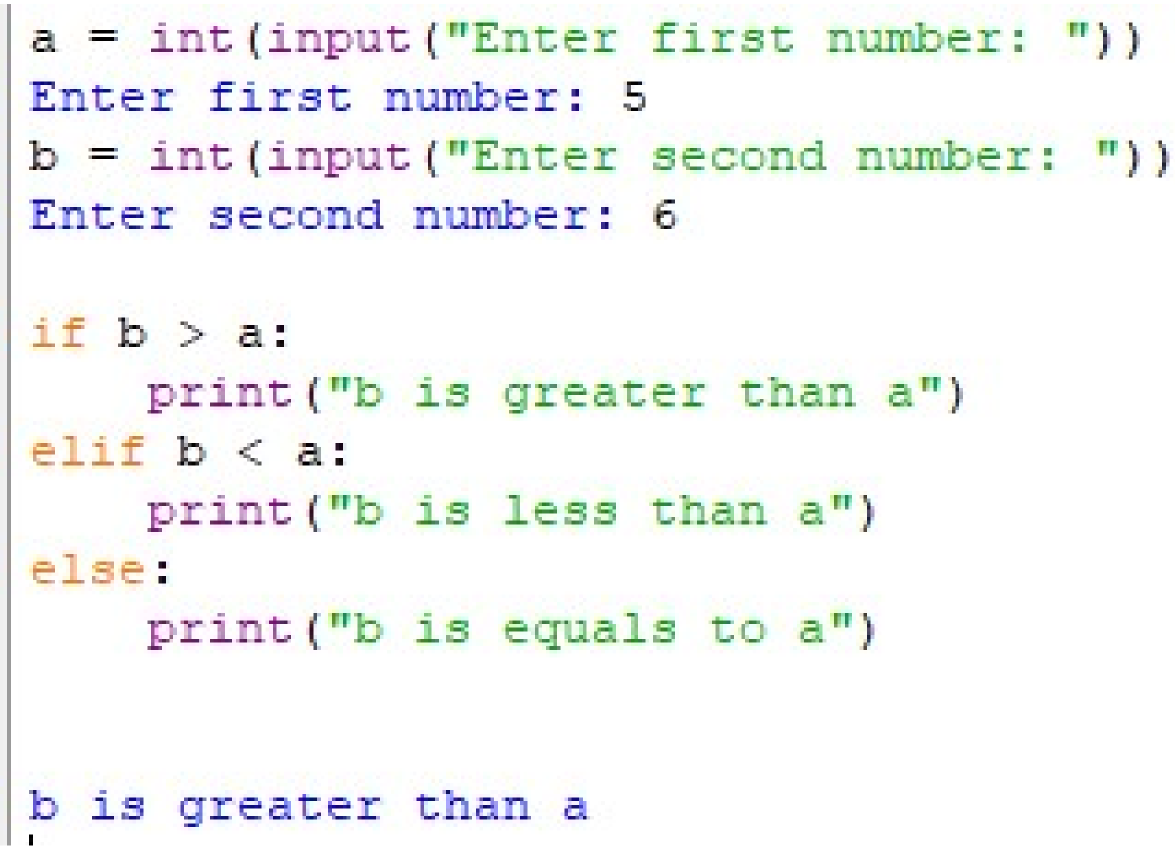
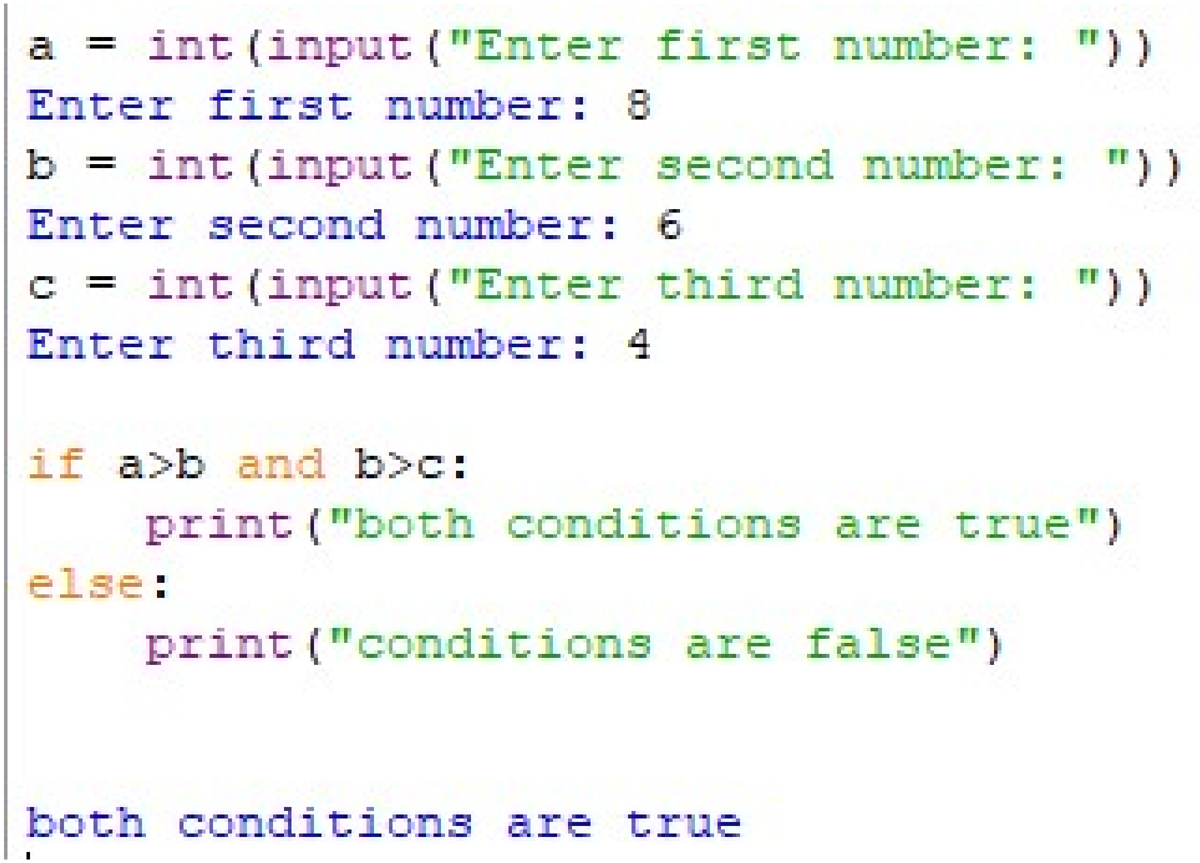


```
a = int(input("Enter first number: "))
Enter first number: 5
b = int(input("Enter second number: "))
Enter second number: 6

if b > a:
    print("b is greater than a")
elif b < a:
    print("b is less than a")
else:
    print("b is equals to a")


b is greater than a
```



```
a = int(input("Enter first number: "))
Enter first number: 8
b = int(input("Enter second number: "))
Enter second number: 6
c = int(input("Enter third number: "))
Enter third number: 4

if a>b and b>c:
    print("both conditions are true")
else:
    print("conditions are false")


both conditions are true
```

