



[Input] → f (param1, param2, param3) → [Output]

"function run / input into the function" → set params

Result → results = function_name (value1, value2, value3)
even if return None | * keyword argument
function_name (param1=value1)

↳ Case: function return multiple values

① results = function_name()
(value1, value2, value3) ↗
Tuple ≈ list

↳ results = (value1, value2, value3)
↳ results[0] → index → results[0]

② var1, var2, var3 = function_name()
(value1, value2, value3)

↳ var1, var2, var3 are assigned to value1, value2, value3

Định nghĩa của một function

input → f(x) → output
↑ Arguments ↑ return

* function & arguments (input) ⇒

* function & return (return) ⇒

Ex 1
- function definition
↳ argument: Nut

↳ return: Nut

def print_name():
 print("Nut") ↗
 ↗ input
print_name() ↗
 ↗ return
 ↗ input

↳ return: `Yield`

Ex 2 (in function)
function from
input from user

- arguments: y_2
- return: y_1

Ex 3

- function សំណួល
នៅក្នុងនៃនាមីតិ៍
បានបាននា
- arguments: ចំណាំ
- return : រាយក្រឹង

Ex 4

- function បន្ថែម
នូវវិធាន សម្រាប់
នាំវ
- arguments : 20%
- return : 1 នូវ

~~EXS:~~

- function
 - gives out a value
 - function
 - w, h
- arguments: 2 or
 (w, h)
- return: 2 or
area, perimeter

```

def add_print(a3, b5):
    total = a+b = 8
    print(total)
add_print(3, 5)
    arguments
    ↗
    ↘ [8]
  
```

```
def add(a,b):  
    total = a+b  
    return total  
  
result = add(3,5)  
print(result) → 8
```

```

def rectangle_properties(w, h):
    area = w * h = 50
    perimeter = 2 * (w + h) = 30
    return area, perimeter

width = 5
height = 10
a, p = rectangle_properties(width, height)
print(f"area = {a}, peri = {p}")

```

area, perimeter | $a_{\text{rect}} = \text{rectangle_properties}(\text{width}, \text{height})$
 $\text{print}(f"\text{Area} = {a} \text{, peri} = {p}")$

Main function

- function works like a box
- can call functions
- can return values

```
✓ def msg():
    print("Hi")
    msg()
msg()

def main():
    msg()
    msg()
main()
```

Ex 1 function to BMI + main function to run user

definition: 1 function = 1 job = 1 mini program

Running function

① BMI_Cal(w,h)

- arguments: 2 float (w, h)
- return: 1 float

② main():

- arguments: None
- return: None
- takes from user!

```
def BMI_Cal(w, h):
    bmi = w / (h**2)
    return bmi
```

```
def main():
    weight = float(input())
    height = float(input())
    bmi = BMI_Cal(weight, height)
    print(bmi)
```

Ex 2

function
- user input list
our_sum

- function to mean
my list
our_mean

- function main
no return value

User enters
list

to sum

↳ calculate sum

```
def our_sum(nums):
    total = 0
    for num in nums:
        total += num
    return total
```

```
def our_mean(nums):
    mean = our_sum(nums) / len(nums)
    return mean
```

```
def main():
    n = int(input())
    my_list = []
    for i in range(n):
```

Note:
our_sum(nums)

↳ argument: 1 list
↳ return: 1 float

our_mean(nums)

↳ argument: 1 list
↳ return: 1 float

↳ return: 1 float

↳ សូន្យ

```

    ↳ បានរៀននាំ
    ↳ Main()
    ↳ my_list = []
    ↳ for i in range(n):
        ↳     my_list.append(int(input()))
    ↳ mean = our_mean(my_list)
    ↳ print(mean)
  
