

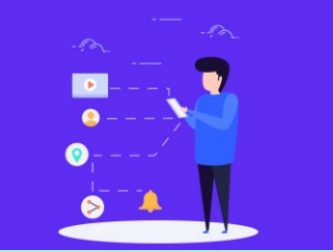


Introduction to Computer Programming

ACCA4020

Functions

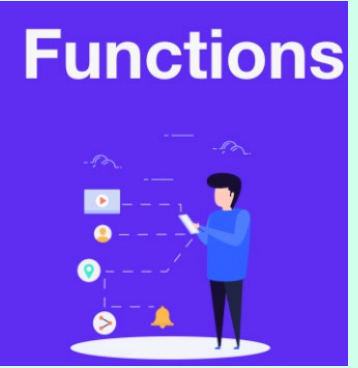
Dr. Seena Joseph



Functions

Learning outcomes

- Understanding the concepts of functions as modular blocks of code
- Defining a function using the appropriate syntax (Example Python)
- Calling a function by providing the function name and arguments



Functions

`len("Information")`

`pow(x, y)`

`print(x)`

`len(s)`

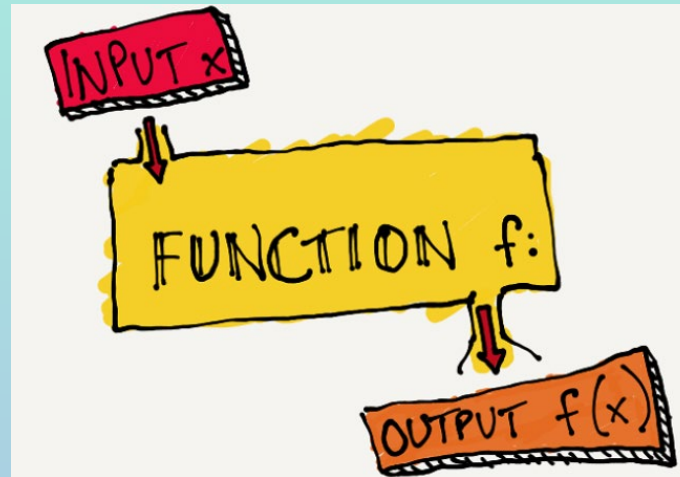
`pow(3,2)`

`print("Hello")`

`abs(-5)`

Functions

- Fundamental building block in programming
- Allow you to group a set of instructions under a given name



Organize

Modular

Reusable

Functions

Organize

- Functions break large programs into smaller, manageable parts.
- Easier to debug and maintain.

Modular

- Code becomes cleaner and more readable.

Reusable

- Functions can be called multiple times without rewriting the code.

Functions

What is a function?

- A self-contained block of code that performs a specific task or a set of tasks.
- Functions become increasingly important as the application grows in size and complexity.

Example

pow(5,2) = ?
pow(3,3) = ?
min(5,2,6)=?
max(5,2,6)=?
len("computer")= ?
len("Hello")= ?

Python code

Finding the power of a number

```
BNumber=int(input("Enter the base value : "))
ENumber=int(input("Enter the exponent : "))

ans=1
for i in range(ENumber):
    ans*=BNumber
print(str(BNumber) + " to the power " + str(ENumber) + " is " + str(ans));
```

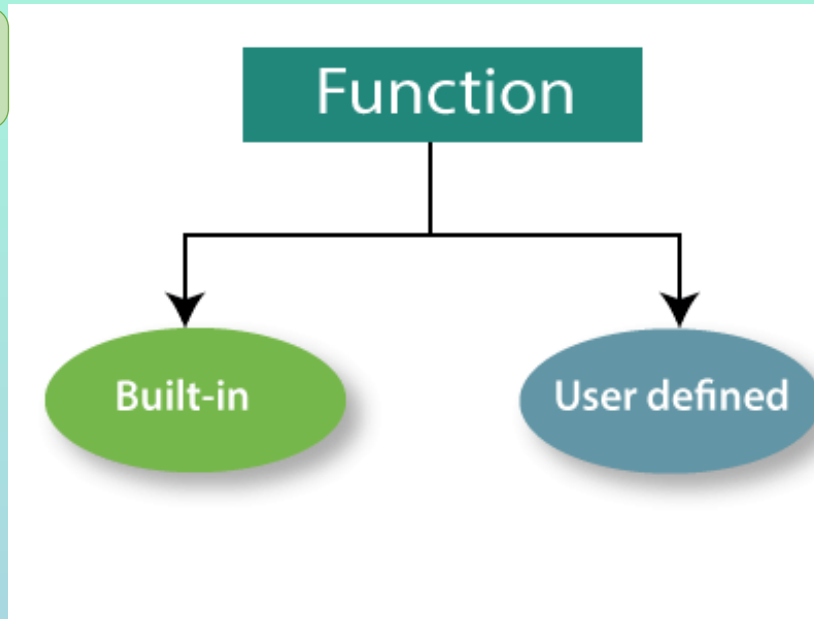
```
Enter the base value : 4
Enter the exponent : 4
4 to the power 4 is 256
```

Built-in function

[defined in the programming language; perform *general* task; ready to use]

Functions

Built-in function examples?



