Function

In [ ]:

```
Definition : A function is a reusable block of code that perform specific task,
                returns the output.


Core Usages:

    1- Function hide implemenation details and provide a simple interface
    2- Function is use for reusability(write ones and use multiple times)
    3- Modularity: Break complex promplex problem into smaller pieces.

Synax:

def <name_of_the_function>(parametes):
    <body_of_statement>
    return <returns_the_value>
```

Type of Function:

1- Built-in Function

2- User defined Function

3- Anonymous Function

In [ ]:

```
# 1- Built-in Function

print("Hello")
len("john")
type()
input()
int()
str()
dict()
```

In [1]:

```
# 2- User defined Function

# 1

def greet():
    return f"Hello world"

greet()
```

Out[1]:

```
'Hello world'
```

In [10]:

```
# 2

# Function
def greet(name):
    return f"Hello : {name}"
```

```
# calling the function
print(greet("John"))
print(greet("Bob"))
```

```
Hello : John
Hello : Bob
```

In [ ]:

```
# Defined input Paraters types

# 1- Default arguments(parameters)

# 2- Required arguments

# 3- keyword arguments

# 4- variable-length arguments
```

In [17]:

```
# 1- Default arguments(parameters)

def user_name(name = "Bob"):
    return name

print(user_name())

# 2
def greet_with_title(name = "John",title = "Mr"):
    return f"Hello {title} {name}"

greet_with_title("bob")
```

```
Bob
```

Out[17]:

```
'Hello Mr bob'
```

In [20]:

```
# 2- Required arguments(parameters)

def add(value1, value2):
    sum_output = value1 + value2
    return sum_output

add(10,20)
```

Out[20]:

```
30
```

In [22]:

```
# 3- Keyword arguments

def print_the_name(name, age):
    return f" {name} {age}"

print_the_name(name = "John Cena", age = 40)
```

Out[22]:

```
' John Cena 40'
```

In [25]:

```
# 4 Variable length arguments

def names(*names):
    for name in names:
        print(name)
```

```
Lst = ["amit","pranav","deepak","raghav"]
names(Lst)
```

```
['amit', 'pranav', 'deepak', 'raghav']
```

```
In [ ]:
```