

# Let's Write Python

[1]

ဒါလိုမျိုးလေးရေးဖို့ idea ရှိတာတော့ကြာပြီပဲ ဟိုလုပ်ဒီလုပ်နဲ့ ဖြစ်နေတော့ဘာမှသေချာမလုပ်ဖြစ်ခဲ့ဘူး။ အခုတော့လုပ်မယ်ကွာလို့ စိတ်ဆုံးဖြတ်ချက်ချပြီးရေးလိုက်ပါပြီ။ နောက် post တင်တာလဲ ဥရုတ်ငုရုတ်ခြားတစ်ခုလောက်ပဲဖြစ်မယ်ထင်တယ်။ (အတိအကျကြီးတော့မဟုတ်ဘူးပေါ့ဗျာ) ဂျွန်တော်လဲအခုတလကတခြားလုပ်စရာလေးတွေရှိနေလို့ပါ။ ဒီလိုပြောလို့ စာအရေးအသားညံ့သွားမယ်လို့ မပြောဘူးနော် အရှင်းနိုင်ဆုံးရှင်းအောင်ရေးပါမယ်။ 😊

Python Environment Installation တွေဘာတွေတော့တခြားသူတွေတင်ထားတာရှိပြီးသားဆိုတော့ပြန်မပြောနေတော့ပါဘူး။

PS: Windows သမားတွေဆို syntax highlight အတွက် notepad++ လေးကိုဒေါင်းလုပ်ဆွဲပြီးအသုံးပြုပေးကြပါ။ Linux သမားတွေကတော့ gedit နဲ့ လဲအဆင်ပြေပါတယ်။

ဂျွန်တော်ရေးသားတွေသည် Python 2 ပုံစံနဲ့ သာရေးသားသွားမှာဖြစ်လို့ Python Environment Download ဆွဲရင် Python 2 ကိုဆွဲဖို့ တော့သတိပြုစေချင်ပါတယ်။

နောက်ဆုံးမှာခွင်တာလေးကတော့ ဖတ်လာတာနဲ့ အမှုခက်ခဲတာမျိုးကြုံလာရင် မကျော်ပါနဲ့ ခေါင်းခါပြီးထွက်မသွားပါနဲ့။ Python နဲ့ ပတ်သက်တဲ့ community တွေအများကြီးရှိကြပါတယ်။ Documentation တွေလဲရှိပါတယ်။ ဂျွန်တော်ရေးတာလောက်နဲ့ လဲရပ်မနေပါနဲ့။ ဂျွန်တော်ကလမ်းကြောင်းပေါ်ရောက်အောင်ဆွဲခေါ်ဖို့ လောက်ထိပဲတတ်နိုင်ပါတယ် ကိုယ်ထိုင်ထပြီးမရောက်သရွေ့ ရှေ့ရောက်မှာလဲမဟုတ်ပါဘူး။ 🙏

## 1. Variables, expression and statements

### 1.1 Values and Types

Value ဆိုတာကတော့ program အလုပ်လုပ်ရာမှာပါဝင်တဲ့ အခြေခံအရာလေးတစ်ခုပါပဲ။ ဥပမာ numbers (1, 2) ဒါမှမဟုတ် letters ("Hello World") ပေါ့လေ။ ဟုတ်ပြီဗျာ ဒါကဂျွန်တော်တို့ ခြုံကြည့်တဲ့သဘောနော်။ အခု Programming လေ့လာနေတာဆိုတော့ဒါကိုနဲ့နဲ့ထပ်ပြောကြည့်ရအောင်။ ဂျွန်တော်အခုပြောလိုက်တဲ့ numbers နဲ့ letters တွေကတကယ်တော့ မတူညီတဲ့အမျိုးအစားလေးတွေပါပဲ။

Number ထဲမှာလဲအမျိုးမျိုးရှိပါသေးတယ်။ integers, floating point စသဖြင့်ပေါ့နုနုရေးလိုက်တဲ့ 1, 2 တို့ ဆိုတာကတော့ integers (မြန်မာလိုဆို ကိန်းပြည့်ပေါ့နော်) ထားပါ။ E လိုသာမှတ်လိုက်ပိုရှင်းတယ်။ နောက် double quotes လေးနဲ့ ရေးထားတဲ့ "Hello World" ဆိုတာလေးကတော့ string ပေါ့။ Python မှာဘာ type မှန်းမသိရင် interpreter ကိုမေးလို့ ရပါတယ်။ အရင်ဆုံး Python Prompt လေးကိုဖွင့်လိုက်ပါ။ ပြီးရင် အောက်ကလိုရိုက်ပါ။