User defined function examples

AreaofCircle(r)

AreaofRectangle(l,b)

isPalindrome(sWord)

isPrime(i)

Organize

Modular

Reusable

Functions

Why **user-defined** functions?

Write a program in Python that accepts an integer and determines if it is even or odd.

Even or Odd - Single number

```
iNum1=int(input("Enter the number : "))

print('*****')

if iNum1%2==0:
    print(str(iNum1) + " is Even")
else:
    print(str(iNum1) + " is Odd")
```

```
Enter the number : 36
*****
36 is Even
```

Write a program in Python that accepts three numbers and determines if the input numbers are even or odd.

Even or Odd - Three numbers

```
iNum1=int(input('Enter the first number :'))
iNum2=int(input('Enter the second number :'))
iNum3=int(input('Enter the third number :'))
print('*****')
if iNum1%2==0:
    print(f'{iNum1} is Even')
else:
    print(f'{iNum1} is Odd')

if iNum2%2==0:
    print(f'{iNum2} is Even')
else:
    print(f'{iNum2} is Odd')

if iNum3%2==0:
    print(f'{iNum3} is Even')
else:
    print(f'{iNum3} is Odd')
```

```
Enter the first number :456
Enter the second number :341
Enter the third number :34
*****
456 is Even
341 is Odd
34 is Even
```

Functions

User-defined functions

Defined by the programmer/user

- Not built in the programming language
- Assist in modularizing the code, reusability, and enhancing code organization
- Steps in defining a user-defined function
 - Function definition
 - Function header - [Function name, parameters]
 - Function body
 - Call/invoke the function

Functions

User-defined functions – Anatomy

```
def function_name (param1, param2):  
    Function body    # statements to get the output  
    Return result
```

parameter(s)
One or more variables that a function expects as input

function definition

```
def EvenOdd(iNum):  
    if iNum%2==0:  
        print(str(iNum) + " is Even")  
    else:  
        print(str(iNum) + " is Odd")
```

Function Header

Function name and parameters

Function Body

Functions

User-defined functions – Anatomy

function definition

```
def EvenOdd(iNum):  
    if iNum%2==0:  
        print(str(iNum) + " is Even")  
    else:  
        print(str(iNum) + " is Odd")
```

Function definition

Input

```
iNum1=int(input("Enter the First number : "))  
56
```

Enter the First number : 26

56

Function call

Function Call

```
EvenOdd(iNum1)
```

56 is Even

```
EvenOdd(69)
```

69 is Odd

Functions

Without a user-defined function

Write a program in Python that accepts three numbers and determines if the input numbers are even or odd.

With a user-defined function

Even or Odd - Three numbers

```
iNum1=int(input('Enter the first number :'))
iNum2=int(input('Enter the second number :'))
iNum3=int(input('Enter the third number :'))
print('*****')
if iNum1%2==0:
    print(f'{iNum1} is Even')
else:
    print(f'{iNum1} is Odd')

if iNum2%2==0:
    print(f'{iNum2} is Even')
else:
    print(f'{iNum2} is Odd')

if iNum3%2==0:
    print(f'{iNum3} is Even')
else:
    print(f'{iNum3} is Odd')
```

```
Enter the first number :456
Enter the second number :341
Enter the third number :34
*****
456 is Even
341 is Odd
34 is Even
```

function definition

```
def EvenOdd(iNum):
    if iNum%2==0:
        print(str(iNum) + " is Even")
    else:
        print(str(iNum) + " is Odd")
```

Modular

Organize

Reusable

```
iNum1=int(input("Enter the First number : "))
iNum2=int(input("Enter the Second number : "))
iNum3=int(input("Enter the Third number : "))
```

```
EvenOdd(iNum1)
EvenOdd(iNum2)
EvenOdd(iNum3)
```

```
Enter the First number : 26
Enter the Second number : 34
Enter the Third number : 897
26 is Even
34 is Even
897 is Odd
```

Functions

Think/Share

Practical Task 1 – Class activity

Write a user-defined function in Python to add two numbers

```
def sum(in1, in2):  
    return in1+in2
```

Find the function name, parameters , return value.
Identify the function call

```
isum=sum(26,45)
```

```
print(isum)
```

71

```
print(sum(2.5,2.5))
```

5.0

Functions

Think /share

Practical Task 1 – Class activity

1. Write a user-defined function in Python to find the product of three numbers.
2. Rewrite the function EvenOdd with the following changes:
 - a. Function name should be *isEven*
 - b. The function should return Boolean value (True or False)

Conclusion

Function - reusable pieces/blocks of code

Functions are essential building blocks in programming languages that allow you to create **modular**, **reusable**, and organized code by encapsulating specific tasks or operations.

- Functions are not run in a program until they are **called/invoked**
- Function characteristics:
 - has a **name**
 - has **parameters** (0 or more) - **input**
 - has a **body**
 - has return value - **output**

Exercise

- Write a user-defined function in Python to calculate and return area of a rectangle using length and width.
- Write a user-defined function in Python check if a number is positive, negative or zero.
- Write a user-defined function in Python to check if the given string is palindrome or not.