

# Let's Write Python

## [3]

အပိုင်းခုကိုမှာတော့ Tuple အကြောင်းကိုလေ့လာကြပါမယ်။

### Tuples

#### 3.1 Tuples are immutable

အရင်ကလဲကွန်စတန်ပြောခဲ့ပါတယ်။ Tuples တွေကို constant ဖြစ်စေခွင့်တဲ့ Sequence တွေမှာသုံးကြပါတယ်။ Tuple ဆိုတာဟာလဲ Values တွေကို Sequence လိုက်သတ်မှတ်ထားတာပါပဲ။ Values တွေကကြိုက်တဲ့ Type ဖြစ်နိုင်ပါတယ် နောက် Lists တုံးကလိုပါပဲ integers နဲ့ ပဲ indexed ထားတာပါ။ Lists နဲ့ တူတဲ့အချက်တွေပေါ့လေ မတူတာဆိုလို့ Mutable မဖြစ်တာပါ။ Tuples တွေကိုရေးရင် parentheses မပါပဲ comma (,) ကုဒ်ပြီးရေးလိုက်တာနဲ့ တင် tuple ဖြစ်ပါတယ်။

#### Code:

```
>>> a = 'Hello',
```

ဒါပေသိအကုန်ဖြစ်ပြီး ကြည့်ရတာရှင်းရှင်းလင်းလင်းဖြစ်အောင်တော့ထည့်ရေးပေးပါ။

#### Code:

```
>>> a = ('Hello', 'World', 1)
>>> type(a)
<type 'tuple'>

>>> type(a[0])
<type 'string'>
>>> type(a[2])
<type 'int'>

>>> b = tuple('MSF')
>>> b
```

Lists မှာတုံးကပြောခဲ့တဲ့ indexing နဲ့ slicing က tuple မှာလဲအလုပ်လုပ်ပါတယ်။

#### Code:

```
>>> a
('Hello', 'World', 1)
>>> a[2]
1

>>> a[:2]
('Hello', 'World')
```

Tuple တွေသည် immutable ဖြစ်တဲ့အတွက် modify လုပ်လို့ မရပါ ဒါပေသိ replace လုပ်လို့ ငှောရပါတယ်။

#### Code:

```
>>> a[2] = 2
TypeError: 'tuple' object doesn't support item assignment

>>> a = tuple('MSF')
>>> a
('M', 'S', 'F')
```

### 3.2 Tuple Assignment

Variable သုထက်ပိုတာကို တပြိုင်ထဲ assign လုပ်ခွင် comma ကုဒ်ပြီးတွဲပေးသွားလို့ ရပါတယ်။ ဒါပေသိ left hand side နဲ့ right hand side က အရေအတွက်တူညီရမှာပါ။

#### Code:

```
>>> a, b = 1, 2
>>> c, d = 1, 2, 3
ValueError: too many values to unpack

>>> address = 'helloworld@python.org'
>>> username, domain = address.split('@')
```

အရင်က split() ဆိုတာကိုသုံးပေးခဲ့ဖူးပါတယ်။ သဘောပေါက်မယ်ထင်ပါတယ်။

### 3.3 Tuples as return values

Function တစ်ခုဟာ Value တစ်ခုထဲကိုပဲ Return ပြန်နိုင်ပါတယ်။ ဒါပေမဲ့ Value က tuple ဖြစ်နေရင် ဒါဟာ multiple values ကို return ပြန်ပေးသလိုပါပဲ။ သင်္ချာမှာ Quotient (စားလဒ်) နဲ့ Remainder (အကြွင်း) ဆိုတာကြားဖူးမယ်ထင်ပါတယ်။ ဒီအတွက် Python မှာ divmod()

ဆိုတဲ့ built-in function တစ်ခုရှိပါတယ်။ သူက argument ၂ခုကိုယူမယ်နောက်ပြီးရင် quotient နဲ့ remainder ကို tuple အဖြစ်နဲ့ ပြန်ပေးပါတယ်။