**Code:**

```
>>> type('Hello World!')
```

ဒါဆို interpreter ကဒီလိုလေးပြန်ဖော်ပြမှာပါ။ <type 'str'> str ဆိုတာကတော့ string ကိုပြောတာပါ။ ဒါဆိုနောက်တစ်ခုစမ်းမယ်ဗျာ။

**Code:**

```
>>> type(404)
```

<type 'int'> လို့ ပြမှာပါ။ int ဆိုတာကတော့ integer ပါ။ နောက်ဆုံး 4.04 ဆိုရင်ကော?

ဒါဆိုရင်တော့ floating point ဖြစ်တဲ့ <type 'float'> လို့ ဖော်ပြမှာဖြစ်ပါတယ်။ ကျောင်းမှာတုံးကတော့ floating point ကို ဂျွန်တော်တို့က

decimal number လို့ ပြောတာပေါ့လေ တူတူပါပဲ။  
ဒါဆိုနောက်တစ်ခုထပ်ရေးမယ်ဗျာ။ ဒီလိုလေးပါ။

**Code:**

```
>>> type('404')
```

<type 'str'> တိန်ဘာလို့ string လို့ ပြန်တာလဲဗျ။ 404 ဆိုတာနံပါတ်ကြီးမဟုတ်ဘူးလား? 404 ဆိုတာတော့ number မှန်ပါတယ်။ ဒါပေသိသူ့ကို single quote နဲ့ ပိတ်ထားပါတယ်။ (double quote အသုံးပြုလဲရပါတယ်။) ဒီလိုရေးသားလိုက်ခြင်းဟာလဲ string တစ်ခုဖြစ်သွားစေပါတယ်။ ဂျွန်စော်တို့တွေစာရေးရင် digit ဃုလုံးပြီးတိုင်း comma (,) ကုဒ်ကြတယ်မလား။ ဒါဟာ python မှာလုပ်လို့ မရပါဘူး။ ရေးလိုက်ရင် Error တော့မတက်ပါဘူး သို့သော် ကိုယ်တွေ့ထားသလိုဖြစ်မှာမဟုတ်ပါဘူး။

**Code:**

```
>>> 1,000,000
```

(1, 0, 0) လို့ ပြန်ဖော်ပြမှာဖြစ်ပါတယ်။ Python Interpreter က tuple တစ်ခုအဖြစ်ပြန်ဖော်ပြလိုက်ပါတယ်။  
(You will learn the tuple later in this tutorial.)

## 1.2 Variables

Variables တွေကိုကောင်းကောင်း manipulate ပြုလုပ်နိုင်ခြင်းသည်လဲ programming language တစ်ခုအတွက်အားတစ်ခုပါပဲ။

**Code:**

```
>>> message = 'I am a message.'
>>> num = 17
>>> pi = 3.142
```

ပထမစာကြောင်းမှာက message ဆိုတဲ့ new variable တစ်ခုစီကို string တစ်ခု assign လုပ်လိုက်ပါတယ်။ ဒုတိယစာကြောင်းကလဲထိုကဲ့သို့ပါပဲ num ဆိုတဲ့ new variable ထဲကို integer value assign လုပ်လိုက်ပါတယ်။ pi ကတော့ floating point value ကို assign လုပ်လိုက်တာပေါ့လေ။

ဒီက Bro တွေကိုတစ်ခုစဉ်းစားခိုင်းမယ်နော်  
Python Interpreter မှာ အောက်ကလိုရိုက်ကြည့်ပါ။

**Code:**

```
>>> print 02132
```

ဘာနံပါတ်ထွက်လာလဲဆိုတာရိုက်ပြီးရင်တွေ့ ရမှာပါ။ အဲ့တာဘာလို့ လဲဘာအတွက်အကြောင့်အခုလိုဖြစ်တယ်ဆိုတာကို PM ပို့ ပြီးဖြေကြည့်စေချင်ပါတယ်။

(Hint: try to print 01, 010, 0100, 01000)

အဖြေကိုတော့ဒုတိယပိုင်းကြရင်ဂျွန်စော်ပြောပြပေးပါမယ်။ ဒီကြားထဲမှာတော့ဖြေကြည့်ပေးစေချင်ပါတယ်။

 FIGHTINGGGG!!!! 

