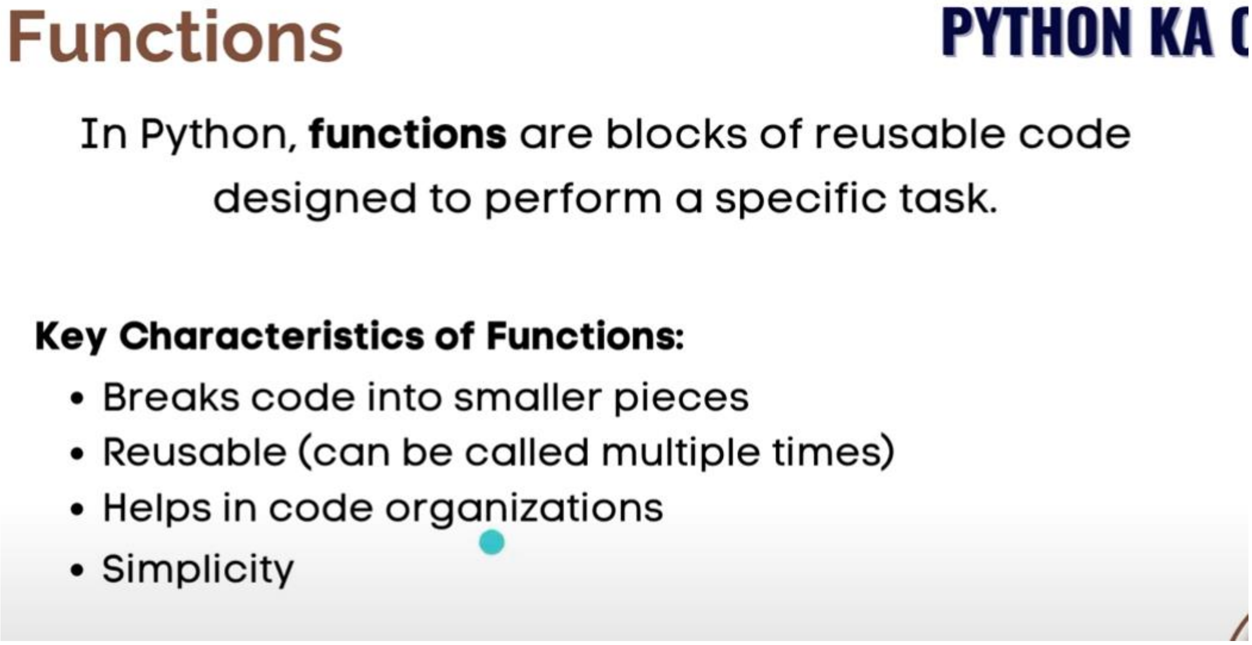


Functions

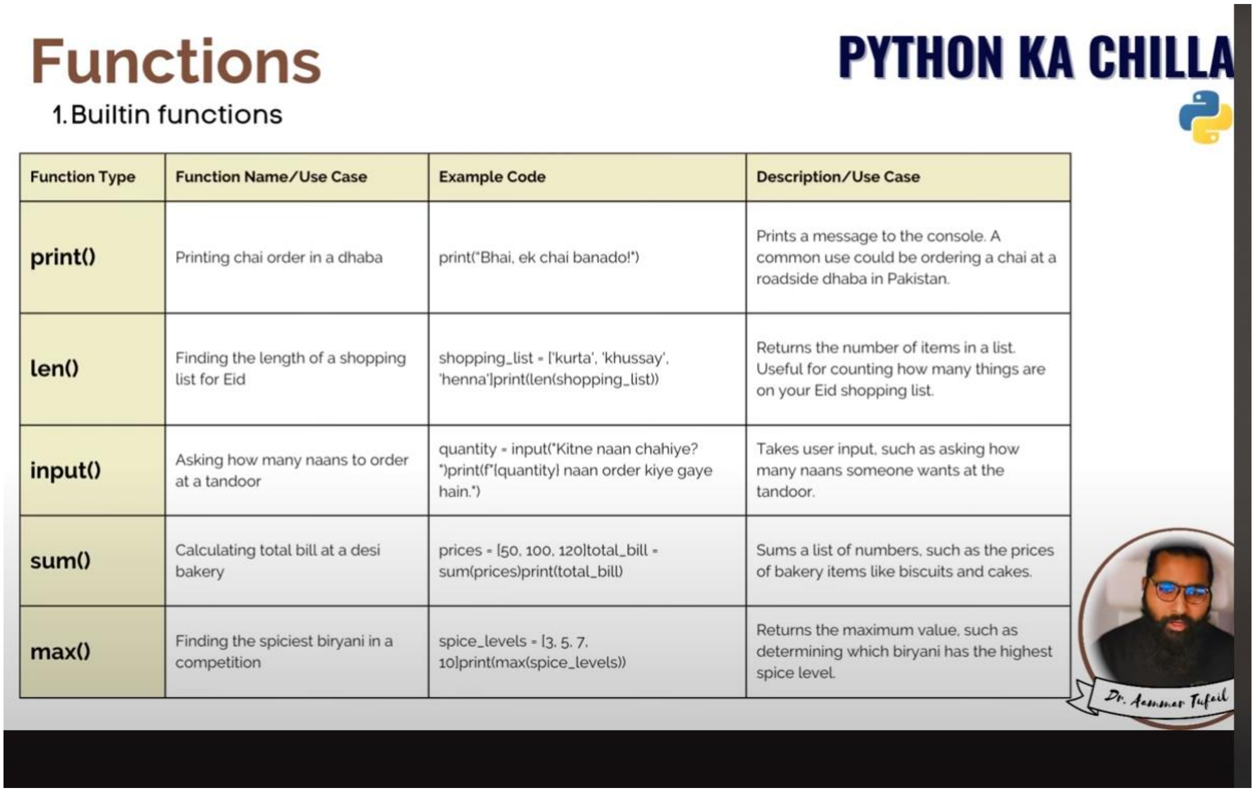
PYTHON KA C

In Python, **functions** are blocks of reusable code designed to perform a specific task.

Key Characteristics of Functions:

- Breaks code into smaller pieces
- Reusable (can be called multiple times)
- Helps in code organizations
- Simplicity

Built in function is already define in the system

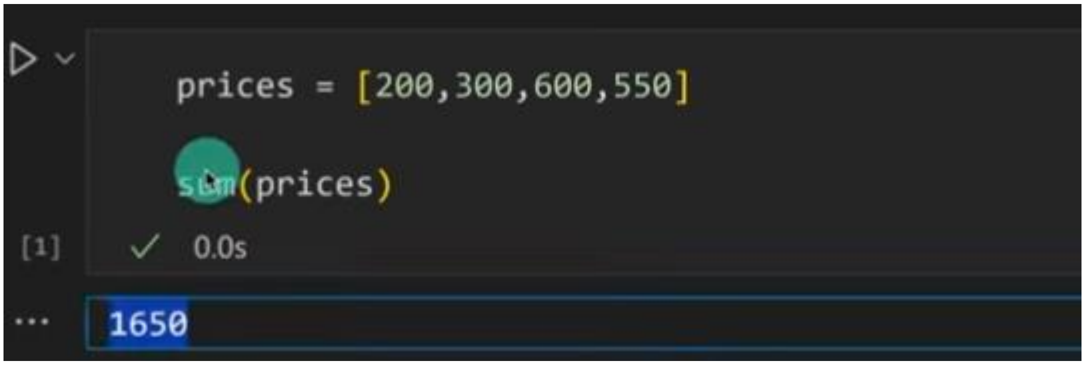


Functions

PYTHON KA CHILLA

1. Builtin functions

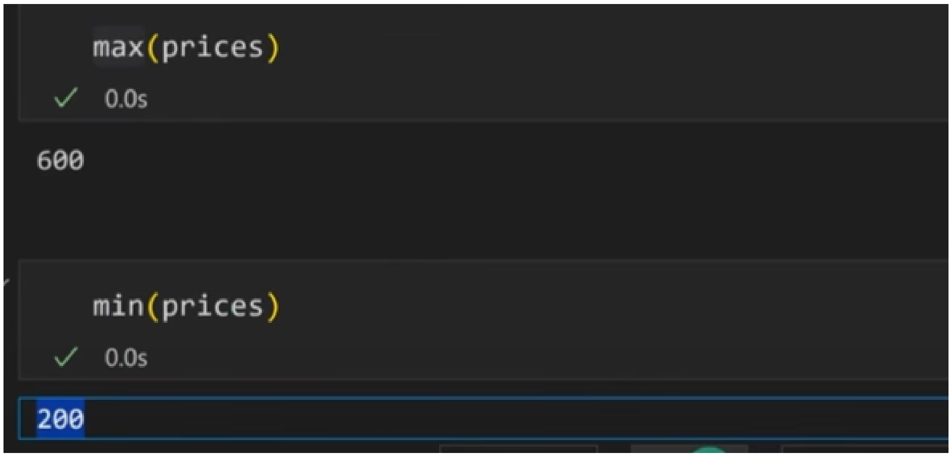
Function Type	Function Name/Use Case	Example Code	Description/Use Case
print()	Printing chai order in a dhaba	print("Bhai, ek chai banado!")	Prints a message to the console. A common use could be ordering a chai at a roadside dhaba in Pakistan.
len()	Finding the length of a shopping list for Eid	shopping_list = ['kurta', 'khussay', 'henna']print(len(shopping_list))	Returns the number of items in a list. Useful for counting how many things are on your Eid shopping list.
input()	Asking how many naans to order at a tandoor	quantity = input("Kitne naan chahiye? ")print(f"{quantity} naan order kiye gaye hain.")	Takes user input, such as asking how many naans someone wants at the tandoor.
sum()	Calculating total bill at a desi bakery	prices = [50, 100, 120]total_bill = sum(prices)print(total_bill)	Sums a list of numbers, such as the prices of bakery items like biscuits and cakes.
max()	Finding the spiciest biryani in a competition	spice_levels = [3, 5, 7, 10]print(max(spice_levels))	Returns the maximum value, such as determining which biryani has the highest spice level.



```
prices = [200,300,600,550]
sum(prices)
```

```
[1]  ✓ 0.0s
```

```
1650
```



```
max(prices)
```