Decision Making Structure Tasks:

- ❑ Write program to check of the given number is positive, negative or zero.
- ❑Ask the user for a grade percentage and display the corresponding letter grade (A, B, C, D, F)

  A-Grade Range (80(inclusive) - 100(inclusive))
  B-Grade Range (70(inclusive) - 80(exclusive))
  C-Grade Range (60(inclusive) - 70(exclusive))
  D-Grade Range (50(inclusive) - 60(exclusive))
  Below 50 Fail
- ❑ Write program that displays: Kamran Akmal on output, if score > 30, Shoaib Akhtar, if 20<score<30 and Shahid Afridi, if 10<score<20.

Decision Making Structure Task:

❑ Write a program that takes positive integer as input from user and checks whether the number is even or odd and display the appropriate message on screen.

- Iterative Structure (For, While Loops)
  - A loop is a programming structure that repeats a block of code multiple times.
  - For loop is used to iterate over a sequence (that is either a list, a tuple, a dictionary, a set, or a string) – how many times you want to repeat the block of code.
  - While loop is used to repeat block of code until condition becomes False.

For loop structure:

```
for variable in sequence:
        # block of code
```

While loop structure

```
while condition:
        # block of code
```

Functions:

- A function is a group of statements made to execute them more than once in a program. A function has a name.
- Functions can compute a result value and can have parameters that serves as function inputs which may differ each time when function is executed.
- Functions are used *to reduce the size of code as it increases the code reusability* and *split a complex problem into multiple modules (functions) to improve manageability*.
- Sequential codes are easy for small scale programs. It becomes harder to keep track of details when code size exceeds.

