

Range

`range()` function က နံပါတ်တွေကို အစဉ်လိုက်ထုတ်ပေးတဲ့ function ဖြစ်ပါတယ်။

```
test.py x
1 numbers = list(range(10))
2 print(numbers)

Shell x
>>> %Run test.py
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>>
```

line 1 မှာ ရေးထားတဲ့ကုဒ်က 0 ကနေ 10 အထိ နံပါတ်တွေကို ထုတ်ပေးမှာပါ။ သတိထားရမဲ့အချက်က *range* ထဲမှာ ကိုယ်ရချင်တဲ့ကိန်းဂဏန်းတစ်လုံးပဲ ထည့်ရေးရင် 0 ကနေစပြီး ထည့်ပေးလိုက်တဲ့ ဂဏန်းအထိ နံပါတ်စဉ်ထုတ်ပေးမှာပါ။ 0 ကနေ စပြီးလိုချင်တာမဟုတ်ဘူးဆိုရင် () ထဲမှာ argument တန်ဖိုး နှစ်ခုထည့်ပြီး အောက်ပါပုံအတိုင်း ရေးသားနိုင်ပါတယ်။

```
test.py x
1 numbers = list(range(3, 8))
2 print(numbers)
3
.

Shell x
>>> %Run test.py
[3, 4, 5, 6, 7]
>>> |
```

သတိပြုရမဲ့အချက်က *range* ဟာ ပထမဂဏန်းကနေ စတင်ဖော်ပြပေးပြီး ဒုတိယဂဏန်းအထိ မရောက်ဘူးဆိုတာပဲ ဖြစ်ပါတယ်။ ပုံမှာ နံပါတ် 3 ကနေစတင်ပေးပြီး နံပါတ် 7 မှာ ရပ်သွားပါတယ်။ list ထဲမှာပါတဲ့ဂဏန်းကို ရေတွက်ကြည့်ရင် စုစုပေါင်း ၅ လုံး ရှိတာကိုတွေ့ရမှာပါ။ (8-3 = 5) ဖြစ်ပါတယ်။

range ဖန်တီးတဲ့အခါမှာ **list** ထဲကိုထည့်ပေးမှသာ အခုလို မြင်ရမှာပါ။ **list** ထဲကိုမထည့်ပေးရင် **range** တန်ဖိုးအတိုင်းရှိနေမှာ ဖြစ်ပါတယ်။ အောက်ပါပုံကို ကြည့်ပါ။

```
test.py x
1 number_1 = range(10)
2
3 number_2 = range(3, 8)
4
5 print(number_1)
6 print(number_2)
7

Shell x
>>> %Run test.py
range(0, 10)
range(3, 8)
>>> |
```

Range တန်ဖိုးတွေ ဖန်တီးတဲ့အခါမှာ 1,2,3,4 အစဉ်လိုက်မဟုတ်ဘဲ ကိန်းဂဏန်းအရေအတွက် တစ်ခုနဲ့ တစ်ခု ခြားပြီး ဖန်တီးနိုင်ပါတယ်။

```
test.py x
1 numbers = list(range(5, 10, 2))
2 print(numbers)
3

Shell x
>>> %Run test.py
[5, 7, 9]
>>> |
```

range (5, 10, 2) မှာ range (5, 10) က ပုံမှန် 5,6,7,8,9 အထိ တန်ဖိုးတွေ ဖန်တီးတာဖြစ်ပါတယ်။ သို့ပေမဲ့ နောက်မှာ 2 ကို ထည့်ထားပြီး range(5, 10, 2) လို့ ရေးထားတဲ့အတွက် အစဆုံးဂဏန်းကနေ 2 ခြားပြီးမှ ရလာတဲ့ဂဏန်းတွေကိုပဲ range တန်ဖိုးတွေအဖြစ် ဖန်တီးမှာဖြစ်ပါတယ်။ 5, 7, 9 ဆိုပြီး ရလာပါတယ်။ နမူနာပြခဲ့တဲ့ 2 ရဲ့ နေရာမှာ အခြားဂဏန်းတွေနဲ့ ဖန်တီးနိုင်ပါတယ်။ အနှုတ်တန်ဖိုးကိုလည်း ထည့်နိုင်ပါတယ်။

```
test.py x
1 numbers = list(range(20, 5, -2))
2 print(numbers)
3

Shell x
>>> %Run test.py
[20, 18, 16, 14, 12, 10, 8, 6]
>>> |
```

`range` နဲ့ တွဲဖက်ပြီးအသုံးပြုလေ့ရှိတာက `for` loop ပတ်တဲ့အခါမှာ ဖြစ်ပါတယ်။ `for` loop ကို မိမိလိုချင်တဲ့အကြိမ်ရေအတိုင်း ပတ်စေချင်တဲ့အခါမှာ အကြိမ်ရေအတွက်သတ်မှတ်ဖို့အတွက် `range` ကို သုံးလေ့ရှိပါတယ်။

```
test.py x
1 for i in range(5):
2     print("Hello!")
3

Shell x
>>> %Run test.py
Hello!
Hello!
Hello!
Hello!
Hello!
>>> |
```