### 1.3 Variable names and keywords

Programmers တိုင်းကသူတို့ ရဲ့ Variable names တွေကို ပြုလုပ်မယ့် function ဒါမှမဟုတ်သက်ဆိုင်ရာနာမည်မျိုးတွေပေးတတ်ကြပါတယ်။ Variable တစ်ခုရဲ့နာမည်တွင် Letters ကော Numbers တွေကောပါဝင်နိုင်ပါတယ် ဒါပေသိမှတ်ထားရမှာလေးက Variables တွေသည် Letters ဖြင့်သာစရေးရမှာဖြစ်ပါတယ်၊ Number နဲ့ စလို့ မရပါဘူး။ Variable Name တစ်ခုရဲ့ Letter ကို Uppercase နဲ့ စရေးနိုင်ပေမယ့်ပုံမှန်ဆိုရင်တော့ lowercase နဲ့ စတာအသင့်တော်ဆုံးပါပဲ။

နောက် name ထဲမှာ underscore character ( \_ ) တွေပါဝင်နိုင်ပါတယ်။ Variable Words တွေရဲ့ကြားမှာ space ခြားခွင့်မရှိပါဘူး။ Interpreter မှာဒါလေးရေးကြည့်ပါ။

#### Code:

```
>>> Hades y2k = 1
```

သေချာပါတယ် Error တက်လာမှာပါ။ ဒါဆိုဘယ်လိုရေးမလဲ Hades\_y2k လို့ သုံးရင်အဆင်ပြေပါတယ်။ ဒါလေးတွေလဲစမ်းရေးကြည့်ကြပါ။

#### Code:

```
>>> 4KumA = 'Akuma mean Demon in Japanese'
>>> minn@ = 'Everyone'
>>> finally = 'This is final'
```

အကုန်လုံး invalid syntax ဆိုပြီး Error တက်လာမှာပါ။ ပြောစရာဆိုလို့ နောက်ဆုံးကောင်ပဲ finally ကဘာလို့ Error တက်ရတာလဲပေါ့ သူက letter နဲ့ လဲစတာပဲကို.....။

တခြားမဟုတ်ပါဘူး finally သည် python ရဲ့ keyword ဖြစ်နေလို့ပါ။ keywords တွေကိုလဲ variable name တွေအဖြစ်ယူသုံးလို့ မရပါဘူး။ Python 2 တွင် Keywords 31 ခုရှိပါတယ်။ အဲတာတွေကတော့

## Python 2 has 31 keywords:

and	del	from	not	while
as	elif	global	or	with
assert	else	if	pass	yield
break	except	import	print	
class	exec	in	raise	
continue	finally	is	return	
def	for	lambda	try	

In Python 3, `exec` is no longer a keyword, but `nonlocal` is.

### 1.4 Operators and operands

Operators ဆိုတာကတကယ်တော့ Special Characters တွေပါပဲ ဘာလုပ်ဆောင်တာလဲဆိုတော့ Addition, Multiplication ပေါင်းနုတ်မြောက်စား ဒါကိုဆိုလိုတာပါ။ ဒီ Mathematical Operations လုပ်ဆောင်ဖို့ Assign လုပ်လိုက်တာတွေကို operands လို့ ခေါ်တာပါ။  
 + ကတော့လူတိုင်းသိတဲ့အတိုင်းအပေါင်း၊ - က subtraction ခေါ်အနုတ်၊ Multiplication လို့ ခေါ်တဲ့အမြောက်ကတော့ \* နောက် အစားလို့ ခေါ်ကြတဲ့ Division ကတော့ / နဲ့ Exponentiation (မြန်မာလိုမပြန်တတ်တော့ဘူး :3 ) ကတော့ \*\* ပါ။  
 တချို့ Language တွေမှာတော့ Exponentiation ကို ^ နဲ့ သုံးပါတယ်။ ဒါပေမဲ့ Python မှာတော့ ^ သည် XOR ဆိုတဲ့ Bitwise Operator တစ်ခုပါ။