```

Local Variables

- ↳ អាជីវកម្មនៃ function នៅក្នុង function នឹងបាន
- ↳ តាមរយៈនូវ local variables

Global variables

- ↳ function នឹងការពារលើ គេហទ័រនៃខ្លួន
- ↳ បានបានការពារលើ global នូវការពារនៃខ្លួន
- ↳ ឬនៅក្នុងខ្លួនឯងបានការពារលើ global នូវ function
- ↳ នានាដែលនៅក្នុងខ្លួនឯងបានការពារលើ global នូវការពារ
- ↳ Syntax: global variable_name

```

    ➤ x = 5
  answrs
    def do_x():
        ↳ x = 7
    do_x()
    print(x)
  ➤ 5
  
```

9

```

    ➤ x = 5
  answrs
    def do_x():
        global x
        x = 7
    do_x()
    print(x)
  ➤ 7
  
```

ឧបសម្ព័ន្ត function នៃ
បច្ចនុយនុយនិយោជន៍
នូវ!

Library និង import function

- ↳ បានរាយការ "import" function នៃ បច្ចនុយនិយោជន៍
ឬ library /

- ↳ បើរាងការ "import" ទូទៅ ^{ស្ថាបន្ទាន់} ↳ សម្រាប់ library / module
 - ↳ Syntax: import library_name
 - ↳ ចាប់បូន្មាន function: library_name.function-name()
-

random library

- ↳ នីមួយៗនៃវិធាននេះ គឺជាពុំ
- ↳ randint(start, end)
- ↳ ជីវិត នៅលើក នឹងបង្ហាញការស្វែងរករាយ (start, end) (equal probability)
- ↳ ex: randint(1, 5) →

1	2	3	4	5
P:	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$

ចាប់ពី end ហើយ

~~Ex 1~~ import random
 result = random.randint(1,5)
 print(result)

~~Ex 2~~  # ធម៌លូកដោា n គ្រឿង លូកដោា => 1 - 6
 import random
 n = int(input('Enter n: '))
 for i in range(n):
 die = random.randint(1,6)
 print(die)

→ Enter n: 5
 4
 1
 4
 2
 3

Seed

- ↳ នៅក្នុងការ set នឹងបានបង្កើតឡើង

- ↳ $\text{d}_{\text{f}} \text{ set } \text{d}_{\text{f}}$ m_{f} u_{f} v_{f} w_{f} x_{f} y_{f}
- ↳ d_{f} m_{f} u_{f} v_{f} w_{f} x_{f} y_{f} \rightarrow d_{f} m_{f} u_{f} v_{f} w_{f} x_{f} y_{f} z_{f}
- ↳ Syntax: random, seed (optional)

```
[23] # ຄອຍລູກເຕົາ ກ ຄັ້ງ ລູກເຕົາ => 1 - 6 [24] # ຄອຍລູກເຕົາ ກ ຄັ້ງ ລູກເຕົາ => 1 - 6
✓ 1s ✓ 10s
import random
random.seed(777) Set ທີ່ກ່ອນກຳນົດ
n = int(input('Enter n: '))
for i in range(n):
    die = random.randint(1,6)
    print(die)

[23] Enter n: 5
2
4
4
3
5
[24] Enter n: 5
2
4
4
3
5
ທີ່ກ່ອນກຳນົດ
```

Math library

Log & math functions

↳ Syntax: import math

functions

math. $\sin(x)$

math. cos(x)

math.tan(x)

math. $\exp(x) \rightarrow e^x$

math. pow(x,y) $\rightarrow x^y$

math.floor(x) → ⌊x⌋

`math.ceil(x) → Fx7`

math. $\log(x)$

math, $\sqrt{x} \rightarrow x$ $\rightarrow (x) \approx 0.5$

Ex import math

$$x = 10$$

10

Y = main, struc, etc (var)

print(y)

Constant

Math.e → 2.71...

(924

```
e_value = math.e
```

$\text{math.e} \rightarrow 2.71\dots$

$\text{math.pi} \rightarrow 3.141\dots$

```
import math
```

```
e_value = math.e  
pi_value = math.pi  
print(e_value, pi_value)
```

```
→ 2.718281828459045 3.141592653589793
```

```
▶ import math  
r = 10  
area = math.pi*r**2  
print(area)
```

```
→ 314.1592653589793
```