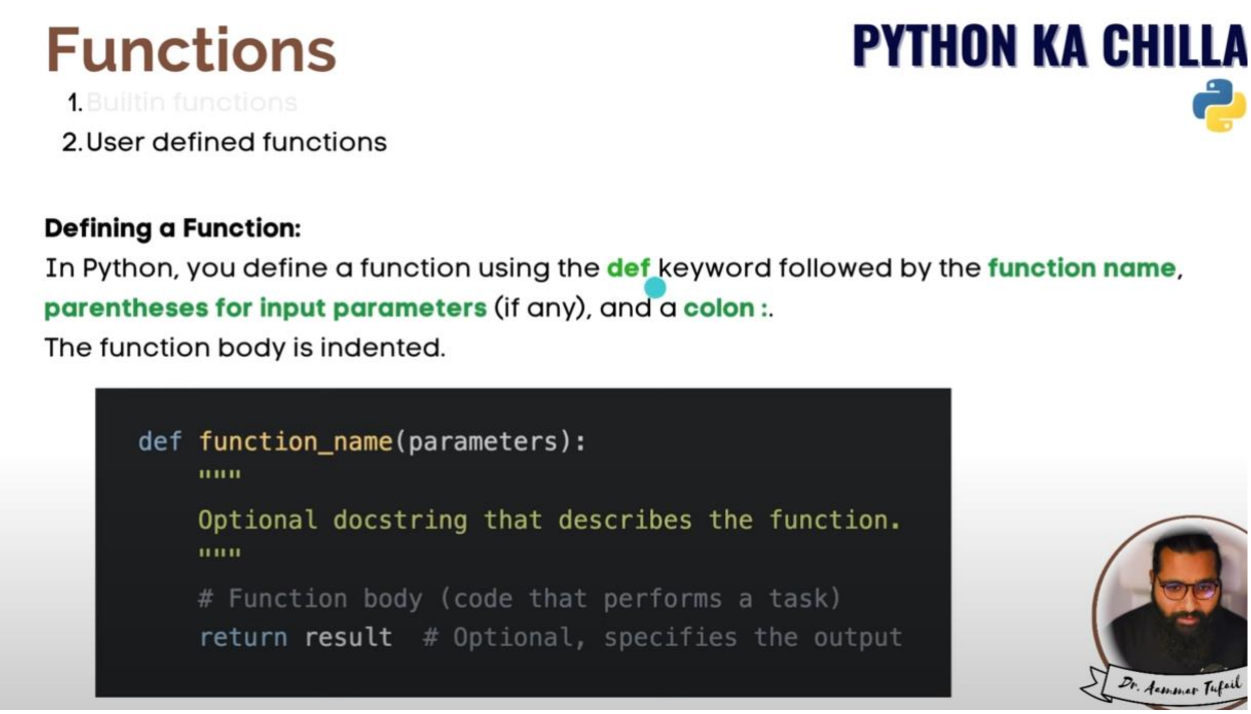
```
✓ 0.0s
```

```
600
```

```
min(prices)
```

```
✓ 0.0s
```

```
200
```



