## Python Default Parameters

Function parameters can have default values in Python. We can provide a default value to a parameter by using the assignment operator (=). Here is an example.

```
def wish(name, wish="Happy Birthday"):

"""This function wishes the person with the provided message. If
the message is not provided, it defaults to "Happy Birthday" """

    print("Hello", name + ', ' + wish)

greet("Rohan")
greet("Hardik", "Happy New Year")
```

**Output**

```
Hello Rohan, Happy Birthday
Hello Hardik, Happy New Year
```

In this function, the parameter `name` does not have a default value and is required (mandatory) during a call.

On the other hand, the parameter `wish` has a default value of `"Happy Birthday"`. So, it is optional during a call. If an argument is passed corresponding to the parameter, it will overwrite the default value, otherwise it will use the default value.

### Important Points to be kept in mind while using default parameters:

- Any number of parameters in a function can have a default value.
- The conventional syntax for using default parameters states that once we have passed a default parameter, all the parameters to its right must also have default values.
- In other words, non-default parameters cannot follow default parameters.

For example, if we had defined the function header as:

```
def wish(wish = "Happy Birthday", name):
...
```

We would get an error as:

```
SyntaxError: non-default argument follows default argument
```

Thus to summarise, in a function header, any parameter can have a default value unless all the parameters to its right have their default values.