Passing Functions as Arguments

Let us understand how to pass functions as arguments using Python as programming language.

- The function which takes other functions as arguments is typically called as higher order function and the function which is passed as argument is called as lower order function.
- You need to define all the functions you want to pass as argument for the higher order functions.
- For simple functionality, we can also pass unnamed functions or lambda functions on the fly. We will see as part of the next topic.
- Let us take the example of getting sum of integers, squares, cubes and evens related to passing functions as arguments.

> Regular Functions

```
list(range(1, 10))
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
list(range(1, 10, 2))
```

```
[1, 3, 5, 7, 9]
```

```
list(range(1, 10, 3))
```

```
[1, 4, 7]
```

```
def sum_of_integers(lb, ub):
    total = 0
    for i in range(lb, ub + 1):
        total += i
    return total
```

```
sum_of_integers(5, 10)
```

```
45
```

```
def sum_of_squares(lb, ub):
    total = 0
    for i in range(lb, ub + 1):
        total += i * i
    return total
```

```
sum_of_squares(5, 10)
```

```
355
```

```
def sum_of_cubes(lb, ub):
    total = 0
    for i in range(lb, ub + 1):
        total += i * i * i
    return total
```

```
sum_of_cubes(5, 10)
```

```
2925
```

```
def sum_of_evens(lb, ub):
    total = 0
    for i in range(lb, ub + 1):
        total += i if i % 2 == 0 else 0
    return total
```

```
sum_of_evens(5, 10)
```

```
24
```

Using Functions as arguments

```
def my_sum(lb, ub, f):
    total = 0
    for e in range(lb, ub + 1):
        total += f(e)
    return total
```

```
def i(n): return n
```

```
def sqr(n): return n * n
```

```
def cube(n): return n * n * n
```

```
def even(n): return n if n % 2 == 0 else 0
```

```
my_sum(5, 10, i)
```

```
45
```

```
my_sum(5, 10, sqr)
```

```
355
```

```
my_sum(5, 10, cube)
```

```
2925
```

```
my_sum(5, 10, even)
```

```
24
```