Functions

PYTHON KA CHILLA

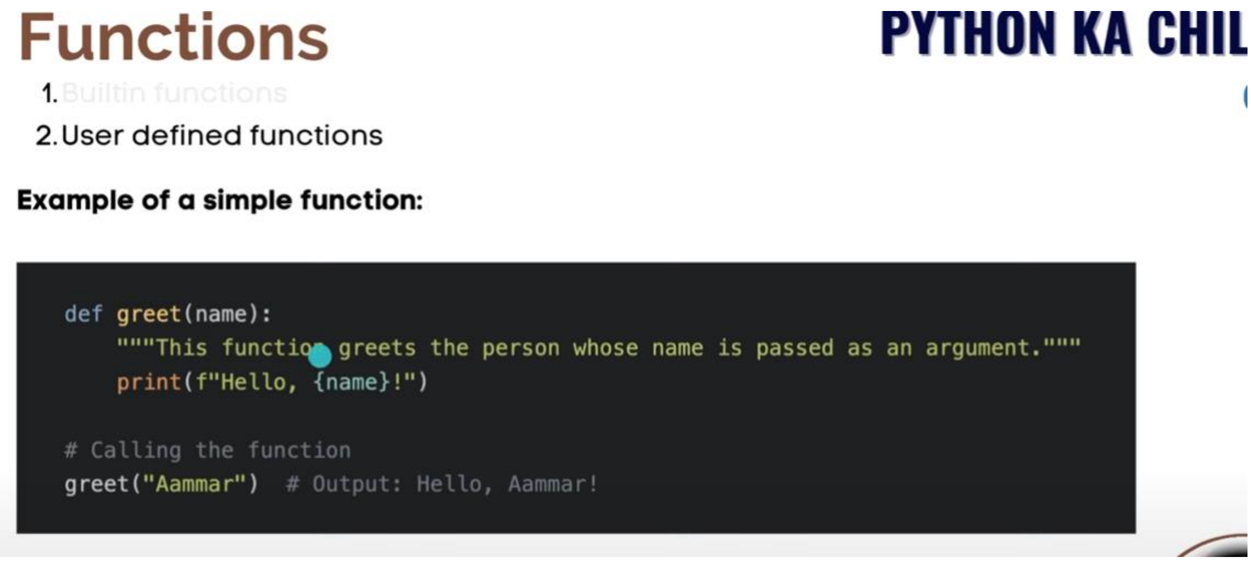
1. Builtin functions
2. User defined functions

Defining a Function:

In Python, you define a function using the **def** keyword followed by the **function name**, **parentheses for input parameters** (if any), and a **colon :**.
The function body is indented.

```
def function_name(parameters):
    """
    Optional docstring that describes the function.
    """
    # Function body (code that performs a task)
    return result  # Optional, specifies the output
```

“”” ””” is a doCS string



Functions

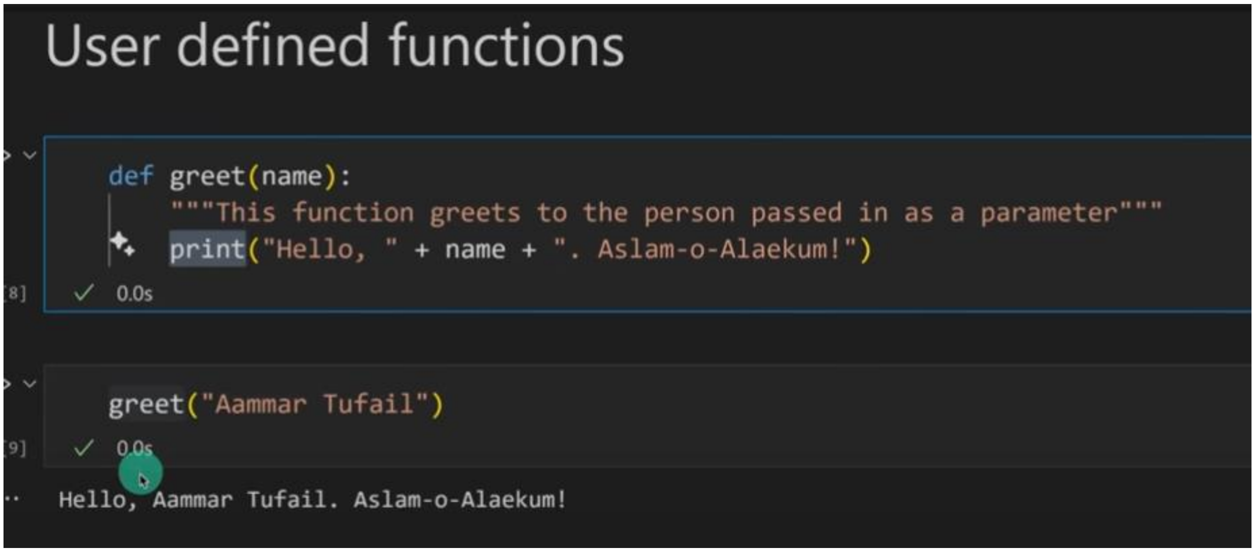
PYTHON KA CHIL

1. Builtin functions
2. User defined functions

Example of a simple function:

```python
def greet(name):
    """This function greets the person whose name is passed as an argument."""
    print(f"Hello, {name}!")

# Calling the function
greet("Aammar")  # Output: Hello, Aammar!
```



