျိုးရှိပါတယ်။ ၎င်းတို့မှာ -

1. Integer = 1234(ကိန်းပြည့်)
2. Float = 1.2 (ဒဿမကိန်း)
3. long = 123456lab(ရှည်လျားကိန်း)
4. complex = 123.12ab(ရှုပ်ထွေးနေတဲ့ကိန်း)

(Note: Numeric Data Type တွေကို single quote or double quote နဲ့ အသုံးမပြုပါ။)

Example 4.1.0

```
>>> print(1)
1
>>> █
```

Explain

print(1) ဆိုပြီး 1 ဆိုတဲ့ integer data type ကို output ထုတ်ပြတာဖြစ်ပါတယ်။

Example 4.1.2

```
>>> float=1.4
>>> print(float)
1.4
>>> █
```

Explain

float ဆိုတဲ့ Variable ထဲကို 1.4 ဆိုတဲ့ float data type ကို သိုလှောင်လိုက်တာဖြစ်ပါတယ်။ ထို့နောက် print Function နဲ့ ပြန်ပြီး output ထုတ်ပါတယ်။

4.2 Type Function

Type function ဆိုတာ ကိုယ်သုံးတဲ့ data type အမျိုးအစားကို သိချင်ရင် type function ကို အသုံးပြုပါတယ်။


```
print(type("String"))
print(type(1))
print(type(1.4))
```

main x

```
/root/PycharmProjects/mama/venv/bin/python /root/PycharmPro
<class 'str'>
<class 'int'>
<class 'float'>

Process finished with exit code 0
```

Explain

ကိုယ်သုံးတဲ့ Data type အမျိုးအစားကို ကြည့်ချင်ရင် Type Function ကို အသုံးပြုပါတယ်။

Line 1: string ဟုတ်မဟုတ် သိချင်လို့ type function နဲ့ စစ်ကြည့်တာဖြစ်ပါတယ်။

Line 2: 1 က integer ဖြစ်တဲ့အတွက် output မှာ class 'int' ဆိုတာ integer ဖြစ်တယ်လို့ပြောတာပါ။

Line 3: 1.4 က float type ဖြစ်တဲ့အတွက် type နဲ့ print ထုတ်တဲ့အခါ float လို့ ရလဒ်ထွက်ပါတယ်။

4.3 Python Arithmetic Operator

Python Operator ထဲ က Arithmetic Operator အကြောင်းကိုဖော်ပြသွားမှာပါ။ ကျန် Operators များကို နေရာအလိုက်ဖော်ပြသွားမှာပါ။ Arithmetic Operators တွေမှာ -

Symbol(Operator)	Name(Description)	Example
+	Addition	a+b
-	Subtraction	a-b
*	Multiplication	a*b
/	Division	a/b
%	Modulus	a%b

Exponentiation

ab**

//

Floor Division

a//b

Arithmetic Operators တွေကိုကြည့်ယုံနဲ့သိမယ်ထင်ပါတယ်ခင်ဗျာ။

Example ==>

```
~ : python3 — Konsole
File Edit View Bookmarks Settings Help
(soloninja@kali)-[~]
$ python3
Python 3.9.1+ (default, Feb  5 2021, 13:46:56)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> a=3
>>> b=7
>>> a+b
10
>>> b*a
21
>>> b-a
4
>>> b/a
2.3333333333333335
>>> b%a
1
>>> a**b
2187
>>> a//b
0
>>> b//a
2
>>> 
```

Operators တွေရဲ့ အလုပ်လုပ်ပုံက တော့ အထက်ပါပုံအတိုင်းဖြစ်ပါတယ်ခင်ဗျာ။

C-7

Num1 ဆိုတဲ့ variable ထဲမှာ 5 ဆိုတဲ့ integer data type ကိုသိုလှောင်ပါ။

Num2 ဆိုတဲ့ variable ထဲမှာ 23 ဆိုတဲ့ integer data type ကိုသိုလှောင်ပါ။

Challenge One

Num1 နဲ့ Num2 ပေါင်းထားတဲ့ ဂဏန်းကို ans ဆိုတဲ့ variable ထဲမှာသိုလှောင်ပါ။ output ပြန်ထုတ်ပါ။

C-8

Num2 ထဲကနေ Num1 ကိုနုတ်ထားတဲ့ ဂဏန်း ကို ans လို့အမည်ပေးထားတဲ့ variable ထဲမှာ သိုလှောင်ပါ။ ထို့နောက် output ပြန်ထုတ်ပါ။

(Note အပေါ်က ပေးထားတဲ့ Arithmetic operators နဲ့အတူ တွဲဖတ်လေ့ကျင့်ပါ။)

Solution

C-7

```
Num1=5
```

```
Num2=23
```

```
ans=Num1+Num2
```

```
print(ans)
```

C-8

```
Num1=5
```

```
Num2=23
```

```
ans=Num2-Num1
```

```
print(ans)
```

CHAPTER-5

5.0 User Input Function

ကျနော်တို့ user က ထည့်သွင်းတဲ့ လိုက်တဲ့ data တွေကို လက်ခံဖို့ user input function ဆိုတာရှိပါတယ်။

Example 5.0.0

```
name=input("Enter Your Name: ")
```

အပေါ်က code ကိုကြည့်မယ်ဆိုရင် user name ကို input ထည့်သွင်းခိုင်းတဲ့ code ဖြစ်ပါတယ်။

နားလည်သွားအောင် အောက်ပါ example ကိုကြည့်ပါ။

```
# python3
Python 3.9.1+ (default, Feb  5 2021, 13:46:56)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> name=input("Enter Your Name: ")
Enter Your Name: Htet Arkar Kyaw
>>> print(name)
Htet Arkar Kyaw
>>>
```

Input Function နဲ့ user name ကို ထည့်သွင်းခိုင်းသည်။ ထို့နောက် name ဆိုတဲ့ variable ထဲကို သိုလှောင်လိုက်သည်။ print function နဲ့ name ကို ပြန်ပြီး output ထုတ်သည်။
(Note: input function ထဲ ထည့်လိုက်သမျှကို python က string (စာသား) အနေနဲ့ပဲ သိပါတယ်)

5.1 Input Function

a=str(input("Enter Your Mother Name: ")) ဆိုရင် user ထည့်သွင်းတဲ့ အချက်အလက်ကို string data type အနေနဲ့ သိမ်းဆည်းတယ်။

b=int(input("Enter A Number: ")) ဆိုရင် user ထည့်သွင်းတဲ့ data ကို integer data type အနေနဲ့ သိမ်းဆည်းပါသည်။

C-10 Input နဲ့ သင့်ရဲ့အမည်ကို ထည့်သွင်းပါ။ ထို့နောက် ရလဒ်ပြန်ထုတ်ပြပါ။

C-11 Input နဲ့ သင့်ချစ်သူနာမည်ကိုထည့်သွင်းပါ။ ထို့နောက် အရမ်းချစ်တယ် ဆိုတဲ့ စာသားနဲ့ ရလဒ် ထုတ်ပြပါ။

C-12 ကိန်းဂဏန်း နှစ်ခုကို input function နဲ့ ထည့်သွင်းပါ။ ထိုနှစ်ခုပေါင်းထားတဲ့ရလဒ်ကို ပြန် ထုတ်ပြပါ။

C-13 ပိုက်ဆံအိတ်ထဲရှိတဲ့ ငွေပမာဏကိုထည့်သွင်းပါ။ ထပ်မံပြီး သင့်ဝယ်လိုက်တဲ့ ပစ္စည်း ရဲ့ ငွေ ပမာဏ ကို ထည့်သွင်းပါ။ ထို့နောက် ပစ္စည်းရဲ့ တန်ဖိုးကို ပိုက်ဆံအိတ်ထဲက နုတ်ပါ။ result ပြန် ထုတ်ပါ။

C-14 ပုံစား ထမင်းတစ်ပွဲရဲ့ တန်ဖိုးကို ထည့်သွင်းပါ။ စားသောက်တဲ့ လူအရေအတွက်ကို ထည့်သွင်း ပါ။ “စုစုပေါင်းကုန်ကြေး = “ ဆိုတဲ့စာသားနဲ့အတူ ရလဒ်ပြန်ထုတ်ပြပါ။

Solution

C-10

```
name=input("Enter Your Name: ")
print(name)
```

C-11

```
mylove=input("Enter Your Girlfriend name: ")
print("I love you so much", mylove)
```

C-12

```
a=int(input("Enter First Number: "))
b=int(input("Enter Second Number:"))
c=a+b
print(c)
```

C-13

```
a=int(input("Money in the my pocket: "))
b=int(input("The things money: "))
c=a-b
print("my money is ",c)
```

C-14

```
rice=int(input("ပုံးစားထမင်းတစ်ပွဲရဲ့ငွေပမာဏ: "))
people=int(input("စားသုံးသူလူဦးရေ: "))
amount=rice*people
print("စုစုပေါင်းကုန်ကြေး =", amount))
```

(Note: English လို့ထည့်ရေးလို့ရပေမဲ့ တစ်ချို့သောလူတွေအတွက်အဆင်ပြေအောင် ယခုကဲ့သို့ရေးသားခြင်းဖြစ်သည်)

CHAPTER - 6

Tuple,List,Dictionary

6.0 Tuple

Tuple ဆိုတာ data တွေကို သိုလှောင်ထားတဲ့ list တစ်ခုပဲဖြစ်သည်။ Tuple ထဲ ထည့်ထားတဲ့ data တွေကို ပြုပြင်ပြောင်းလဲလို့မရပါဖူး။ ဒါကြောင့် tuple ကို ပြုပြင်စရာမလိုတဲ့နေရာမှာပဲ အသုံးပြုပါတယ်။

6.1 List

List ဆိုတာ က data ကို သိုလှောင်ထား list data type တစ်ခုပဲဖြစ်ပါ။ Tuple နဲ့ ကွာခြားတဲ့ အချက်က List ထဲက Data တွေကိုပြုပြင်ပြောင်းလဲနိုင်ပါတယ်။

6.2 Dictionary

Dictionary ဆိုတာ key နဲ့ value နှစ်ခုတွဲပြီးသိုလှောင်ထားတဲ့ data type တစ်မျိုးပဲဖြစ်ပါတယ်။

6.3 Tuple

Tuple ဆိုတာ Data တွေကို သိုလှောင်ထားတဲ့ list တစ်ခုဖြစ်တယ်ဆိုတာ သိနှင့်ပြီးသားဖြစ်မှာပါ။ Tuple lists ထဲက data တွေကို round bracket ကြားမှာရေးပါတယ်။ တစ်ခုနဲ့ တစ်ခုကို comma နဲ့ ခြားထားပါတယ်။

Example 6.3.0

```
>>> tuple=("apple","orange","banana")
>>> print(tuple)
('apple', 'orange', 'banana')
>>>
```

Description

tuple ဆိုတဲ့ variable ထဲကို String data types တွေသိုလှောင်လိုက်တာဖြစ်ပါတယ်။ ထို့နောက် print function နဲ့ ပြန်ပြီး output ထုတ်ပြပါတယ်။

6.4

ကိုယ်လိုချင်တဲ့ item တစ်ခု သို့ items တွေကို ကြည့်တဲ့အခါမှာသုံးတဲ့ function ကိုပြောပြပါမယ်။

```
>>> phone=("Vivo","Oppo","Sony")
>>> print(phone.index("Oppo"))
1
>>> print(phone[1])
Oppo
>>> print(phone[0:])
('Vivo', 'Oppo', 'Sony')
>>> print(phone[0:2])
('Vivo', 'Oppo')
>>>
```

Description

Line 1: Phone ဆို variable တစ်ခုကိုတည်ဆောက်လိုက်တာဖြစ်ပါတယ်။

Line 2: index("Oppo") ဆိုတာ Oppo ရဲ့ numeric key ရှိတဲ့နေရာကို output ထုတ်ပြတာဖြစ်ပါတယ်။ list နဲ့ Tuple မှာ 0 ကနေစတင်တာဖြစ်သောကြောင့် Oppo သည် 1 နေရာတွင်ရှိနေသည်။

Line 3: ထို့ကြောင့် တတိယကြောင်းတွင် 1 လို့ပေါ်နေခြင်းဖြစ်သည်။

Line 4: ကျနော်တို့ items တစ်ခုတည်းကိုရွေးချယ်ကြည့်ချင်တဲ့အခါမှာ square bracket ကို အသုံးပြုပါတယ်။ item ရဲ့ location ကို [] ကြားထဲမှာ ထည့်ရေးရပါတယ်။ ယခု စာကြောင်း print(phone[1]) ဆိုတာ phone tuple ထဲမှာရှိတယ် နံပါတ် ၂ နေရာမှာ ရှိတဲ့ items ကို Display ပြသတာဖြစ်ပါတယ်။

Line 5: print(phone[0:]) ဆိုတာက tuple ရဲ့ 0 ကနေ နောက်ဆုံးနေရာထိ output ထုတ်ပြတာဖြစ်ပါတယ်။

Line 6: print(phone[0:2]) ဆိုတာက 0 နဲ့ 2 ကြားက items တွေကို output လုပ်ချင်တဲ့ အခါမှာအသုံးပြုပါတယ်။

(Note: ယခု Functions တွေကို list data type မှာလဲ အသုံးပြုနိုင်ပါသည်။)

C-15

သင့်သိတဲ့ အသီးတွေကို tuple list တစ်ခုပြုလုပ်ပြီး fruit ဆိုတဲ့ variable ထဲ သို့လှောင်လိုက်ပါ။ ထို့နောက် output ပြန်ထုတ်ပြပါ။ သစ်သီးအနည်းဆုံး ငါးလုံးလောက်ရှိပါစေ။

C-16

fruit variable ထဲက အသီးသုံးလုံးတစ်လုံးတည်းကို Output ထုတ်ပြပါ။

C-17

Fruit Variable ထဲက အသီး ၃ လုံးမြောက်ကနေစပြီး နောက်ကဟာတွေ အကုန်လုံးကို output ထုတ်ပြပါ။

C-18

Fruit Variable ထဲက အသီး 0 ကနေ 4 ကြားထဲရှိ အသီးတွေကို output ထုတ်ပြပါ။

Solution

C-15

```
fruit=("Mengo","Banana","Strawberry","Apple","Orange")  
print(fruit)
```

C-16


```
fruit=("Mengo","Banana","Strawberry","Apple","Orange")
print(fruit[3])
```

C-17

```
fruit=("Mengo","Banana","Strawberry","Apple","Orange")
print(fruit[2:])
```

C-18

```
fruit=("Mengo","Banana","Strawberry","Apple","Orange")
print(fruit[0:4])
```

6.5 List

List မှာဆို items တွေကို square bracket ကြားမှာ သိုလှောင်ပါတယ်။ တစ်ခုနဲ့ တစ်ခုကို comma နဲ့ခြားထားပါတယ်။ List ထဲ က items တွေကိုတော့ပြုပြင်ပြောင်းလဲလို့ရပါတယ်။

```
>>> list=["a","b","c","d",1,2,3]
>>> print(list)
['a', 'b', 'c', 'd', 1, 2, 3]
```

Description

strings and integers စတဲ့ Data type နှစ်ခုလုံးကို စာရင်းစီပြီး list ဆိုတဲ့ Variable ထဲမှာ သိုလှောင်လိုက်ပါတယ်။ထို့နောက် Output ပြန်ထုတ်ပြပါတယ်။

6.6

list ထဲမှာ ရှိတဲ့ data တွေပြောင်းလဲပုံကိုရှင်းပြမှာဖြစ်ပါတယ်။ list ထဲ က items တွေကို ထုတ်ကြည့်ပုံကို Tuple က ပုံစံနှင့်တူသောကြောင့်ထပ်မံရှင်းပြတော့ပါ။

```
>>> list=["a","b","c","d",1,2,3]
>>> print(list)
['a', 'b', 'c', 'd', 1, 2, 3]
>>> list.insert(4,'e')
>>> print(list)
['a', 'b', 'c', 'd', 'e', 1, 2, 3]
>>>
```

Description

Line :1 နဲ့ Line 2 အပေါ်မှာရှင်းပြခဲ့ပြီးဖြစ်ပါပြီ။

Line :3 insert() function က data အသစ်ထည့်တဲ့နေရာမှာအသုံးပြုပါတယ်။ အရှေ့က နံပါတ် 4 က နေရာပါ။ ကျနော်တို့ရဲ့ စာရင်းအသစ်ကိုထည့်မဲ့နေရာပါ။ နောက်က e ဆိုတာက ကျနော်တို့ထည့်တဲ့ data or item တစ်ခုပင်ဖြစ်သည်။

Line 4: list ပြန်ထုတ်ကြည့်တဲ့အချိန်မှာ d ရဲ့ နောက်တွင် e ရှိနေတာကိုတွေ့ပါလိမ့်မယ်။

6.7

```
>>> print(list)
['a', 'b', 'c', 'd', 'e', 1, 2, 3]
>>> list.append('4')
>>> print(list)
['a', 'b', 'c', 'd', 'e', 1, 2, 3, '4']
```

Description

Append() ဆိုတာ data items အသစ်ကိုနောက်ဆုံးမှာထည့်ချင်တဲ့အခါမှာအသုံးပြုပါသည်။

တစ်ခုထူးခြားတာက output ထုတ်ကြည့်တဲ့အခါမှာ 4 မှာ single quote နဲ့ရှိနေတာကို တွေ့ရပါလိမ့်မယ်။ ဒါက integer နဲ့ အနေနဲ့ သိုလှောင်လိုက်တာမဟုတ်ပဲ string အဖြစ် သိုလှောင်လိုက်တာဖြစ်ပါတယ်။ Append() နဲ့ သိုလှောင်တည်းက single quote ထည့်ထားသောကြောင့်ဖြစ်ပါသည်။ သတိထားနိုင်ပါစေ။

6.8

List နောက်တစ်ခုကို ထည့်ဖြည့်ခြင်း။

```
>>> list.extend([5,6,7,8])
>>> print(list)
['a', 'b', 'c', 'd', 'e', 1, 2, 3, '4', 5, 6, 7, 8]
```

Description

Extend([]) ဆိုတာက list ရဲ့ နောက်ဆုံးနေရာမှာ list တစ်ခုကိုပေါင်းထည့်ချင်းဖြစ်ပါသည်။

6.9

list ထဲက item ကို remove လုပ်ပြမှာဖြစ်ပါတယ်။

```
>>> list.remove(8)
>>> print(list)
['a', 'b', 'c', 'd', 'e', 1, 2, 3, '4', 5, 6, 7]
```

Description

Remove() ဆိုတာက list ထဲက item ကို ဖယ်ထုတ်လိုက်တာဖြစ်ပါတယ်။

6.9.1

```
1.4  
>>> list=["a","b","c"]  
>>> del list  
>>> print(list)  
<class 'list'>  
>>>
```

Description

del ဆိုတာက list တစ်ခုလုံးကို Delete လုပ်လိုက်တာဖြစ်ပါတယ်။ ပြီးနောက် Output ထုတ်ပြပါတယ်။

Challenge

C-19

new_family ဆိုတဲ့ variable နဲ့အတူ အသစ်ဖြစ်လာနိုင်တဲ့ လူတစ်ယောက်အမည်ကို input ထည့်ခိုင်းပါ။ထို့နောက် သင့်မိသားစုအမည်တွေနဲ့အတူ list လုပ်ပြီး family ဆိုတဲ့ variable တစ်ခုကိုဖန်တီးပါ။ output ပြန်ထုတ်ပြပါ။

C-20

သင့်သူငယ်ချင်းတွေရဲ့ အမည်တွေနဲ့အတူ friend ဆိုတဲ့ List တစ်ခုကို ဖန်တီးပါ။ ထို့နောက် Append function နဲ့အတူ သင့်အသစ်ရလာတဲ့ သူငယ်ချင်းတစ်ယောက်ရဲ့အမည် ကို နောက်ဆုံးမှာ ထည့်ပါ။output ပြန်ထုတ်ပြပါ။

C-21

New_friends ဆိုပြီး သင့်အသစ်ရလာတဲ့ သူငယ်ချင်းအမည် ကို list လုပ်ပါ။ထို့နောက် အရင် friend ဆိုတဲ့ list နဲ့ ပြန်ပေါင်းပါ။ output ပြန်ထုတ်ပြပါ။

C-22

Friend list ထဲမှာ သင့်နဲ့မတွေ့ဖြစ်တော့တဲ့ သူငယ်ချင်းတစ်ယောက်ကို remove လုပ်ပါ။ Output ပြန်ထုတ်ပြပါ။

C-23

Friend list တစ်ခုလုံးကို del လုပ်ပြီး output ထုတ်ပြပါ။၏

Solution==>

C-19

```
new_family=str(input("Enter New Family Name: "))
family=["Father","Mother","Brother","Me",new_family]
print(family)
```

C-20

```
friend=["Yai Naing","Kaung Kaung","Min Khant"]
friend.append("Kyaw Zin Aung")
print(friend)
```

C-21

```
new_friend=["Kyaw Kyi","Ya Wai","Ye Bhone Naing","Aung Kyaw Zaw"]
friend=["Ye Naing","Kaung Kaung","Min Khant"]
friend.extend(new_friend)
print(friend)
```

C-22

```
friend=["Ye Naing","Kaung Kaung","Min Khant","Kyaw Kyi","Ya Wai","Ye Bhone Naing","Aung Kyaw Zaw"]
friend.remove("Ye Naing")
print(friend)
```

C -23

```
friend=["Ye Naing","Kaung Kaung","Min Khant","Kyaw Kyi","Ya Wai","Ye Bhone Naing","Aung Kyaw Zaw"]
del friend
print(friend)
```

6.10.0

Dictionary

Dictionary ဆိုတာက key နဲ့ Value နှစ်ခုတွဲပြီးသိုလှောင်ခြင်းဖြစ်ပါသည်။ key နဲ့ value ကို : ခွဲခြားထားပါတယ်။ ဥပမာ ပြောရရင် name က key ဆိုရင် Hter Arkar(ကျနော်အမည်) Value ပေါ့။ နားလည်မယ်လို့မျှော်လင့်ပါတယ်။ Dictionary data type ကလဲ ပြုပြင်ပြောင်းလဲလို့ရပါတယ်။

Example 6.10.0

```
>>> info={"name":"Htet Arkar Kyaw","age":18}
>>> print(info)
{'name': 'Htet Arkar Kyaw', 'age': 18}
```

Description

info ဆိုတဲ့ dictionary ကို ဖန်တီးလိုက်တာပဲ ဖြစ်ပါတယ်။ Dictionary ထဲက data တွေကို curly brackets ကြားမှာရေးပြီး တစ်ခုနဲ့ တစ်ခုကို full column နဲ့ခြားထားပါတယ်။ Key နဲ့ Value နှစ်ခုတွဲပြီးဖော်ပြထားကိုမြင်ရပါလိမ့်ပါမယ်။ Value သည် String or integer စတဲ့ Data Type ဖြစ်ပါတယ်။ Dictionary သည်လည်းပြုပြင်ပြောင်းလဲလို့ရပါတယ်။

Example 6.10.1

```
>>> student={}
>>> student["name"]="Kaung Kaung"
>>> student["age"]=19
>>> print(student)
{'name': 'Kaung Kaung', 'age': 19}
>>>
```

Description

ဒါကတော့ Dictionary ကိုနောက်ထပ်ဖန်တီးနည်းတစ်ခုပဲဖြစ်သည်။ student variable ထဲကို ဘာ data(key နဲ့ value) မထည့်ဘဲ ဖန်တီးထားတာကို တွေ့မှာပါ။ ထို့နောက် student["name"] ဆိုတာ name ဆိုတဲ့ key ကို ဖန်တီးလိုက်တာဖြစ်ပြီး၊ kaung kaung ဆိုတဲ့

string data type ကတော့ value ကို ဖန်တီးလိုက်ပြီး student ဆိုတဲ့ variable ထဲမှာသိုလှောင် လိုက်တာဖြစ်ပါသည်။

Example 6.10.2

Dictionary ထဲ က key မှာ list or tuple ဆိုတဲ့ data type တွေထည့်သွင်းလို့ရပါတယ်။

```
>>> family={"family_name":("U Tun","Daw Mya","Arkar")}
>>> print(family)
{'family_name': ('U Tun', 'Daw Mya', 'Arkar')}
>>> family={"member":["Arkar","Wai Yan","Ingyin","Mya Mya"]}
>>> print(family)
{'member': ['Arkar', 'Wai Yan', 'Ingyin', 'Mya Mya']}
>>>
```

Description

- Line 1: family ဆိုတဲ့ Dictionary ကိုဖန်တီးလိုက်ပါတယ်။ထို့နောက် family_name ဆိုတဲ့ Key နဲ့ အတူ Value နေရာမှာ လူအမည်တွေရေးထားတဲ့ tuple ကိုထည့်ထားပါတယ်။
- Line 2: ထို့နောက် print keyword နဲ့အတူ Dictionary ပြန်လည်ကြည့်ရှုခြင်းဖြစ်သည်။
- Line 3: family ဆိုတဲ့ Dictionary ကိုဖန်တီးလိုက်တယ်။ပြီးနောက် member ဆိုတဲ့ key နဲ့ အတူ member name တွေရေးထားတဲ့ list data type ကို Value နေရာမှာထည့်ထားပါတယ်။

6.11.0

Dictionary ထဲ က Data တွေကိုလိုသလို ပြုပြင်ပြောင်းလဲကြည့်ကြမယ်။

```

>>> member={"name":"Arkar","age":20}
>>> print(member["name"])
Arkar
>>> print(member.get("age"))
20
>>> print(member.keys())
dict_keys(['name', 'age'])
>>> print(member.values())
dict_values(['Arkar', 20])
>>> member["weight"]=110
>>> print(member)
{'name': 'Arkar', 'age': 20, 'weight': 110}
>>> member["age"]=21
>>> print(member)
{'name': 'Arkar', 'age': 21, 'weight': 110}
>>> member.pop("weight")
110
>>> print(member)
{'name': 'Arkar', 'age': 21}

```

Line 1: member ဆိုတဲ့ dictionary တစ်ခုကိုဖန်တီးလိုက်ပါတယ်။

Line 2: member["name"] နဲ့ name key ရဲ့ value ကို output ထုတ်ကြည့်လိုက်ပါတယ်။

Line 3: get method ကလဲ age key ရဲ့ value ကို output ထုတ်ပြခြင်းဖြစ်ပါတယ်။

Line 4: member.keys() ဆိုတာ dictionary ထဲမှာရှိတဲ့ key only ကိုပဲ output ထုတ်ကြည့်တာဖြစ်ပါတယ်။

Line 5: member.values() ကတော့ value ပဲထုတ်ကြည့်တာဖြစ်ပါတယ်။

Line 6: member["weight"] =110 ဆိုတာ weight ဆိုတဲ့ key နဲ့ 110 ဆိုတဲ့ integer data type ကိုထည့်သွင်းလိုက်တာဖြစ်ပါတယ်။

Line 7: သူကတော့ print ထုတ်ပြီးသက်သေပြလိုက်တာပေါ့။

Line 8: Line 6 မှာကျတော့ အသစ်ထည့်တာနော် line 8 ကတော့ပြုပြင်ပြောင်းလဲတာ။ age 20 ကနေ နဲ့ 21 ကိုပြောင်းလဲသွားတာကိုတွေ့ပါလိမ့်မယ်။

Line 9: print ပြန်ထုတ်ပြထားပါတယ်။

Line 10: pop() ဆိုတာ remove လိုက်တာပါ။ weight ကို remove လိုက်တာကိုတွေ့ပါလိမ့်မယ်။

Line 11: မှာ print ပြန်ထုတ်ပြထားပါတယ်။

Challenge ==>

C-24

member ဆိုတဲ့ dictionary data type ကိုတည်ဆောက်ပါ။ name age weight phone ဆိုတဲ့ key လေးခုနဲ့အတူ value ထည့်ပါ။ ထို့နောက် output ထုတ်ပြပါ။

C-25

name ဆိုတဲ့ user input တစ်ခုကိုတည်ဆောက်ပါ။ age ဆိုတဲ့ user input ကို တည်ဆောက်ပါ။ phone ဆိုတဲ့ integer data type တစ်ခုကိုတည်ဆောက်ပါ။ ထို့နောက် info ဆိုတဲ့ variable name နဲ့အတူ dictionary data type တစ်ခုတည်ဆောက်ပါ။ အထဲတွင် name age phone ဆိုတဲ့ key နဲ့အတူ ကိုက်ညီတဲ့ variable name(value) စသည့်တို့ကို ထည့်ပါ။ ထို့နောက် output ပြန်ထုတ်ပြပါ။

C-26

my_data ဆိုတဲ့ variable name နဲ့အတူ dictionary data type တစ်ခုတည်ဆောက်ပါ။ အထဲတွင်ရှိသော သင့်ရဲ့ weight နဲ့ age ကိုပြောင်းလဲပါ။

Challenge

family ဆိုတဲ့ dictionary တစ်ခုတည်ဆောက်ပါ။ Dictionary ထဲတွင် key ကို name လို့အမည်ပေးပါ။ key ကို list တစ်ခုအနေဖြင့်ထားပါ။ list ထဲတွင် ကိုယ်မိသားစုနာမည်တွေထည့်ပါ။ထို့နောက် ouptut ပြန်ထုတ်ပြပါ။

Solution

C-24

```
member={"name":"Naing Kyaw","age":19,"weight":100.4,"phone":"+959884"}  
print(member)
```

C-25

```
name=str(input("Enter Your Name: "))  
age=int(input("Enter Your age: "))  
phone=int(input("Enter Your Phone Number: "))  
info={"name":name,"age":age,"phone":phone}  
print(info)
```

C-26

```
my_data={"name":"Htet Arkar Kyaw","age":20,"weight":120}  
my_data["age"]=21  
my_data["weight"]=135
```



```
print(my_data)
```

C

```
family={"name":["Htet Wai Yan","Kyaw Kyi","Htet Kyi"]}
```

```
print(family)
```

CHAPTER-7

Comparison Operators

Operator	Description	Example
==	Equal To	a==b
!=	Not Equal To	a!=b
>	greater than	a>b
<	less than	a=	Greater than Equal to	a>=b
<=	less than Equal To	a<=b

Logical Operators

Operator	Description	Example
and	Both statements are true	a==1 and b<=6
or	one of the statements is true	a<3 or a>5
not	Reverse the result	not(b>a)

Boolean

Boolean Data Type သည် python တွင် built-in ရှိပြီးသားဖြစ်သည်။ သူက အခြေအနေ တစ်ခုမှန်လား ၊ မှားလား စသည့်တို့ကို ရည်ညွှန်းတာဖြစ်ပါတယ်။

```
>>> a=5
>>> b=6
>>> a==b
False
>>> a<b
True
>>> █
```

