Functions in Python

- A function is a reusable block of code, which performs a specific task.
- It simply means - write once and utilise as much as required.
- Function provides readability, modularity, and reusability.

lets see how we define a function in python.

- We define a function in python using a keyword "def" followed by the function name and paranthesis.

For example:

```
                ↓ ← function name
keyword → def greet():
              print(F"Hellow! there") ← function body
```

Now let's see how to invoke this function.

- Invoking a function means, calling a function to perform the required task.

We simply call it as:

```
greet()
```

And it executes the code inside the function.

Now, that we have just printed something inside a function. Likewise, a function can return a value.

For example:

```
                ← function parameter.
def add(x, y):
    return x+y  ← return value.
```

Now here, When we invoke this function, it basically requires you to pass 2 arguments.
Basically the two numbers, which you want to perform addition of.

Lets invoke this function.

```
add(5,4)
```

Now just invoking the function isn't enough when a function is explicitly returning...
We recieve the returned value on the left...

```
added_value = add(5,4)
```

Now here you get that returned value inside this variable added_value, which you can then use anywhere you wish.

Now, remember that: A function always returns.
If explicitly there isn't anything returned from the function then by default return type is None.

Also remember that in python you can return multiple values.
And when you do so - it becomes a tuple.

For example.

```
def getUserDetails():

    name = input("Enter your name: ")
    age = int(input("Enter your age: "))
    gender = input("Enter your gender: "))

Return name, age, gender
```

Now this return values are returned in form of a tuple.
like this: ("Shailesh", 22, "Male").

If you pass this tuple to another function it'll manually have to extract values from it, using indexing.
So, you can distribute it in positional arguments, by unpacking the tuple when calling this method.

* Just like javascript, In python as well functions are first class citizens.

- You can assign a function to a variable.
- You can pass a function to another as an argument.
- A function can be returned from another function.