User defined functions

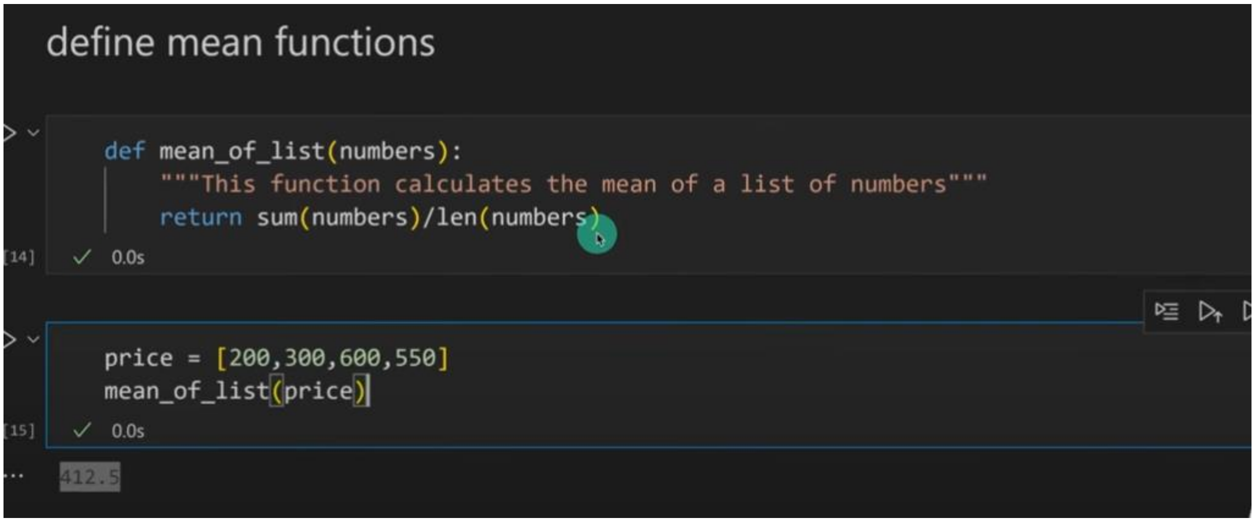
```python
def greet(name):
    """This function greets to the person passed in as a parameter"""
    print("Hello, " + name + ". Aslam-o-Alaekum!")
```

```
[8]  ✓ 0.0s
```

```python
greet("Aammar Tufail")
```

```
[9]  ✓ 0.0s

Hello, Aammar Tufail. Aslam-o-Alaekum!
```

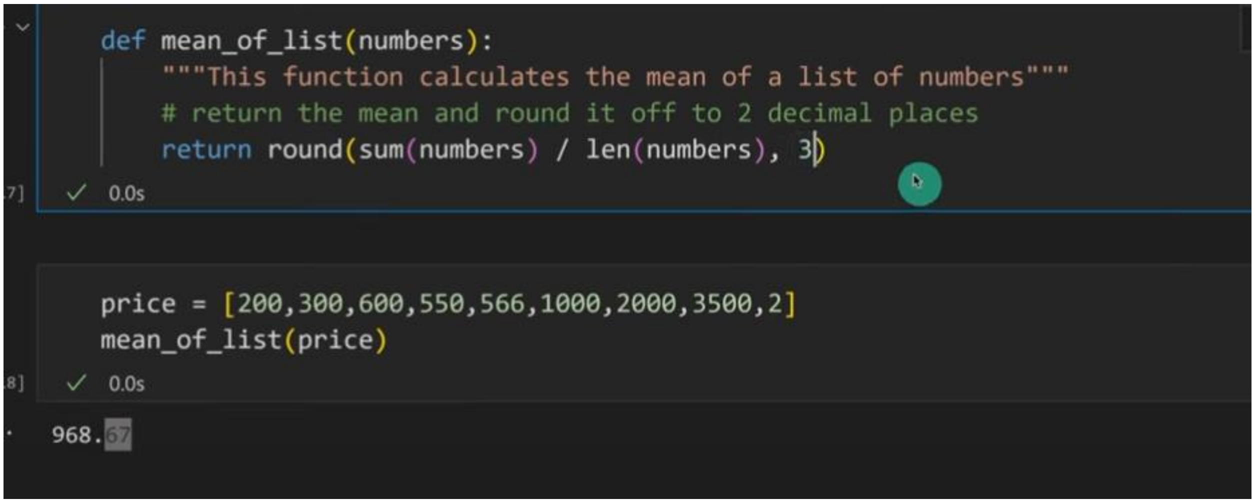


define mean functions

```
def mean_of_list(numbers):
    """This function calculates the mean of a list of numbers"""
    return sum(numbers)/len(numbers)
```

```
price = [200,300,600,550]
mean_of_list(price)
```