Description

ပုံကိုကြည့်မယ်ဆိုရင် a တန်ဖိုးကို 5 လို့ပေးထားသည်။ b တန်ဖိုးကို 6 လို့ပေးထားတယ်။ Comparison operator သုံးပြီး သူတို့နှစ်ယောက်ကို ညှိကြည့်တယ်။ 5 နဲ့ 6 ဖြစ်တဲ့ အတွက် သူတို့အခြေအနေက မှားနေတယ် ဒါကြောင့် False လို့ output ထုတ်ပြီး a က b ထက်ငယ်တော့ကြောင့် True လို့ထုတ်ပေးတယ်။

7.0

IF Else statement

if statement သည် condition မှန် မမှန် စစ်ဆေးသည်။မှန်တဲ့ condition ကို အလုပ်ဆက်လုပ်ပါသည်။ (နားလည်အောင် ဒီလိုပဲမှန်ထားပါ။)

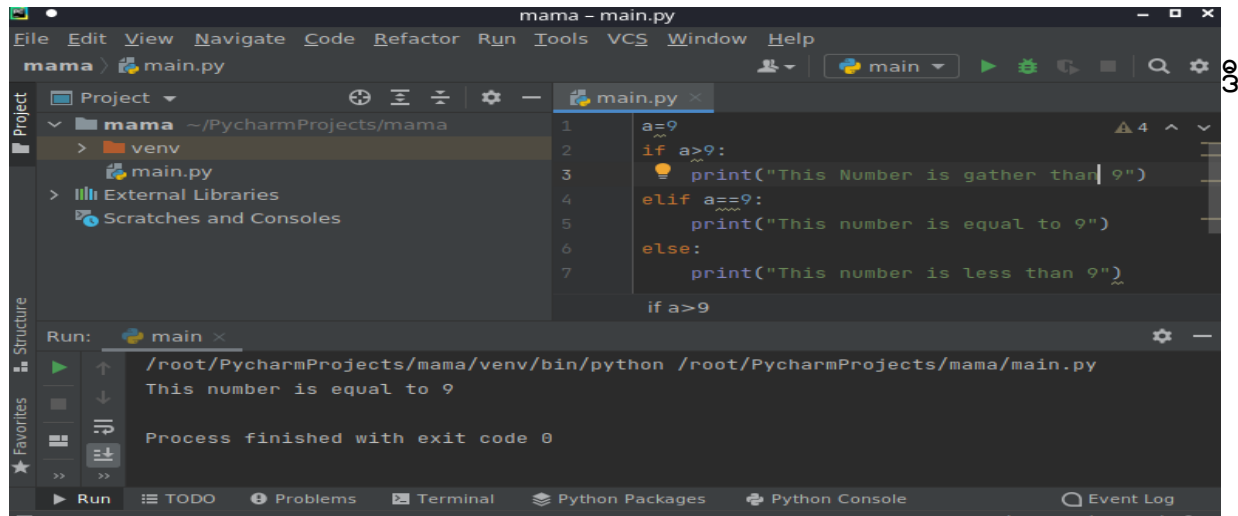
```
>>> a=10
>>> if a>11:
...     print("False")
... else:
...     print("True")
...
True
```

Line1 - a ရဲ့ တန်ဖိုး ကို 10 လို့သတ်မှတ်ထားသည်။
 Line2 - if a>11: ဆိုတာ a က 11 ထက်ကြီးခဲ့မယ်ဆိုရင်
 Line3 - False ဆိုတဲ့ စာသား ကို ထုတ်ပြပါ။
 Line4 - else ဆိုတာ a က 11 ထက်ငယ်ထက်မယ်ဆိုရင်(11 ထက်မကြီးတော့ဖူးဆိုရင်)
 Line5 - True ဆိုတဲ့ စာသားကိုထုတ်ပြပါ။ထို့ကြောင့် a က 11 ထက်ငယ်သောကြောင့် True ဆိုတဲ့ စာသားကို ထုတ်ပြခြင်းဖြစ်သည်။(မှန်တဲ့ condition ကို ဆက်လုပ်ပြီး မှားတဲ့ condition ကိုမလုပ်ပဲ ကျော်ခဲ့သည်။)

7.0.1

If elif else statement

Condition များတွဲအခါမှာ elif ကိုထည့်သုံးရပါတယ်။ နားလည်အောင်အောင်ပါ code ကိုကြည့်ပါ။



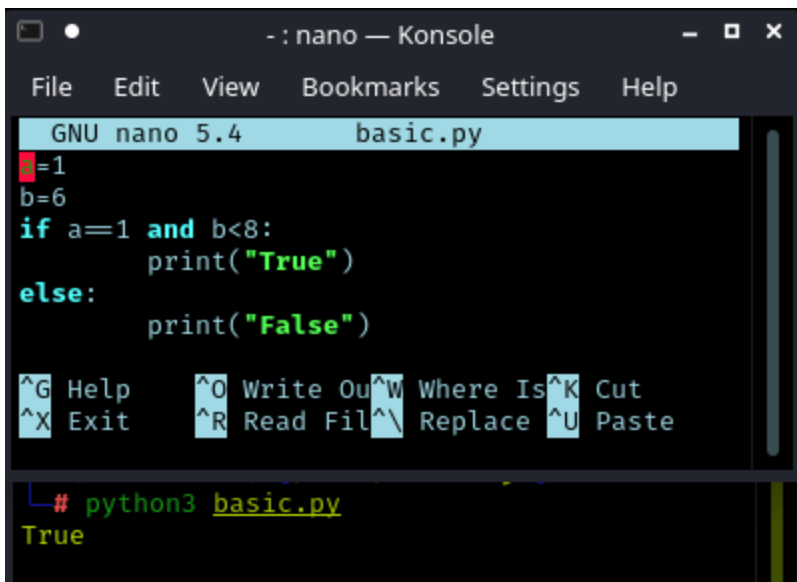
လောက်ဆိုရင် Comparison operators အကြောင်းကို နားလည်သင့်သလောက်နားလည်ပြီလို့ ယူဆရပါတယ်။

- Line 1 - a ကို 9 လို့သတ်မှတ်ထားလိုက်တယ်။
- Line 2 - if a> 9 (a က 9 ထက်ကြီးခဲ့မယ်ဆိုရင်)
- Line 3- “This Number is gather than 9” လို့ ရလဒ်ထုတ်ပြပါ။
- Line 4 - elif a==9: (အဲ့ဒီလိုမဟုတ်ဘဲ a က 9 နဲ့ ညီခဲ့မယ်ဆိုရင်)
- Line 5 - “This Number is equal to 9” လို့ အဖြေထုတ်ပြပါ။
- Line 6 - else: (အပေါ်က code တွေမှားနေခဲ့ရင်)
- Line 7 - This number is less than 9 လို့အဖြေထုတ်ပါ။

7.0.2

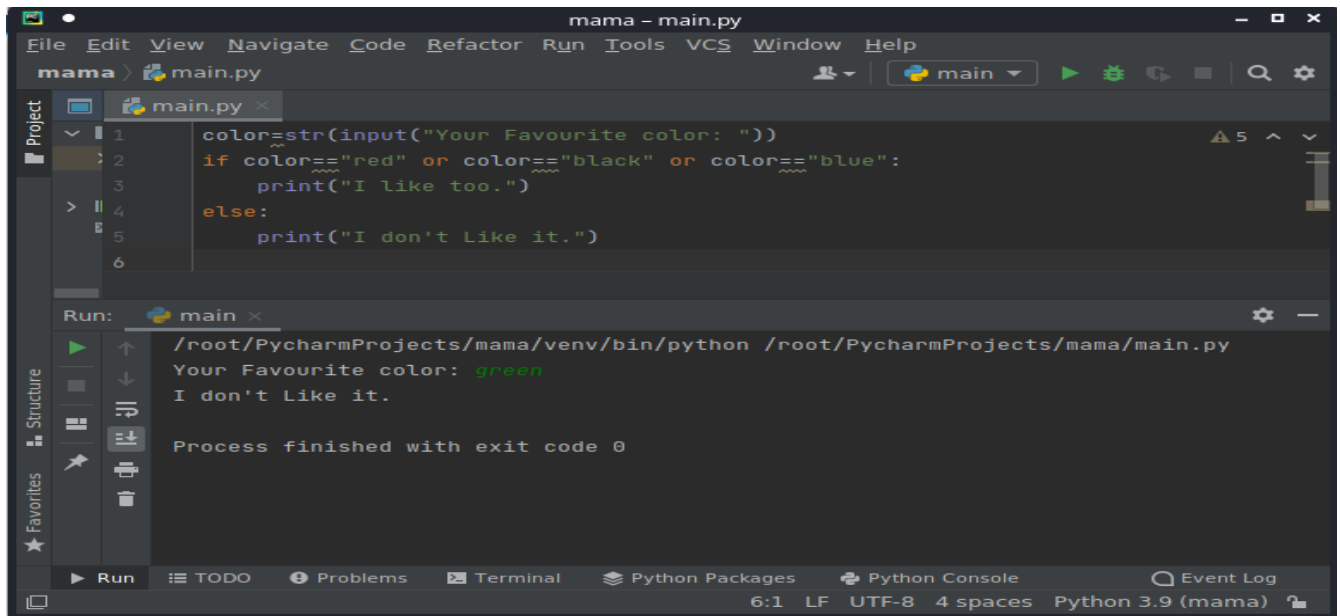
Logical Operators တွေနဲ့ တွဲသုံးပြသွားမှာပါ။

Example 4.2.1



Line 1: a=1 လို့သတ်မှတ်ထားပါ။
Line 2: b=6 လို့သတ်မှတ်ထားပါ။
Line 3: if a==1 and b<8: (a က 1 နဲ့လည်း ညီမျှပြီး 8 ထက်လည်းငယ်မယ်ဆိုရင်)
Line 4: True လို့ output ထုတ်ပြပါ။
Line 5: else:(အဲ့လိုမဟုတ်ခဲ့ရင်)
Line 6: False လို့ output ထုတ်ပြပါ။

Example 7.0.3



The screenshot shows the PyCharm IDE interface. The main editor window displays a Python script named `main.py` with the following code:

```
1 color=str(input("Your Favourite color: "))
2 if color=="red" or color=="black" or color=="blue":
3     print("I like too.")
4 else:
5     print("I don't Like it.")
6
```

The Run window at the bottom shows the execution of the script. The command used is `/root/PycharmProjects/mama/venv/bin/python /root/PycharmProjects/mama/main.py`. The output is:

```
Your Favourite color: green
I don't Like it.
```

The status bar at the bottom indicates the process finished with exit code 0. The bottom status bar also shows the file encoding as UTF-8, 4 spaces, and the Python version as 3.9 (mama).

Line 1: color ကို input ထည့်ခိုင်းတယ်
Line 2: ထည့်လိုက်တဲ့ color တွေထဲက (red,black,blue)တစ်ခုခုနဲ့ညီခဲ့ရင်
Line 3: “I like too” လို့ ရလဒ်ထုတ်ပြပါ။
Line 4: color မတူခဲ့ရင်
Line 5: “I don't like it” လို့ output ထုတ်ပြပါ။

Challenge

C-27

user ကို name နဲ့ password ထည့်ခိုင်းပါ။ တူညီခဲ့ရင် success ပြပြီး၊ မတူညီရင် incorrect လို့ပြပါ။

C-28

User (ဝယ်သူရဲ့) အသက် ကို input ထည့်ခိုင်းပါ။ 18 နှစ်နဲ့ ညီခဲ့ရင် "you can drink bear" ဟု ရလဒ် ထုတ်ပြပါ။ 20 နဲ့ ညီကြီးခဲ့ရင် "you can drink alcohol" လို့ output ထုတ်ပြပြီး၊ တစ်ခုမှမတူညီခဲ့ရင် "You are baby လို့ output ထုတ်ပြပါ။

C-29

User ကို input function နဲ့ သူကြိုက်တဲ့ ဖုန်းအမျိုးအစားထည့်ခိုင်းပါ။ သူကြိုက်တဲ့ phone ရှိခဲ့ရင် phone name နဲ့အတူ "you can buy" လို့ output ထုတ်ပြပြီး၊ မရှိခဲ့ရင် phone name နဲ့အတူ "Sorry, This model does not exit." လို့ output ထုတ်ပြပါ။

Solution

C-27

```
uname=str(input("Enter Your Name: "))
upasswd=str(input("Enter Your Password: "))
if uname=="Arkar" and upasswd=="123@##":
    print("success")
else:
    print("incorrect")
```

C-28

```
age=int(input("Enter Your age: "))
if age==18:
    print("You can drink bear")
elif age>=20:
    print("You can drink alcohol")
else:
    print("You are baby")
```

C-29

```

phone=str(input("Enter Your Favourite Phone Name: "))
if phone=="Vivo" or phone=="oppo" or phone=="sony" or phone=="saungsum":
    print("you can buy",phone)
else:
    print("Sorry,This modle does not exit.",phone)

```

Chapter – 8

Membership operators

Symbol	Description
in	to check if the value exists in a variable or not
not in	exact opposite in operator

(နားလည်အောင် အောင်က example တွေကိုကြည့်ပါ။)

```

>>> fruit=["apple","orange","mango"]
>>> 'mango' in fruit
True
>>>

```

Description

fruit ဆိုတဲ့ variable နဲ့အတူ အသီးသုံးမျိုးပါတဲ့ list တစ်ခုတည်ဆောက်လိုက်တယ်။ပြီးနောက် in operator နဲ့ mango ကို fruit ထဲရှိမရှိ စစ်ပါတယ်။ရှိတဲ့အတွက် true လို့ပြပါတယ်။

```

>>> 'banana' not in fruit
True
>>>

```


banana ဆိုတဲ့ အသီးကို not in operator နဲ့ စစ်တယ်။ မရှိတဲ့အတွက် true လို့ပြတယ်။ in operator က ရှိတဲ့အတွက် true လို့ပြတာ။ ပြောင်းပြန် သဘောကို သိစေချင်တာပါ။

For loop

for loop သည် အကြိမ်အရေအတွက်နဲ့အတူတူ အလုပ်လုပ်ပါသည်။

(statement တစ်ကြောင်းကို အကြိမ်တစ်ရာနှစ်ရာ စသဖြင့် ပေးထားတဲ့ အကြိမ်အရေအတွက် အတိုင်း အလုပ်လုပ် စေချင်တဲ့အခါမှာအသုံးပြုပါသည်။)

for i(variable) in sequence:

statement

i(ကို variable တစ်ခုအဖြစ်သတ်မှတ်ပြီး sequence ထဲက value တွေကို in keyword နဲ့ စစ်တယ် ရှိရင် i(ထဲမှာ သိုလှောင်တယ်။ sequence ဆိုတာ အကြိမ်အရေအတွက်၊စကားအစုတစ်ခု(string) စသည့်တို့ဖြစ်သည်။ ထို့နောက် အလုပ်လုပ်စေချင်တဲ့ statement ကို အောက်မှာထည့်ပေးရသည်။ ပေးထားတဲ့ အကြိမ်အရေအတွက် အတိုင်း ထိုstatement ကို run နေအုံးမည့်ဖြစ်သည်။ထို့ကြောင့် ၎င်းကို counting loop လို့လဲခေါ်ဆိုကြသည်။

Example 8.0.1

```
>>> for i in "I Love You":  
...     print(i)  
...  
I  
  
L  
o  
v  
e  
  
Y  
o  
u  
>>>
```

Description

For Keyword နဲ့ for loop ကိုစတင်အသုံးပြုမည်ဖြစ်သည်။ i(ကို variable တစ်ခုအဖြစ်သတ်မှတ်တယ် in operator နဲ့ sequence ရှိမရှိစစ်တယ်။ရှိတဲ့အတွက် sequence ထဲ က စကားလုံးတွေကို i(variable ထဲ သိုလှောင်တယ်။ထို့နောက် ပထမ တစ်လုံး ကို output ထုတ်တယ်။ ထို့နောက် ဒုတိယတစ်လုံးကို i(ထဲ သိုလှောင်တယ်။ပြန်ပီးတော့ output ထုတ်တယ်။ နောက်ဆုံးစကားလုံး u ကို i(ထဲ သိုလှောင်တယ်။ထို့နောက် output ပြန်ထုတ်တယ်။

(Enter ခေါက် အစက်သုံးစက် ရောက်ရင် tab key ခုန်ပါ။ ထို့မှသာ for loop မှာအလုပ်လုပ်မည် ဖြစ်သည်)

Range Function

Range Function ကို for loop နှင့်အတူ အလုပ်လုပ်ပါသည်။ range function တွေ စတင် လုပ်ဆောင် မည့် အရေအတွက်(starting number)၊ နောက်ဆုံးလုပ်မည့်အရေအတွက်(ending number) နှင့် အရေအတွက်ကွာခြားထားခြင်း(step) တို့ပါရှိသည်။

Example 8.0.2


```
>>> for i in range(0,10):
...     print("I love you")
...
I love you
I love you
I love you
I love you
I love you
I love you
I love you
I love you
I love you
I love you
I love you
```

Description

for နဲ့စတင်ရေတွက်မည့်ဖြစ်သည်။ ထိုနောက် 0 (အကြိမ်အရေအတွက်နော်) နဲ့ စတင်ရေတွက်ပြီး 10(ရောက်ရင် ရပ်တန့်သွားမည့်ဖြစ်သည်။ အကြိမ်အရေအတွက် တွေကို in keyword နဲ့စစ်တယ်။ ထို့နောက် i(ထဲကိုသိုလှောင်လိုက်သည်။ print နဲ့ " I love you ဆိုတဲ့ စာကြောင်းတစ်ကြောင်းချင်းကို ၁၀ ကြောင်းမြှောက်တဲ့အထိ loop ပတ်၍ output ထုတ်ပြမည့်ဖြစ်သည်။ 10 နဲ့ ညီသွားတဲ့အခါ loop ပတ်တာရပ်တန့်သွားမည့်ဖြစ်သည်။ ဒီနေရာမှာ 0 နဲ့ စတင်ရေတွက်ပြီး 10 မှာဆုံးတာဖြစ်သည်။ ထို့ကြောင့် 10 နဲ့ညီသွားရင်ရပ်သွားမည်ဖြစ်သောကြောင့် ၁၁ ကြောင်းမြှောက်ကို (0-10ကို outputထုတ်ပြမည်မဟုတ်ပါ။ နားလည်အောင် print("I love you",i)နဲ့ လုပ်ကြည့်ပါ။ တခြား ပုံသေနည်းအရပြောရမယ်ဆိုရင်range(n,n-1)ဖြစ်သည်။

Example 8.0.3

```
>>> for i in range(0,20,3):
...     print(i)
...
0
3
6
9
12
15
18
>>>
```

Description

အပေါ်မှာရှင်းပြထားလို့အရှည်ကြီးတော့ရှင်းမပြတော့ပါဖူး။ range(0,20,3) မှာ ဆို နောက်ဆုံး number ဖြစ်တဲ့ 3 သည့် အရေအတွက်တစ်ခုနဲ့တစ်ခုကြား ၃လုံးအကွာတွင်ရှိသော variable ကို သာ ထုတ်ပြ ခိုင်းခြင်း အလုပ်လုပ်စေခြင်းဖြစ်သည်။

For loop with ass.

Challenge==>

C-30

Input function နဲ့ user ရဲ့အမည်ကို ထည့်သွင်းခိုင်းပါ။ထို့အမည်သုံးကြိမ် output ထုတ်ပြပါ။

C-31

Hello Python ဆိုတဲ့ စာကြောင်းမှ for loop ကို အသုံးပြု၍ စကားလုံးတစ်လုံးစီ output ထုတ်ပြပါ။

C-32

Variable name ကို x လို့သတ်မှတ်ပါ။ range function မှာ starting number မှာ 30 ၊ ending number မှာ 2 နှင့် step number ကို -4 အဖြစ်ထားပါ။ ထို့နောက် x ကို output ထုတ်ပြပါ

Solution ==>

C-30

```
name=str(input("Enter Your Name: "))  
for i in range(0,3):  
    print(name)
```

C-31

```
for i in "Hello Python":  
    print(i)
```

C-32

```
for x in range(30,2,-4):  
    print(x)
```

CHAPTER-9

Assignment Operators

Symbol	Example	Description
x=	x=5	x=5
+=	x+=5	x=x+5
-=	x-=5	x=x-5
=	x=5	x=x*5
/=	x/=5	x=x/5
%=	x%=5	x=x%5
//=	x//=5	x=x//5
=	x=5	x=x**5

```
>>> x=3
>>> x+=5
>>> print(x)
8
>>> x=x+5
>>> print(x)
13
>>>
```

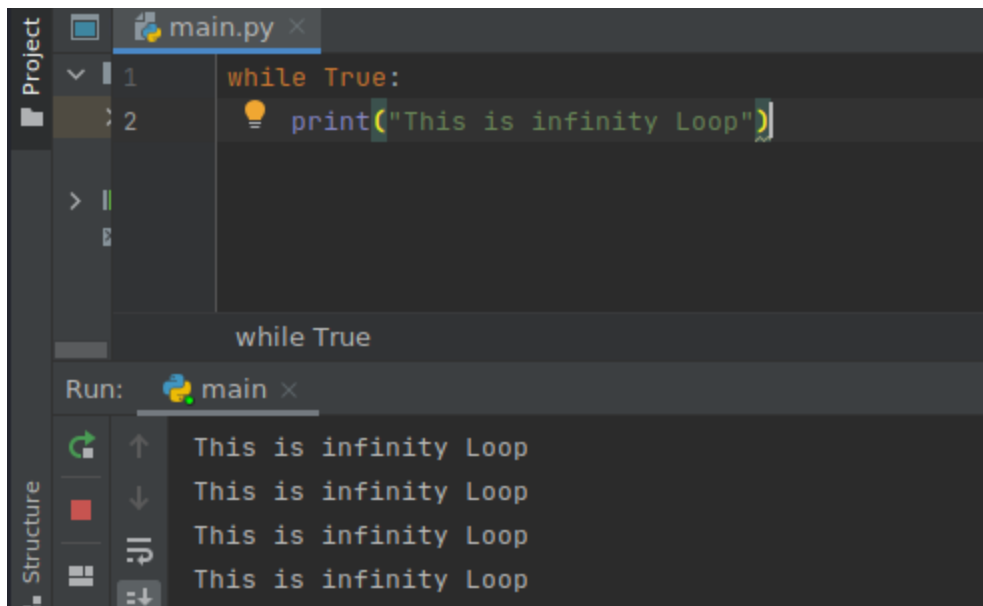
ဒီတစ်ခုနဲ့တင် နားလည်မယ်လို့မျှော်လင့်ပါတယ်

x တန်ဖိုးကို ၃ လို့သတ်မှတ်တယ်။ ထို့နောက် x+=5 ဆိုတာ x တန်ဖိုး 3 ကို 5 နဲ့ပေါင်းတယ်။ ပေါင်းထားတဲ့ ရလဒ်ကို x ထဲမှာသိုလှောင်လိုက်တယ်။ ထို့နောက် x တန်ဖိုးကို print keyword နဲ့ output ပြန်ထုတ်တဲ့အခါ နှစ်ခုပေါင်းရလဒ် 8 ကိုထုတ်ပေးတယ်။ x=x+5 ဆိုတာ ရလဒ်တဲ့ x ရဲ့ တန်ဖိုး 8 နဲ့ 5 ကိုပေါင်းပြီး x ထဲမှာသိုလှောင်လိုက်တယ်။ပြီးနောက် x ရဲ့ တန်ဖိုးကို output ပြန်ထုတ်တယ်။

9.0 WHILE LOOP

While loop သည် condition မှန်နေသမျှ အလုပ်လုပ် နေမည်ဖြစ်သည်။

example 9.0.0 Infinity Loop



```
main.py x
1 while True:
2     print("This is infinity Loop")

while True
Run: main x
This is infinity Loop
This is infinity Loop
This is infinity Loop
This is infinity Loop
```

Description

Line 1: while True: ဆိုတာ condition true ဖြစ်နေသမျှ

Line 2: "This is infinity Loop" ဆိုတဲ့ စာသားကို output ထုတ်နေပါ။

Example 9.0.1

```
>>> x=0
>>> while x<5:
...     print("I'm robot.")
...     x+=1
...
I'm robot.
I'm robot.
I'm robot.
I'm robot.
I'm robot.
```

Description

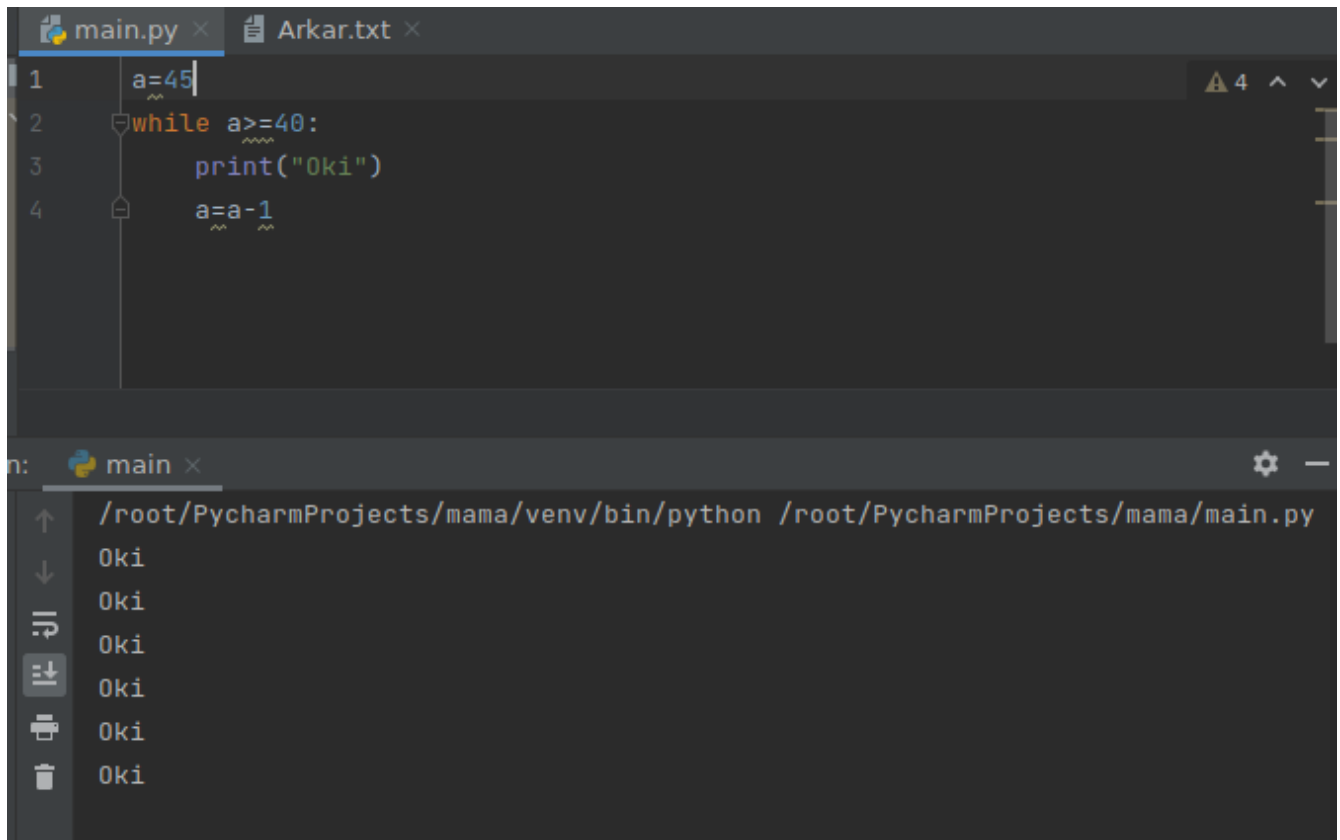
Line 1: x ရဲ့တန်ဖိုးကို 0 လို့သတ်မှတ်လိုက်တယ်။

Line 2: ထို့နောက် while loop စ၍လုပ်ဆောင်ပါတယ်။ condition အရ x သည် 5 ထက်ငယ်နေသမျှ အလုပ်လုပ်ရမည်ဖြစ်သည်။

Line 3: condition မှန်နေသမျှ output ထုတ်ရမည်စာကြောင်းပင်ဖြစ်သည်။

Line 4: x ထဲကို 1 ပေါင်းထည့်၍ loop ထပ်ပတ်သည်။ x တန်ဖိုး 5 ဖြစ်သွားတဲ့အခါ loop သည် ရပ်သွားမည်ဖြစ်သည်။

Example 9.0.2



The screenshot shows a Python IDE with two tabs: 'main.py' and 'Arkar.txt'. The 'main.py' tab is active and contains the following code:

```
1 a=45
2 while a>=40:
3     print("0ki")
4     a=a-1
```

Below the code editor, the output console shows the result of running the script. The command prompt is '/root/PycharmProjects/mama/venv/bin/python /root/PycharmProjects/mama/main.py'. The output consists of six lines, each displaying '0ki'.