ဂျွန်တော်ပြောတာထက်ပိုရှင်းရှင်း Python interpreter မှာ help('divmod') လို့ ရိုက်ကြည့်လိုက်ပါ။ သူတွက်ချက်တဲ့ formula လေးကို x နဲ့ y အနေနဲ့ ဖော်ပြပေးထားမှာပါ။

#### Code:

```
>>> a = divmod(9, 3)
>>> a
(3, 0) #first argument is quotient and second argument is remainder
```

### 3.4 Lists and tuples

Python မှာ zip() ဆိုတဲ့ built-in function လေးတစ်ခုရှိတယ်။ သူက ၂ခုအိမ်မဟုတ် ၂ခုထက်ပိုတဲ့ sequence တွေကို tuple တစ်ခုချင်းစီနဲ့ List ပြုလုပ်ပေးပါတယ်။ လည်သွားပြီလားမသိ... ဥပမာလေးကြည့်လိုက်ရအောင်...

#### Code:

```
>>> a = 'abc'
>>> b = [1, 2, 3]
>>> zip(a, b)
[('a', 1), ('b', 2), ('c', 3)]
```

ရှင်းတယ်မလား ဒါဆိုဆက်မယ် အကယ်လို့ zip ထဲထည့်မယ့် argument ၂ခုက length မတူရင်ကော 🙄

#### Code:

```
>>> c = 'Hello'
>>> d = 'MSF'
>>> e = zip(c, d)
>>> e
[('H', 'M'), ('E', 'S'), ('L', 'F')]
```

မြင်တယ်မလား။ Error မတက်ပါဘူး။ သူမှာရှိသလောက်နဲ့ ပဲတွဲပေးသွားမှာပါ။ ဒါဆိုအဲ့ဒီ e ဆိုတာကို for loop ပတ်ပြီးတော့ တွဲရက် argument တွေကိုတစ်ကြောင်းချင်းဖော်ပြမယ့် function ရေးမယ်ဗျာ။

#### Code:

```
>>> for x, y in e:
...     print x, y
...
H M
e S
1 F
```

### 3. 5 Comparing Tuples

Tuple တွေကို compare လုပ်မယ်ဆိုရင် left hand side နဲ့ right hand side ရှိတယ်ဆိုပါစို့။ LHS ဘက်က first element ကို RHS ဘက်က first element နဲ့ ကိုစပြီး compare လုပ်ပါတယ်။ equal ဖြစ်တယ်ဆိုမှနောက် element ကိုသွားပြီး differ ဖြစ်တဲ့အထိဒီအတိုင်းလုပ်ဆောင်သွားပါတယ်။ Subsequence တွေကိုတော့ထည့်မတွက်ပါဘူး။

#### Code:

```
>>> (0, 1, 2) < (0, 3, 4)
True
>>> (0, 1, 10) < (0, 3, 4)
True
>>> (0, 10, 2) < (0, 3, 4)
False
>>> (10, 1, 2) < (0, 3, 4)
False
```

sort function ကလဲအဲ့သလိုပဲပါ first element ကိုစလုပ်ပါတယ်။ အကယ်လို့ တူနေခဲ့ရင် second element နဲ့ sort သွားပါတယ်.... နောက်ပြောစရာရှိတာလေးကတော့ DSU patten ပေါ့။ အရှည်ခေါက်ကတော့ Decorate, Sort and Undecorate

Decorate = အရင်ဆုံး sequence ကို tuples ပြောင်းနောက်သက်ဆိုင်ရာ sequence ထဲက element ရှိသက်ဆိုင်ရာ key နဲ့ အတူ List ထဲ ထည့်လိုက်ပါတယ်။ ပြောရရင် List of tuples ပေါ့။

Sort = အဲ့ List ထဲက tuple တွေကို sorting လုပ်ပါတယ်။

Undecorate = နောက် sorted လုပ်ထားတဲ့ sequence ကို extract ပြန်လုပ်ပါတယ်။

အောက်ကဥပမာလေးကြည့်လိုက်ရအောင်ပိုရှင်းအောင်ပေါ့ သူက ထည့်လိုက်တဲ့ string ကို List အနေနဲ့ ပြောင်းပြန်ပြပေးမှာပါ။