Advantages:

→ Modularity

→ Abstraction

→ Code reusability

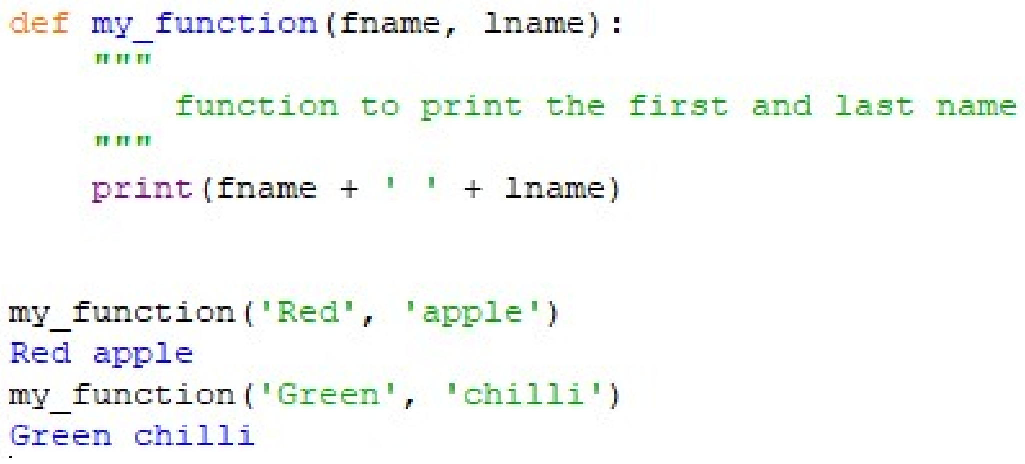
- A function is a block of code which only runs when it is called.
- You can pass data, known as parameters, into a function.
- A function can or cannot return data as result.

Function Components:

- *Function signature*
  - *Function Name*
  - *Function Arguments (optional)*
- *Doc string*
- *Function Body*
- *Function return statement*

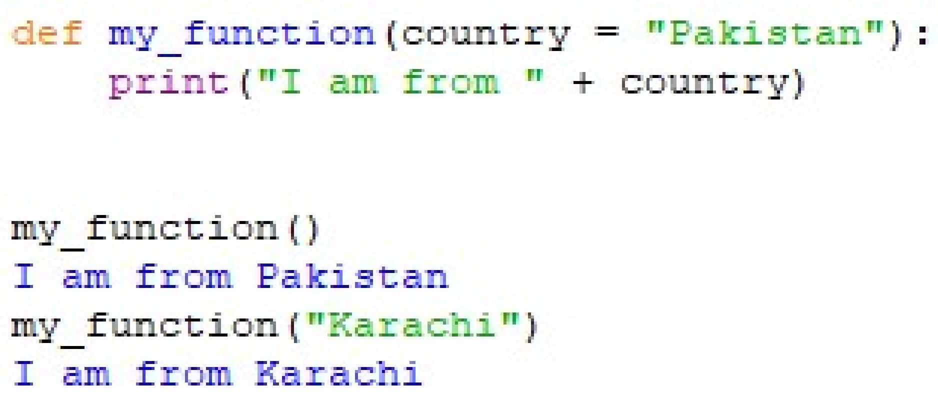
Syntax:

```
def function_name(arguments):
        doc string
        body
        return statement
```



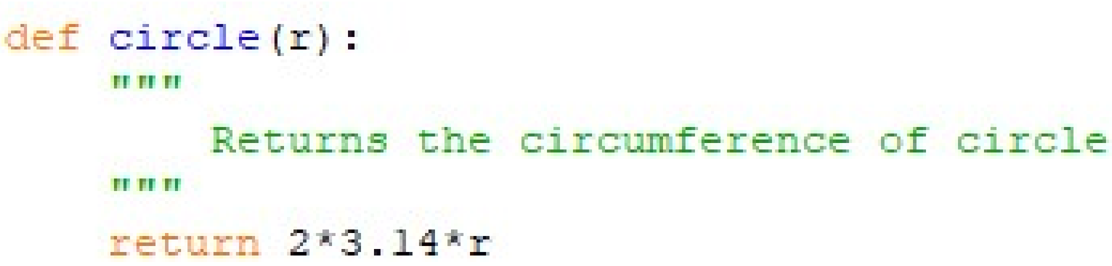
```
def my_function(fname, lname):
    """
        function to print the first and last name
    """
    print(fname + ' ' + lname)


my_function('Red', 'apple')
Red apple
my_function('Green', 'chilli')
Green chilli
```

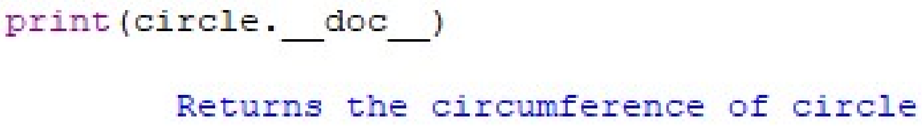


```
def my_function(country = "Pakistan"):
    print("I am from " + country)


my_function()
I am from Pakistan
my_function("Karachi")
I am from Karachi
```