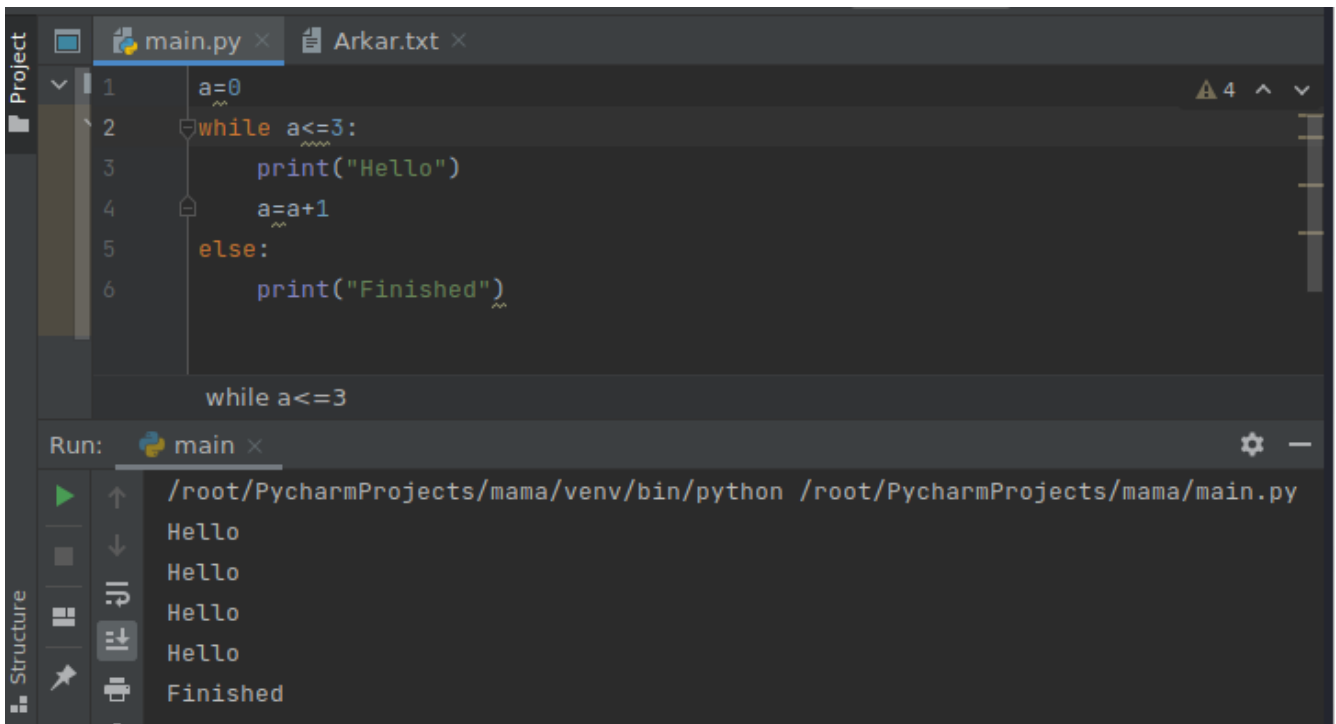
Description

a ရဲ့တန်ဖိုးကို 45 လို့သတ်မှတ်လိုက်တယ်။ while loop စတင်လုပ်ဆောင်တယ်။ condition အရဆိုရင် a က 40 နဲ့ညီ သို့မဟုတ် ကြီးနေသမျှ အလုပ်လုပ်မည်ဖြစ်သည်။ ထို့နောက် condition မှန်နေသမျှ oki လို့ output ထုတ်နေမည်ဖြစ်သည်။ a ထဲက တန်ဖိုးကို -1 နုတ်သည်။ condition မှန်ရင်အလုပ်ဆက်လုပ်သည်။ condition မှားသွားတဲ့အခါ(false) loop ရပ်သွားမည်ဖြစ်သည်။

While else loop

While else loop မှာ loop ကို condition false ဖြစ်သွားတဲ့အခါမှာ အလုပ်လုပ်တဲ့ loop ဖြစ်ပါတယ်။

Example.



The screenshot shows the PyCharm IDE interface. At the top, there are two tabs: 'main.py' and 'Arkar.txt'. The 'main.py' tab is active, displaying a Python script with the following code:

```
1 a=0
2 while a<=3:
3     print("Hello")
4     a=a+1
5 else:
6     print("Finished")
```

Below the code editor, there is a 'Run' section. It shows the command used to run the script: `/root/PycharmProjects/mama/venv/bin/python /root/PycharmProjects/mama/main.py`. The output of the script is displayed below the command, showing four 'Hello' messages followed by 'Finished'.

Description

a ရဲ့တန်ဖိုးကို 0 လို့သတ်မှတ်လိုက်ပါတယ်။ ထို့နောက် while loop ပတ်တယ်။ a က 3 နဲ့ ညီ သို့မဟုတ် သွားတဲ့ loop ရပ်သွားပါတယ်။ hello လို့ output ထပ်မံထွက်တော့ပါ။ condition false ဖြစ်သွားသောကြောင့် Finished ဆိုတဲ့ စာသားကို output ထုတ်လိုက်ပါသည်။

Challenge

C==> 33

a ရဲ့တန်ဖိုးကို 2 လို့သတ်မှတ်ပါ။ while loop မှာ a က နံပါတ် 40 ထက်ငယ်ပါစေ။ ထို့နောက် အောက်တစ်ကြောင်းတွင် This number is ဆိုတဲ့စာသားနဲ့အတူ variable a ကိုပါ print ထုတ်ပြပါ။ ထို့နောက် a ကို 2 နဲ့မြှောက်၍ a ထဲမှာပြန်သိုလှောင်ပါ။

c==>34

a ရဲ့တန်ဖိုးကို 3 လို့ထားလိုက်ပါ။ while loop မှာ a ရဲ့ တန်ဖိုးကို 3 ထက်ငယ်ပါစေ။ ထို့နောက် this number is ဆိုတဲ့စာသားနဲ့အတူ variable a ကိုပါ output ထုတ်ပြပါ။ ထို့နောက် $a // 2$ ၍ တန်ဖိုးတွက်ပါ။ ထို့နောက် else loop ပတ်ပြီး Finished ဆိုတဲ့စာသားကို output ထုတ်ပါ။

Solution

```
c==>33
```

```
a=2
```

```
while a<40:
```

```
    print("This number is",a)
```

```
    a*=2
```

```
c==>34
```

```
a=30
```

```
while a>3:
```

```
    print("This number is",a)
```

```
    a//=2
```

```
else:
```

```
    print("Finished")
```

Nested Loop

Nested Loop ဆိုတာ loop အဆင့်ဆင့်အလုပ်လုပ်တာကို ဆိုလိုခြင်းဖြစ်ပါတယ်။

Example :

```
1 a=str(input("သင့်ဒီပစ္စည်းကိုလိုချင်ပါသလား။Y/N: "))
2 if a=="Y" or a=="y":
3     price=int(input("သင့်တက်နိုင်တဲ့ငွေအမောက်ကိုထည့်ပါ: "))
4     if price>=100000:
5         print("ကျေးဇူးအများကြီးတင်ပါတယ်ခင်ဗျာ")
6     elif price>=50000:
7         print("ကျေးဇူးတင်ပါတယ်ခင်ဗျာ")
8     else:
9         print("ယခုပစ္စည်းမရရှိနိုင်ပါ။")
10 else:
11     print("ယခုပစ္စည်းမရရှိနိုင်ပါ။")
```

Run: main ×

/root/PycharmProjects/mama/venv/bin/python /root/PycharmProjects/mama/main.py

သင့်ဒီပစ္စည်းကိုလိုချင်ပါသလား။Y/N: Y

သင့်တက်နိုင်တဲ့ငွေအမောက်ကိုထည့်ပါ: 1000000000

ကျေးဇူးအများကြီးတင်ပါတယ်ခင်ဗျာ

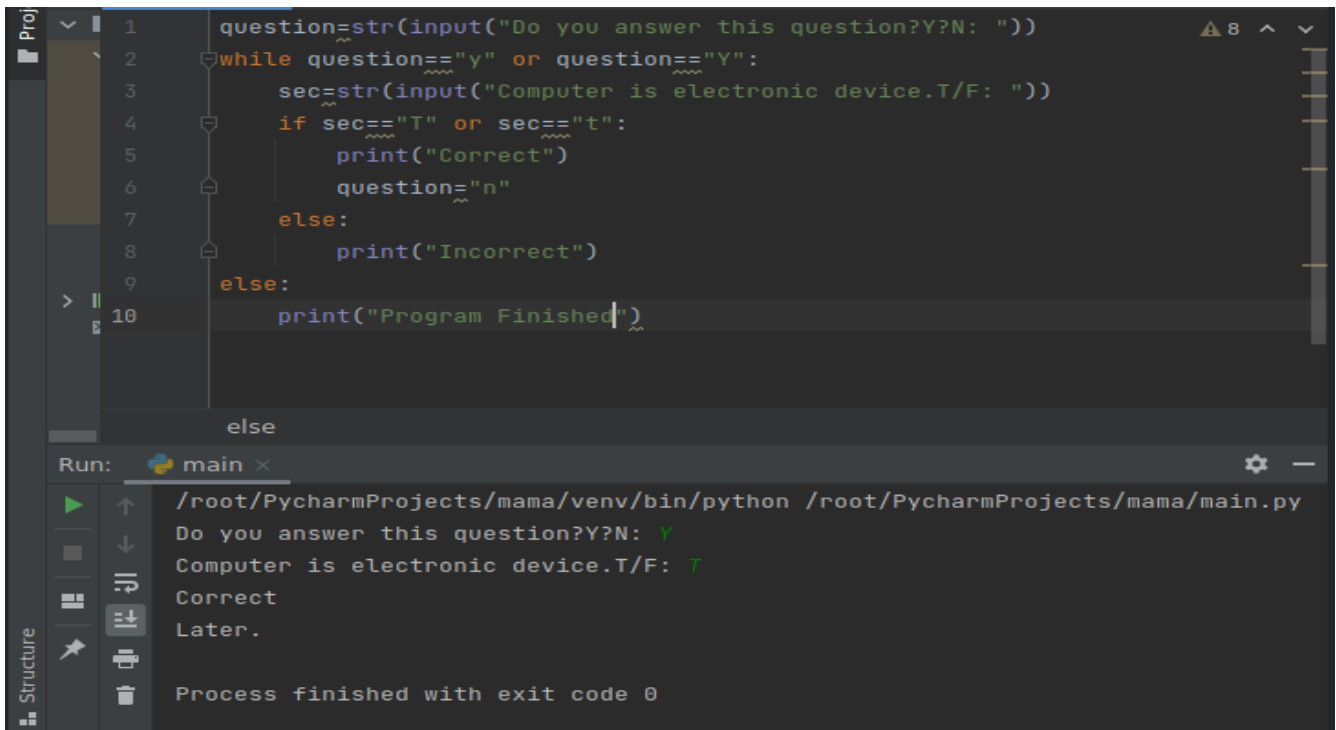
Process finished with exit code 0

English လို့မရေးတက်တော့လို့။(စတာပါ မြန်မာရေးလဲ ရတယ်ဆိုတာ သိချင်စေတာပါ။)

Description

ပထမဦးဆုံး a variable နဲ့ user ကို ပစ္စည်းဝယ်မလား မဝယ်ဖူးလားကို (Y/N) နဲ့ မေးတယ်။ user က y ထည့်လိုက်ရင် price ဆိုတဲ့ variable ထဲကို user တက်နိုင်တဲ့ ငွေအမောက်ကိုထည့်ခိုင်းတယ်။ အကယ်၍ တက်နိုင်ငွေအမောက်က ငွေ 100000 ထက်ကျော် သို့ ညီ ခဲ့ရင် "ကျေးဇူးအများကြီးတင်ပါတယ်လို့ output ထုတ်ပါလိမ့်မယ်။ဒီလိုမဟုတ်ဘဲ 50000 ထဲ ညီ သို့မဟုတ် ကြီးခဲ့မယ်ဆိုရင် "ကျေးဇူးတင်ပါတယ်ခင်ဗျာ" ပေါ့။ အဲ့ထက်ငယ်ခဲ့ရင် "ယခုပစ္စည်းမရရှိနိုင်ပါပေါ့။ ဒီလိုမဟုတ်ဘဲ အစတည်း က (N) ရိုက်ခဲ့ရင် ထိုနည်းတူ "ယခုပစ္စည်းမရရှိနိုင်ပါ။"

The Next Example



```
1 question=str(input("Do you answer this question?Y?N: "))
2 while question=="y" or question=="Y":
3     sec=str(input("Computer is electronic device.T/F: "))
4     if sec=="T" or sec=="t":
5         print("Correct")
6         question="n"
7     else:
8         print("Incorrect")
9 else:
10    print("Program Finished")
```

Run: main x

/root/PycharmProjects/mama/venv/bin/python /root/PycharmProjects/mama/main.py

Do you answer this question?Y?N: Y

Computer is electronic device.T/F: T

Correct

Later.

Process finished with exit code 0

Description

Line 1 User ကို မေးခွန်းဖြစ်မလားမဖြေဘူးလား အရင်မေးတယ်။

Line 2 while နဲ့ ဖြေမဖြေစစ်တယ်။ ဖြေခဲ့တယ်ဆိုရင်

Line 3 ကွန်ပျူတာ က electronic device ဟုတ်မဟုတ်မေးတယ်။

Line 4 အဖြေမှန်ခဲ့တယ်ဆိုရင်

Line 5 Correct လို့ output ထုတ်ပေးလိမ့်မယ်။

Line 6 question="n" ဆိုတာ while loop ပတ်တာကြောင့် နောက်တစ်ကြိမ် question ထပ်မပြ
အောင် တားလိုက်ခြင်းဖြစ်သည်။(တစ်ခြားပုံစံနဲ့ရေးလို့ရပါသေးတယ်။)

Line 7 အဖြေမှန်ခဲ့ရင်

Line 8 incorrect လို့ output ထုတ်မည်ဖြစ်သည်။

Line 9 program finished ဆိုပြီး ပြီးဆုံးကြောင်း output ထုတ်ပြမည်ဖြစ်သည်။

(Note ယခုလာမည့် challenges များသည် အနည်းငယ်ခက်ခဲ(ရှုပ်ထွေး)နိုင်သည်။ ထို့ကြောင့် အရင်
ဖြေကြည့်လို့မရမှ အဖြေကြည့်ရန် လေးစာစွာ အကြံပေးအပ်ပါသည်။)

Loop Control Statements

Loop Control Statements သုံးမျိုးရှိသည်။၎င်းတို့မှာ continue ၊ break နှင့် pass တို့ဖြစ်သည်။

Continue Statement

```
>>> for i in 'Python':
...     if i=='h':
...         continue
...     print("The current Letter",i)
...
The current Letter P
The current Letter y
The current Letter t
The current Letter o
The current Letter n
>>>
```

Description

for loop တစ်ခုကို တည်ဆောက်လိုက်တယ်။ Python ဆိုတဲ့ string sequence ကို တစ်ခုချင်းစီ ပတ်မည်ဖြစ်သည်။ သို့သော် if နဲ့ i က h နဲ့ ညီသွားခဲ့ရင် ကျန်တဲ့ sequence ကို ဆက်လုပ်မည်ဖြစ်သည်။ ထို့ကြောင့် output တွင် h ပျောက်နေသည်ဖြစ်သည်။

Break Control Statement

```
>>> list=["Kyaw Kyaw","Aung Aung","Zaw Zaw","Mya Mya","Aye Aye"]
>>> for i in list:
...     if i=='Mya Mya':
...         break
...     print("The boy is",i)
...
The boy is Kyaw Kyaw
The boy is Aung Aung
The boy is Zaw Zaw
>>>
```

Description

Line 1: နာမည်တွေထည့်ထားတဲ့ list တစ်ခုကို တည်ဆောက်လိုက်တယ်။

Line 2: list ထဲ က items တွေကို for loop နဲ့ အလုပ်လုပ်စေမည်။

Line 3: i(က Mya Mya နဲ့ ညီသွားရင်

Line 4: break statement နဲ့ loop ကိုရပ်၍ ကျန်သော items များကို ဆက်၍ loop ပတ်တော့မည်မဟုတ်ပေ။

Line 5: output ထုတ်ပေးထားဖြစ်ပါသည်။

Pass Control Statement

Pass Statement သည် တခြား statement တွေလိုမဟုတ်ပါ။ သူသည် ဘာမှမရှိပါ။ သူကို အဓိက အားဖြင့် error တွေကို ကျော်စေချင်တဲ့ နေရာမှာအသုံးပြုသည်။

```
>>> for i in "hello":
...
...File "<stdin>", line 2
...
...^
IndentationError: expected an indented block
>>>
```

ပုံကိုကြည့်မယ်ဆိုရင် Indentation Error တက်တာကိုမြင်ရပါလိမ့်မယ်။

အောက်ကပုံကို ဆက်ရှုကြည့်ပါ။

```
>>> for i in "Hello":
...     pass
...
...
>>>
```

For loop မှာ pass ကိုထည့်ဖြစ်လိုက်တော့ကြောင့် error ကိုမပြတော့ပါ။ ဒီလောက်ဆိုနားလည်မယ်လို့ မျှော်လင့်ပါတယ်။

Challenge ==>

c==>35

User ရဲ့ အမည်ကို input ထည့်ခိုင်းပါ။ user password ကို ထက်ပြီး input ထည့်ခိုင်းပါ။ ထို့နောက် if loop နဲ့ user နဲ့ name password ညီမညီ စစ်ပါ။ ညီခဲ့ရင် ဒုတိယ password (two factor) ထည့်ခိုင်းပါ။ ထို password ကို if နဲ့ပဲ ထည့်စစ် မှန်ပါ။ မှန်ခဲ့ရင် success လို့ output ထုတ်ပြပြီး။ မအောင်မြင်ခဲ့ရင် fail လို့ output ထုတ်ပြပါ။ ဒီလိုမှမဟုတ်ဘဲ အစတည်းက မှားခဲ့ရင် incorrect လို့ output ထုတ်ပြပါ။

Challenge==>36

a ရဲ့ တန်ဖိုးကို 0 လို့သတ်မှတ်ပါ။ ထို့နောက် a ရဲ့တန်ဖိုး 0 ဖြစ်နေရင် love ဆိုတဲ့ variable နဲ့အတူ user ကို "Do you love me?Y/N" ဆိုတဲ့နေရာမှာ y or n ဖြေခိုင်းပါ yes ဖြစ်ခဲ့ရင် for loop ကို အသုံးပြုပြီး "I love you too" ဆိုတဲ့ စသားကို မိမိနှစ်သက်သလောက်အကြိမ်ရေထုတ်ပါ။ အောက်တစ်ကြောင်းတွင် a=1 လို့သတ်မှတ်လိုက်ပါ။ ထို့မှသာ a တန်ဖိုး 1 ဖြစ်သွား၍ နောက်တစ်ကြိမ် ဆက်၍ အလုပ်ဆက်လုပ်တော့မည်မဟုတ်ပေ။ ထို့နောက် N ဖြစ်ခဲ့ရင် "Are you sure" ဆိုတဲ့ စသားကို output ထုတ်ပါ။ ထို့ကြောင့် a=0 ဆက်ထားသောကြောင့် နောက်တစ်ကြိမ် ချစ်မချစ်မေးမည်။ ကိုယ်တိုင် run ကြည့်မှ သိမယ်ဗျ။

challenge==>37

Passwords တွေပါတဲ့ list တစ်ခုကို တည်ဆောက်ပါ။ for loop ကို အသုံးပြုပါ။ loop အောက်တွင် list ထဲမှာပါတဲ့ password ကို တစ်ခုကို variable name တစ်ခုထည့်မှာပါသို့လှောင်ထားပါ။ if loop သုံး၍ variable က password ညီသွားခဲ့ရင် password ကိုရပ်၍ ၎င်း password ကို output ထုတ်ပြပါ။

Solution

C==> 35

```
user=str(input("Enter Your Name: "))
password=str(input("Enter Your Password: "))
if user=="kyawkyi" and password=="123@#abc":
    second=int(input("Enter Second Password: "))
    if second==123476:
        print("suceess")
    else:
        print("fail")
else:
    print("incorrect")
```

C==>36

```
a=0
while a==0:
    love=input("Do you love me?Y/N: ")
    if love=="y" or love=="Y":
        for i in range(0,100):
            print("I love you too.")
        a=1
    else:
        print("Are you sure.")
```

c==>37

```
p_list=["akr333","ak44","ak12345","@#$123abc","234!@#4"]
```

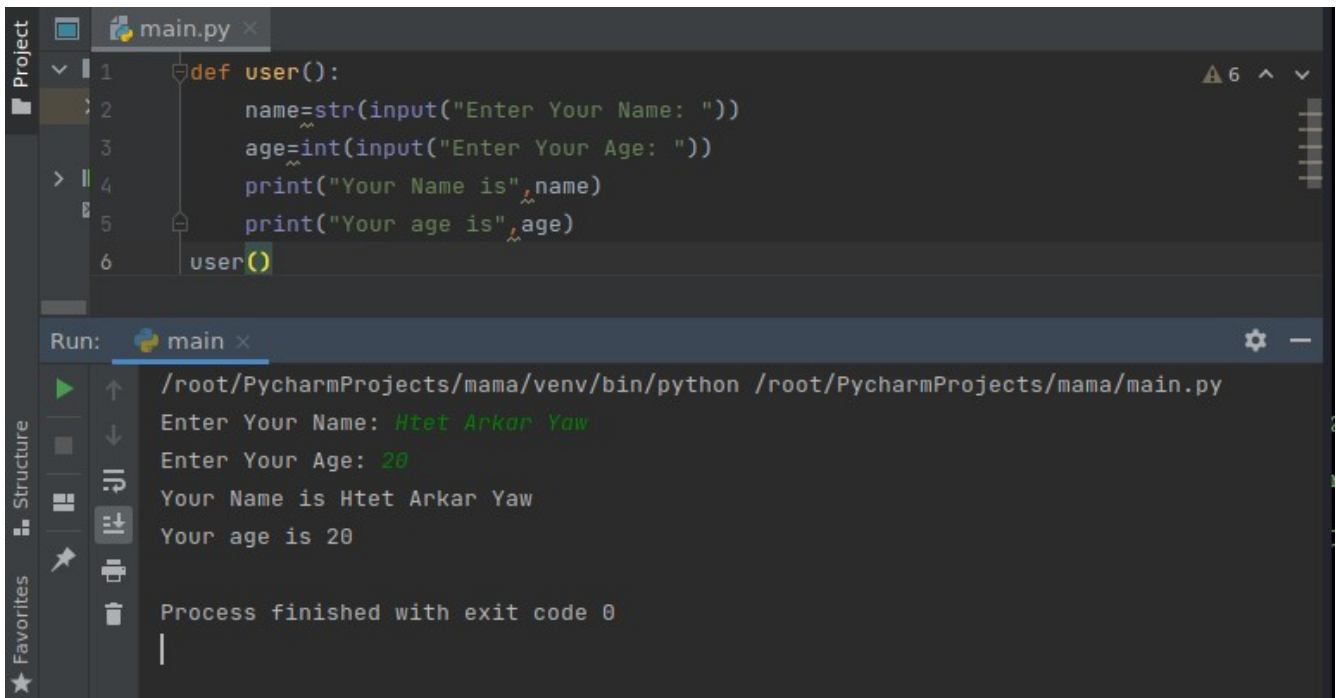


```
for p in p_list:
    password="ak12345"
    if p==password:
        print(password)
        break
```

Chapter – 10

User define Function

User Define Function ဆိုတာ subprogram တစ်ခုဖြစ်သည်။ Subprogram ဆိုတာ code အစုအဝေး (blocks of code) ပင်ဖြစ်သည်။ ကျနော်လိုချင်တဲ့ အပိုင်းတွေကို ခွဲပြီး ရေးလိုက်တာဖြစ်ပါတယ်။ ဥပမာ ပြောရရင် အိမ်တစ်အိမ်က ကျနော်တို့ရဲ့ main program ကြီးပေါ့ဗျာ။ အိမ်ထဲမှာ ဘုရားခန်း၊ အိပ်ခန်း၊ မီးဖိုခန်း၊ ဧည့်ခန်း စသည့်ဖြင့် အခန်းတွေခွဲထားတယ်မလား။ ထိုခွဲထားတဲ့ အခန်းတွေက ကျနော်တို့ရဲ့ subprogram တွေပေါ့ဗျာ။ အခန်းမခွဲထားရင် ကြည့်ရရှုပ်နေမှာပေါ့။ ကျနော်တို့ရဲ့ ကိုယ်ပိုင် function ကို တည်ဆောက်ရာတွင် def keyword နဲ့ အတူ name(): ယခုလိုရေးပြီးနောက် အောက်တွင် ကျနော်တို့ရဲ့ statement တွေ code တွေကိုရေးနိုင်ပါတယ်။



The screenshot shows the PyCharm IDE interface. The top pane displays a Python script named `main.py` with the following code:

```
1 def user():
2     name=str(input("Enter Your Name: "))
3     age=int(input("Enter Your Age: "))
4     print("Your Name is",name)
5     print("Your age is",age)
6     user()
```

The bottom pane shows the execution output for the `main` script:

```
/root/PycharmProjects/mama/venv/bin/python /root/PycharmProjects/mama/main.py
Enter Your Name: Htet Arkar Yaw
Enter Your Age: 20
Your Name is Htet Arkar Yaw
Your age is 20
Process finished with exit code 0
```

Description

- Line 1: `def` keyword အတူ `user` function တစ်ခုကို တည်ဆောက်လိုက်တယ်။
- Line 2: `name` variable အတူ `user` name ကို `input` ထည့်ခိုင်းတယ်။
- Line 3: `age` variable အတူ `user` ရဲ့ အသက် ကို `input` လုပ်ခိုင်းတယ်။
- Line 4: `Your name is` ဆိုတဲ့ စာသားနဲ့ အတူ `user` ရဲ့ အမည်ကို `output` ထုတ်ခိုင်းတယ်။
- Line 5: `Your age is` ဆိုတဲ့ စာသားနဲ့ အတူ `user` ရဲ့ အသက်ကို `output` ထုတ်ခိုင်းတယ်။
- Line 6: `user()` ဆိုတာ ကျနော်တို့ရဲ့ subprogram ကို `run` လိုက်ခြင်းဖြစ်ပါတယ်။

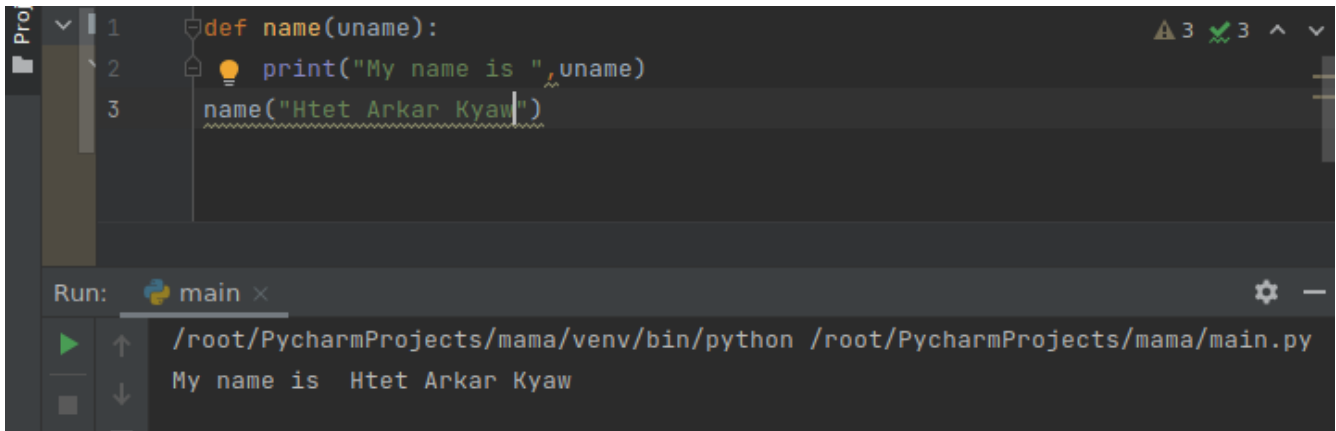
6.1

User Define Function မှာ parameter နဲ့ argument တို့ထည့်လို့ရတယ်။Parameter ဆိုတာ variable (argument name) ဖြစ်ပြီး argument ဆိုတာ value ဖြစ်သည်။နားလည်အောင်အောက်က ဥပမာကြည့်ပါ။

```
def user(parameter):  
    print("Your name is")
```

```
user('arguments')
```

Example



```
1 def name(uname):
2     print("My name is " + uname)
3     name("Htet Arkar Kyaw")
```

Run: main x

/root/PycharmProjects/mama/venv/bin/python /root/PycharmProjects/mama/main.py

My name is Htet Arkar Kyaw

Description

def name ဆိုပြီး function တစ်ခုတည်ဆောက်လိုက်တယ်။ Function ထဲတွင် uname ဆိုတဲ့ parameter (variable) ကို ထည့်ထားတယ်။ ထို့နောက် My name is ဆိုတဲ့ စာသားနဲ့အတူ parameter ကို output ထုတ်ခိုင်းတယ်။ထို့နောက် name("Htet Arkar Kyaw") ဆိုပြီး function ကို run စေပြီး parameter နဲ့တန်ဖိုး value ကို သတ်မှတ်လိုက်သည်။ ထို့ကြောင့် parameter ရဲ့ တန်ဖိုးက Htet Arkar Kyaw ဖြစ်ပြီး output တွင် My name is Htet Arkar kyaw လို့ထွက်လာခဲ့ချင်းဖြစ်သည်။

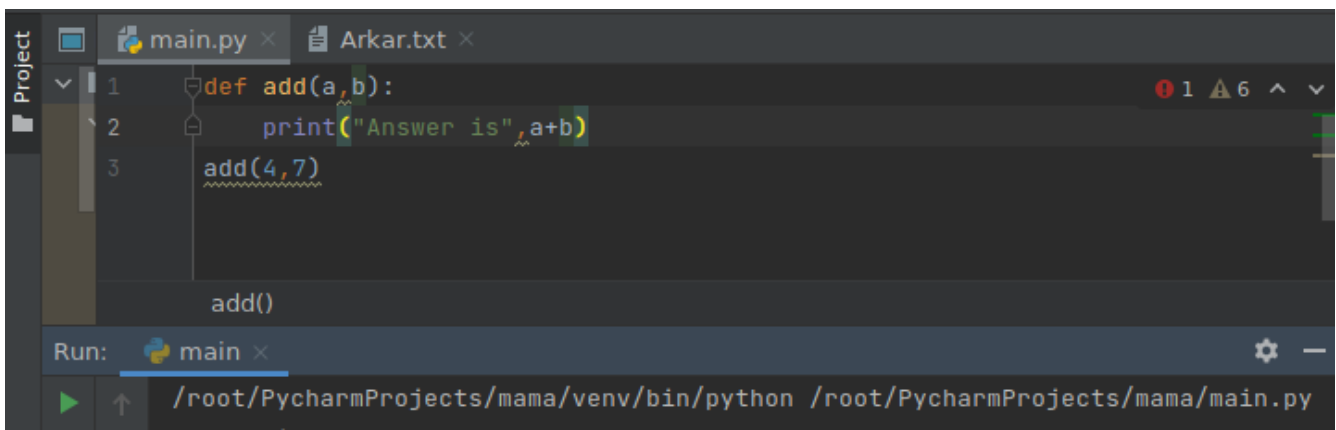
{Note:

uname="Htet Arkar Kyaw"

uname ဆိုတဲ့ variable ထဲကို Htet Arkar Kyaw ဆိုတဲ့ string data type သိုလှောင်လိုက်သည်။

print function နဲ့ ပြန်ပြီး output ထုတ်သည်။(နားလည်မယ်လို့မျှော်လင့်ပါတယ်)}

The next example



```
1 def add(a,b):
2     print("Answer is",a+b)
3     add(4,7)
```

Run: main x

/root/PycharmProjects/mama/venv/bin/python /root/PycharmProjects/mama/main.py

Answer is 11

Description

add ဆိုတဲ့ function တစ်ခုတည်ဆောက်လိုက်တယ်။ထို့နောက် parameter a b ကိုထည့်လိုက်သည်။ ထို့နောက် Answer is ဆိုတဲ့ စာသားနဲ့အတူ a နဲ့ b ပေါင်းထားတဲ့ number ကို output ထုတ်ခိုင်းသည်။

argument အနေနဲ့ a ရဲ့ တန်ဖိုး 4 ၊ b ရဲ့တန်ဖိုး 7 လို့သတ်မှတ်လိုက်သည်။ထို့ကြောင့် output အနေနဲ့ The number is 11 ပေါ်ခြင်းဖြစ်သည်။

