mindX
Technology & Startup School

Function, Scope and Modules

Lesson 8

TABLE OF CONTENTS

mindX
Technology & Startup School

01

Introduction to Functions

Characteristics of a function

$$f(x)$$

```
def heron_formula(a, b, c):
    s_p = (a + b + c) / 2
    return math.sqrt(s_p * (s_p - a)
                         * (s_p - b)
                         * (s_p - c))
```

A block of code which only runs when it is called

Can pass data, known as parameters, into a function

A function can return data as a result

Declaring and using a function

Function's arguments (parameters)
Optional
Separate by comma

Function's name

"def" keyword

Colon symbol

```
def greeting(name):
    print("Hello", name)

    # do other stuffs
```

Function's body

```
greeting("John") # Hello John
greeting("Kelly") # Hello Kelly
```

Calling the function and passing values

Properties of function

Passing value:

- You can pass a number, string, list...
- Positional arguments => **Order matters!!**
- Keyword Arguments

Make an argument optional:

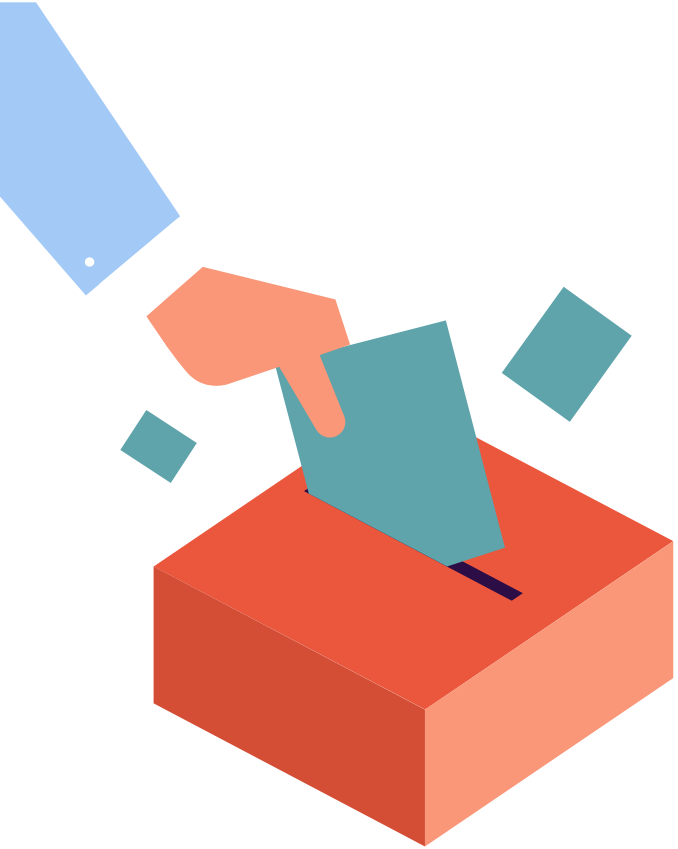
- Default value (must be one of the most left arguments)

Return value

```python
def area_of_triangle(width, height):
    return width * height * 0.5

print(area_of_triangle(5, 10)) # 25.0
print(6 * area_of_triangle(2, 4.5)) # 27.0
```

- To let a function return a value (number, string, list...), use the return statement
- If the function has a return statement in the middle of the body, it will exit the function, regardless of any code after it
- A function returning nothing, or don't return anything will have **None** value



mindX
Technology & Startup School

02

Scope of the variables

Scope of the variables

Any variable declared inside a function, will disappear when the function end

```
def function_a():
    # a is declared inside function_a
    a = 12

# a doesn't exists outside of function_a
print(a)  # Error: name ‘a’ is not defined
```

The *global* keyword

The *global* keyword allows you to modify the variable outside of the current scope.

It is used to create a global variable and make changes to the variable in a local context.

Rules of the *global* Keyword

When we create a variable inside a function, it is local by default.

When we define a variable outside of a function, it is global by default. You don't have to use *global* keyword.

We use global keyword to read and write a *global* variable inside a function.

Use of *global* keyword outside a function has no effect.

mindX
Technology & Startup School

03

Python Modules

What is a module?

Consider a module to be the same as a code library

A file containing a set of functions you want to include in your application

Creating and using a module

STEP 1

Save a function, variables, list,... in a another .py file, the filename will be module name

STEP 2

In the file which the module will be used, add
import *module_name*
To the top of the file

STEP 3

To use the module's function, variables,..; use the syntax module_name.name

Tips on modules

Module can contains lots of things

- The module can contain functions, as already described, but also variables of all types (arrays, dictionaries, objects etc)

Rename on import

- You can create an **alias** when you import a module, by using the as keyword
- E.g: import my_module as md

dir()

- The dir() function lists all the function names or variables of the module
- import platform print(dir(platform))

Some commons built-in modules

random

pseudo-random number generators for various distributions

math

gives developers access to the mathematical functions defined by the C standard

datetime

delivers tools that make it easier to work with **dates and times**

mindX
Technology & Startup School

THANKS!

See you in the next lesson!