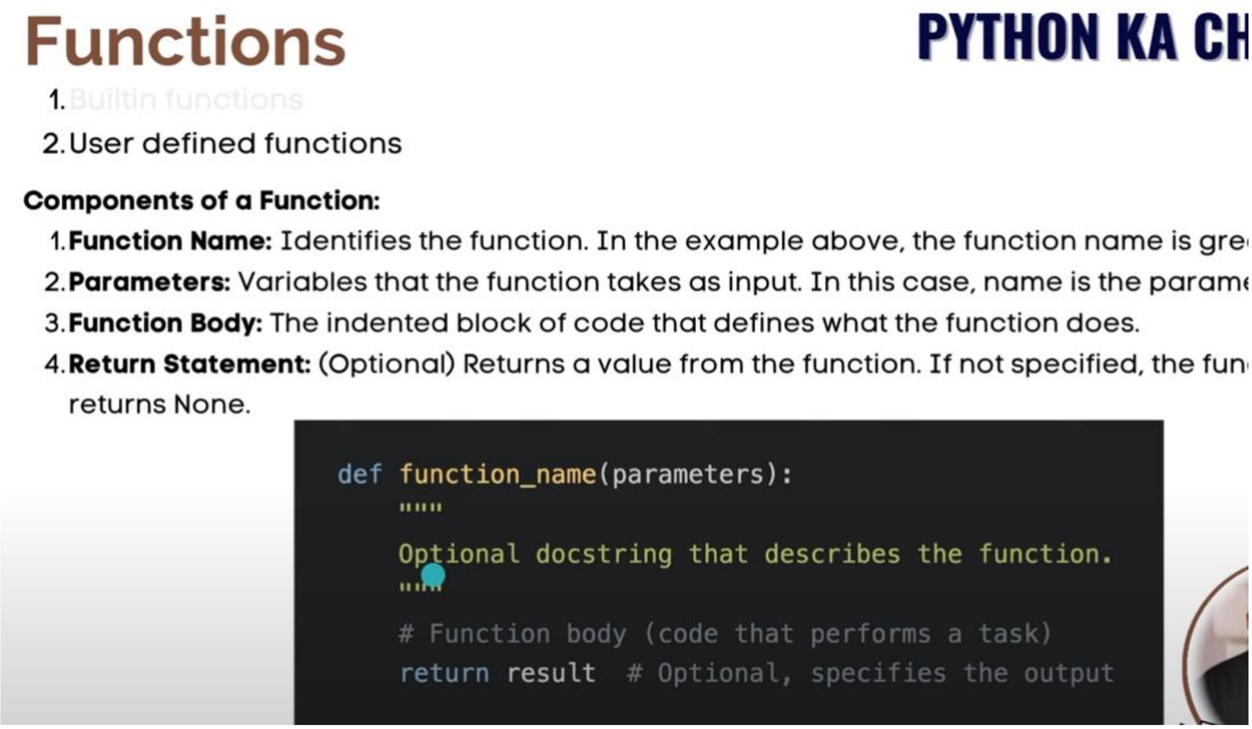
```
412.5
```



```
def mean_of_list(numbers):
    """This function calculates the mean of a list of numbers"""
    # return the mean and round it off to 2 decimal places
    return round(sum(numbers) / len(numbers), 3)
```

```
price = [200,300,600,550,566,1000,2000,3500,2]
mean_of_list(price)
```

```
968.67
```



Functions

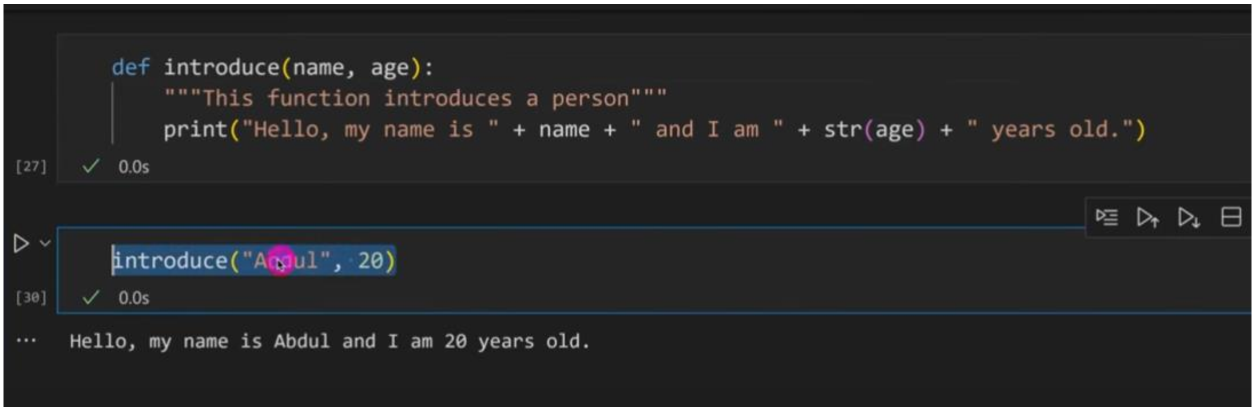
PYTHON KA CH

1. Builtin functions
2. User defined functions

Components of a Function:

1. **Function Name:** Identifies the function. In the example above, the function name is gre
2. **Parameters:** Variables that the function takes as input. In this case, name is the parame
3. **Function Body:** The indented block of code that defines what the function does.
4. **Return Statement:** (Optional) Returns a value from the function. If not specified, the fun returns None.