ဒါဆိုဥပမာလေးတွေကိုသွားလိုက်ရအောင်ဗျာ။ Interpreter ခေါ်လိုက်ပါ။ ပြီရင်အောက်ကလိုရေးလိုက်ပါ။

#### Code:

```
>>> first_number = 2
>>> second_number = 10
>>> first_number/second_number
>>> 0
```

ဟိုက်လာစားပြီ။ 2 ကို 10 နဲ့ စားတာဘာလို့ Decimal Number ဆိုတဲ့ 0.2 ခေါ် 1/5 Floating Point မပြရတာပဲနဲ့ 0 ဆိုတဲ့ integer value လာပြရတာလဲ။ အစမှာထဲက floating point မယူခွဲလို့ အဖြေကိုလဲ fraction ပိုင်းကိုဖြတ်ပြီး 0 ယူလိုက်တာပါ။ အစမှာထဲက interger အစား 2 ကိုဖြစ်ဖြစ် 10 ကိုဖြစ်ဖြစ် floating point 2.0 or 10.0 လို့ ပြောင်းရင်အဆင်ပြေသွားမှာပါ။ ဒါကလဲသတိထားစရာလေးတစ်ခုပါ။

### 1.5 Expressions and statement

Values, Variables နဲ့ Operatos တွေကိုပေါင်းထားတာကို Expression လို့ ခေါ်ပါတယ်။ Statement ဆိုတာကတော့ Python interpreter က execute လုပ်နိုင်တဲ့ code ကိုပြောတာပါ။ Statement ဆိုတာ၂မျိုးရှိပါတယ်။ Print နဲ့ Assignment ပါ။

အဲဒါကယ်တော့ Expression ဆိုတာကလဲ statement ပါပဲ တစ်ခုပဲကွာတာပါ။ Expression တွင် value ရှိပါတယ် ဒါပေမယ့် statement မှာတော့မရှိပါဘူး။

### 1.6 Interactive mode and script mode

Python ကဲ့သို့ Interpreted language က C တို့ လိုတွေထက်အားသာချက်ကတော့ interactive mode မှာ code လေးတွေကိုတစ်ခုချင်းစီ မီးနှိုင်းတာပါပဲ။

ဣန္ဒာဏီတို့ Python ကို Calculator လို့ သဘောထားကြည့်လိုက်စာ...

#### Code:

```
>>> miles = 26.2
>>> miles * 1.61
42.182
```

ပထမလိုင်းက miles ဆိုတဲ့ variable ကို 26.2 ဆိုတဲ့ floating value assign လုပ်ပါတယ်။ ဒုတိယလိုင်းက expression ကို execute လုပ်ပါတယ်ပြီးတော့ result ကိုထုတ်ပြပါတယ်။

ဒါကို script mode မှာ ချရေးရင်ဘာမှမြင်ရမှာမဟုတ်ပါဘူး။ Script mode မှာဆို expression ကဘာမှပြန်မဖော်ပြပါဘူး။ ပြောရရင် Python ကို expression ကို execute လုပ်ပါတယ် သို့သော်သင်ကမဖော်ပြခိုင်းသရွေ့ သူကလဲ Result ကိုဖော်ပြမှာမဟုတ်ပါဘူး။

Script mode မှာဆိုအောက်ကလိုရေးရမှာပါ။ ပြီးရင်တော့ .py ဆိုတဲ့ extension နဲ့ save ပြီး cmd/terminal ကနေ run ကြည့်ရမှာပေါ့။

#### Code:

```
miles = 26.2
print miles * 1.61
```

Script တစ်ခုဆိုတာတကယ်တော့ statements တွေပေါင်းစည်းထားတာပါပဲ။ statement ကတစ်ကြောင်းထက်ပိုနေရင်လဲ ဖော်ပြရမဲ့ results တွေကိုတပြိုင်ထဲဖော်ပြပါတယ်။

#### Code:

```
print 1
x = 2
print x
```

ဒါဆိုရင် 1 ကော x value ကောကိုတပြိုင်ထဲဖော်ပြသွားမှာပါ။

### 1.7 Order of operations

Expression ထဲမှာ operators တွေသည်တစ်ခုထက်ပိုနေရင် order of evaluation က precedence rules အတိုင်းပဲလုပ်ဆောင်သွားမှာပါ။

သင်္ချာမှာသင်ခဲ့ရသလိုဆို + နဲ့ \* ကို \* ကိုအရင်လုပ်ဆောင်သွားမှာပါ။ ဒါမှီပေါ့...။

- Parentheses ဆိုတာကိုအသုံးပြုပြီးတော့ သင်လုပ်ဆောင်စေချင်တဲ့ Expression ကိုသင့်စိတ်ကြိုက် order ပေးလုပ်ဆောင်နိုင်ပါတယ်။

ဥပမာ..

