Functions

Monday, December 13, 2021 3:24 PM

USER DEFINED FUNCTIONS

When programming, you will often find yourself in situations where you will have to perform operations repeatedly. You can write these operations over and over again but the optimal solution is to write them in a function which can be used as many times as you need.

To create and define a function in Python, we just use def followed by the name of the function with open and close parenthesis and a colon, with any arguments (inputs) within the parenthesis, see the example below.

```
In [ ]: def example_function():
```

For the first function we will simply pass. Pass is how we tell the Python interpreter to skip a line a do nothing.

Any code within a function must be indented because we group code blocks by indentation which we explored earlier, this next example shows this.

```
In [1]: def example_function():
            pass
```

To call a function we simply type the name of the function and empty parenthesis, example below:

```
In [2]: example_function()
```

This function when run returns nothing because we used pass.

If we trying to call the function without adding parenthesis, the Python interpreter will display that it is a function, see example:

```
In [3]: example_function
Out[3]: <function __main__.example_function()>
```

Next we can write a basic function that returns the string "Hello World".

We start again by typing def followed by the function name with parenthesis. This function will have zero argument but will return the value of "Hello World". We can do this by simply typing return followed by the value, one again we clal the function by typing the function name followed by parenthesis, see example:

```
In [4]: def hello_world():
            return "Hello World"

In [5]: hello_world()
Out[5]: 'Hello World'
```

As you can see the function printed the string "Hello World" however we can store the value in a variable should we desire. We can do this by simply typing a variable name followed by the assignment operator =, then the function return method, see the example where we assign the value stored in the function to a variable called x which we then print:

```
In [6]: x = hello_world()
        print(x)
Out[7]: 'Hello World'
```

Next we can create a function that is a bit more advanced, one that calculates the volume of a sphere. First we need to import the math module `import math` so we can use math.pi.

Like before we can create a function but this time pass a variable called r as an argument within the parenthesis. This will ask for an input when calling the function.

Next we can add a doc string to tell the user what the function does and then we can calculate the volume and save it into a variable called v, then print the variable, see the example below.

```
In [8]: import math

        def volume(r):
            """Returns volume of a sphere"""
            v = (4.0/3.0) * math.pi * r**3
            return v

        volume(2)
```

```
Out[8]: 33.510321638291124
```

Note that when calling the function that has an argument like mentioned earlier, we need to provide an input in the parenthesis otherwise the Python interpreter returns an error which informs you to use an argument when calling the function.