CLASS 4

FUNCTIONS

Functions:

- They are **blocks of code that can be reused** at various parts of the code as well as various files.

- **Functions are not run** in a program **until they are "called" or "invoked" in a program**.

- If no return statement then Python returns the value **`None`** - represents absence of a value.

Function Characteristics:

aCubeIT
Institute of Artificial Intelligence

Python understands you are writing a function seeing **"def"** . It's a key word

Name of the function.
Otherwise, How can we tell Python which function to use?

Parameters are values given as input to function.
Ex: if we want to do sum of 2 numbers, function should know which numbers to add

```
def function_name(parameters [0 or more]):
"""docstring"""
    statement(s)
    return(something )
```

Document String to understand what that function does. **Not mandatory but good practice**

Body of function **Operation we want to perform**

Return **the output to the main program** **Optional in python**

How to write and Invoke/Call a function

- Writing a function

```
def is_even(i):
    """
    Input: i, a positive int
    Returns True if i is even, otherwise False
    """
    print("Inside is_even")
    return i%2 == 0
```

```
# Invoking the function
is_even(3)
```

```
Inside is_even

False
```

How to write and Invoke/Call a function

aCubeIT
Institute of Artificial Intelligence

- Formal parameter gets bound to the value of actual parameter when function is called.
- **New scope / frame / environment created when you enter a function**
- Scope is mapping of names to objects

```
def f(x):
    x = x + 1
    print("In f(x): x = ", x)
    return x
```

```
x = 3
z = f(x) #(Invoke/call)
print(z)
print(x)
```

```
In f(x): x =  4
4
3
```

Functions: Exercises

- Write a function to compute the square of a number

```
def square(num):
    out = num**2
    return(out)
```

```
square(3)
```

```
9
```

- **Try this code**

```
def square(num):
    out = num**2
```

```
Sq_3 = square(3)
print(Sq_3)
```

Functions: Exercises

- Write a code to compute the **factorial** of a number

```
def factorial(n):
    if n>1:
        return n*factorial(n-1)
    else:
        return n

fact = factorial(5)
print(fact)
```

```
120
```

Functions: Exercises

aCubeIT
Institute of Artificial Intelligence

- Write a function to check whether x is a factor of y.

  Example: (x,y) = (2,9)

```
def  is_factor(x,y):

    if y%x == 0:
        print("x is a factor of y")
    else:
        print("x is not a factor of y")
    return(0)
```

```
# Invoke:
is_factor(2,9)
```

```
x is not a factor of y

0
```

Functions: Exercises: HW

1. Write a function to check whether x is any power of y.
   Example: (x,y) = (2,8)

2. Write a Python function to find the Max of three numbers.

3. Write a Python program to reverse a string.
   Sample String : "1234abcd"
   Expected Output : "dcba4321“

4. Write a Python function to calculate the factorial of a number (a non-negative integer). The function accepts the number as an argument.

5. Write a Python function to check whether a number is in a given range.

6. Write a Python function that accepts a string and calculate the number of upper case letters and lower case letters. Go to the editor

Sample String : 'The quick Brow Fox'

Expected Output : No. of Upper case characters : 3

No. of Lower case Characters : 12

7. Write a Python function that checks whether a passed string is palindrome or not.

8. Write a function that returns the sum of multiples of 3 and 5 between 0 and **limit** (parameter). For example, if limit is 20, it should return the sum of 3, 5, 6, 9, 10, 12, 15, 18, 20.