Escape Character

Symbol	Description
\\	backslash
\'	single quote
\\	double quote
\n	new line
\t	tab

Beginner Level အနေနဲ့ ဒီလောက်ဆိုအဆင်ပြေပါတယ်။တခြား character တွေကို ကိုယ်ဘာသာ လေ့လာကြည့်ပါဗျ။

Example==>

```
>>> print("Hello\tArkar")
Hello  Arkar
>>> print("Hello\nArakr")
Hello
Arakr
>>>
```

Description

Hello နဲ့ Arkar ကြားက \t ဆိုတာ tab တစ်ခါခုန်တာပါ။ ဒါကြောင့်အောက်က output မှာ Hello နဲ့ Arkar က tab တစ်ခါခြားစာရှိနေတာဖြစ်ပါတယ်။

နောက်တစ်ကြောင်း \n ဆိုတာ new line နောက်တစ်ကြောင်းကိုဆင်းပါ လို့ဆိုလိုခြင်းဖြစ်ပါသည်။ ဒါကြောင့် Hello ပြီးနောက် အောက်တစ်ကြောင်းရောက်မှ Arkar လို့ပေါ်ခြင်းဖြစ်သည်။

Next Example

```
>>> print('I'm arkar')
File "<stdin>", line 1
    print('I'm arkar')
          ^
SyntaxError: invalid syntax
>>>
```

ယခု ဥပမာ ကျနော်တို့အနေနဲ့ I'm arkar လို့ ပေါ်ရမည်ဖြစ်သော်လည်း single quote နှစ်ခုကြားမှာ single quote ထည့်ရေးလို့မရပါ။ထို့ကြောင့်လဲ Escape character ကိုအသုံးပြုခြင်းဖြစ်သည်။

Example

```
>>> print('I\'m arkar')
I'm arkar
>>> 
```

Description

\' ဆိုတဲ့ escape character ထည့်လိုက်သောကြောင့် အပေါ်က လို့ error တက်သောမည် မဟုတ်ပေ။ထို့ကြောင့် I'm arkar ဆိုတဲ့ output ကိုထုတ်ပေးခြင်းဖြစ်သည်။အကယ်၍ double quote ကြားမှာ double quote ထည့်ချင်တယ်ဆိုရင်လဲ ယခုသကဲ့သို့အသုံးပြုရမည်။ဖြစ်သည်။

Example

```
>>> print("She asked \"How are you\"")
She asked "How are you"
>>> 
```

Description

\” ဆိုတဲ့ escape character ထည့်လိုက်သောကြောင့် error မတက်ပေ။ထို့ကြောင့် She asked “How are you” လို့ output ထွက်ချင်းဖြစ်သည်။

Triple Quote

ကျနော်တို့တွေ စာပိုဒ်တစ်ခု သို့ menu စသည့်ဖြင့်ပြုလုပ်တဲ့အခါမှာ \n \t စသကဲ့ escape characters တွေကိုအသုံးပြုချင်တဲ့အခါမှာအသုံးပြုပါတယ်။

```

1 def menu():
2     print("""
3         Students's Information
4         1.  Kaung Kaung
5         2.  Aung Aung
6         3.  Naing Kyaw  """)
7     menu()

```

main x

```

/root/PycharmProjects/mama/venv/bin/python /root/PycharmProjects/m

```

```

        Students's Information
        1.  Kaung Kaung
        2.  Aung Aung
        3.  Naing Kyaw

```

Process finished with exit code 0

Description

ကျနော်တို့ menu ဆိုတဲ့ function တစ်ခုကိုတည်ဆောက်လိုက်တယ်။ ထို့နောက် print function ကို triple quote နဲ့ students's information ခေါင်းစဉ် နဲ့ အတူ ကျောင်းသားသုံးယောက်ရဲ့ အမည်နဲ့အတူ menu တစ်ခုတည်ဆောက်လိုက်တယ်။ triple quote နဲ့ရေးထားသောကြောင့် ကျနော်တို့ရေးထားတဲ့ပုံစံအတိုင်းပေါ်မည်ဖြစ်သည်။ နောက်တစ်ဆင့်အနုစေ့နဲ့ color တွေထည့်ကြည့်တာပေါ့။

Text color In python3

Symbol	l	Name
\033[0;30m		Black
\033[0;31m		Red
\033[0;32m		Green
\033[0;33m		Brown
\033[0;34m		BLUE
\033[0;35m		Purple
\033[0;36m		Cyan
\033[0;37m		Light Gray
\033[1;33m		Yellow

Example ==>


```
1 def menu():
2     print("""
3         \033[0;31mStudents's Informations
4         \033[0;32m 1.  Kaung Kaung
5         \033[0;34m 2.  Aung Aung
6         \033[1;33m 3.  Naing Kyaw  """)
7     menu()

menu()
```

main x

/root/PycharmProjects/mama/venv/bin/python /root/PycharmProjects/mama/main.py

```
Students's Informations
1.  Kaung Kaung
2.  Aung Aung
3.  Naing Kyaw
```

Process finished with exit code 0

Description

Menu ဆိုတဲ့ function တစ်ခုတည်ဆောက်လိုက်ပါတယ်။ခေါင်းစဉ်စာသားကို အနီရောင် ပြောင်းထားပါတယ်။အောက်စာသားများနဲ့ စာကားလုံးအရောင်ပြောင်းထားပါတယ်။

Challenge

c==>38

question ဆိုတဲ့ function တစ်ခုတည်ဆောက်ပါ။ ပထမဆုံးတွင် a ရဲ့တန်ဖိုးကို 0 လို့သတ်မှတ်ပေးပါ။
ထို့နောက် user ကို question ကိုဖြေမဖြေမေးပါ။ while loop နဲ့ဖြေမဖြေစစ်ပါ။ဖြေခဲ့တယ်ဆို user
input နဲ့ Y/N မေးခွန်းတစ်ခုမေးပါ။ အဖြေမှန်ခဲ့ရင် "your answer is correct. Your mark is 1 ဆိုပြီး
output ထုတ်ပါ။ မှားခဲ့ရင်လဲ အမှတ်နဲ့အတူ မှားကြောင်း output ထုတ်ပေးပါ။ ထို့နောက် while loop
ရဲ့ else မှာ finished လို့ output ထုတ်ပေးပါ။

C==>39

info ဆိုတဲ့ function တစ်ခုတည်ဆောက်ပါ။ အထဲတွင် students's information ဆိုတဲ့ ခေါင်းစဉ်နဲ့
အတူ ကျောင်းသားနှစ်ယောက်နဲ့ အမည်ကိုထည့်ပါ။ ထို့နောက် user ကို သိချင်တဲ့ ကျောင်းသားရဲ့
နံပါတ်ကို input ထည့်ခိုင်းပါ။ထည့်ခိုင်းတဲ့ ကျောင်းသားရဲ့ အချက်အလက်ကို ပြသပေးပါ။ ထို့ကြောင့်

ကျောင်းသားနှစ်ယောက် အချက်အလက် ကို function တစ်ခုစီ တည်ဆောက်လိုက်ပါ။ စကားလုံးတွေ ကို ကာလာထည့်ပါ။

Solution

C==>38

```
def question():
    a=0
    que=str(input("Do you want to answer this question?Y/N: "))
    while que=="Y" or que=="y":
        ans=str(input("Computer is a electronic devcie.Y/N: "))
        if ans=="Y" or ans=="y":
            a+=1
            print("Your answer is correct.", "Your mark is", a)
        else:
            a=0
            print("Your answer is incorrect.Your mark is", a)
        else:
            print("Finished")
    question()
```

C==>39

```
def info():
    print("""
        \033[0;31mStudent's Information
        1. Kyaw Kyi
        2. Maung Zaw
        3. Exit
```

```

"""
a=int(input("\033[0;32mEnter Number: "))
if a==1:
    kyaw()
elif a==2:
    zaw()
else:
    exit()
def kyaw():
    print("""\033[0;34m
    Name = Kyaw Kyi
    Age = 14
    Phone = 0933424645444
    """)
def zaw():
    print("""\033[0;30m
    Name= Maung Zaw
    Age = 16
    Phone = 093345678203
    """)
info()

```

(Note ကျနော်တို့ function ကိုခေါ်မှ အလုပ်လုပ်မည်ဖြစ်သည်။ ဒါကြောင့် if a==1 ရဲ့အောက်မှာ kyaw() ဆိုပြီး kyaw လို့အမည်ပေးထားတဲ့ function ကို အလုပ်လုပ်လာရန်ခေါ်လိုက်ခြင်းဖြစ်သည်။ အောက်တွင် kyaw ဆိုတဲ့ function ကိုတည်ဆောက်ထားတာတွေ့ပါလိမ့်မယ်။exit() ဆိုတာ program က နေ ထွက်တဲ့နေရာမှာအသုံးပြုပါတယ်။ အောက်ဆုံး info() က info function ကို အလုပ်လုပ်ရန် (run) ရန် ခေါ်ဆိုလိုက်ခြင်းဖြစ်သည်။(ကိုယ်တိုင်လုပ်ကြည့်ရင်သိပါလိမ့်မယ်ဗျ။)

Chapter -11

Error Handle

user က input ထဲ မှားနိမ့်မိရင်ဖြစ်ဖြစ် program က ကို error တက်ရင်ဖြစ်ဖြစ် error တက်တဲ့ နေရာကိုပြပြီး ပြန်ပြီး run စေတဲ့နေရာမှာအသုံးပြုနိုင်သလို ပြန်ပြီး လဲ run ပြလို့ရပါတယ်။

```
>>> if a==0:
...     print("Zero")
...
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'a' is not defined
>>>
```

Description

ဒီမှာဆို ကျနော်က a ကို variable အဖြစ်သတ်မှတ်ခဲ့ပါဖူး။ ထို့နောက် if မှာ a ကိုညီခိုင်းစေတယ်။ ထို့နောက် zero ကို output ထုတ်ခိုင်းခဲ့တယ်။ a ဆိုတာကို ဘာမှမသတ်မှတ်ခဲ့ သောကြောင့် name error တက်ခဲ့ပါတယ်။ Program နှစ်ခုနဲ့ အောက်မှာရှင်းပြပါမယ်။

```
>>> a=1
>>> if a==1:
...     print("Zero")
... else:
...     print("one")
...
Zero
>>>
```

a ကို 1 လို့သတ်မှတ်ခဲ့တယ် တစ်နည်းပြောရရင် a ကို variable name အဖြစ်သတ်မှတ်ခဲ့ပြီးဖြစ်သည်။ ထို့နောက် a က 1 နဲ့ ညီခဲ့ရင် 0 မညီခဲ့ဖူးဆိုရင် one လို့ output ထုတ်လိမ့်မည်။ a က 1 နဲ့ ညီလို zero လို့ output ထုတ်ခဲ့ပါတယ်။

```
>>> if b==1:
...     print("Hello")
... else:
...     print("Bye")
...
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'b' is not defined
>>>
```

b ကို variable name အဖြစ်သတ်မှတ်ခဲ့ဖူး ဒါကြောင့် Name error တက်တာဖြစ်ပါတယ်။ Try except နဲ့ ရေးကြည့်အောင်။

```
>>> try:
...     if b==1:
...         print("Hello")
...     else:
...         print("Bye")
... except NameError:
...     print("Please defined b")
...
Please defined b
>>>
```

try နဲ့ ကျနော်တို့ရဲ့ code ကိုစတင် ကိုင်တွယ်ထားလိုက်တယ်(run လိုက်တယ်)။ ထို့နောက် b က 1 နဲ့ ညီခဲ့ရင် Hello ပေါ့။မညီခဲ့ရင် Bye ပေါ့။ကျနော်အပေါ်က ပြောခဲ့သလို b ကို variable တစ်ခုခုအဖြစ်မ သတ်မှတ်ခဲ့တဲ့အတွက် name error တက်တယ်။ဒါကြောင့် program အနေနဲ့ Name error တက်ခဲ့ရင် Please defined b လို့ output ထုတ်ပေးပါပေါ့။

(Note: Programmer တစ်ယောက်အနေနဲ့ ကျနော်တို့ရဲ့ code ကို error တက်အောင်ရေးမည်မဟုတ်။ ထို့ကြောင့် user က input ထည့်လိုက်တဲ့ error handling ပိုင်း နှင့် module error ပိုင်း ဦးစားပေးသွား မှာပါ။)

i

```
>>> num=int(input("Enter Number: "))
Enter Number: qwe
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: invalid literal for int() with base 10: 'qwe'
>>>
```

Description

Program အရဆိုရင် ကျနော်တို့က user ကို int ဆိုတဲ့ ကိန်းပြည့်ကိုသာ ထည့်ခိုင်းတာပါ။ သို့ပေမဲ့ user က number မထည့်ဘဲ qwe ဆိုပြီး string data type ကိုထည့်လိုက်သော်ကြောင့် ValueError တက်လာသည်။ တကယ်လို့ user က ယခုသကဲ့သို့ int မဟုတ်ဘဲ string ထည့်လို့ ၎င်း ကဲ့သို့ error အရှည်ကြီးပြလာရင် programmer တစ်ယောက်အနေနဲ့ မသင့်တော်ပါဖူး။ နောက် example ကို ဆက်ကြည့်ပါ။

Example

```
>>> try:
...     num=int(input("Enter Number: "))
...     print("Number is",num)
... except ValueError:
...     print("Please,enter only number")
...
Enter Number: abc
Please,enter only number
```

Description

try နဲ့ကျနေတဲ့တို့ရဲ့ code ကို စတင်ကိုင်တွယ်ထားတယ်။ထို့နောက် user ကို integer data type ကို input သွင်းခိုင်းတယ်။ user က int မဟုတ်ဘဲ str(string) ကိုရိုက်ချလိုက်တယ်။ အမှန်ဆို အပေါ်က ValueError စသဖြင့် အရှည်ကြီး ပေါ်ရမှာပါ။ သို့ပေမဲ့ except ValueError ဆိုတာ အကယ်၍ ValueError တက်လာခဲ့ရင် user ကို Please,enter only number လို့ output ထုတ်ပေးပါ။ user ဘက်ကလဲ အငယ်နည်းအထင်ကြီးသွားနိုင်တာပေါ့။ ဒီလောက်ဆိုနားလည်မယ်လို့မျှော်လင့်ပါတယ်။ Module နဲ့ ပတ်သတ်တဲ့ error လဲ ရှိပါသေးတယ်။နောက်ပိုင်းမှာဖော်ပြထားပါတယ်။

Challenge-39

အင်္ကျီတစ်ထည် ကို 1500 ကျပ်။ user လိုချင်တဲ့ အထည်အရေအတွက်ကို input function နဲ့ ထည့်ခိုင်းပါ။အကယ်၍ input မှာ တခြား string ထည့်ခဲ့မယ်ဆိုရင် only enter number ဆိုတဲ့ error message ကိုပြပြီး code ကို ပြန် auto ပြန် run ရမည်ဖြစ်သည်။

Challenge-40

try နဲ့ code ကို run ရမည်ဖြစ်သည်။ a ရဲတန်ဖိုးကို 30 ထားပါ။ num variable နဲ့အတူ user ကို ကိန်းပြည့်တန်ဖိုးသွင်းခိုင်းပါ။ထို့နောက် a ကို num နဲ့ စားပြီး c variable ထဲမှာ သိုလှောင်ရမည်ဖြစ်သည်။ ပြီးနောက် c ကို output ပြန်ထုတ်ပြပါ။ ထို့နောက် zero division error တက်ခဲ့ရင် Except ဖြင့် zero unavailable လို့ output ထုတ်ပြရမည်ဖြစ်သည်။

Challenge-41

try နဲ့ code ကို စ run ရမည်။ user ကို ပန်းသီးအရေအတွက်ထည့်ခိုင်းပါ။ထို့နောက် ပန်းသီးအရေအတွက် နဲ့ c ကိုမြှောက်ပြီး b variable ထဲသိုလှောင်လိုက်ပါ။ထို့နောက် output ထုတ်ပြပါ။ c ကို define မလုပ်ခဲ့အတွက် name error တက်မည်ဖြစ်သည်။ ထို့ကြောင့် except ဖြင့် name error တက်ခဲ့ရင် please define c ဆိုတဲ့ စာသားကို output ထုတ်ပြရမည်ဖြစ်သည်။

Solution-39

try:

price=1500

num=int(input("Enter Number of clothes: "))

sum=price*num

print(sum)

except ValueError:

print("Enter Only Number")

Solution=40

try:

a=30

num=int(input("Enter Number: "))

c=a/num

print(c)

except ZeroDivisionError:

print("zero unavailable")

Solution -41

try:

num=int(input("Enter Number of apples: "))

b=1*c

print(b)

except NameError:

print("Please Define c")

Chapter-12

ကျနော်တို့ module အကြောင်းမသွားခင် PIP အကြောင်းသွားကြတာပေါ့။ PIP က Python module or package အတွက် package manager ဖြစ်သည်။ Package or module ဆိုတာ package module နဲ့ ပတ်သတ် တဲ့ Function တွေကို ထည့်ထားတာပါ။

Packager or Module ကို install လုပ်မယ်ဆိုရင်တော့

`pip install <package name>`

time module ကို အသုံးပြုမှာဖြစ်သောကြောင့် terminal မှာ

`pip install time`

ဆိုပြီး time module ကို install လုပ်လိုက်သည်။

Example

```
# pip install time
```

Module

Module ဆိုတာဘာလဲ Module ဆိုတာ library တစ်ခုပါ။ အဲ့ library ထဲမှာ သူနဲ့သက်ဆိုင်တဲ့ function တွေ ထည့်ထားပါတယ်။

Time module

module တစ်ခုကို အသုံးပြုမည့်အကြောင်း ကို ကျနော်တို့ကြေညာပေးဖို့လိုပါတယ်။ အောက်က ဥပမာ ကိုကြည့်ပါ။

Example

```
>>> import time
>>> time.ctime()
'Thu May 20 21:29:16 2021'
>>>
```

Description

time function တွေကို အသုံးပြုချင်သောကြောင့် `import time` ဆိုပြီး time module ကို ကြေညာလိုက်သည်။ထို့နောက် `time.ctime()` ဆိုပြီး ယခုရက်စွဲကို output ထုတ်ခိုင်းခြင်းဖြစ်သည်။

Renaming Module

time လို့ module ရေးတာထဲ t လို့ ရေးတာက ပိုပြီး အချိန်သက်သာတဲ့ မလား။

Example

```
>>> import time as t
>>> t.ctime()
'Thu May 20 21:34:31 2021'
>>> █
```

time module ကို import လုပ်လိုက်တယ်။ ထို့နောက် time လို့ အရှည်ကောက်ထက် t ဆိုပြီး အမည် ပြောင်းလိုက်တယ်။ ထို့နောက် t.ctime() ဆိုပြီး ရက်နေ့ date ကို output ထုတ်လိုက်တယ်။

module ထဲက function တစ်ခုကို အသုံးပြုခြင်း

ကျနော်တို့ ctime ဆိုတာ time module ထဲက function တစ်ခုမဟုတ်လား။ အဲ့ module ကိုပုံမှန်စာ ကြောင်းအတိုင်း အသုံးပြုချင်တဲ့အခါမှာအသုံးပြုတယ်။

Example

```
>>> from time import ctime
>>> ctime()
'Thu May 20 21:51:29 2021'
>>> █
```

Description

line 1: time module ထဲက ctime function ကို အသုံးပြုပေးပါ။

line 2: ctime() ဆိုပြီး ယနေ့ date ကို output ထုတ်ပေးမည်ဖြစ်သည်။

(နားလည်မယ်လို့မျှော်လင့်ပါတယ်)

Time sleep Function

ကျနော်တို့ program တစ်ခုကို run နေချိန် ခန့်ရပ်စေချင်တဲ့အခါ မှာအသုံးပြုပါတယ်။

Example

```
import time as t
print("Starting.....")
t.sleep(5)
print("Ended")
```

main x

/root/PycharmProjects/mama/venv/bin/python /root/PycharmProjects/mama

Starting.....

Ended

Process finished with exit code 0

Description

Line 1 time ကို t အသေးအဖြစ်သတ်မှတ်ထားတယ်။

Line 2 Starting ဆိုတဲ့ စာသားကို output ထုတ်တယ်။

Line 3 t.sleep(5) ဆိုပြီး 5 second ခန့် ရပ်နားထားလိုက်သည်။

Line 4 Ended ဆိုတဲ့ စာသားကို output ထုတ်သည်။(run ကြည့်ပါ)

(Note တခြား functions တွေရှိပါသေးတယ် ကိုယ်ဘာသာ ဆက်၍ လေ့လာပါ)

Challenge-42

ကျောင်းသားကျောင်းသူတွေရဲ့ အချက်အလက် ကို dictionary ဖြင့်ဖန်တီးပါ။ထို့နောက် list တစ်ခုလုပ်ပါ။ list ကို for loop ဖြင့်ပတ်ပါ။အကယ်၍ list ထဲက student တစ်ယောက်ရဲ့ အချက်အလက်နဲ့ ညီသွားခဲ့ရင် သုံးစက္ကန့်ကြာတာနဲ့ ထို့ကျောင်းသားရဲ့ အချက်အလက်ကို output ထုတ်ပေးရမည်။

Challenge-43

Company ဝန်ထမ်းတွေရဲ့ data တွေကို dictionary ဖြင့် ဖန်တီးထားပါ။ထို့နောက်သူတို့ရဲ့ အချက်အလက် တွေကို ၃ စက္ကန့်ခြားစီ output ထုတ်ပြပါ။(ကျနော်တော့အငယ်နည်းပဲ လုပ်မယ်နော်)

Challenge=44

Auto message program တစ်ခုပေါ့။ ကျနော်တို့ စာတစ်ကြောင်းရေးရင် auto reply တစ်ခုလာမယ်။ အပျော်ပေါ့။

Solution-42

```
import time
a={"name":"Htet Arkar Kyaw","Age":15,"Class":"Grade(10)-B"}
b={"name":"Aye Mya","Age":14,"Class":"Grade(9)-A"}
c={"name":"Mauang Maung","Age":15,"Class":"Grade(10)-A"}
d=[a,b,c]
for i in d:
    if i==b:
        time.sleep(3)
        print(i)
```

Solution-43

```
import time
a={"Name":"Arkar Kyaw","Position":"Manager","Salary":1000000,"Phone Number":"0992983944"}
b={"Name":"Zin Min Ko","Position":"Clerk","Salary":250000,"Phone Number":"09393029200"}
```


Chapter-13

OS Module

Os module က တကယ်အသုံးဝင်တဲ့ module တစ်ခုဖြစ်ပါတယ်။ ကျနော်ရဲ့ စာအုပ်မှာ တော့ function s တစ်ချို့ပဲဖော်ပြမယ်ဆိုပေမဲ့ ကျန်တာ ဆက်လေ့လာဖို့ အထောက်အကူပြုပါတယ်။ Os module သည် operating system နဲ့ အပြန်အလှန် ချိတ်ဆက် အသုံးပြုပါတယ်။

Example

```
>>> import os
>>> os.getcwd()
'/home/soloninja'
>>>
```

Description

Line 1: os module အသုံးပြုရန်အတွက် import လုပ်လိုက်ခြင်းဖြစ်သည်။

Line 2: os.getcwd() က ကျနော်ယခုရောက်ရှိတဲ့ current Directory ကိုဖော်ပြပေးတာဖြစ်ပါသည်။

```
>>> print(dir(os))
['CLD_CONTINUED', 'CLD_DUMPED', 'CLD_EXITED', 'CLD_KILLED', 'CLD_STOPPED', 'C
', 'EX_NOHOST', 'EX_NOINPUT', 'EX_NOPERM', 'EX_NOUSER', 'EX_OK', 'EX_OSERR',
EX_USAGE', 'F_LOCK', 'F_OK', 'F_TEST', 'F_TLOCK', 'F_ULOCK', 'GRND_NONBLOCK',
TLB', 'MFD_HUGE_16GB', 'MFD_HUGE_16MB', 'MFD_HUGE_1GB', 'MFD_HUGE_1MB', 'MFD_
', 'MFD_HUGE_512MB', 'MFD_HUGE_64KB', 'MFD_HUGE_8MB', 'MFD_HUGE_MASK', 'MFD_H
D', 'O_ASYNC', 'O_CLOEXEC', 'O_CREAT', 'O_DIRECT', 'O_DIRECTORY', 'O_DSYNC',
O_NONBLOCK', 'O_PATH', 'O_RDONLY', 'O_RDWR', 'O_RSYNC', 'O_SYNC', 'O_TMPFILE',
FADV_NORMAL', 'POSIX_FADV_RANDOM', 'POSIX_FADV_SEQUENTIAL', 'POSIX_FADV_WILLN
', 'PRIO_PROCESS', 'PRIO_USER', 'P_ALL', 'P_NOWAIT', 'P_NOWAITO', 'P_PGID', '
D_LAZY', 'RTLD_LOCAL', 'RTLD_NODELETE', 'RTLD_NOLOAD', 'RTLD_NOW', 'RWF_DSYNC
', 'SCHED_IDLE', 'SCHED_OTHER', 'SCHED_RESET_ON_FORK', 'SCHED_RR', 'SEEK_CUR',
', 'ST_NOATIME', 'ST_NODEV', 'ST_NODIRATIME', 'ST_NOEXEC', 'ST_NOSUID', 'ST_RD
WCOREDUMP', 'WEXITED', 'WEXITSTATUS', 'WIFCONTINUED', 'WIFEXITED', 'WIFSIGNA
', 'WUNTRACED', 'W_OK', 'XATTR_CREATE', 'XATTR_REPLACE', 'XATTR_SIZE_MAX', 'X_O
', '__loader__', '__name__', '__package__', '__spec__', '_check_methods', '_e
nvef', '_walk', '_wrap_close', 'abc', 'abort', 'access', 'altsep', 'chdir', '
copy_file_range', 'cpu_count', 'ctermid', 'curdir', 'defpath', 'device_enco
eclp', 'execlp', 'execlpe', 'execv', 'execve', 'execvp', 'execvpe', 'extsep',
thconf', 'fsdecode', 'fsencode', 'fspath', 'fstat', 'fstatvfs', 'fsync', 'ftr
rminal_size', 'getcwd', 'getcwddb', 'getegid', 'getenv', 'getenvb', 'geteuid',
'getpgrp', 'getpid', 'getppid', 'getpriority', 'getrandom', 'getresgid', 'get
lpg', 'lchown', 'linesep', 'link', 'listdir', 'listxattr', 'lockf', 'lseek',
kfifo', 'mknod', 'name', 'nice', 'open', 'openpty', 'pardir', 'path', 'pathco
six_fadvise', 'posix_fallocate', 'posix_spawn', 'posix_spawnnp', 'pread', 'pre
t_fork', 'remove', 'removedirs', 'removexattr', 'rename', 'renames', 'replace
ched_getaffinity', 'sched_getparam', 'sched_getscheduler', 'sched_param', 'sc
ler', 'sched_yield', 'sendfile', 'sep', 'set_blocking', 'set_inheritable', 's
', 'setregid', 'setresgid', 'setresuid', 'setreuid', 'setsid', 'setuid', 'set
wnvp', 'spawnvpe', 'st', 'stat', 'stat_result', 'statvfs', 'statvfs_result',
ids', 'supports_fd', 'supports_follow_symlinks', 'symlink', 'sync', 'sys', 's
', 'times', 'times_result', 'truncate', 'ttyname', 'umask', 'uname', 'uname_re
aitid', 'waitid_result', 'waitpid', 'waitstatus_to_exitcode', 'walk', 'write'
>>>
```

Description

Line 1: ကျနော်တို့ os module မှာ အသုံးပြုလို့ရတဲ့ function တွေကို ကြည့်ချင်းဖြစ်ပါသည်။

Line 2: os module မှာ အသုံးပြုလို့ရတဲ့ functions တွေပဲဖြစ်ပါတယ်။

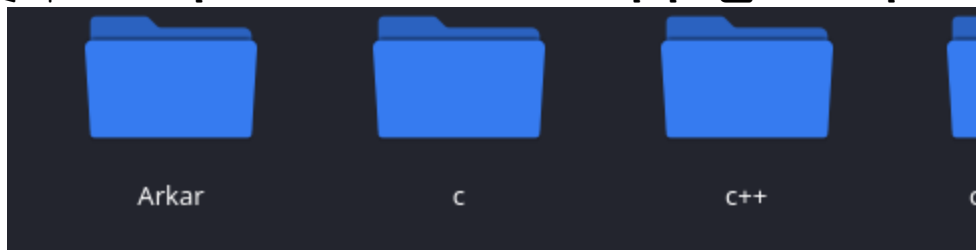
(Note: ကျနော်စာအုပ်အတွက်လိုအပ်တာကို သုံးပြသွားမှာဖြစ်ပါတယ်)

Create Folder(Directory)

ကျနော်တို့ Windows နဲ့ android မှာ ဆို folder လို့ခေါ်ပြီး၊ linux မှာဆိုရင် directory လို့ခေါ်ပါတယ်။

```
>>> os.mkdir("Arkar")
>>>
```

ဒါကတော့ ကျနော် Arkar လို့ခေါ်တဲ့ Directory (folder) တစ်ခုကိုတည်ဆောက်လိုက်တာဖြစ်ပါတယ်။



ကျနော်ရဲ့ current Directory အောက်ပါအတိုင်း directory တစ်ခုရှိနေတာကိုးတွေ့ရမှာပါ။

Directory ကိုဖျက်ခြင်း

```
>>> os.rmdir("Arkar")
>>>
```

os.rmdir က ကျနော်တို့ရဲ့ directory ကို ဖျက်လိုက်တာဖြစ်ပါတယ်။ မှုရင်းနေရာကိုပြန်သွားကြည့်ရင် ပျက်သွားတာကိုတွေ့ပါလိမ့်မယ်။(ကိုယ်တိုင်လေ့လာမှသိလိမ့်မယ်ဗျ)

လက်ရှိတဲ့ရောက်နေတဲ့ Folder နဲ့တွေ့ကို list လုပ်ကြည့်ချင်း

```
>>> os.listdir()
['software', 'publickey', 'Videos', 'a.py', '.vscode', 'Desktop',
'sh_history', '.bash_logout', '.profile', 'c', 'windowapp', 'c++',
'che', '.xsession-errors', 'Download', 'VidMate', '.BurpSuite', '.
', '.packettracer', 'Downloads', '.bashrc.original', '.gtkrc-2.0',
'VMs', '.Xauthority', '.local', '.gnupg', 'Documents', 'compTIA',
>>>
```