3 \* ( 4 + 1 ) ဆို 15 ထွက်လာမှာပါ။ ငယ်ငယ်ကသင်ခဲ့တဲ့သင်္ချာသဘောတရားအတိုင်းပါပဲ။ Parentheses တွေကိုအသုံးပြုခြင်းအားဖြင့် ကြည့်

ရတာရှင်းလင်းလာစေပါတယ်။

- Exponentiation ဆိုတဲ့ `**` က လဲ highest precedence ဖြစ်ပါတယ်။ `2 ** 1 + 1` ဆို 3 ထွက်မှာပါ။ 4 မထွက်ပါဘူး။ `3 * 1 ** 3` ဆို 3 ပါ 27 မဟုတ်ပါဘူး။
  - Same precedence ရှိတဲ့ Operators တွေဆို Left to Right အလုပ်လုပ်ပါတယ်။ (Exponentiation ကလွဲရင်ပေါ့)
- ဒါကအလွတ်ကျက်နေစရာမလိုပါဘူး မသဲကွဲဘူးဆိုရင် Parentheses ကိုအသုံးပြုပြီး operators တွေကိုစမ်းသွားလို့ ရပါတယ်။

### 1.8 String Operation

ပုံမှန်အရဆို string တွေကို mathematical operations တွေပြုလုပ်ခွင့်မရှိပါဘူး။ ဥပမာပေးရမယ်ဆို `'3' - '1'` ပြီးတော့ `'hello'/'world'` + ဆိုတဲ့ Operators ကတော့ strings တွေနဲ့ အလုပ်လုပ်ပါတယ် ဒါပေသိ Addition ဆိုတဲ့ပေါင်းတာမဟုတ်ပါဘူး Strings တွေကို concatenation ပြုလုပ်ပေးတာပါ။ တနည်းပြောရရင်ဆက်ပေးတယ်ပေါ့ဗျာ။

#### Code:

```
first = 'Hello'
second = 'World'
print first + second
```

Hello World လို့ ပြမှာဖြစ်ပါတယ်။ နောက် + လိုပဲ multiplication ဖြစ်တဲ့ \* ကလဲ string တွေနဲ့ အလုပ်လုပ်ပါတယ်။ ဒါပေသိ string မှာ တော့ multiply လုပ်တာမဟုတ်ပါဘူး repeat လုပ်တဲ့သဘောတရားပါ။ ပြောရရင် `'Hello' * 4` ဆိုတာက `'Hello' + 'Hello' + 'Hello' + 'Hello'` နဲ့ တူတူပါပဲ။

### 1.9 Comments

Program တစ်ခုရေးသားရာမှာ Comments တွေကအရမ်းကိုအရေးကြီးလှပါတယ်။ Source Code Sharing လုပ်ရာမှာ မိမိရေးထားတဲ့ functions တွေရဲ့အလုပ်လုပ်ပုံကို human language နဲ့ ဘေးမှာရေးသားတာမျိုးတွေပါ။

Python မှာတော့ Comments ကို `#` နဲ့ စရေးပါတယ်။ `#` နောက်ကရေးသမျှတွေသည် Program အလုပ်လုပ်ရာမှာထည့်ရေးမှာမဟုတ်ပါဘူး။

ဒီနေ့ အတွက်ကတော့ဒီလောက်ပါပဲ။ Python လေ့လာရတာကိုမျှော်စွဲ မယ်လို့ ယူစပါတယ်။ အပေါ်မှာစဉ်းစားခိုင်းခဲ့တဲ့ print 010 ဆိုရင်ဘာ လို့ တခြားဟာထွက်တယ်ဆိုတာကိုလဲစဉ်းစားပြီး PM ပေးကြပါအုံးဗျာ။ နောက်ညီကိုတို့ အမြင်လေးတွေလဲ Comment ရေးပေးပါအုံးဗျာ။  
Ref: How to be a Computer Scientist.

Enjoy the Beauty of Python