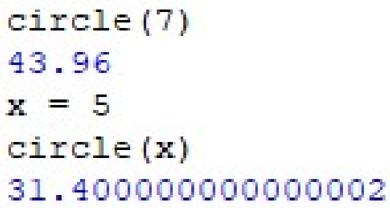


```
def circle(r):
    """
        Returns the circumference of circle
    """
    return 2*3.14*r
```

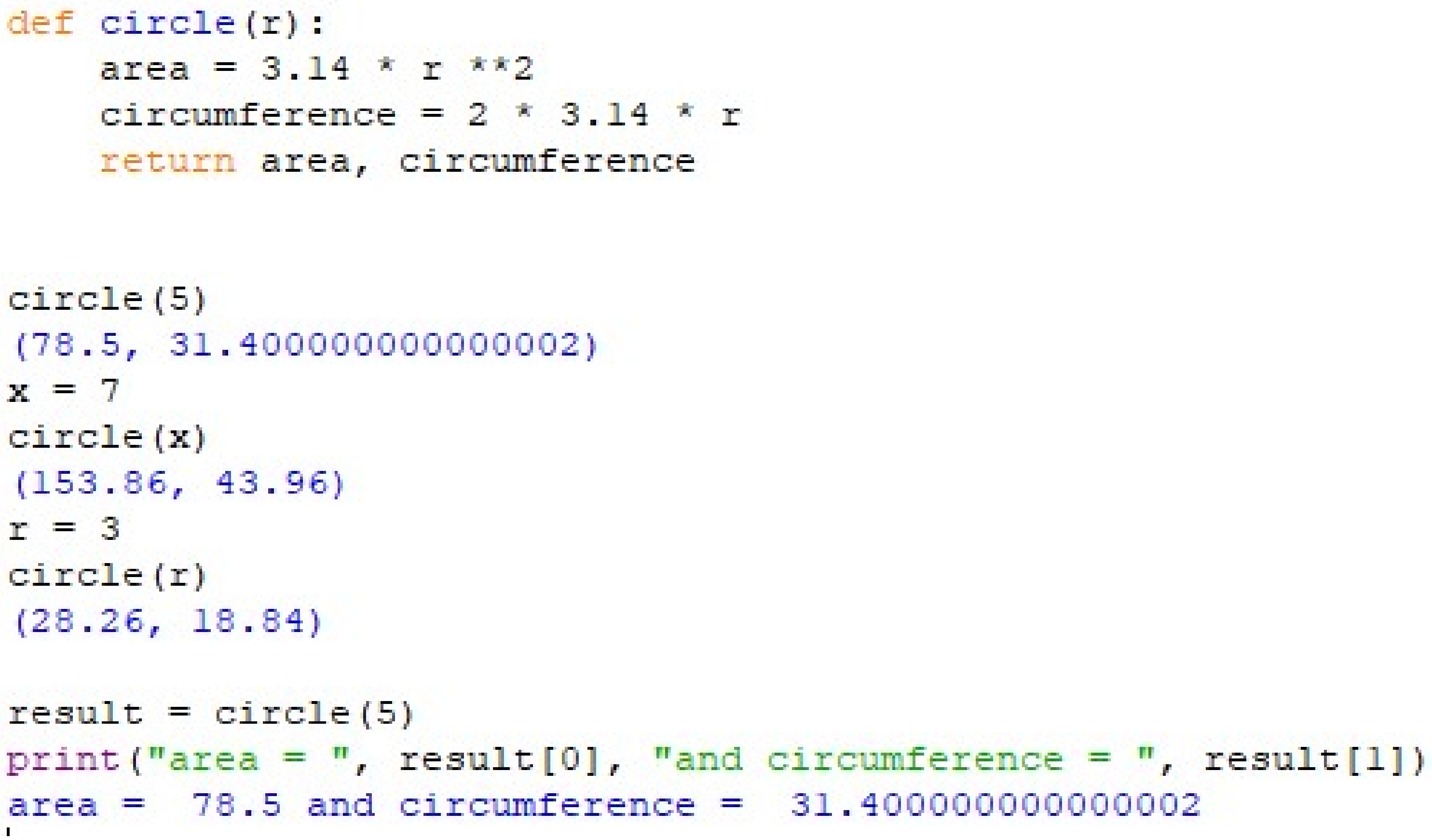


```
print(circle.__doc__)

        Returns the circumference of circle
```



```
circle(7)
43.96
x = 5
circle(x)
31.400000000000002
```



```
def circle(r):
    area = 3.14 * r **2
    circumference = 2 * 3.14 * r
    return area, circumference


circle(5)
(78.5, 31.400000000000002)
x = 7
circle(x)
(153.86, 43.96)
r = 3
circle(r)
(28.26, 18.84)

result = circle(5)
print("area = ", result[0], "and circumference = ", result[1])
area =  78.5 and circumference =  31.400000000000002
```