ပုံပါအတိုင်းကျနော်လက်ရှိရောက်နေတဲ့ Directory အောက်က folder file တွေကို တွေ့ရမှာဖြစ်ပါတယ်။

os.system ကို အသုံးပြုချင်း

os.system() က ကျနော်တို့ command line prompt နဲ့ တိုက်ရိုက်အလုပ်လုပ်တဲ့ဗျ။ round brackets ကြားမှာ အလုပ်လုပ်စေချင်တဲ့ command ကို ထည့်လိုက်တာနဲ့ အဲ့ command အတိုင်းအလုပ်လုပ်ပါတယ်။ အောက်က ဥပမာကိုကြည့်ပါ။

```
>>> os.system("ls")
'android studio'  Arduino  c++      compTIA  Download  Music    pentesting
a.py              basic.py 'ccna(class)' Desktop  Downloads network  Pictures
arduino          c        code    Documents lab      osecurity privatekey

>>> os.system("clear")
```

Description

ls ဆိုတာ linux မှာဆိုရင် list ထုတ်ကြည့်တာဖြစ်ပါတယ်။ဒါကြောင့် os.system function သုံးပြီးကျနော် list ထုတ်ကြည့်တာဖြစ်ပါတယ်။

clear ဆိုတာ screen ကိုရှင်းလိုက်တာပါ။ဒါကြောင့် os.system function သုံးပြီး system ကို clear လုပ်လိုက်တာဖြစ်ပါတယ်။(ကိုယ်တိုင်လေ့ကျင့်မှ ချလုပ်မှာ program ကို နားလည်လွယ်တာဗျ) ကျနော်တို့ဖတ်ခဲ့တာတွေကို အသုံးပြုပြီး project တစ်ခုလုပ်အောင်ဗျ။ နောက်ပြီး တူညီ project link တွေကိုလဲ လေ့လာဖို့ပေးထားပါမယ်ဗျ။

Challenge-45

Wait... ဆိုတဲ့ စာသား ကို ဖော်ပြပါ။ 3 စက္ကန့်နေတာနဲ့ ထိုစာသား ပျက်ရမည်။ ထို့နောက် welcome to myapp ဆိုတဲ့ စာသားကို output ထုတ်ပြပါ။

Challenge-46

Starting ဆိုတဲ့စာသား သည် ငါးစက္ကန့်ပေါ်ပြီးနောက်ကွယ်ပျောက်သွားရမည်ဖြစ်သည်။ယနေ့အချိန်ကို output ထုတ်ပြရမည်။ထို့နောက်မေးခွန်းဖြေ/မဖြေ မေးရမည်။ဖြေလျှင် ပထမဆုံးမေးခွန်းကျလာမည်။ မေးခွန်းဖြေပြီးတာနဲ့ မေးခွန်းပျောက်သွားရမည်ဖြစ်သည်။ အမှတ်တွေကို auto ပေါင်းပေးသွားရမည်။ မေးခွန်းသုံးခုခန့်အနည်းဆုံးပါရမည်ဖြစ်သည်။

Challenge Solution=45

Solution

```
import time
import os
print("Waiting.....")
time.sleep(3)
os.system("clear")
print("Welcome To Myapp")
(Note: ဖွင့်လှစ်ရန် run ဝါး example python name.py)
```

Solution=46

```
import time
import os
print("starting.....")
time.sleep(2)
os.system("clear")
```

```
ans=str(input("Do you Want to Exam?Y/N: "))
if ans=="Y" or ans=="y":
    os.system("clear")
    print(time.ctime())
    mark=0
    print("Questin one")
    b=str(input("Computer is A Electronic Device.Y/N: "))
    if b=="Y" or b=="Y":
        mark=mark+1
        print("Correct")
        os.system("clear")
    else:
        print("incorrect.")
        os.system("clear")

print("question two")
```

```
c=str(input("Python is a Programming.Y/N: "))
if c=="Y" or c=="y":
    mark=mark+1
    print("correct")
    os.system("clear")

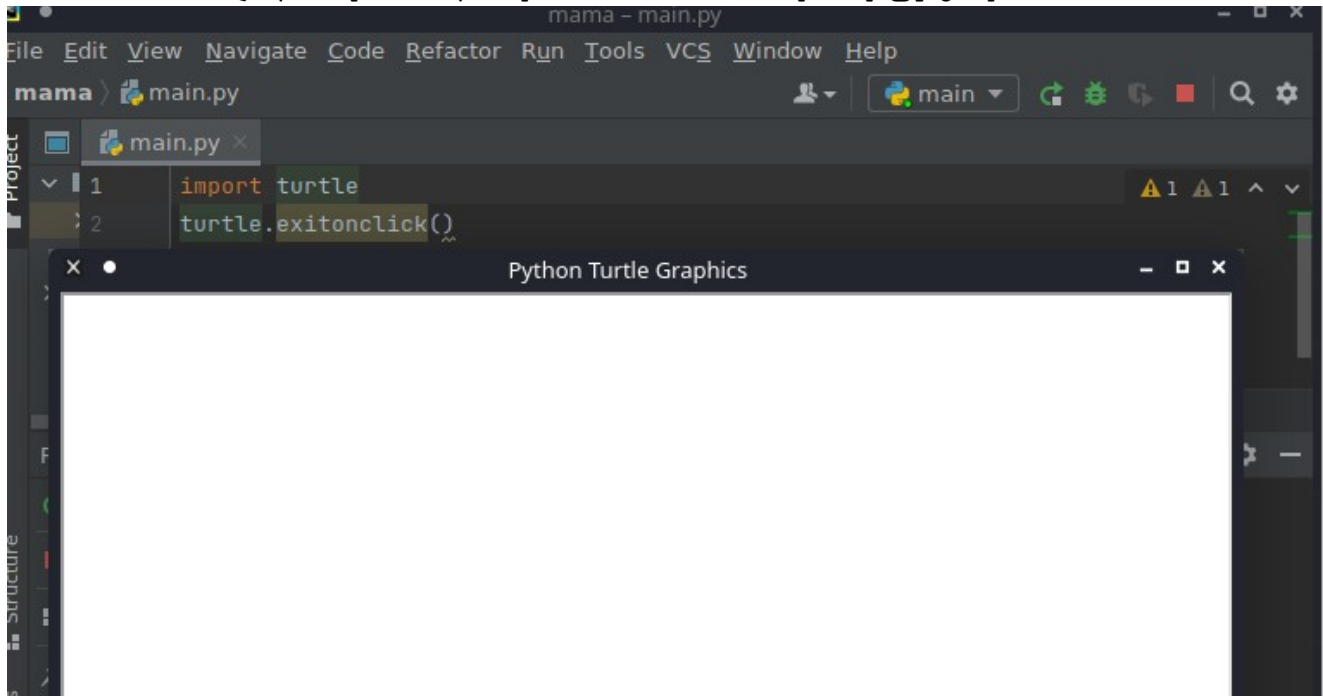
else:
    print("Incorrect")
    os.system("clear")

print("question three")
d=str(input("Mouse is a input device.Y/N: "))
if d=="Y" or d=="y":
    mark=mark+1
    print("correct")
    os.system("clear")
else:
    print("Incorrect")
    os.system("clear")
print("Your mark is",mark)
print("Exam Ended")
```

Chapter -14

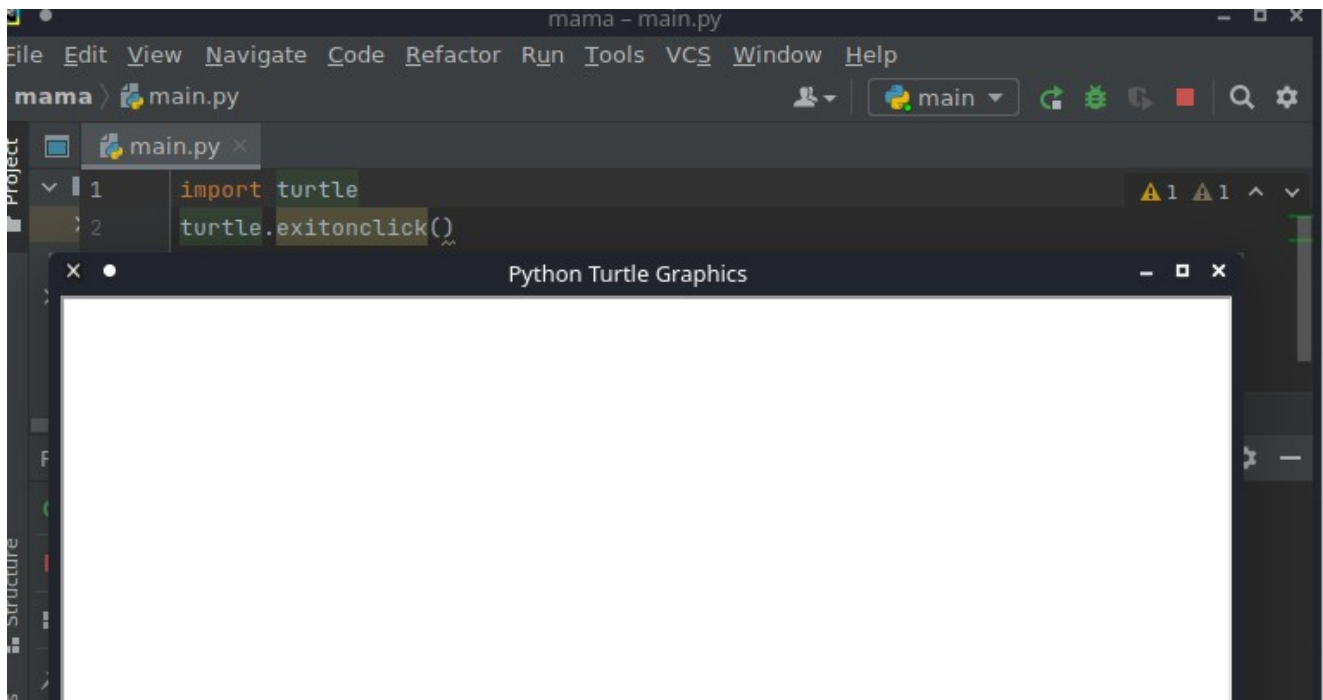
Turtle

ကျနော်ယခု သင်ခန်းစာမှာ turtle module ကိုအသုံးပြုသွားမှာပါ။



Description

import turtle ဆိုတာ turtle function တွေအသုံးပြုခွင့်တောင်းလိုက်တာဖြစ်ပါတယ်။ turtle.exitonclick() ဆိုတာ turtle garaphic interfaces ကို ဖန်တီးလိုက်တာဖြစ်ပါတယ်။ တကယ်လို့ turtle.exitonclick() function မပါလာခဲ့ရင် turtle က run တုန်းခနဲ GUI(gragphic user interface) ကိုပြသမှာဖြစ်ပါတယ်။ ဒါကတော့ ကိုယ်တိုင် run ကြည့်ရင် သိလာပါလိမ့်မယ်။



Turtle Function များ

`t=turtle.Turtle()` ဆိုတာ turtle function ကိုအရှည်မရေးချင်လို့ `t` ဆိုတဲ့ variable အဖြစ်သတ်မှတ်လိုက်တာဖြစ်ပါတယ်။

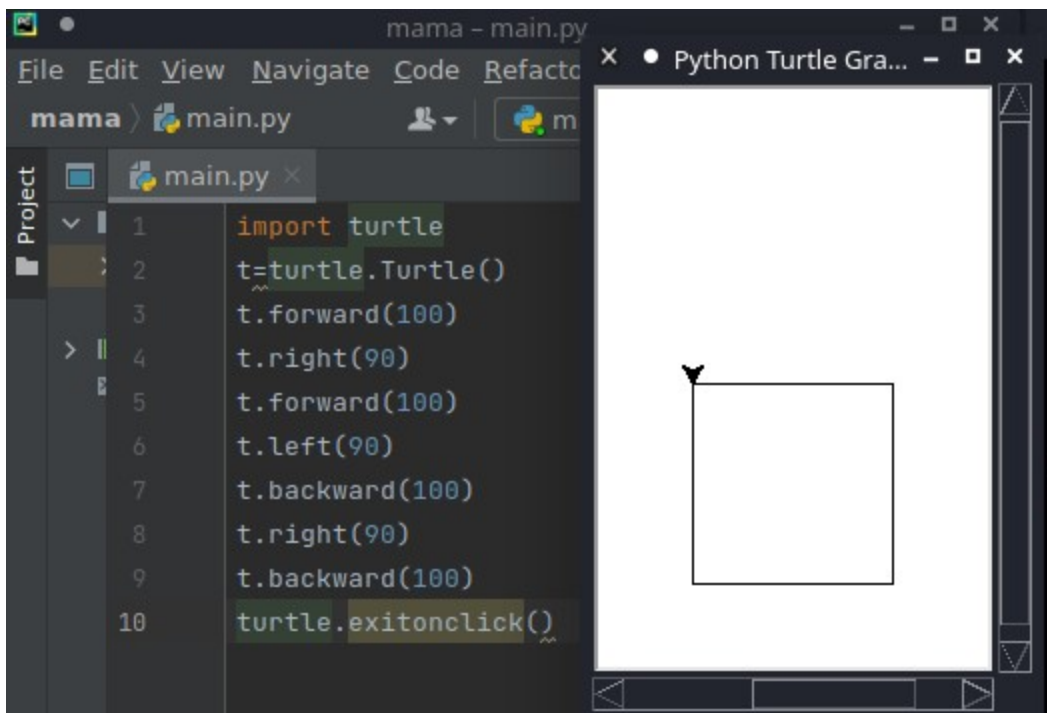
`t.forward(90)` ဆိုတာ ရှေ့သို့ 90 pixel အကွာရှိတဲ့ မျဉ်းကြောင်းတစ်ကြောင်းကိုဆွဲလိုက်ခြင်းဖြစ်သည်။(t.fd() အနေနဲ့လဲ အသုံးပြုနိုင်ပါတယ်။)

`t.backward(100)` ဆိုတာ နောက်သို့ 100 pixel အကွာရှိတဲ့ မျဉ်းကြောင်းတစ်ကြောင်းကိုရေးဆွဲလိုက်တာပါ။(t.bk() အနေနဲ့အသုံးပြုနိုင်ပါတယ်)

`t.right(90)` ဆိုတာ ညာဘတ်သို့ 90 စောင်းချင်းဖြစ်သည်။(t.rt() အနေနဲ့အသုံးပြုနိုင်ပါတယ်)

`t.left(90)` ဆိုတာ ဘယ်ဘတ်သို့ 90 degree စောင်းခြင်းဖြစ်သည်။(t.lt() အနေနဲ့လဲအသုံးပြုနိုင်ပါတယ်)

Example



ExampleDescription

- Line 1: turtle module ကို import ထုတ်လိုက်ပါတယ်။
- Line 2: t=turtle.Turtle() နဲ့ turtle function ကို t အသေးဖြစ်သတ်မှတ်လိုက်ပါတယ်။
- Line 3: t.forward(100) ဆိုတာ ရှေ့သို့ 100 pixels ရှည်တဲ့ မျဉ်းကိုရေးဆွဲလိုက်တာဖြစ်ပါတယ်။
- Line 4: t.right(90) ဆိုတာ ညာဘက်သို့ 90 degree စောင်းလိုက်တာဖြစ်ပါတယ်။
- Line 5: t.forward(100) ဆိုတာက ရှေ့သို့ 100 pixels ရှည်တဲ့မျဉ်းကိုဆွဲလိုက်တာပါ။
- Line 6: ညာဘက်သို့ 90 degree စောင်းလိုက်တာဖြစ်ပါတယ်။
- Line 7: အနောက်သို့ 100 pixels ရှည်တဲ့ မျဉ်းကိုရေးဆွဲလိုက်တာဖြစ်ပါတယ်။
- Line 8: ညာဘက်သို့ 90 degree စောင်းလိုက်တာ ဖြစ်ပါတယ်။
- Line 9: အနောက်သို့ 100 pixels ရှည်တဲ့ မျဉ်းကိုဆွဲလိုက်တာဖြစ်ပါတယ်။
- Line 10: ကတော့ program ကို အဆုံးသတ်လိုက်တာဖြစ်ပါတယ်။

Turtle Other Functions

Turtle နဲ့ပတ်သတ်ပြီး တစ်ခြားလုပ်ဆောင်ချက်တွေကို Theory အရဖော်ပြသွားပါ။ Theory နဲ့အတူ challenge တွေနဲ့အတူ တွဲဖတ်လေ့ကျင့်နိုင်ပါတယ်။

title("Title name") ကတော့ turtle နဲ့ နာမည် ကို change လိုက်ခြင်းဖြစ်သည်။

circle(60) ဆိုတာ 60 Degree ရှိတဲ့ စက်ဝိုင်းကိုရေးဆွဲလိုက်တာဖြစ်ပါတယ်။

dot(20) ဆိုတာ 20 Degree ရှိတဲ့ အချင်းကို ကာလာ ဖြည့်ပြီးသား စက်ဝိုင်းကိုရေးဆွဲလိုက်တာဖြစ်ပါတယ်။

bgcolor("green") ဆိုတာ နောက်ခံကာလာကို အရောင်းပြောင်းလိုက်တာဖြစ်ပါတယ်။

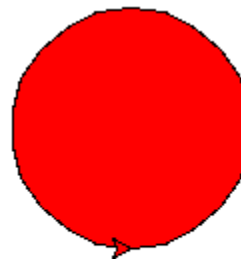
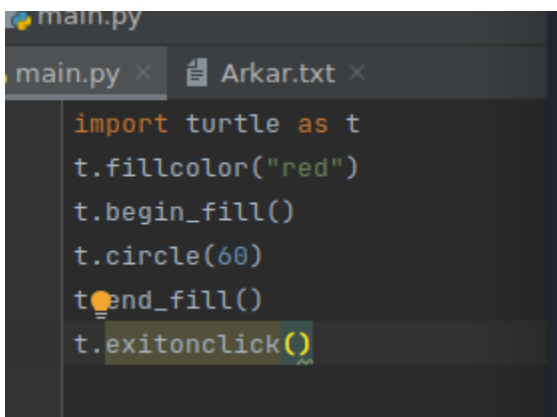
pensize(10) ဆိုတာ pen ရဲ့ အရွယ်အစားကို ပြောင်းလဲလိုက်တာဖြစ်ပါတယ်။
pencolor("green") ဆိုတာ pen ရဲ့ color ကိုပြောင်းလဲလိုက်တာဖြစ်ပါတယ်။
fillcolor("red") ဆိုတာ color ကိုပြောင်းလိုက်တာဖြစ်ပါတယ်။
t.begin_fill() ဆိုတာ color ကို စတင်ဖြည့်လိုက်တာဖြစ်ပါတယ်။
t.end_fill() ဆိုတာ color ချယ်ပီးပြီ ဖြစ်ပါတယ်။
clear() screen ကို clear လုပ်လိုက်တာဖြစ်ပါတယ်။



Description

Line 1: Turtle module ကို import လုပ်လိုက်ပါတယ်။
Line 2: ၎င်းကတော့ background color ပြောင်းလိုက်တာဖြစ်ပါတယ်။
Line 3: turtle ရဲ့ title name ကိုပြောင်းလိုက်တာဖြစ်ပါတယ်။
Line 4: pensize အရွယ်အစားကို ပြောင်းလဲလိုက်တာဖြစ်ပါတယ်။
Line 5: pen ရဲ့ color အရောင်ကိုပြောင်းလဲလိုက်တာဖြစ်ပါတယ်။
Line 6: အချင်း 70 ရှိတဲ့ စက်ဝိုင်းကိုရေးဆွဲလိုက်ချင်းဖြစ်ပါသည်။
Line 7: program ကို screen မှာ ဆက်လက်လုပ်ဆောင်နေရပ်ဖြစ်သည်။

The next example



Description

Line 1: turtle ကို import လုပ်ပြီး t အနေနဲ့ အသုံးပြုမည်ဖြစ်သည်။

Line 2: drawing လုပ်မဲ့ ကာလာအရောင်ကို အနီရောင်ပြောင်းပါ။

Line 3: ကာလာကို စတင်ဖြည့်ပါ(အရောင်ချယ်ပါ)

Line 4: 60 အချင်း ရှိတဲ့ စက်ဝိုင်း ကို ဆွဲပါ။

Line 5: ကာလာချယ်တာကိုး အဆုံးသတ်ပါ။

Line 6: သူက screen ကို exit လုပ်မှ ပိတ်ရန်အတွက်အသုံးပြုတာဖြစ်ပါတယ်။

Challenge- 47

Draw a triangle with green color.

Challenge-48

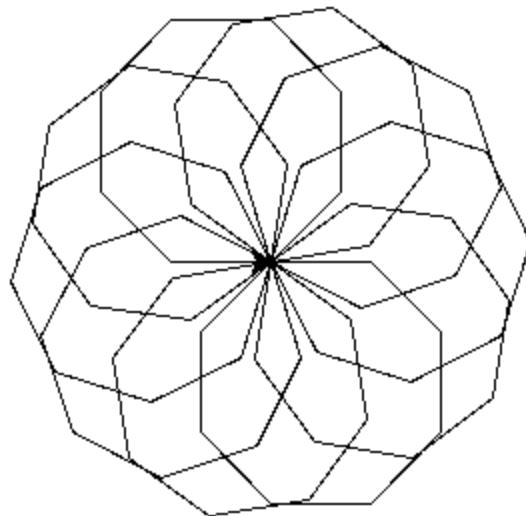
Draw a square with red color.

No-49

Draw a five-point star.

Challenge-50

No 4. ယခုပုံစံအတိုင်းရအောင်ဆွဲပေါ့။



Solution-47

```
import turtle as t
t.color("black","green")
t.begin_fill()
for i in range(0,3):
    t.forward(100)
    t.left(120)
t.end_fill()
t.exitonclick()
```

solution-48

```
import turtle as t
t.color("black","red")
t.begin_fill()
for i in range(0,4):
    t.forward(100)
    t.right(90)
t.end_fill()
t.exitonclick()
```

solution-49

```
import turtle as t
for i in range(0,5):
    t.forward(100)
    t.right(144)
t.exitonclick()
```

solution-50

```
import turtle
for x in range (0,10):
    for i in range(0,8):
        turtle.forward(50)
        turtle.right(45)
```



```
turtle.right(36)
turtle.exitonclick()
```

Chapter-15

Python3 with file

ကျနော်တို့ အပေါ်မှာ data တွေကို variable ထဲမှာယာယီသိုလှောင်ပြီးနောက် program ပြီးဆုံးသွားတာကို သိခဲ့ရပါသည်။ ယခု ကျနော်တို့ရဲ့ data တွေကို file တစ်ခုထဲမှာ save တော့မှာဖြစ်ပါတယ်။ File reading နဲ့ writing ပိုင်းကိုအရင်သွားရပါမယ်။ ယခုသင်ခန်းစာသည့် Computer Data Science ပိုင်းကိုစတင်လေ့လာရမည့်အပိုင်းဖြစ်သည်။ open () function သည် file အတူတူအလုပ်လုပ်သည်။ open function မှာ file name နဲ့ mode ဆိုပြီးအပိုင်းနှစ်ပိုင်းရှိတယ်။

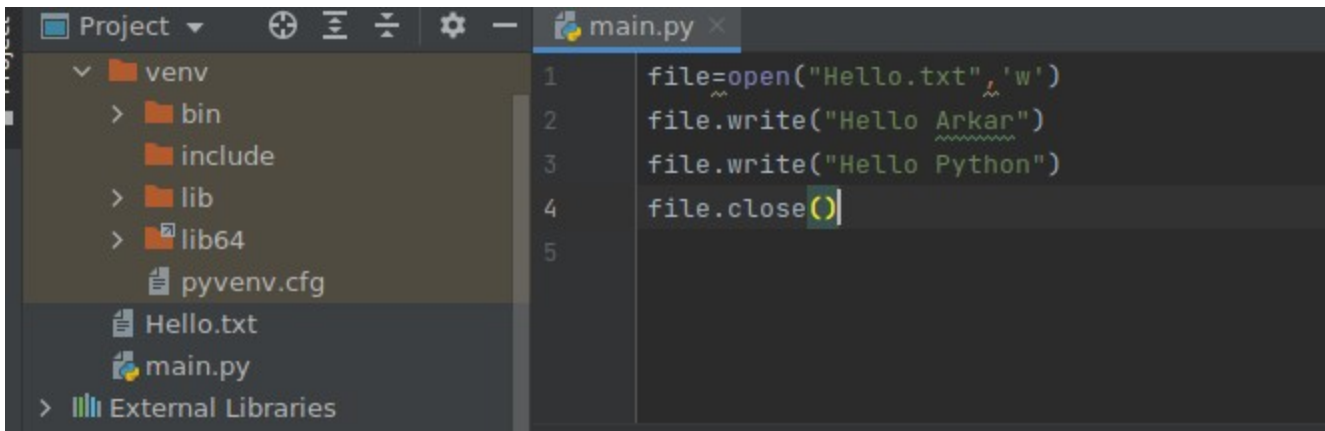
Example:

```
open("a.txt",'r')
```

a.txt က file ဖြစ်ပြီး r က mode ဖြစ်သည်။

File ပတ်သတ်ပြီး သိထားရမှန်အချက်သုံးချက်

symbol	Example	Description
w	open("a.txt",'w')	a.txt file ကို ဖန်တီးလိုက်ခြင်းဖြစ်သည်။
r	open("a.txt",'r')	a.txt file ကိုဖတ်ခြင်းဖြစ်သည်။
a	open("a.txt","a")	Data တွေကို အသစ်ထည့်တဲ့နေရာမှာအသုံးပြုပါတယ်။



```
1 file=open("Hello.txt",'w')
2 file.write("Hello Arkar")
3 file.write("Hello Python")
4 file.close()
```

Description

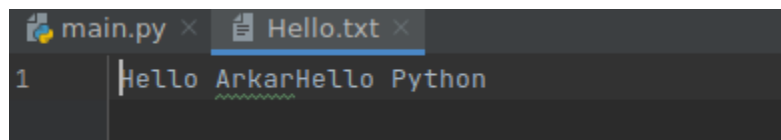
Line one: က Hello.txt ဆိုတဲ့ file ကို တည်ဆောက်လိုက်တာဖြစ်ပါတယ်။

(Note: တကယ်လို့ file name တူရှိခဲ့ရင် ထိုဖိုင်ကိုဖျက်ပစ်ပြီး အသစ်လုပ်မှာဖြစ်ပါတယ်။)

Line two: write() function က Hello Arkar ဆိုတဲ့ စာသားကို Hello.txt file ထဲကိုရေးထည့်လိုက်တာဖြစ်ပါတယ်။

Line three: က Hello Python ဆိုတဲ့ စာသားကို Hello.txt file ထဲကို ထည့်ရေးလိုက်တာဖြစ်ပါတယ်။

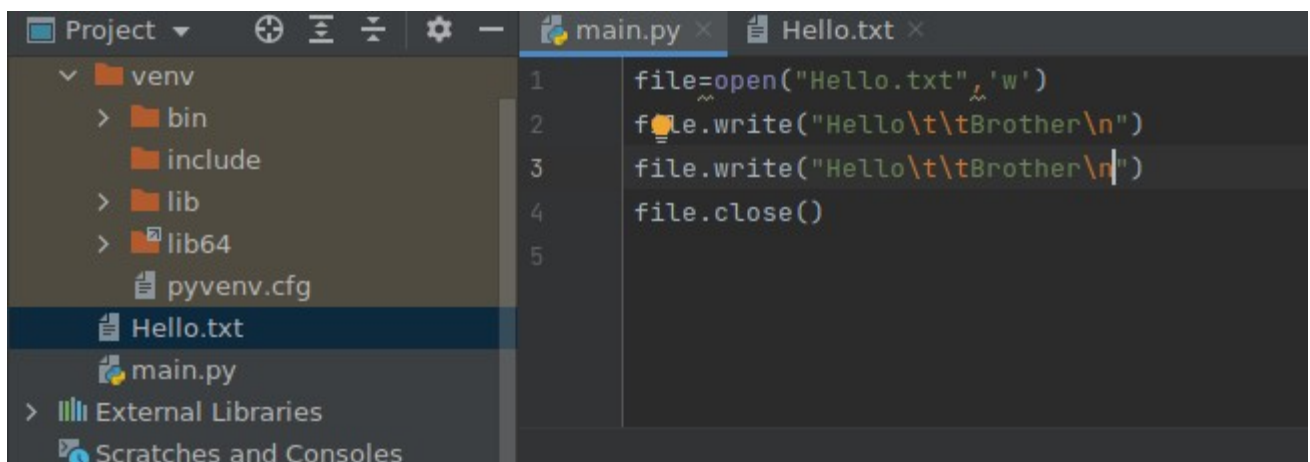
Line four : ကတော့ လုပ်ဆောင်ချက်ပီးစီး၍ ဖိုင်ကိုပိတ်လိုက်တာဖြစ်ပါတယ်။



```
1 Hello ArkarHello Python
```

Hello.txt ဖိုင်ကို ကြည့်လိုက်တဲ့အခါ အထက်ပါအတိုင်းမြင်ရမှာဖြစ်ပါတယ်။ဒီနေရာတွင် ကျနော်တို့ escape characters သည် များစွာအသုံးဝင်သွားပါသည်။

Example



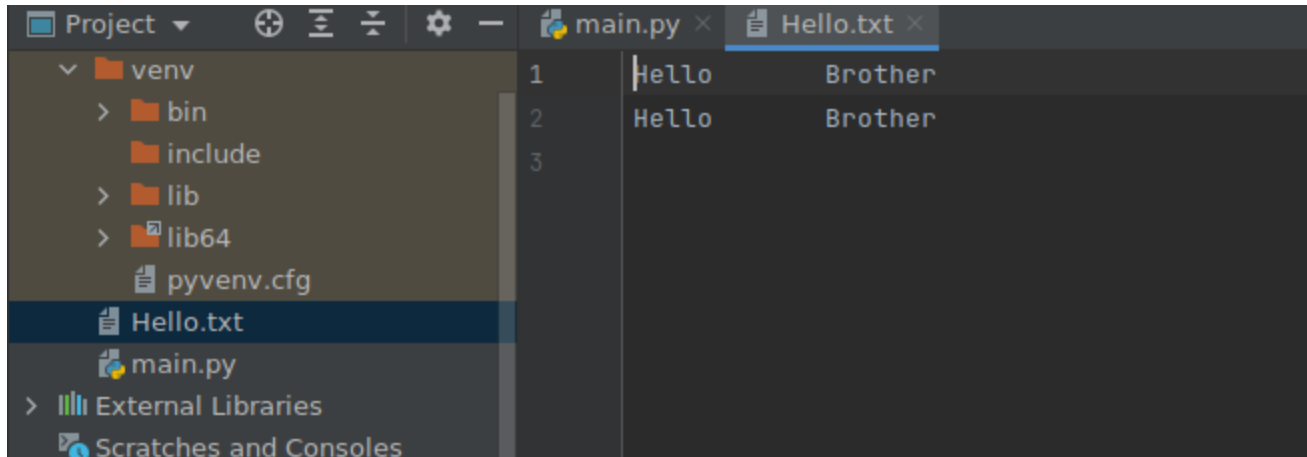
```
1 file=open("Hello.txt",'w')
2 file.write("Hello\t\tBrother\n")
3 file.write("Hello\t\tBrother\n")
4 file.close()
```

