Programming Fundamental A Hitchhiker Guide to Coding with Python

Lesson 5: Functions

Akarate Singta

Department of Mathematics
Faculty of Science and Technology, RMUTT

Lesson Outline

The concept of function

User-Defined Functions

Section Inputs and Outputs

The concept of function



Image Source: John Sturtz,

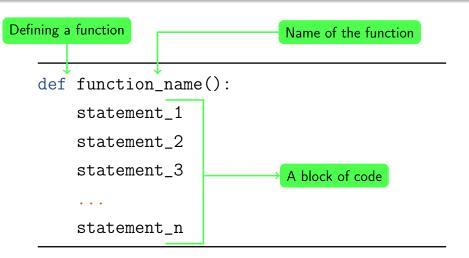
- A function is a block of code that only runs when called.
- Like a sub-program, functions can accept inputs and provide outputs.
- A function often only accomplishes one certain task.
- Functions increase the reusability and modularity of code.

Python Built-In Functions

Here are some functions you have already seen.

Function	Description
<pre>print()</pre>	Prints to the standard output device
input()	Allowing a user input and returning a string from it
<pre>int()</pre>	Return an integer number
float()	Return a floating-point number
str()	Return a string
len()	Returns the length of an object

User-Defined Functions



```
line_fn.py
```

```
def line():
        x = ' - ' * 30
        print(x)
 5
    x = 10
    print('x = ', x)
    line()
    x = x + 10
    print('x = ', x)
10
    line()
11
```

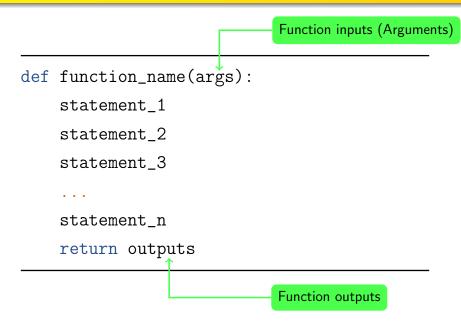
```
>_ python line_fn.py
x = 10
-----x = 20
```

```
hi_offer.py
```

```
def say_hi():
        print('Hello')
        print('Nice to meet you!')
4
    def offer_meal():
6
        print('Please have some meal')
        print('It is very good pasta')
10
    guest = 'Jame Doe'
    print('Here is', guest)
11
12
    say_hi()
    offer_meal()
13
```

```
>_ python hi_offer.py
Here is Jame Doe
Hello
Nice to meet you!
Please have some meal
It is very good pasta
```

Function Inputs and Outputs



```
line_fn.py
```

```
def line(symbol):
        x = symbol * 25
        print(x)
 5
    x = 10
    print('x = ', x)
    line('-')
    x = x + 10
    print('x = ', x)
10
    line('*')
11
```

```
line_fn.py
```

```
def line(symbol, n):
        x = symbol * n
        print(x)
5
    x = 10
    print('x = ', x)
    line('#', 15)
    x = x + 10
    print('x = ', x)
10
    line('0', 25)
11
```

```
>_ python line_fn.py
x = 10
############
x = 20
@@@@@@@@@@@@@@@@@@@@@@
```

```
line_fn.py
```

```
def line(symbol, n):
        x = symbol * n
        return x
    x = 10
    print('x = ', x)
    print(line('#', 15))
    x = x + 10
    print('x = ', x)
10
    print(line('0', 25))
11
```

```
>_ python line_fn.py
x = 10
############
x = 20
@@@@@@@@@@@@@@@@@@@@@@@
```

circle_fn.py

```
def circle_area(radius):
        pi = 3.14159265359
         area = pi * (radius ** 2)
        return area
    def line(symbol, n):
        x = symbol * n
        return x
10
    radius_list = [3.44, 1.56, 6.88]
11
12
    for r in radius list:
         print('Radius is', r)
13
         a = circle_area(r)
14
        print('Area is', a)
15
         print(line('-', 25))
16
```

```
>_ python circle_fn.py
Radius is 3.44
Area is 37.17635082552262
Radius is 1.56
Area is 7.645379881776625
Radius is 6.88
Area is 148.70540330209047
```

circle_fn.pv

```
def circle_cal(radius):
        pi = 3.14159265359
         area = pi * (radius ** 2)
         perimeter = 2 * pi * radius
        return area, perimeter
    def line(symbol, n):
        x = symbol * n
        return x
10
    radius_list = [3.44, 1.56, 6.88]
11
12
    for r in radius list:
         print('Radius is', r)
13
         a, p = circle_cal(r)
14
        print('Area is', a)
15
         print('Perimeter is', p)
16
         print(line('-', 25))
17
```

```
>_ python circle_fn.py
Radius is 3.44
Area is 37.17635082552262
Perimeter is 21.6141574566992
Radius is 1.56
Area is 7.645379881776625
Perimeter is 9.8017690792008
Radius is 6.88
Area is 148.70540330209047
Perimeter is 43.2283149133984
```

hi_offer.py

```
def say_hi(name):
        print('Hello', name)
        print('Nice to meet you!')
    def offer meal(name):
        if name == 'Lucy':
             print(name, 'needs no meal')
        else:
             print('Please have some meal')
10
             print('It is very good pasta')
11
12
    guests = ['James', 'Lucy', 'Susan']
13
    for g in guests:
14
        sav_hi(g)
15
        offer_meal(g)
16
```

```
>_ python hi_offer.py
Hello James
Nice to meet you!
Please have some meal
It is very good pasta
Hello Lucy
Nice to meet you!
Lucy needs no meal
Hello Susan
Nice to meet you!
Please have some meal
It is very good pasta
```

Exercise

Write a Python function absolute to find the absolute value |x| of a real number x, i.e.,

$$|x| = \begin{cases} x & \text{if } x \ge 0, \\ -x & \text{if } x < 0. \end{cases}$$

The function takes an argument (input), which is a real number, and returns an output of its absolute value.

Command	Result
absolute(5)	5
absolute(-4.234)	4.234
absolute(0)	0
absolute(-2.14e-3)	2.14e-3

Exercise

Write a Python function solve_quad to solve a quadratic equation $ax^2+bx+c=0$ where $a\neq 0$, with the formula

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

The function takes 3 arguments (inputs), which are the coefficients a,b and c in a quadratic equation, and returns outputs of the solutions to the equations.

Command	Result
solve_quad(1, -5, 6)	2.0 3.0
solve_quad(-2, 2, 1)	1.366 -0.366
solve_quad(1, 2, 1)	-1.0 -1.0
solve_quad(0, 5, 6)	None

Exercise

Write a Python function solve_poly to solve a polynomial equation

$$ax^2 + bx + c = 0.$$

The function takes 3 arguments (inputs), which are the coefficients a, b and c in the equation, and returns outputs of the solutions.

Command	Result
solve_poly(1, -5, 6)	2.0 3.0
solve_poly(-2, 2, 1)	1.366 -0.366
solve_poly(1, 2, 1)	-1.0
solve_poly(0, 5, 6)	-1.2