Functions:

Functions are building blocks of Python code. They package reusable logic and make your programs more organized, efficient, and easier to maintain. Let's dive into this key concept together!

Made with Gamma

Types of Functions

Built-in Functions

These are ready-to-use functions provided by Python. They perform common tasks like printing, converting data types, or finding the length of a list.

User-defined Functions

These are functions you create to perform specific tasks. They can be customized to your needs and improve code modularity.

Made with Gamma

Function Creation and Usage

Define a Function

We use the **def()** to create a function.

def function_name():

- def xyz(): #Declare a function
  print("hello")
- xyz() #Call a function

Made with Gamma

Function Parameters and Arguments

1 Parameters

These are variables defined(pass) in the function declaration that act as placeholders for data inputs

def function_name(parameter1, parameter2,.....)

2 Arguments

These are the actual values passed to the function when it is called

def function_name(argument1, argument2,.....)

3 Example

```
def sumthis(a,b):       #parameters
        print(a+b)
sumthis(10,20)          #arguments
```

Made with Gamma

Types of Parameters and Arguments

1) Positional

arguments passed to a function in correct positional order.

```
def xyz(a,b):
    print(a+b)
xyz(10,20)
```

2) Keyword

arguments passed with specific parameters name, Can be in any order.

```
def save(x,y,z):
    print(f"name:{x}, age: {y}, id: {z}")
save(y = 20, z = 17234, x = "morris")
```

3) Default

parameters with present values if no argument passed.

```
def save(x = 10,y=20,z=25):
    print(x+y+z)
save(10)
```

4.1) *args

It allows a function to handle an unknown no. of positional arguments passed to it during calls or we don't known how much value user pass. The arguments are available as a **tuple** inside the function body.

```
def sumthis(*args):
    print(args)
sumthis()
sumthis(10,2)
```

4.2) **kwargs

It allows the function to accept an arbitrary number of keyword arguments, which are gathered into a dictionary that can be accessed in the function body.

```
def savedata(**kwargs):
    print(kwargs)
savedata()
savedata(name = "isha", age = 22)
```

Made with Gamma