#### Code:

```
def sortby_length(x):
    a = []
    for word in x:
        a.append((len(word), word)) # Decorate
    a.sort(reverse=True) # Sort
    res = []
    for length, word in a:
        res.append(word)
    return res # Undecorate
```

### 3.6 Sequence of sequences

Strings, lists နဲ့ tuples တွေကိုလိုအပ်တဲ့နေရာမှာသင့်တော်သလိုအသုံးပြုကြပါတယ်။ String ကနေစပြောရမယ်ဆို strings တွေဟာ immutable ဖြစ်ပါတယ်နောက် elements တွေဟာလဲ character တွေဖြစ်ကိုဖြစ်ရပါမယ်။ အကယ်လို့ new strings လဲမလုပ်ခင်ပဲ characters တွေကို changes ပြုလုပ်မယ်ဆို List ကိုအသုံးပြုရပါမယ်။

Lists တွေက tuples တွေထက်ပိုတော့ ရတတ်ပါတယ်။ mutable ဖြစ်တာကလဲတစ်ပိုင်းပေါ့ ဒါပေသိ တချို့အခြေအနေမျိုးတွေမှာ Tuples တွေကိုအသုံးပြုရမယ့်အခါမျိုးတွေရှိတတ်ပါတယ်။ ဘယ်လိုအချိန်အခါမျိုးလဲဆိုတော့.....

၁. Return statement လိုနေရာမျိုးမှာဆို List တစ်ခုလုပ်ရတာထက် Tuple ပဲအသုံးပြုတာကပိုအဆင်ပြေပါတယ် သို့သော် exception ဝေးတွေတွေရှိတာပေါ့..

၂. Dictionary အကြောင်းတော့မပြောရသေးဘူးပေါ့... (အကြမ်းဖျင်းပြောရမယ်ဆို Dictionary မှာ keys and values ဆိုပြီးရှိပါတယ်) ဒါ့ဒီ Keys တွေကိုတော့ constant ဖြစ်ဖို့ အတွက် immutable types တွေဖြစ်တဲ့ strings ဒါမှမဟုတ် tuples ကိုပဲအသုံးပြုရပါတယ်။

၃. Function ထဲကို sequence တစ်ခုအား argument အနေနဲ့ ထည့်တဲ့အခါနဲ့မှာ tuples ကိုအသုံးပြုတာက aliasing မှာ error ဖြစ်နိုင်ခြေနည်းသွားစေပါတယ်။

ဒါပေသိ Tuples တွေက immutable ဖြစ်တဲ့အတွက် Lists တွေလို sort တို့ လို modify လုပ်တဲ့ကောင်တွေတော့မရဘူးပေါ့။ ကြိုတင်ပြောသွားအုံးမယ် sort() က ငယ်စဉ်ကြီးလိုက်စီသလို ကြီးစဉ်ငယ်လိုက်ဆို reverse() ကိုသုံးနိုင်ပါတယ်။ ဒါပေသိ Python မှာ Tuple ကို List လို sort တို့ reverse တို့ လုပ်နိုင်အောင် built-in function လေး ၂ခုရှိပါသေးတယ်။ ဒါ့တာကတော့ sorted() နဲ့ reversed() ပါ။

မတူဘူးနော်... အခုဟာက ed နဲ့ ဆုံးတာနော်။ နောက်အဲ့ကောင်တွေက list အနေနဲ့ ပြန်တာပါ ဒီတော့ tuple ပြန်ဖြစ်စေခွင့် tuple() ဆိုတဲ့ function နဲ့ တွဲသုံးရမှာပါ။ ဥပမာလေးရေးပြပါမယ်။

#### Code:

```
>>> a = (1, 3, 2)
>>> a = tuple(sorted(a))
>>> a
(1, 2, 3)
```

Tuple အကြောင်းကတော့ဒီလောက်ပါပဲ သိပ်ပြောစရာမရှိပါဘူး။ နောက်တစ်ခန်းကွာရင်တော့ကွန်တို Dictionary အကြောင်းကိုလေ့လာကြကပါမယ်။