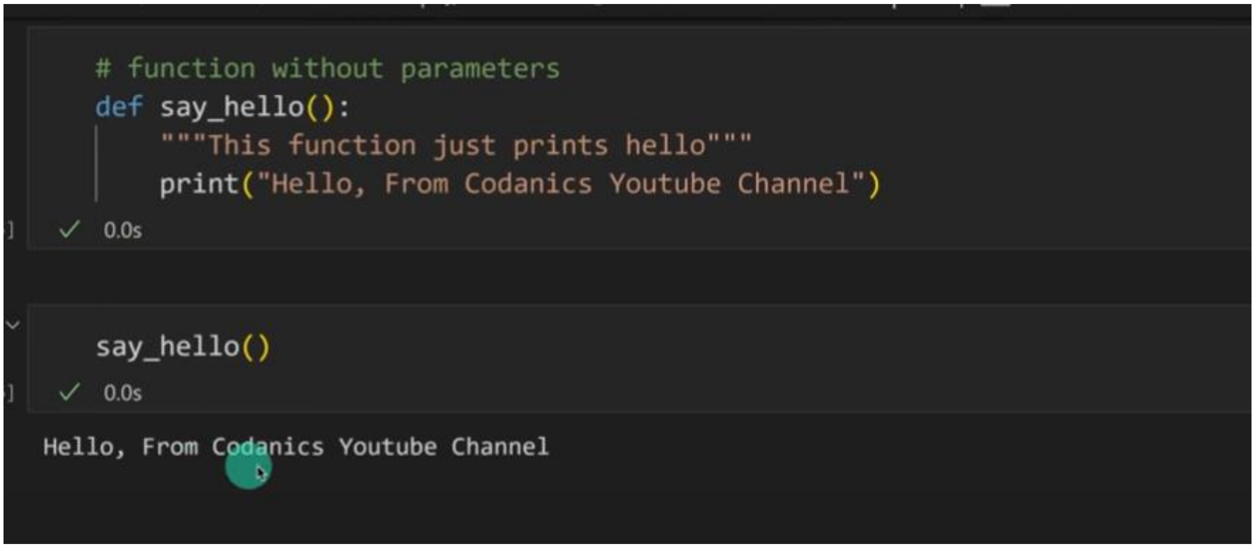
```
def function_name(parameters):
    """
    Optional docstring that describes the function.
    """
    # Function body (code that performs a task)
    return result  # Optional, specifies the output
```



```
def introduce(name, age):
    """This function introduces a person"""
    print("Hello, my name is " + name + " and I am " + str(age) + " years old.")
```

```
introduce("Abdul", 20)
```

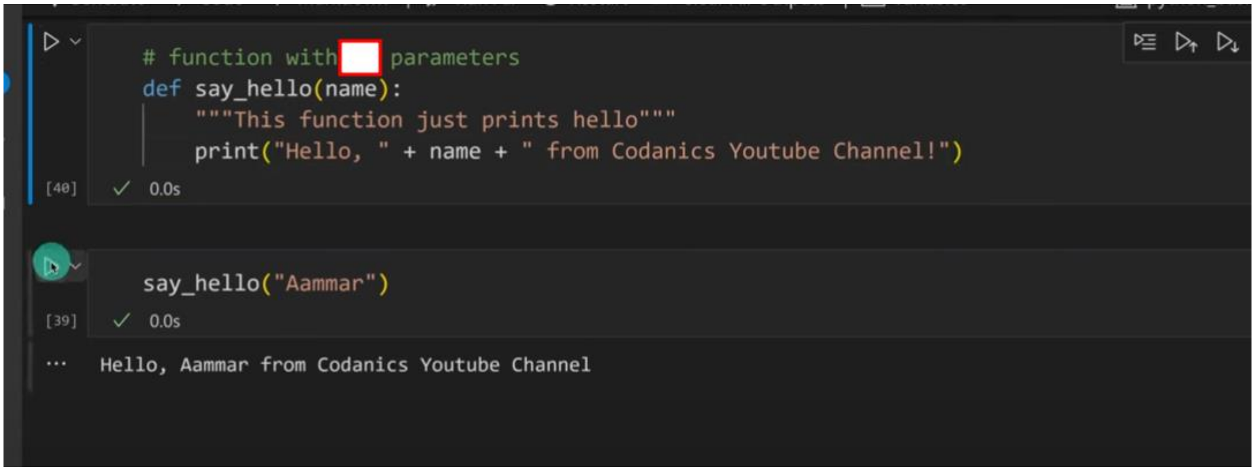
```
Hello, my name is Abdul and I am 20 years old.
```



```python
# function without parameters
def say_hello():
    """This function just prints hello"""
    print("Hello, From Codanics Youtube Channel")
```

```python
say_hello()
```