အခြေခံနားလည်ရင်ရပါပီ ဒါပေမဲ့နည်းနည်းရှင်းပြလိုက်မယ်။

Line 2: Hello နဲ့ Brother ကြားက tab နှစ်ခုခန့်စာ ခြားထားပါတယ်။ထို့နောက်လာမဲ့စာသားကို အောက်တစ်ကြောင်းမှာသာဖော်ပြရန်ဖြစ်သည်။

Line 3: Line 2 ကိုပြန်ဖတ်ပါ။

Output ကိုကြည့်ရင် နားလည်သွားမှာပါ။

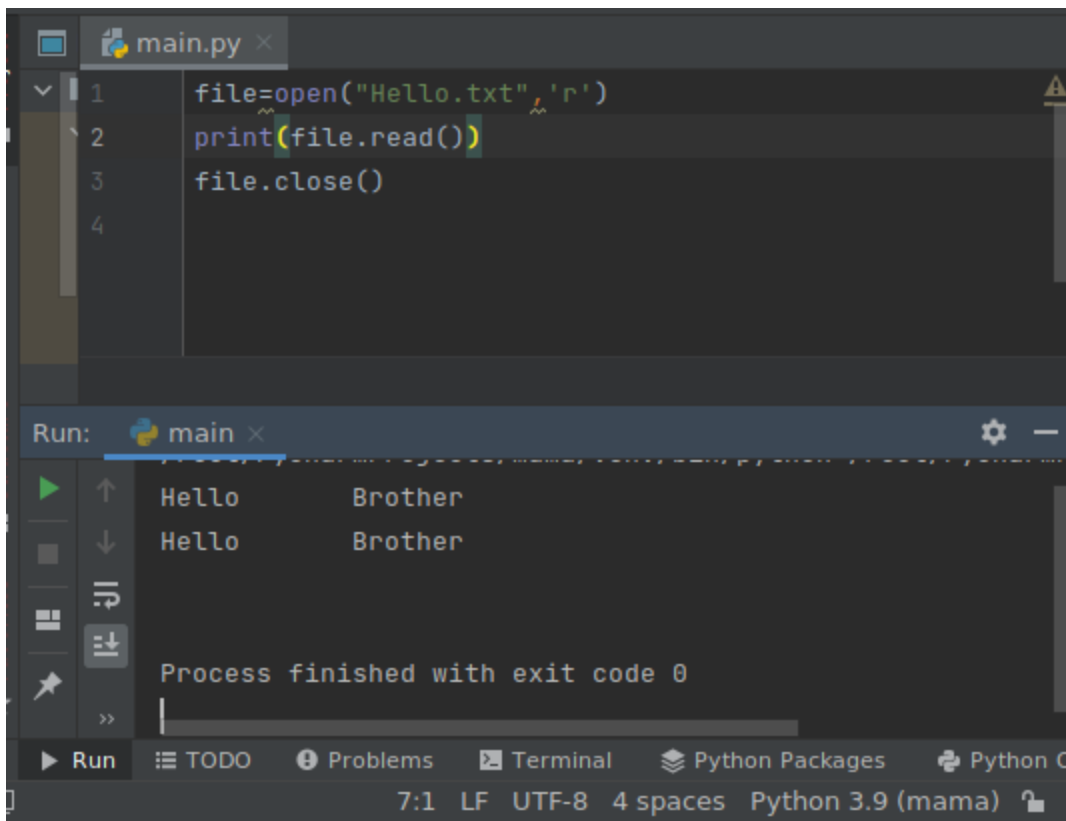
A screenshot of a code editor interface. On the left, a 'Project' sidebar shows a file tree with folders like 'venv', 'bin', 'include', 'lib', 'lib64' and files like 'pyvenv.cfg', 'Hello.txt', and 'main.py'. The 'Hello.txt' file is selected. The main editor area shows the content of 'Hello.txt' with two lines of text: 'Hello Brother' on line 1 and 'Hello Brother' on line 2. The text is in a monospaced font, and there is a tab character between 'Hello' and 'Brother' on each line.

နားလည်မယ်လို့မျှော်လင့်ပါတယ်။

File Writing ပိုင်းအခြေခံသဘောဖြစ်နားလည်သွားမယ်လို့မျှော်လင့်ပါတယ်။ယခု file reading ပိုင်းကို ဆက်သွားအောင်ဗျ။

File reading

ကျနော်တို့ file ထဲက data တွေကို ဖတ်ဖို့အတွက်ဆို read() function ကို အသုံးပြုပါတယ်။ output ပါထုတ်စေချင်တဲ့ဆိုရင် print() function ပါတွဲသုံးရပါတယ်။



```
main.py x
1 file=open("Hello.txt", 'r')
2 print(file.read())
3 file.close()
4

Run: main x
Hello Brother
Hello Brother

Process finished with exit code 0

Run TODO Problems Terminal Python Packages Python C
7:1 LF UTF-8 4 spaces Python 3.9 (mama)
```

Description

ကျနော်တို့ Hello.txt အနောက်က w နေရာမှာ r ပြောင်းသွားတာကိုးမြင်ရမှာပါ။ ဒါက write မဟုတ်ဘဲ read မယ်လို့ပြောတာပါ။ print(file.read()) ဆိုတာက Hello.txt file ထဲက data တွေကို output ထုတ်ပြခိုင်းတာဖြစ်ပါတယ်။ file.close() ဆိုတာက ကျနော်တို့ဖတ်ပြီးတော့ file ကိုပိတ်လိုက်တာဖြစ်ပါတယ်။

File Appending

append() function ကို data အသစ်ထည့်တဲ့နေရာမှာ ကျနော်တို့အသုံးပြုပါတယ်။



```
main.py x Hello.txt x
1 file=open("Hello.txt", 'a')
2 file.write("Hello Programmer")
3 file.close()
4
```

Description

Line 1: mode မှာ r ကနေ a ပြောင်းသွားတာတွေ့ပါလိမ့်မယ်။ ဒါက ကျနော်တို့ data အသစ်တွေကို ထည့်တဲ့နေရာမှာအသုံးပြုပါတယ်။

Line 2: Hello Programmer ဆိုတဲ့ စာသားကို ရေးထည့်လိုက်ခြင်းဖြစ်ပါတယ်။

Line 3: File ကို ပိတ်လိုက်တာဖြစ်ပါတယ်။

(Note: read function နဲ့ ကိုယ်ဘာသာပြန်ကြည့်ကြည့်ပေါ့ဗျာ။)

Challenge No 51

လူလေးယောက်အမည် ကို text file တစ်ခုထဲသို့ထည့်ပါ။

Challenge No 52.

text file တစ်ခုထဲ က data တွေ Output ထုတ်ပြပါ။

Challenge No 53.

text file ထဲသို့ input function သုံးပြီး name အသစ်ထဲပါ။

Challenge No 54.

text file ထဲက အမည်တွေကို ၃ စက္ကန့်ခြားတစ်ခါစီဖော်ပြပါ။

Challenge No.55

menu တစ်ခုကိုဖန်တီးပါ။ Menu မှာ file အသစ် ကိုလဲ ဖန်တီးလို့ရတယ်။ file ထဲက data ကိုလဲ ဖတ်လို့ရတယ်။ file ထဲကို လဲ data အသစ်တွေထည့်လို့ရမယ်။

(ယခု challenge သည် python with excel အခန်းနဲ့ ဆက်စပ်မှုရှိပါသည်။)

Solution 51

```
a=open("name.txt","w")
a.write("Htet Arkar Kyaw\n")
a.write("Aung Naing Oo\n")
a.write("Thant Zin Aung\n")
a.write("Aye Aye Aung")
a.close()
```

Solution 52

```
a=open("name.txt","r")
```

```
print(a.read())
```

Solution 53

```
a=open("name.txt","a")
name=str(input("Enter Name: "))
a.write("\n"+name)
a.close()
```

solution-54

```
import time
a=open("name.txt","r")
for i in a:
    print(i)
    time.sleep(3)
```

solution-55

```
def menu():
    print("""
        Main Menu
        (1) Create File
        (2) Display The File
        (3) Add New items to The File """)
    try:
        a=int(input("Enter Number:"))
        if a==1:
            cf()
        elif a==2:
            df()
        elif a==3:
            af()
        else:
            print("This number does not exit")
            menu()
```

```
except ValueError:  
    print("Enter Only Number")  
    menu()
```

```
def cf():  
    fname=str(input("Enter File Name: "))  
    file=open(fname,'w')  
    file.write("Name\t")  
    file.close()
```

```
def df():  
    fname=str(input("Enter File Name: "))  
    file=open(fname,'r')  
    print(file.read())
```

```
def af():  
    fname=str(input("Enter File Name: "))  
    file=open(fname,'a')  
    name=str(input("Enter Your Name: "))  
    file.write("\n"+name)  
    file.close()
```

Chapter-16

Array, Numpy and matplotlib

တခြားသော programming language တွေမှာ Array က built in ပါပြီးသားဆိုပေမဲ့ python မှာ ဆိုရင် သူ့ကိုခေါ်သုံးရပါတယ်။ list မှာဆိုရင် ကျနော်တို့ strings and integers စသည့်မတူညီတဲ့ data type တွေကို list အနေနဲ့ data တွေကိုသိုလှောင်လိုက်တာကို သိရှိပြီးသားဖြစ်မှာပါ။ Array လည်း ထိုနည်းတူပဲ။ ဒါပေမဲ့ array က number တွေပဲသိုလှောင်လို့ရပါတယ်။ ထိုအပြင် သိုလှောင်မှု numeric data type အမျိုးအစားကိုဖော်ပြပေးရပါတယ်။ array က list နဲ့အတူတူ အလုပ်လုပ်ပါတယ်။ အောက်မှာရှင်းပြထားပါတယ်။

Symbol	Name	Description
“i”	integers	မည့်သည် ကိန်းပြည့်မဆို သိုလှောင်နိုင်သည်။
“l”	long	ကိန်းရှည်တွေကိုသိုလှောင်ဖို့ အတွက် (123454903305000)
“f”	float	ဒသမကိန်းတွေကိုသိုလှောင်ဖို့ အတွက် (1.3, 2.4)

```
>>> import array as ar
>>> a=ar.array('i',[1,2,3,4])
>>> print(a)
array('i', [1, 2, 3, 4])
>>>
```

Description

- Line 1: array ကို import လုပ်ပြီး ar အနေနဲ့ အသုံးပြုမှာဖြစ်ပါတယ်။
- Line 2: a ဆိုတဲ့ variable ထဲ ကို array သိုလှောင်လိုက်သည်။ array ထဲမှာ ပါတဲ့ l ကတော့ integer numeric data type ကိုပဲ list ထဲ သိုလှောင်ရမည်ဖြစ်ကြောင်းသတ်မှတ်လိုက်ခြင်းဖြစ်သည်။ နောက်က list ကတော့ integers တွေထဲထားတဲ့ list တစ်ခုပင်ဖြစ်သည်။
- Line 3: array ကိုပြန်ပြီး output ထုတ်တာဖြစ်ပါတယ်။

Next Example


```
4
>>> b=np.array('f',[1.2,3.4,4.4,6.5])
>>> print(b)
array('f', [1.20000000476837158, 3.40000000953674316, 4.4000000095367432, 6.5])
>>>
```

Description

အပေါ်မှာ array ကို import လုပ်ပြီးခဲ့ပြီဖြစ်ပါသည်။နောက်က round brackets ကြားထဲမှာ သိုလှောင်မဲ့ data type က float ဖြစ်ကြောင်းနှင့် list ထဲမှာ float data type တွေ သိုလှောင်ထားခြင်း ဖြစ်သည်။ထို့နောက် array ကို ပြန်၍ output ထုတ်လုပ်သည်။

Long Array

```
>>> c=np.array('l',[11111111111111111,2222222222222222,3333333333333333])
>>> print(c)
array('l', [11111111111111111, 2222222222222222, 3333333333333333])
>>>
```

Description

သိုလှောင်ရမဲ့ data type က long ဖြစ်ကြောင်းနဲ့ နောက်က list ထဲမှာ long data types တွင်သိုလှောင်၍ နောက်တစ်ကြောင်းတွင် output ထုတ်ပြသည်။
(Note: list ထဲ က data တွေကို ပြုပြင်ပြောင်းလဲနိုင်တယ်နော် မမှတ်မိရင် ပြန်ဖတ်)

Numerical Python

Numerical Python ကို numpy အဖြစ်အတိုကောက်သတ်မှတ်သည်။ Numpy သည် mathematical and scientific ဆိုင်ရာ module တစ်ခုဖြစ်သည်။ Numpy module သည် ndarray (N-dimension array) object ပေါ်တွင်အခြေခံထားသည်။

Example

```
>>> import numpy
>>> a=np.array([1,2,3,4])
>>> print(a)
[1 2 3 4]
>>> type(a)
<class 'numpy.ndarray'>
>>>
```

Description

Line 1: numpy ကို အသုံးပြုမည်ဖြစ်ကြောင်းကြေညာလိုက်တာဖြစ်ပါတယ်။
Line 2: a ဆိုတာ variable နဲ့အတူ numpy array ကိုတည်ဆောက်လိုက်တာဖြစ်ပါတယ်။
Line 3: type function နဲ့ array အမျိုးအစားကိုစစ်ဆေးတာဖြစ်ပါတယ်။

2D-Array

```
>>> import numpy as np
>>> arr=np.array([[1,2,3],[4,5,6]])
>>> print(arr)
[[1 2 3]
 [4 5 6]]
>>>
```

Description

Line 1 ==> numpy ကို import လုပ်ပြီး np အနေနဲ့ အသုံးပြုမည်ဖြစ်ကြောင်းကြေညာခြင်းဖြစ်သည်။
Line 2==> arr ဆိုတဲ့ variable နဲ့ အတူ list ၂ ခုနဲ့ array တစ်ခုတည်ဆောက်လိုက်ပါတယ်။
(List နှစ်ခုပါဝင်သောကြောင့် 2D-Array လို့ခေါ်ပါတယ်)
Line 3==> arr ကို output ထုတ်လိုက်တာဖြစ်ပါတယ်။

3D-Array

```
>>> import numpy as np
>>> arr=np.array([[[1,2],[3,4],[5,6]]])
>>> print(arr)
[[[1 2]
 [3 4]
 [5 6]]]
>>>
```

Description

Line 1==> numpy ကို np အဖြစ်အသုံးမည့်ဖြစ်ကြောင်း ကြေညာခြင်း ဖြစ်သည်။
Line 2==> List ၃ ခုနဲ့ အတူ array တစ်ခုတည်ဆောက်လိုက်တာဖြစ်ပါတယ်။
Line 3==> arr ကို output ထုတ်ကြည့်တာဖြစ်ပါသည်။

Numpy Functions

Numpy ရဲ့ Functions တစ်ချို့ကိုအသုံးပြုပြသွားမှာဖြစ်ပါတယ်။ကျန်တာကိုယ်ဘာသာဆက်လေ့လာပါဗျ။

No 1. ndim

ndim function က ndarray ထဲမှာရှိတဲ့ list အရေအတွက်ကို output ထုတ်တဲ့အခါမှာအသုံးပြုပါသည်။

```
>>> import numpy as np
>>> arr=np.array([[1,2,3],[4,5,6],[7,8,9],[2,3,4]])
>>> print(arr)
[[1 2 3]
 [4 5 6]
 [7 8 9]
 [2 3 4]]
>>> print(arr.ndim)
3
>>>
```

Description

Line 1==> numpy ကို np အနေနဲ့ အသုံးပြုမည့်ဖြစ်ကြောင်း ကြေညာခြင်းသည်။

Line 2==> list ၄ ခုရှိတဲ့ 4D array တစ်ခုတည်ဆောက်လိုက်သည်။

Line 3==> output ထုတ်တာဖြစ်ပါတယ်။

Line 4==> array ထဲရှိ list အရေအတွက် ကို output ထုတ်တာဖြစ်ပါတယ်။

(index number 0 ကနေ စရေတွက်ရတာဖြစ်သောကြောင့် 3 မှာ ဆုံးသည်)

(Note: ndim နဲ့ပတ်သတ်၍ တခြားအသုံးဝင်နည်းတွေရှိပါသေးတယ် ကိုယ်ဘာသာဆက်လေ့လာပါ။)

Data Types in numpy

Numpy မှာလဲ Data Types တွေရှိပါတယ်။ array ထဲမှာရှိတဲ့ data or items တွေ က integer သို့မဟုတ် float စသဖြင့်သတ်မှတ်ပေးတာပေါ့။ dtype() function ကို အသုံးပြုသည်။

Example

```
>>> import numpy as np
>>> arr1=np.array(['a','b','c'])
>>> arr2=np.array([1.2,3.4,4.5])
>>> print(arr1.dtype)
<U1
>>> print(arr2.dtype)
float64
>>>
```

Description

Line 1==> numpy ကို np အဖြစ် အသုံးမည်ဖြစ်သည်။

Line 2==> arr1 ဆို variable နဲ့ အတူ array တစ်ခုကို တည်ဆောက်သည်။

Line 3==> arr2 ဆိုတဲ့ variable နဲ့အတူ array တစ်ခုကို တည်ဆောက်သည်။

Line 4==> dtype function နဲ့ arr1 ထဲမှာရှိတဲ့ data type ကို စစ်ဆေးတာဖြစ်ပါတယ်။

Line 5==> dtype နဲ့ arr2 ထဲမှာရှိတဲ့ data type ကို စစ်ဆေးတာဖြစ်ပါတယ်။

Numpy Data Types

Character	Description
.i	integer
.b	boolean
.u	unsigned integer
.f	float
.s	string

(Note ကျန်တာ ကိုယ်ဘာဆက်လေ့လာပါဗျ)

The Next Example

```
>>> import numpy as np
>>> arr1=np.array([1,2,3,4])
>>> arr2=np.array(["a","d","e"])
>>> print(arr1.dtype)
int64
>>> print(arr2.dtype)
<U1
>>>
```

Description

Line 1==> numpy ကို np အဖြစ်အသုံးပြုမည်ဖြစ်သည်။

Line 2==> arr1 ဆိုတဲ့ variable နဲ့ arr တစ်ခုကို တည်ဆောက်လိုက်သည်။

Line 3==> arr2 ဆိုတဲ့ variable နဲ့ arr တစ်ခုကို တည်ဆောက်သည်။

Line 4 and 5 ==> data type အမျိုးအစားကို စစ်ဆေးပြီး output ထုတ်ပြသည်။

Dtype

```
type help , copyright , credits or license for more informa
>>> import numpy as np
>>> arr=np.array([1,2,3,4],dtype='S')
>>> print(arr)
[b'1' b'2' b'3' b'4']
>>> print(arr.dtype)
|S1
>>>
```

Description

Line 1==> numpy ကို np အဖြစ်အသုံးပြုမည်ဖြစ်သည်။

Line 2==> arr variable နဲ့အတူ array တစ်ခုတည်ဆောက်သည်။dtype function နဲ့ items တွေကို string array အဖြစ်သာသတ်မှတ်ထားသည်။

Line 3==> arr ကို output ထုတ်ကြည့်ချင်းဖြစ်ပါသည်။

Line 4==> array ရဲ့ datatype ကို output ထုတ်ကြည့်တာဖြစ်ပါတယ်။

Numpy Array Shape

Array Shape ဆိုတာ array ထဲမှာရှိတဲ့ list အရေအတွက် နဲ့ item အရေအတွက်ကို ဖော်ပြတာဖြစ်ပါတယ်။

Example

```
>>> import numpy as np
>>> arr=np.array([[1,2,3,4,5],[2,3,4,5,6],[3,4,5,6,7]])
>>> print(arr.shape)
(3, 5)
>>>
```

Description

Line 1==> numpy ကို np အဖြစ်အသုံးပြုမည်ဖြစ်သည်။

Line 2==> items 5 ခုစီပါတဲ့ list 3 ခု နဲ့ array တစ်ခုကို တည်ဆောက်လိုက်တာဖြစ်ပါတယ်။

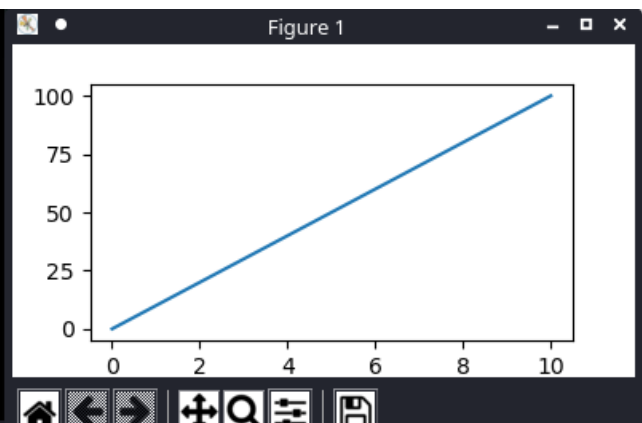
Line 3==> array ထဲရှိ list နဲ့ item အရေအတွက်ကို output ထုတ်ပြတာဖြစ်ပါတယ်။

(Note==> တစ်ခြား function တွေအများကြီး ရှိပါသေးတယ်။ဆက်လေ့လာပါဗျ)

Matplotlib

Graph ဆွဲဖို့အတွက် တီထွင်ထားတဲ့ module တစ်ခုဖြစ်သည်။သူက numpy နဲ့ တွဲဖတ်ပြီး အလုပ်လုပ်ပါတယ်။ Numpy မရရင် matplotlib လဲမရနိုင်ပါဖူး။ ဒီနောက်မှာ ကျနော် pandas module ကိုတမင်ချန်ခဲ့ပါတယ်။

```
AttributeError: module 'matplotlib' has no attribute 'pyplot'
>>> import matplotlib.pyplot as plt
>>> import numpy as np
>>> xpoints=np.array([0,10])
>>> ypoints=np.array([0,100])
>>> plt.plot(xpoints,ypoints)
[<matplotlib.lines.Line2D object at 0x7f8dfae12be0>]
>>> plt.show()
```



Description

Line 1: Matplotlib ကို import လုပ်ပြီး plt အဖြစ်အသုံးမည် ဖြစ်သည်။

Line 2: Numpy ကို np အဖြစ် အသုံးပြုမည်ဖြစ်သည်။

Line 3: xpoints variable တစ်ခု တည်ဆောက်လိုက်သည်။(သူက တော့ x-axis အတွက်)

Line 4: ypoints variable တစ်ခု တည်ဆောက်လိုက်သည်။(သူက တော့ y-axis အတွက်)

Line 5: graph တစ်ခုကိုရေးဆွဲလိုက်တာဖြစ်ပါသည်။

Line 6: graph ကို output ထုတ်ရန်ဖြစ်သည်။

Plot() and Graph

Graph မှာ x ဝင်ရိုးနဲ့ Y ဝင်ရိုး ရှိတဲ့ ဆိုတာ ကျနော်တို့ ငယ်ငယ်တည်းသိပြီးသားဖြစ်မှာပါ။ x-points မှာ ဆို အကြီးဆုံး number မှာ 10 ဖြစ်သောကြောင့် x-axis သည် 10 မှာဆုံးသည်။ ထို့အတူ Y-points မှာ အကြီးဆုံး number မှာ 100 ဖြစ်သောကြောင့် y-axis သည် 100 မှာဆုံးသည်။ arrays ထဲ က number တွေက graph မှာ point ချတာဖြစ်ပါတယ်။ xpints 0 နဲ့ y points 0 မှာ points တစ်ခုချပြီး xpoints 10 နဲ့ ypoints 100 မှာ point တစ်ခုချပြီး သူတို့ကို မျဉ်း တစ်ခုအဖြစ် ဆွဲလိုက်သည်။

plot() function က Graph တစ်ခု ကို ရေးဆွဲလိုက်တာဖြစ်ပါတယ်။

show() function က တော့ ရေးဆွဲလိုက်တဲ့ Graph ကို output ထုတ်ပြတာဖြစ်ပါတယ်။

The Next Example

```
>>> plt.show()
>>> import matplotlib.pyplot as plt
>>> import numpy as np
>>> x=np.array([1,12])
>>> y=np.array([1,31])
>>> plt.plot(x,y,linestyle='')
[<matplotlib.lines.Line2D object at 0x7f48f461da90>]
>>> plt.title("Day and Month")
Text(0.5, 1.0, 'Day and Month')
>>> plt.xlabel("Month")
Text(0.5, 0, 'Month')
>>> plt.ylabel("Day")
Text(0, 0.5, 'Day')
>>> plt.show()
```

Description

Line 1: matplotlib ကို plt အဖြစ် အသုံးမည်။

Line 2: numpy ကို np အဖြစ် အသုံးပြုမည်။

Line 3: x-axis အတွက် point သတ်မှတ်ထားတဲ့ array တစ်ခုဖြစ်သည်။

Line 4: y-axis အတွက် point သတ်မှတ်ထားတဲ့ array တစ်ခုဖြစ်သည်။

Line 5: x,y ပေါင်းပြီး Graph တစ်ခုဆွဲလိုက်သည်။သို့သော် linestyle ကို none(ဖော်မပြရန်) အဖြစ် သတ်မှတ်ထားသည်။

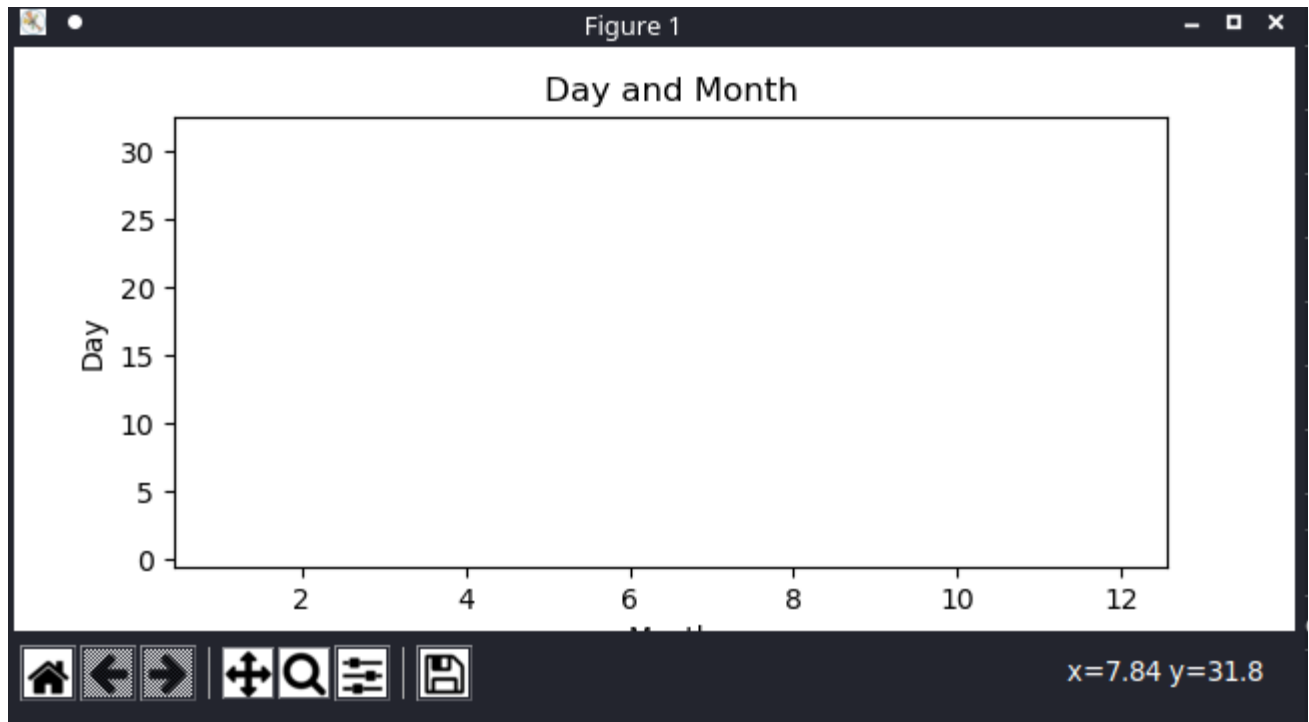
Line 6: Graph ရဲ့ Title ကို အမည် ပေးတာ ဖြစ်ပါတယ်။

Line 7: x-axis ရဲ့ label ကို အမည် ပေးတာ ဖြစ်ပါတယ်။

Line 8: y-axis ရဲ့ label ကို အမည် ပေးတာ ဖြစ်ပါတယ်။

Line 9: ဆွဲထားတဲ့ Graph ကို output ထုတ်ပြတာဖြစ်ပါတယ်။

Output



title() က ခေါင်းစဉ်အမည်ပေးတာဖြစ်ပါတယ်။

xlabel() and ylabel() ကတော့ x-axis နဲ့ y-axis ကို အမည်ပေးတာဖြစ်ပါတယ်။

Line style

ကျနော်တို့ line style ကို တခြားပုံစံမျိုးနဲ့ ဖော်ပြလို့ရပါတယ် အောက်မှာ တချို့ကို ဖော်ပြထားပါတယ်။

Symbol	Description
"o"	Circle
"*" (Shift+8)	Star
"_"	Dash
"."	Dotted

The Next Example

```
>>> import matplotlib.pyplot as plt
>>> import numpy as np
>>> x=np.array([1700,1800,1900,2000,2010,2020])
>>> y=np.array([603,1000,1600,6000,6900,7700])
>>> plt.plot(x,y,linestyle="-")
[<matplotlib.lines.Line2D object at 0x7f48f451ae50>]
>>> plt.title("The World Population")
Text(0.5, 1.0, 'The World Population')
>>> plt.xlabel("Years")
Text(0.5, 0, 'Years')
>>> plt.ylabel("Gobal Population")
Text(0, 0.5, 'Gobal Population')
>>> plt.show()
```

Description

Line 1: matplotlib ကို plt အဖြစ်အသုံးပြုမည်။

Line 2: numpy ကို np အဖြစ်အသုံးပြုမည်။

Line 3: x-axis တစ်ခုကို array နဲ့အတူ တည်ဆောက်ပြီး point သတ်မှတ်ထားသည်။

Line 4: y-axis တစ်ခုကို array နဲ့အတူ တည်ဆောက်ပြီး point သတ်မှတ်ထားသည်။

Line 5: x-axis နဲ့ y-axis နဲ့အတူ graph တစ်ခုရေးဆွဲပြီး point မျဉ်းကြောင်း(line) ကို dashed ဖြင့် ဖော်ပြရန် သတ်မှတ်လိုက်သည်။

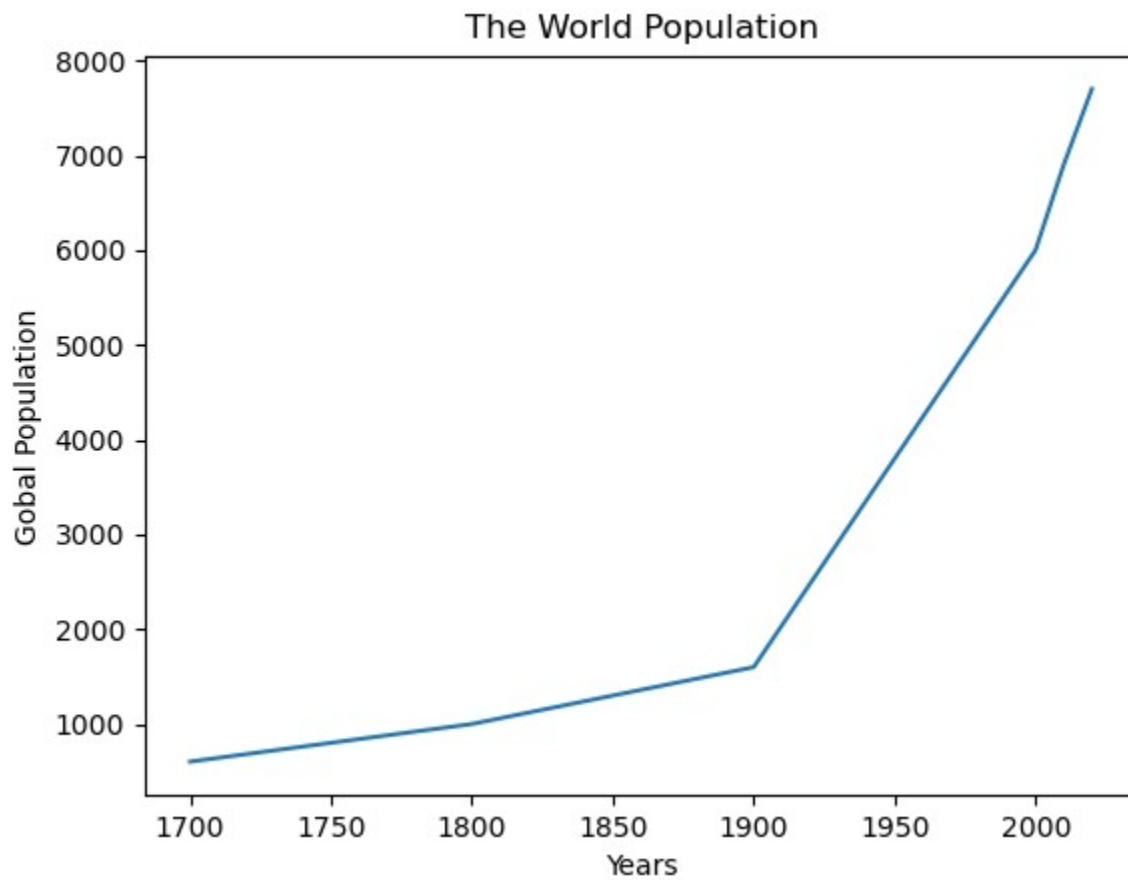
Line 6: Graph ရဲ့ title ကို အမည်ပေးခြင်းဖြစ်သည်။

Line 7: x-axis ကို အမည်ပေးခြင်းဖြစ်သည်။

Line 8: y-axis ကို အမည်ပေးခြင်းဖြစ်သည်။

Line 9: ရေးဆွဲထားသော Graph ကို output ထုတ်ပြခြင်းဖြစ်သည်။

Output-



The Next Example

```
>>> import matplotlib.pyplot as plt
>>> import numpy as np
>>> x=np.array([2017,2018,2019,2020])
>>> y=np.array([300,350,500,700])
>>> plt.plot(x,y,linestyle="dashed",color="green")
[<matplotlib.lines.Line2D object at 0x7f3db4745f10>]
>>> plt.title("School")
Text(0.5, 1.0, 'School')
>>> plt.xlabel("Year")
Text(0.5, 0, 'Year')
>>> plt.ylabel("Student")
Text(0, 0.5, 'Student')
>>> plt.show()
```

Description

Line 1: matplotlib ကို plt အဖြစ် အသုံးပြုမည်။

Line 2: numpy ကို np အဖြစ်အသုံးပြုမည်။

Line 3: ခုနှစ်တွေပါတဲ့ array တည်ဆောက်လိုက်သည်။

Line 4: လူဦးရေပါတဲ့ array တည်ဆောက်လိုက်သည်။

Line 5: Graph တစ်ခုကိုရေးဆွဲလိုက်သည်။ Line ကို dashed ပုံစံနဲ့ ပြသမည်ဖြစ်သည်။ လိုင်းအရောင်ကို အနီရောင်အဖြစ်သတ်မှတ်ထားသည်။

Line 6: Graph title ကို အမည်ပေးလိုက်သည်။

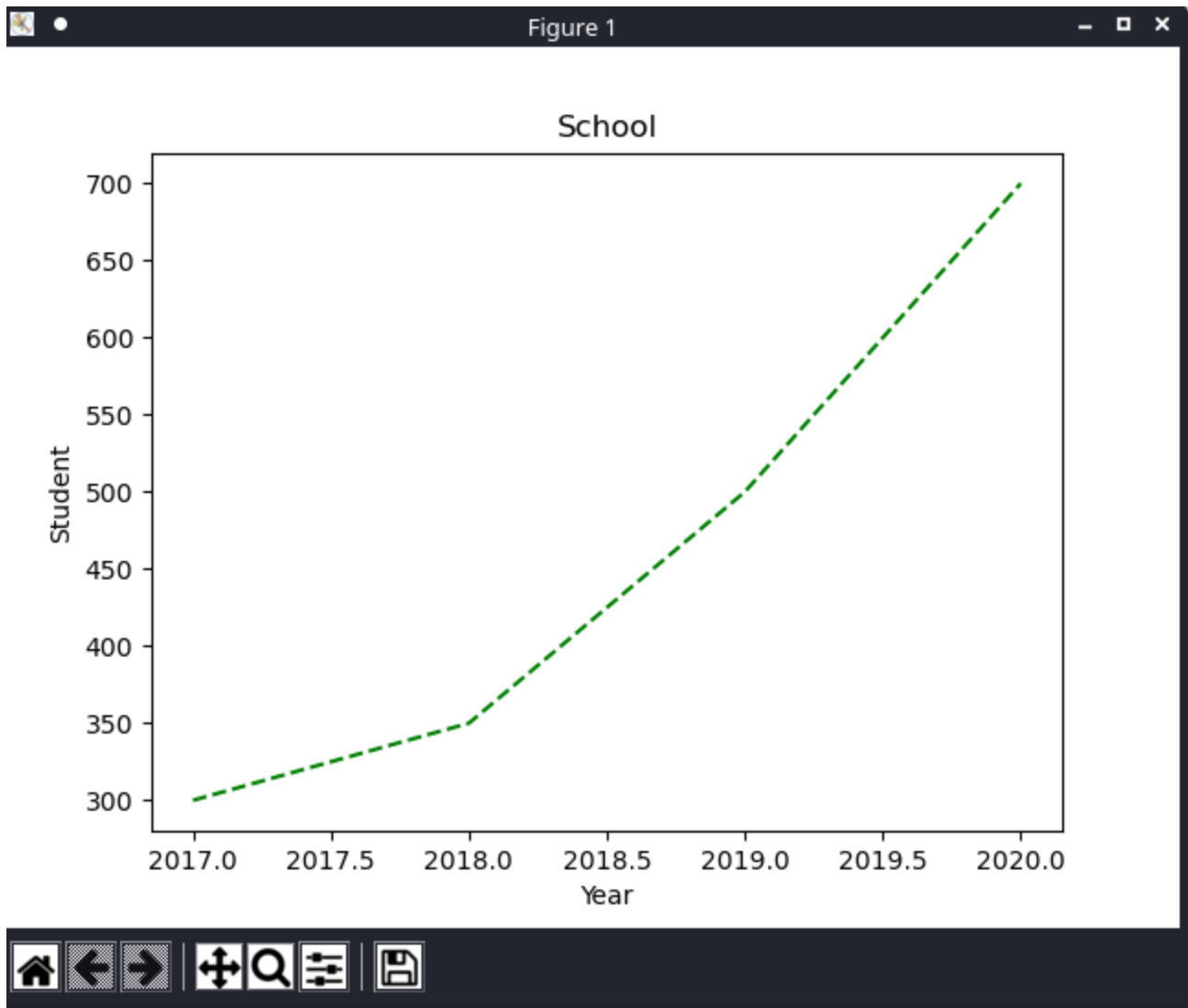
Line 7: x-axis ကို အမည်ပေးလိုက်သည်။

Line 8: y-axis ကို အမည်ပေးလိုက်သည်။

Line 9: Graph ကို output ထုတ်ပြခြင်းဖြစ်သည်။

(Note: Color နေရာမှာ color name လည်း ထည့်နိုင်သလို တခြား RGB နဲ့ ရေးလို့လဲရပါတယ်)

Output:



Bar

Graph ကို Bar အနေနဲ့ ရေးဆွဲလို့ရပါသေးတယ်။ plot နေရာမှာ bar ပြောင်းသွားတာလောက်ပဲရှိပါတယ်။ Example လေးနဲ့ တစ်ချက် ကြည့်လိုက်အောင်။

```
>>> import matplotlib.pyplot as plt
>>> import numpy as np
>>> x=np.array([1990,1995,2000,2005,2010,2015])
>>> y=np.array([29000,31000,34000,37000,40000,38000])
>>> plt.bar(x,y)
<BarContainer object of 6 artists>
>>> plt.title("Myanamr's Rice Export")
Text(0.5, 1.0, 'Myanamr's Rice Export')
>>> plt.xlabel("Years")
Text(0.5, 0, 'Years')
>>> plt.ylabel("Tonnes")
Text(0, 0.5, 'Tonnes')
>>> plt.show()
```

Description

Line 1: meplotlib ကို plt အဖြစ် အသုံးပြုမည်ဖြစ်သည်။

Line 2: numpy ကို np အဖြစ် အသုံးပြုမည်ဖြစ်သည်။

Line 3: x-axis တစ်ခုကို array နဲ့ တည်ဆောက်ထားပြီး point သတ်မှတ်ထားသည်။

Line 4: y-axis တစ်ခုကို array နဲ့ တည်ဆောက်ထားပြီး point သတ်မှတ်ထားသည်။

Line 5: Bar Function ဖြစ်သောကြောင့် bar နဲ့တာဖော်ပြမည်ဖြစ်သည်။

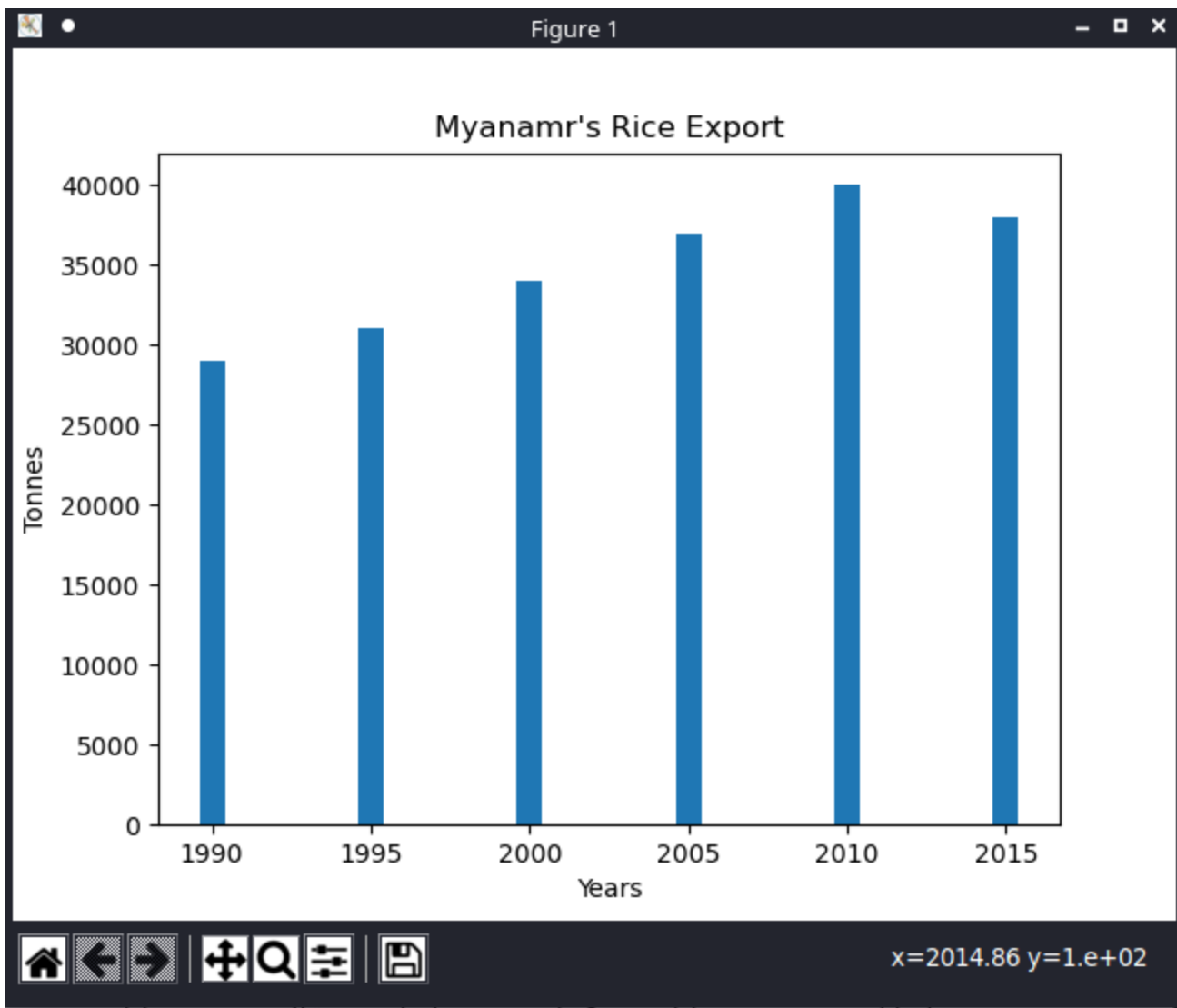
Line 6: Graph ကို name ပေးချင်းဖြစ်သည်။

Line 7: x-axis ကို အမည်ပေးချင်းဖြစ်သည်။

Line 8: y-axis ကို အမည်ပေးချင်းဖြစ်သည်။

Line 9: graph ကို output ထုတ်ပြတာဖြစ်ပါတယ်။

output



(Note အရောင်ပြောင်းချင်ဆိုရင် color="red" ပေါ့)

Basic ကတော့ ဒီလောက်ပါပဲ ကျန်တာဆက်လေ့လာဖို့အဆင်ပြေမှာပါ။

(Challenge မထည့်ပေးတော့ဖူး ဒီခန်းတော့လေ့ကျင့်သွားပါ)

Chapter-17

Python with Excel

File Reading and Writing သင်ခန်းစာမှာတည်းက ကျနော်ဖော်ပြသွားလို့ရပါတယ်။သို့သော် Register software ရေးဖို့အတွက် ယခုလို တွဲဖတ်ဖော်ပြရတာဖြစ်ပါတယ်။ File Reading and writing ရရင် အဆင်ပြေပါတယ်။

Example

```
import csv
file=open("student.csv",'a')
file.write("Name\t")
file.write("Age\t")
file.write("Class\t")
file.close()
```

Description

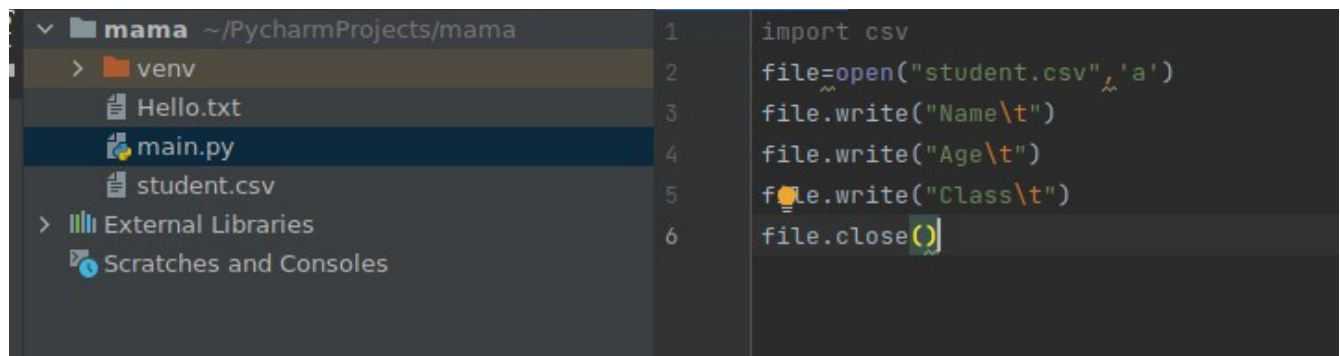
Line 1: Csv file ကို အသုံးပြုမှာဖြစ်သောကြောင့် csv module ကို import လုပ်တာဖြစ်ပါတယ်။

Line 2: student.csv ဆိုတဲ့ file ကို create မှာ ဖြစ်ပြီး file mode ကတော့ append မှာဖြစ်ပါတယ်။

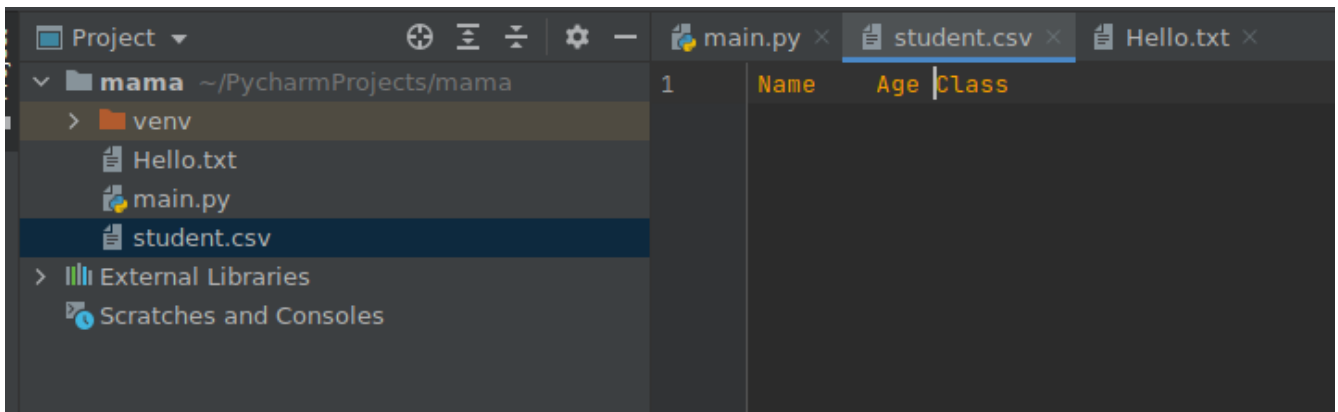
Line 3,4 and 5 ကတော့ ရေးထားတဲ့ စာတွေကို excel file ထဲထည့်လိုက်တာဖြစ်ပါတယ်။

Line 6: file ရေးပြီးကြောင်းကြေညာလိုက်တာဖြစ်ပါတယ်။ရေးပြီးပိတ်လိုက်တဲ့သဘောပါ။

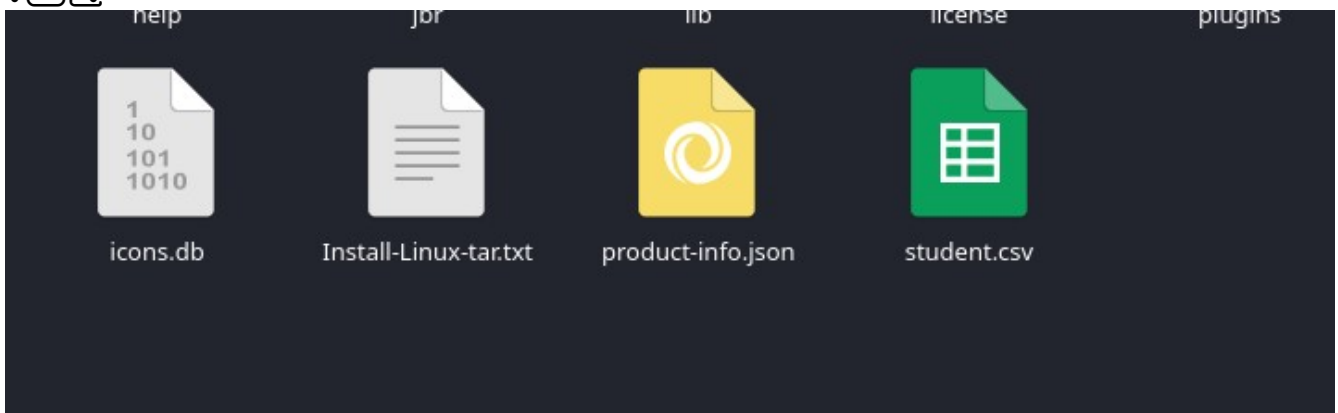
(Note: File reading and writing သင်ခန်းစာနဲ့ ဘာမှ သိမ့်မကွာသွားဖူးနော်)



mama ဆိုတဲ့ Folder အောက် main.py ရဲ့ အောက်မှာ student.csv ဆိုတဲ့ file လေး အလိုအလျောက် ဖန်တီးထားတာကိုး တွေ့မှာပါ။



Student.csv file ကို နိမ့်လိုက်ရင် ယခုသင်္ကေတသို့ ပုံစံလေးတွေ့ရမှာပစ်ပါတယ်။ အပြင်ကနေ file လေးဖွင့်ကြည့်အောင်။



ယခုလို File လေးတွေ့ပါလိမ့်မယ်။

	A	B	C	D	E	F	G
1	Name	Age	Class				
2							
3							
4							
5							
6							
7							
8							
9							
10							

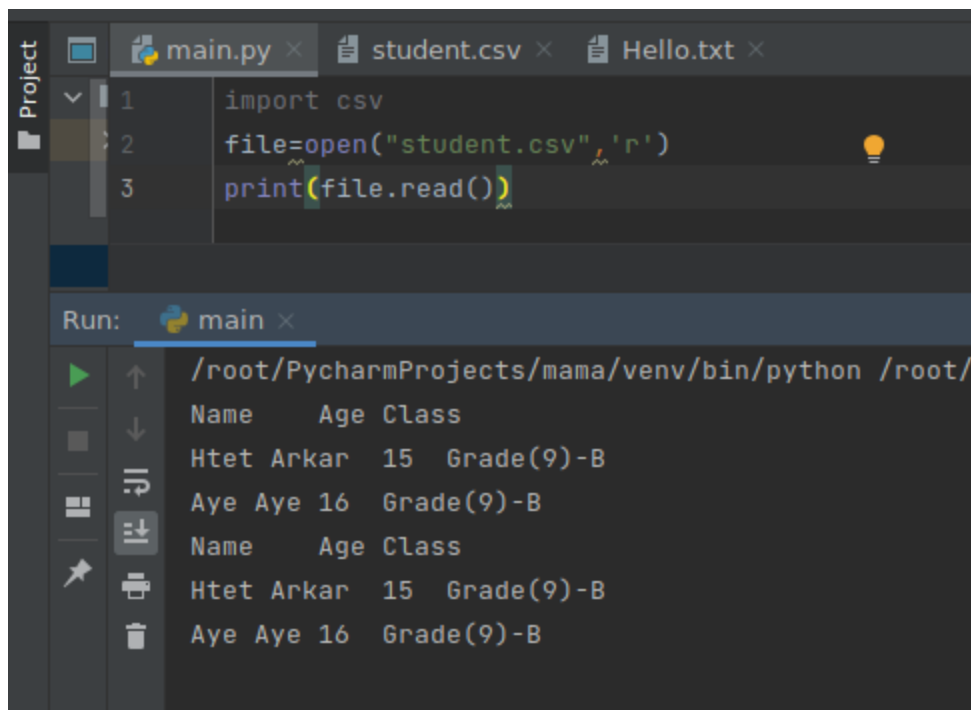
File ကိုဖွင့်လိုက်ရင် ယခုသင်္ကေတသို့ တွေ့ရပါလိမ့်မယ်။ နောက်သင်ခန်းစာ ဆက်သွားကြရအောင်။

တစ်ခုထည့်မှတ်ထားဖို့ /t က ရိုးရိုး file reading and writing မှာ tab ခုတ်တာဖြစ်ပြီး။ csv မှာ တစ်ခြား cell ကိုပြောင်းတာဖြစ်ပါတယ်။ /n က တော့တူတူပါပဲ အောက်တစ်ကြောင်းဆင်းတာဖြစ်ပါတယ်။ ဒီတစ်ခါ ကျောင်းသားနှစ်ယောက်ရဲ့ data တွေထည့်ကြည့်ကြတာပေါ့ဗျာ။

Example

```
import csv
file=open("student.csv",'a')
file.write("Name\t")
file.write("Age\t")
file.write("Class\n")
file.write("Htet Arkar\t")
file.write("15\t")
file.write("Grade(9)-B\n")
file.write("Aye Aye\t")
file.write("16\t")
file.write("Grade(9)-B\n")
file.close()
```

ဒါက တော့ write function နဲ့ data တွေကို ထပ်ဖြည့်တာဖြစ်ပါတယ်။(run ကြည့်ပေါ့)



The screenshot shows the PyCharm IDE with a file named `main.py` open. The code in the editor is as follows:

```
1 import csv
2 file=open("student.csv",'r')
3 print(file.read())
```

Below the editor, the Run console shows the output of the script. The output is displayed in two separate runs, each showing the contents of the `student.csv` file. The first run shows the header and the first data row, and the second run shows the header and the second data row.

```
Run: main x
/root/PycharmProjects/mama/venv/bin/python /root/PycharmProjects/mama/main.py
Name    Age    Class
Htet Arkar  15    Grade(9)-B
Aye Aye  16    Grade(9)-B

Name    Age    Class
Htet Arkar  15    Grade(9)-B
Aye Aye  16    Grade(9)-B
```

`file.read()` function နဲ့ excel file ထဲက data တွေကို output ထုတ်တာဖြစ်ပါတယ်။
student file ကိုဖွင့်ကြည့်ရအောင်နောက်ပိုင်း file တွေကိုတော့ ကိုယ်ဘာသာ ဖွင့်ကြည့်ဗျ။ File က
pycharmProject folder အောက်မှာရှိပါတယ်။

	A	B	C	D	E	F	
1	Name	Age	Class				
2	Htet Arkar	15	Grade(9)-B				
3	Aye Aye	16	Grade(9)-B				
4	Name	Age	Class				
5	Htet Arkar	15	Grade(9)-B				
6	Aye Aye	16	Grade(9)-B				

Project တစ်ခု တူတူလုပ်ကြည့်အောင်။ File name ကိုလဲ Input function နဲ့ ကြိုက်သလို သတ်မှတ်လို့ရတယ်။ data တွေကို ထည့်လို့ရမယ်။ file ထဲက data တွေကိုလဲ output ထုတ်လို့ရမယ်။

Example and Project Code

```
import csv
def menu():
    print("""
        Main Menu
        (1) Create File
        (2) Display The File
        (3) Add New items to The File """)
    try:
        a=int(input("Enter Number:"))
        if a==1:
            cf()
        elif a==2:
            df()
        elif a==3:
            af()
        else:
            print("This number does not exit")
            menu()
    except ValueError:
        print("Enter Only Number")
        menu()
def cf():
    fname=str(input("Enter File Name: "))
    file=open(fname,'w')
```

```
file.write("Name\t")
file.write("Salar\t")
```

```
file.close()
```

```
def df():
```

```
    fname=str(input("Enter File Name: "))
    file=open(fname,'r')
    print(file.read())
```

```
def af():
```

```
    fname=str(input("Enter File Name: "))
    file=open(fname,'a')
    name=str(input("Enter Your Name: "))
    salar=str(input("Enter Your Salar: "))
    file.write("\n"+name+"\t")
    file.write(salar+"\n")
    file.close()
```

```
menu()
```

Output ကတော့ ကိုယ်တိုင် run ကြည့်ရမှ ရမယ်။ Code ကို အပိုင်းနှင့် ရှင်းပြပေးပါမယ်။
အဓိက ဒီ pj က ဝန်ထမ်းအမည် နဲ့ လစာ တွေကို စာရင်းကောက်ယူတာဖြစ်ပါတယ်။
ယခု pj ကိုနားလည်ရင် ရိုးရှင်းတဲ့ register software ရေးလို့ရပါတယ်။

Description

Line 1: ကတော့ csv module ကို import လုပ်တာဖြစ်ပါတယ်။

Line 2: def menu() ကတော့ Menu function တစ်ခုကိုတည်ဆောက်လိုက်တာဖြစ်ပါတယ်။

ထို့နောက် user ကို input လုပ်ခိုင်းတယ်။ Try နဲ့ code တွေကို အလုပ်လုပ်စေပြီး user သွင်းလိုက်တာ 1 နဲ့ညီခဲ့ရင် cf function ကိုအလုပ်လုပ်ပေါ့။ တစ်ခုမှမပါခဲ့ရင် menu စီကိုပြန်သွားပေါ့။ အောက်ဆုံးကြောင်းကတော့ user က input မှားသွားရင် error ပြပြီး code ကို ပြန် run စေတာပါ။

Line 3: def cf() ကတော့ cf function ကို တည်ဆောက် လိုက်တာ ဖြစ်ပါတယ်။သူက file တစ်ခုကို user ကြိုက်တဲ့ name နဲ့ create စေပြီး အထဲမှာ name နဲ့ salar ကို ထည့်ထားပါတယ်။

Line 4: df function ကို တည်ဆောက်လိုက်တာဖြစ်ပါတယ်။သူကတော့ file ထဲdata တွေကို read ဖို့သို့ output ထုတ်ကြည့်ချင်တဲ့ အခါ မှာ အသုံးပြုပါတယ်။

Line 5: af function ကို တည်ဆောက်လိုက်တာဖြစ်ပါတယ်။ သူက data အသစ်တွေထည့်ဖို့အတွက် အသုံးပြုပါတယ်။ကိုယ် Data ထည့်ချင်တဲ့ File Name ကို ထည့်ပါ။(ကျနော်ကတော့ ထည့်ပြီး ပြီပေါ့)။ထို့နောက် ဝန်ထမ်း အသစ်နဲ့ သူရဲ့ လစာကို input ထည့်လိုက်ပေါ့။ (ပြန်ဖတ်ချင်ရင် df function ကို run စေပေါ့။Menu ကနေ 2 ကိုထည့်လိုက်ပေါ့။)

Line 6: menu() function ကို အလုပ်လုပ်ဖို့ ခေါ်လိုက်တာဖြစ်ပါတယ်။

Chapter-18

Tkinter

GUI ဆိုတာ Graphic user interface ဖြစ်သည်။ turtle သင်ခန်းစာမှာလဲကျနော်ပြောပြခဲ့ပြီးပါပြီ။ ကျနော်တို့အပေါ် လုပ်ခဲ့ တဲ့ CLI (Command Line Interface) မှာ button တွေ မပါ screen တွေမပါနဲ့ သိမ့်ပြီး အထာမကျဖူး။ ဒါကြောင့် GUI နဲ့တွဲ ဖတ်ပြီး software ရေးကြတာပါ။ Button တွေဘာညာ တွေကို ဖန်တီးဖို့ Python tkinter module မှာ အသင့်ပါပြီးသားဖြစ်ပါသည်။

Tkinter နဲ့ပတ်သတ်ပြီး function တွေရဲ့ လုပ်ဆောင်ချက်ကို ဖော်ပြသွားပါမယ်။

import tkinter = tkinter နဲ့ ပတ်သတ်တဲ့ functions တွေကို အသုံးပြုဖို့အတွက် module ကို import လုပ်ခြင်းဖြစ်သည်။

Geometry

Geometry function ကို Screen ရဲ့ အကျယ်အဝန်း ကို သတ်မှတ်တဲ့နေရာမှာ အသုံးပြုပါတယ်။

Example

```
tkinter.geometry("200x600")
```

Description

tkinter module ထဲက geometry function နဲ့ အကျယ် 200 pixel ၊ အရှည် 600 pixel ရှိတဲ့ Screen တစ်ခု ကို ဖန်တီးခြင်းဖြစ်သည်။

Title

ကျနော်တို့ SOFTWARE ကို နာမည်ပေးတဲ့နေရာမှာ title function ကို အသုံးပြုပါတယ်။

Example

```
tkinter.title("Hello Python Tkinter")
```

Tkinter module ထဲက title function နဲ့ software ကိုနာမည်ပေးပါ။

Label

ကျနော်တို့ label အညွှန်းတပ်တဲ့နေရာမှာအသုံးပြုပါတယ်။

example

```
Label(text="Hello")
```

Description

Hello ဆိုတဲ့ label အညွှန်းတပ်ပါလို့ဆို လိုချင်းဖြစ်သည်။

Place

place ကို ကျနော်တို့ ရဲ့ label တွေ button တွေ စသဖြင့်နေရာကိုသတ်မှတ်တဲ့ နေရာမှာ အသုံးပြုပါတယ်။ ဥပမာ ဘယ်နေရာမှာ ဘာရှိရမယ်စသဖြင့်ပေါ့။ `place(x=30,y=60)` Graphic ပုံမြင်ရင်နားလည်သွားပါလိမ့်မယ်။(python matplotlib ခန်းကနဲ့တူတူပါပဲ) x နဲ့ 30 အကွာ Y နဲ့ 70 အကွာတွေရှိသောနေရာတွင်နေရာချပါ။

Place Function ကို နေရာချဖို့အတွက်တင်မကပဲ။ အရွယ်အစား size ကို လဲ သတ်မှတ်လို့ရပါတယ်။

Example

```
place(x=30,y=70,weight=60,height=30)
```

x နဲ့ y က နားလည်ပြီး ထည့်ပါတယ်။ weight = 60 ဆိုတာ အကျယ်က 60 pixel ဖြစ်ပြီး အမြင့် 30 pixel ဖြစ်ရမယ်လို့ဆိုလိုတာပါ။

Entry Box

ကျနော်တို့ CLI (Command Line interface) မှာ user ကို data တွေထည့်ခိုင်းဖို့ input function ကို အသုံးပြုသလို GUI(Graphic User Interface) မှာ user ကို data တွေထည့်ခိုင်းဖို့ Entry Box ကို အသုံးပြုလို့ရပါတယ်။

Example

```
Entry("Enter Your Name: ")
```

User name ကို input ထည့်ခြင်းဖြစ်ပါတယ်။

Button

ကျနော်တို့ Button (ခလုတ်) ကို ဖန်တီးရာတွင် Button function ကို အသုံးပြုပါတယ်။

Example

```
Button(text="Click here",command="hello")
```

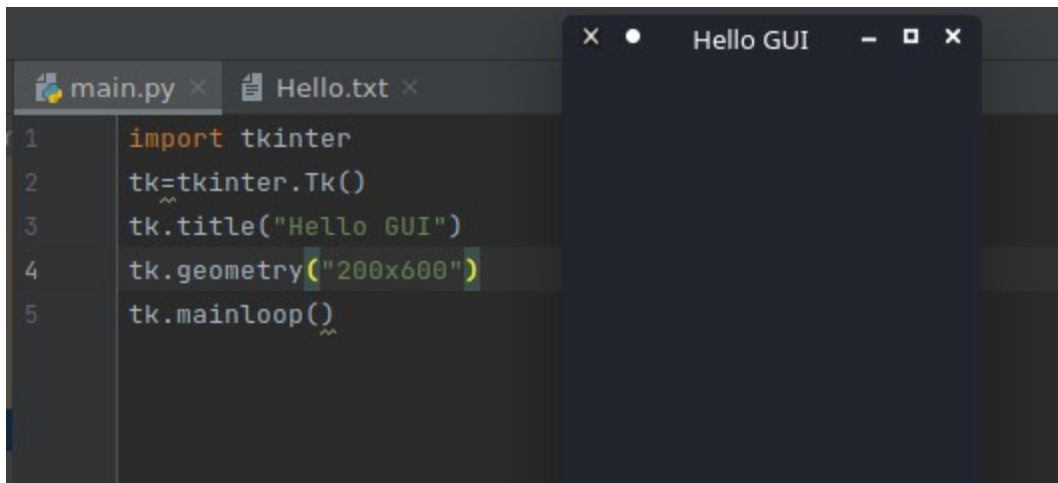
Description

Click here ဆိုတာ Button ပေါ်မှာ မြင်ရမည်စာသားဖြစ်ပြီး၊ hello ဆိုတာ ကျနော်တို့လုပ်ထားတဲ့ subprogram တစ်ခုဖြစ်သည်။ဆိုလိုတာက Click here ဆိုတဲ့ Button ကို နိမ့်တာနဲ့ hello ဆိုတဲ့ function ကို အလုပ်လုပ်မည်ဖြစ်သည်။

Mainloop

Mainloop fuction ကို program ရဲ့နောက်ဆုံးတွင်ရေးမှာဖြစ်ပြီး။ mainloop မပါရင်အလုပ်လုပ်မလုပ်ပါ။

Example

The image shows a code editor window with two tabs: 'main.py' and 'Hello.txt'. The 'main.py' tab is active, displaying five lines of Python code. To the right of the code editor, a separate window titled 'Hello GUI' is visible, which is currently empty. The code in the editor is as follows:

```
1 import tkinter
2 tk=tkinter.Tk()
3 tk.title("Hello GUI")
4 tk.geometry("200x600")
5 tk.mainloop()
```

Description

Line 1: tkinter module ကို import လုပ်လိုက်သည်။

Line 2: tkinter function ကို tk အတိုကောက်အနေနဲ့ အသုံးပြုချင်းဖြစ်သည်။

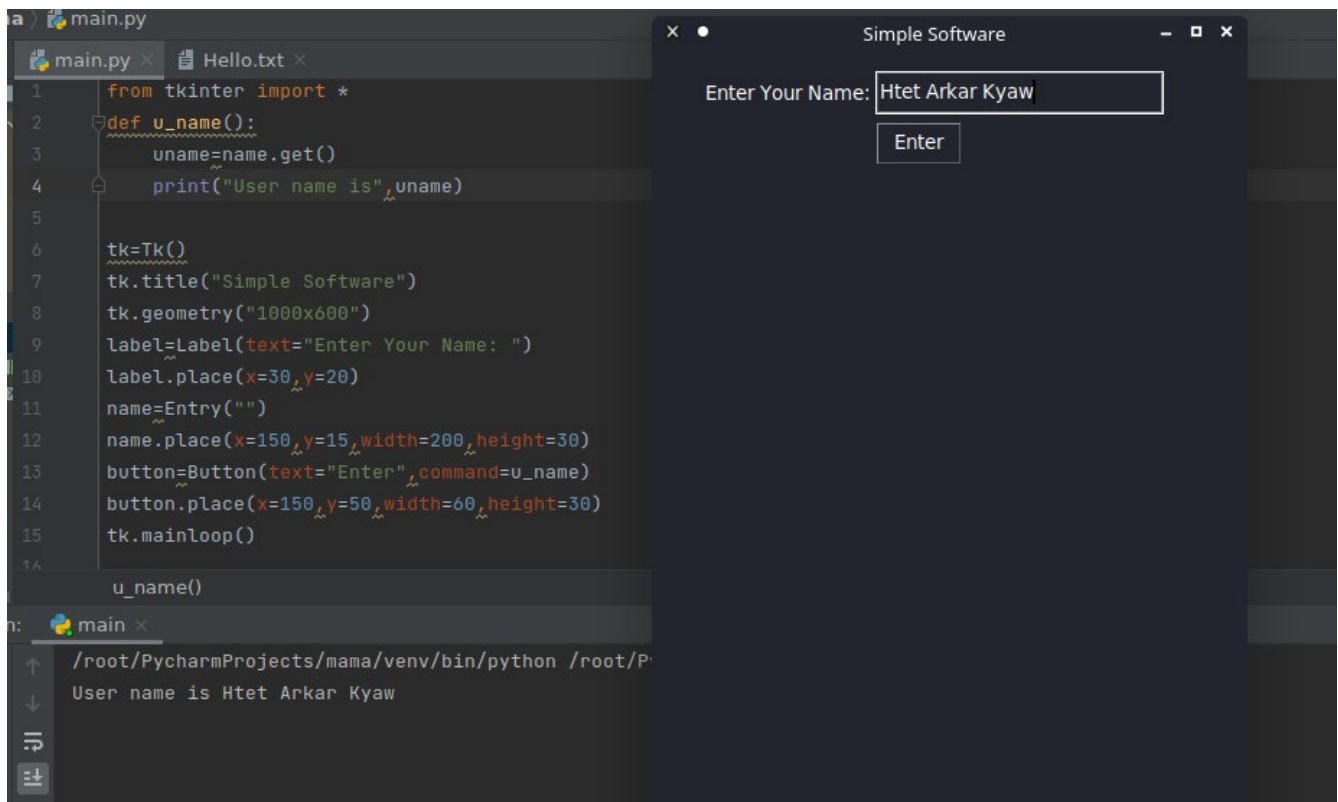
(Note: import tkinter as tk အနေနဲ့လဲ အသုံးပြုနိုင်သည်။)

Line 3: Program ကို Hello GUI လို့အမည်ပေးချင်းဖြစ်သည်။

Line 4: Screen ရဲ့ အကျယ်အဝန်းကို သတ်မှတ်ချင်းဖြစ်သည်။

Line 5: mainloop နဲ့ program ကိုပြီးဆုံးကြောင်းကြေညာလိုက်ခြင်းဖြစ်သည်။

The Next Example.



Description

Line 1: tkinter module ရဲ့ function တွေအကုန်လုံးကို အသုံးပြုမယ်လို့ကြေညာလိုက်တယ်။

Line 2: u_name() function တစ်ခုကိုတည်ဆောက်လိုက်တယ်။

Line 3: uname=name.get() ဆိုတာ name Variable ပေးထားတဲ့ Entry က ပို့လိုက်တဲ့ အမည်ကိုရယူရန်ဖြစ်သည်။

Line 4: user ထည့်လိုက်တော့ name နဲ့အတူ output ထုတ်ရန်ဖြစ်သည်။

Line 5: ကကျော်လိုက်။

Line 6: tkinter ကို tk အတိုကောက်လို့အမည်ပြောင်းမှန်ထားတာဖြစ်ပါတယ်။

Line 7: software ရဲ့ ခေါင်းစဉ်အမည်ပေးခြင်းဖြစ်သည်။

Line 8: screen ရဲ့ အကျယ်ကို သတ်မှတ်လိုက်ခြင်းဖြစ်သည်။

Line 9: Label တပ်ခြင်းဖြစ်သည်။

Line 10: Label ရှိရမည် တည်နေရာကို သတ်မှတ်ခြင်းဖြစ်သည်။

Line 11: name ဆိုတဲ့ variable နဲ့ အတူ Entry ဆိုတဲ့ user input box တစ်ခုဖန်တီးခြင်းဖြစ်သည်။

Line 12: Entry box ရဲ့ တည်နေရာကို သတ်မှတ်ခြင်းဖြစ်သည်။

Line 13: Button တစ်ခုကို ဖန်တီးလိုက်ခြင်းဖြစ်သည်။

Line 14: Button ရဲ့နေရာရော size ရောသတ်မှတ်ထားခြင်းဖြစ်သည်။

Line 15: mainloop နဲ့ program ကိုအဆုံးသတ်လိုက်သည်။

(Note: user က နာမည် ထည့်ပြီး Enter နိမ့်လိုက်တာနဲ့ ၎င်းနာမည်ကို u_name ဆိုတဲ့ function ထဲကို ပို့လိုက်တယ်။ Get နဲ့ နာမည်ကိုလက်ခံပြီး output ပြန်ထုတ်သည်။ကိုယ်တိုင် run ကြည့်ရင်နားလည်ပါ လိမ့်မယ်။ code အမြင်မရှင်းမှာဆိုးလို့အောက်မှာပေးထားပါတယ်)

Code:

```
from tkinter import *  
def u_name():  
    uname=name.get()  
    print("User name is",uname)  
  
tk=Tk()  
tk.title("Simple Software")  
tk.geometry("1000x600")  
label=Label(text="Enter Your Name: ")  
label.place(x=30,y=20)  
name=Entry("")  
name.place(x=150,y=15,width=200,height=30)  
button=Button(text="Enter",command=u_name)  
button.place(x=150,y=50,width=60,height=30)  
tk.mainloop()
```

Other Basic Functions

Message(text="text") သည် စာသားများကို output ထုတ်တဲ့နေရာမှာအသုံးပြုပါသည်။
bg="red" သည် background color ကို အနီရောင်ထားရန်အတွက်အသုံးပြုသည်။
fg="blue" သည် font color ကို အပြာရောင်ပြောင်းလဲချင်းဖြစ်သည်။
listbox() သည် string data types တွေပါဝင်တဲ့ drop-down list တစ်ခုကို တည်ဆောက်ရန်ဖြစ် သည်။
delete(0,End) သည် Entry သို့မဟုတ် list box ထဲမှာ ရှိတဲ့ contents တွေကို ဖျက်ပစ်တာပါ။
get() entry box ထဲက data တွေကို save တာဖြစ်ပါတယ်။
window.wm_iconbitmap("photo.jpg")
software ရဲ့ icon ကို change တာဖြစ်ပါတယ်။
window.configure(background="color name") က background color change တာဖြစ်ပါတယ်။

Example

```
from tkinter import *
window=Tk()
window.geometry("500x200")
def textbox2():
    textbox2=Message(text=data.get())
    textbox2.place(x=155,y=50,width=200,height=25)
    textbox2["bg"]="white"
    textbox2["fg"]="black"
data=StringVar()
label1=Label(text="Enter Your Name: ")
label1.place(x=30,y=20)
textbox1=Entry(textvariable=data)
textbox1.place(x=150,y=20,width=200,height=25)
textbox1["justify"]="center"
textbox1.focus()
button=Button(text="Press Me",command=textbox2)
button.place(x=30,y=50,width=120,height=25)
window.mainloop()
```

Example က code ကို နည်းနည်းလေး ထည့်ဖြည့်လိုက်တာပဲရှိတယ်။

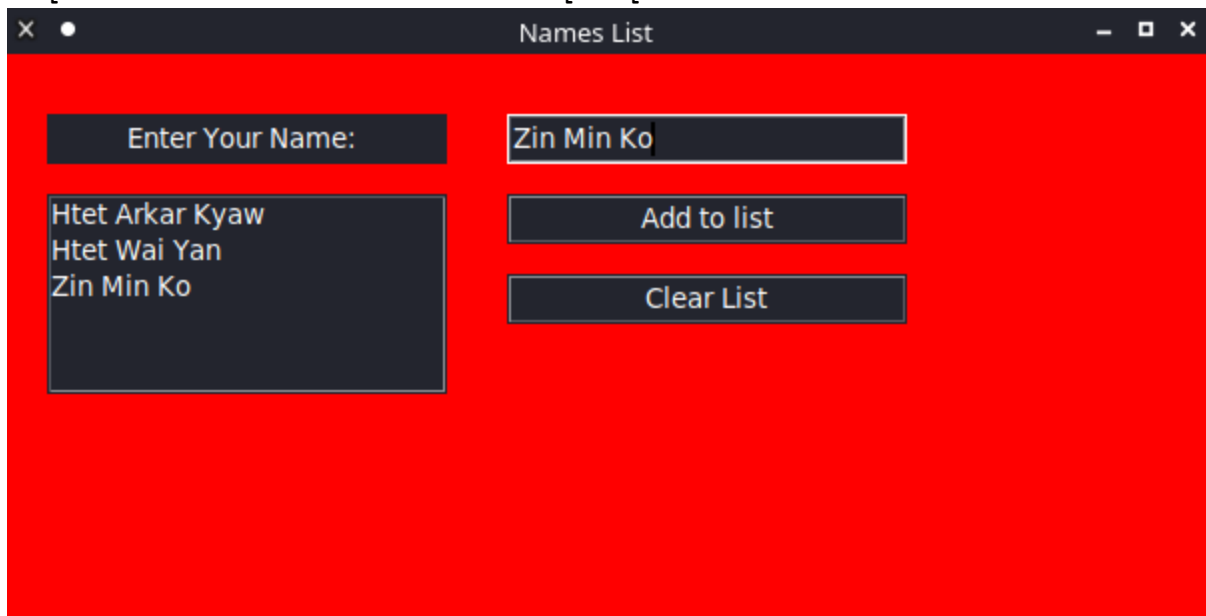
လိုအပ်တဲ့ နေရာလေးပဲရှင်းပြသွားမယ်။

textbox2 function တစ်ခုတည်ဆောက်လိုက်တယ်။

data.get() function က entry box က သွင်းလိုက်တဲ့ data တွေကို လက်ခံပြီး output ပြန်ထုတ်ပြဖို့ ပါ။အောက်ကအကြောင်းက message box ရဲ့တည်နေရာနဲ့ အကျယ်အဝန်းကိုတည်ဆောက်တာဖြစ်ပါတယ်။အောက်ကြောင်းက message box ရဲ့ background color ဖြစ်ပါတယ်။အောက်က တော့ fg ဆိုတာတော့ font color ဖြစ်ပါတယ်။data=StringVar() ခုဆိုတာ Entry box က သွင်းတဲ့ data ကို string data type အဖြစ်သတ်မှတ်လိုက်တာပါ။အောက်နှစ်ကြောင်းက label 1 ရဲ့ တည်နေရာနဲ့ စာသားကို ဖော်ပြပေးတာဖြစ်ပါတယ်။အောက်က textbox1 နဲ့ပတ်သတ်အရောင်ကာလာအကုန်လုံးကို သတ်မှတ်ထားတာပါ။Button ရဲ့ အနေအထားကို ဖန်တီးလိုက်တာဖြစ်ပါတယ်။အောက်ဆုံးအကြောင်းက တော့ mainloop() နဲ့အဆုံးသတ်လိုက်တာပေါ့။

Example

Name တွေကို list တစ်ခုထည့်မယ်။ background-color change မယ်။ List ထဲက name တွေကို clear လုပ်မယ်။ အောက်ပါ software လေးအတိုင်းလုပ်တာပေါ့။



Code and Description

```
from tkinter import *
def name():
    name=name_box.get()
    name_list.insert(END,name)
    name_box.focus()
def clear():
    name_list.delete(0,END)
    name_box.focus()
w=Tk()
w.configure(background="red")
w.title("Names List")
w.geometry("600x400")
label1=Label(text="Enter Your Name: ")
label1.place(x=20,y=30,width=200,height=25)
name_box=Entry(text=0)
name_box.place(x=250,y=30,width=200,height=25)
name_box.focus()
```

```

button1=Button(text="Add to list",command=name)
button1.place(x=250,y=70,width=200,height=25)
name_list=Listbox()
name_list.place(x=20,y=70,width=200,height=100)
button2=Button(text="Clear List",command=clear)
button2.place(x=250,y=110,width=200,height=25)
w.mainloop()

```

Description

Line 1: module import လုပ်တာဖြစ်ပါတယ်။

Line 2: name() function တစ်ခုကိုတည်ဆောက်လိုက်တာဖြစ်ပါတယ်။name variable ထဲကို name_box က လာတဲ့ data တွေကို get function နဲ့ ယူပြီး သို့လှောင်လိုက်တာဖြစ်ပါတယ်။ ထို့နောက် name_list ထဲသို့ list အဖြစ် name က လာတဲ့ data တွေကို ထည့်လိုက်တာဖြစ်ပါတယ်။ အောက်တစ်ကြောင်းကတော့ focus ထားလိုက်တာဖြစ်ပါတယ်။(ကိုယ်ဘာသာလေ့လာကြည့်)

Line 3: clear() function တစ်ခုကို တည်ဆောက်လိုက်တယ်။name_list (list)ထဲရှိ အချက်အလက် အမည်ကို clear or delete လုပ်လိုက်တာဖြစ်ပါတယ်။အောက်ကတော့ ကျနော်ပြောထားတဲ့အတိုင်းပဲ ဖြစ်ပါတယ်။

Line 4: tk function ကို w အနေနဲ့ အသုံးပြုမည်ဖြစ်သည်။

Line 5: background(နောက်ခံကာလာ) ကိုပြောင်းလိုက်တာဖြစ်ပါတယ်။

Line 6: title အမည်ပြောင်းတာဖြစ်ပါတယ်။

Line 7: screen ရဲ့ အကျယ်အဝန်းကို သတ်မှတ်တာဖြစ်ပါတယ်။

Line 8: Label တစ်ခုနဲ့ သူရဲ့တည်နေရာကို ဖန်တီးလိုက်တာဖြစ်ပါတယ်။

Line 9: Entry box တစ်ခုနဲ့ သူရဲ့တည်နေရာကို ဖန်တီးလိုက်တာဖြစ်ပါတယ်။

Line 10: Button တစ်ခုနဲ့ သူရဲ့တည်နေရာကို ဖန်တီးလိုက်တာဖြစ်ပါတယ်။

Line 11: list တစ်ခုနဲ့ သူရဲ့တည်နေရာကို ဖန်တီးလိုက်တာဖြစ်ပါတယ်။

Line 12: button(2) တစ်ခုနဲ့ သူရဲ့တည်နေရာကို ဖန်တီးလိုက်တာဖြစ်ပါတယ်။

Line 13: main.loop() ဖြင့် အဆုံးသတ်လိုက်ခြင်းဖြစ်သည်။

Challenges

(တစ်ချို့နေရာတွေကို အနည်းငယ်ရှင်းပြပေးသွားမှာပါ)

C-

login system တစ်ခုတည်ဆောက်ပါ။

C-

user register form တစ်ခုတည်ဆောက်ပါ။ Data တွေကို csv form ထဲမှာ ထည့်ပြီးသားဖြစ်ပါစေ။

Solution – One

```
from tkinter import *
from tkinter import messagebox
w=Tk()
w.title("Log In System")
w.geometry("400x500")
name=StringVar()
password=StringVar()
def log():
    if name.get()=='admin' and password.get()=='admin':
        messagebox.showinfo(title="Login Status",message="Access Admin")
    else:
        messagebox.showerror(title="Login Error",message="Username/password is
incorrect")
def can():
    status=messagebox.askyesno(title='Question',message="Do you Want to Close the
window?")
    if status==True:
        w.destroy()
    else:
        messagebox.showwarning(title='Warning Message',message="Please Log in Again!")

#label for username
l1=Label(text="Username : ",font=(18))
l1.place(x=20,y=20)
#enter for username
e1=Entry(textvariable=name)
e1.place(x=130,y=20,width=200,height=25)
#label for password
l2=Label(text="Password :",font=(18))
```

```

l2.place(x=20,y=50)
#entry for password
e2=Entry(textvariable=password)
e2.place(x=130,y=50,width=200,height=25)
#button1
b1=Button(text="Login",command=log,font=(18))
b1.place(x=130,y=80,width=80,height=30)
#button2
B2=Button(text="Cancel",command=can,font=(18))
B2.place(x=250,y=80,width=80,height=30)
w.mainloop()

```

output ကိုယ်တိုင် run ကြည့်ပါဗျာ။

```

from tkinter import *
from tkinter import messagebox
w=Tk()
w.title("Log In System")
w.geometry("400x500")
name=StringVar()
password=StringVar()
def log():
    if name.get()=='admin' and password.get()=='admin':
        messagebox.showinfo(title="Login Status",message="Access Admin")
    else:
        messagebox.showerror(title="Login Error",message="Username/password is
incorrect")
def can():
    status=messagebox.askyesno(title='Question',message="Do you Want to Close the
window?")
    if status==True:
        w.destroy()
    else:
        messagebox.showwarning(title='Warnning Message',message="Please Log in Again!")

```

```

#label for username
l1=Label(text="Username : ",font=(18))
l1.place(x=20,y=20)
#enter for username
e1=Entry(textvariable=name)
e1.place(x=130,y=20,width=200,height=25)
#label for password
l2=Label(text="Password :",font=(18))
l2.place(x=20,y=50)
#entry for password
e2=Entry(textvariable=password)
e2.place(x=130,y=50,width=200,height=25)
#button1
b1=Button(text="Login",command=log,font=(18))
b1.place(x=130,y=80,width=80,height=30)
#button2
B2=Button(text="Cancel",command=can,font=(18))
B2.place(x=250,y=80,width=80,height=30)
w.mainloop()

```

Description

line 1: tkinter module ကို import လုပ်ထားတာဖြစ်ပါတယ်။

Line 2: messagebox function ကို အသုံးပြုရန်အတွက် import လုပ်လိုက်သည်။

Line 3: tk function ကို w အဖြစ်အသုံးပြုရန်ဖြစ်သည်။

Line 4: name နဲ့ screen အရွယ်အစားကို သတ်မှတ်ချင်းဖြစ်သည်။

Line 5: name နဲ့ password ကို string variable အဖြစ်သာသတ်မှတ်ထားသည်။

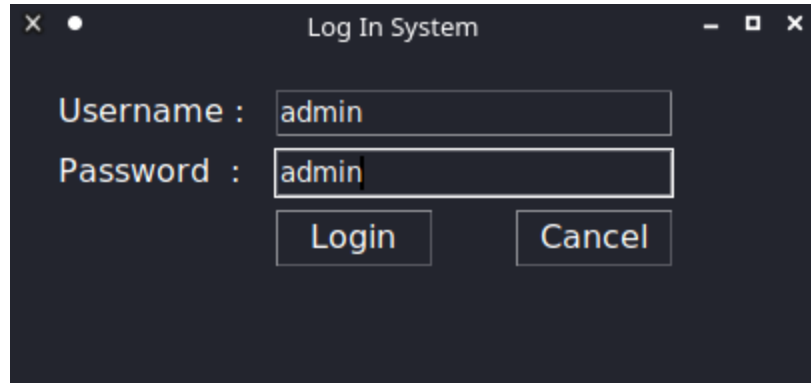
Line 6: log function တစ်ခုတည်ဆောက်လိုက်သည်။ entry box က သွင်းလိုက်တဲ့ username နဲ့ password ကို စစ်ဆေးမယ်။ ညီခဲ့ရင် Login status ခေါင်းစဉ်ရှိတဲ့ messagebox နဲ့အတူ Access admin ဆိုတဲ့ စာသားပါလိမ့်မယ်။ မတူညီခဲ့ဖူးဆိုရင် login error ခေါင်းစဉ်ပါတဲ့ messagebox နဲ့အတူ username/password is incorrect ဆိုတဲ့ စာသားနဲ့အတူ box တစ်ခုကျလာလိမ့်မယ်။

Line 7: can ဆိုတဲ့ function တစ်ခုတည်ဆောက်ထားတယ်။ cancel ဆိုတဲ့ စာသားကိုနှိမ့်တာနဲ့ Do you wan to Close This Window ဆိုပြီး box တစ်ခုကျလာမယ်။ Yes နှိမ့်ရင် software ပိတ်လိုက်မှာ

ဖြစ်သည်။ No နိမ့်ခွဲရင် warning message နဲ့အတူ please log in again ဆိုပြီး login form ကို ပြန်
ရောက်မှာဖြစ်ပါတယ်။

အောက်က code တွေက အပေါ်စာကြေးရင်နားလည်ပါလိမ့်မယ်။

output



Solution

```
from tkinter import *
import csv

def save():
    name=e1.get()
    age=e2.get()
    phone=e3.get()
    file=open("reg.csv","a")
    record=name+","+age+","+phone+"\n"
    file.write(str(record))
    file.close()

w=Tk()
w.title("User Register Form")
w.geometry("500x500")
e1=Entry()
e1.insert(0,'Username')
e1.place(x=30,y=40)

e2=Entry()
e2.insert(0,'Age')
e2.place(x=30,y=65)
```

```
e3=Entry()  
e3.insert(0,'Phone')  
e3.place(x=30,y=90)
```

```
b=Button(text="save",command=save)  
b.place(x=50,y=115,width=100,height=24)
```

```
w.mainloop()
```

Description

insert(0,'age') ဆိုတာ form ထဲမှာ age လို့ပေါ်တာပါ။ကျန်တာနားလည်လိမ့်မယ်လို့မျှော်လင်ပါတယ်
နားမလည်ရင် ပြန်လေ့ကျင့်ပါ။

(Note: GUI ပိုင်းကျနော်အားနည်းသောကြောင့် တင်ပြအားနည်းသွားတဲ့အတွက် တောင်းပန်ပါတယ်။
နောက်စာအုပ်မှာပိုကောင်းအောင်ရေးပါမယ်)