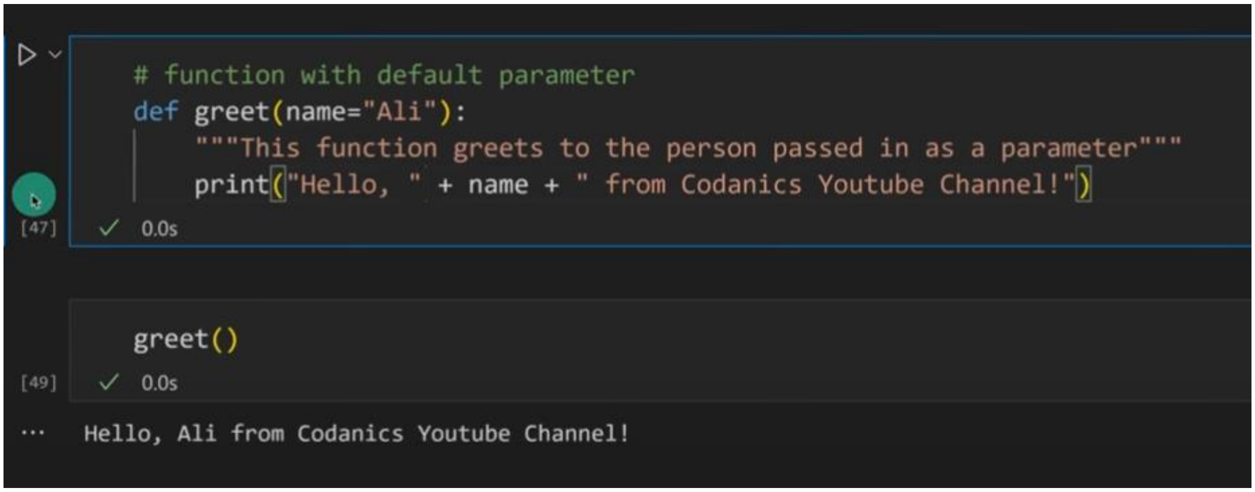
```
Hello, From Codanics Youtube Channel
```



```python
# function with [redacted] parameters
def say_hello(name):
    """This function just prints hello"""
    print("Hello, " + name + " from Codanics Youtube Channel!")
```

```python
say_hello("Aammar")
```

```
Hello, Aammar from Codanics Youtube Channel
```

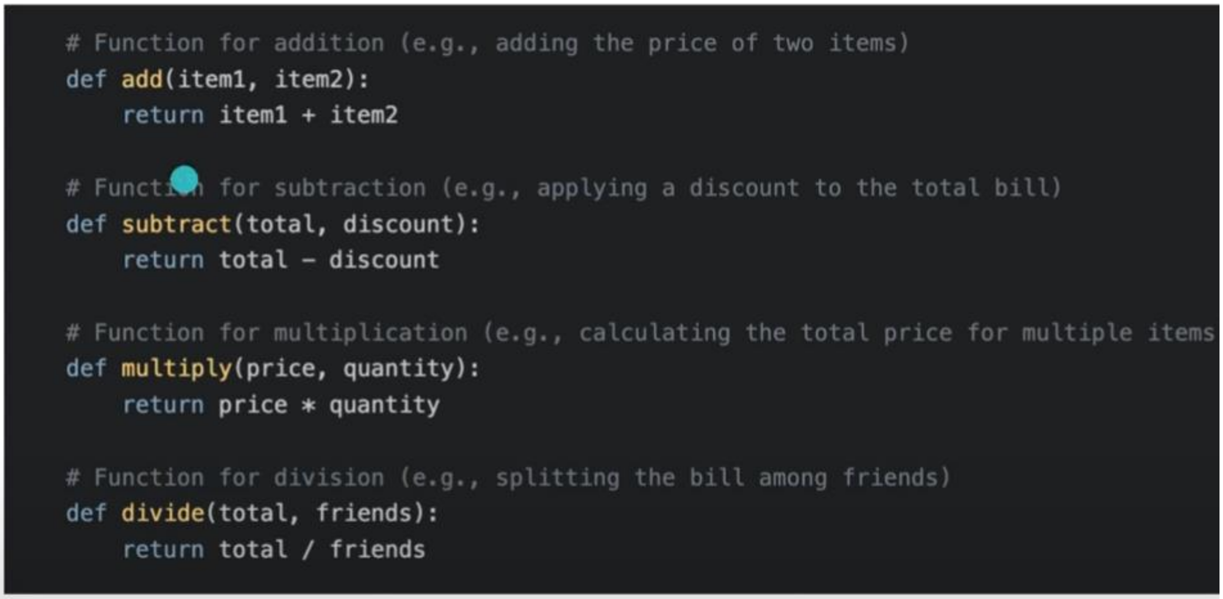


```python
# function with default parameter
def greet(name="Ali"):
    """This function greets to the person passed in as a parameter"""
    print("Hello, " + name + " from Codanics Youtube Channel!")
```

```python
greet()
```

```
Hello, Ali from Codanics Youtube Channel!
```

Functions

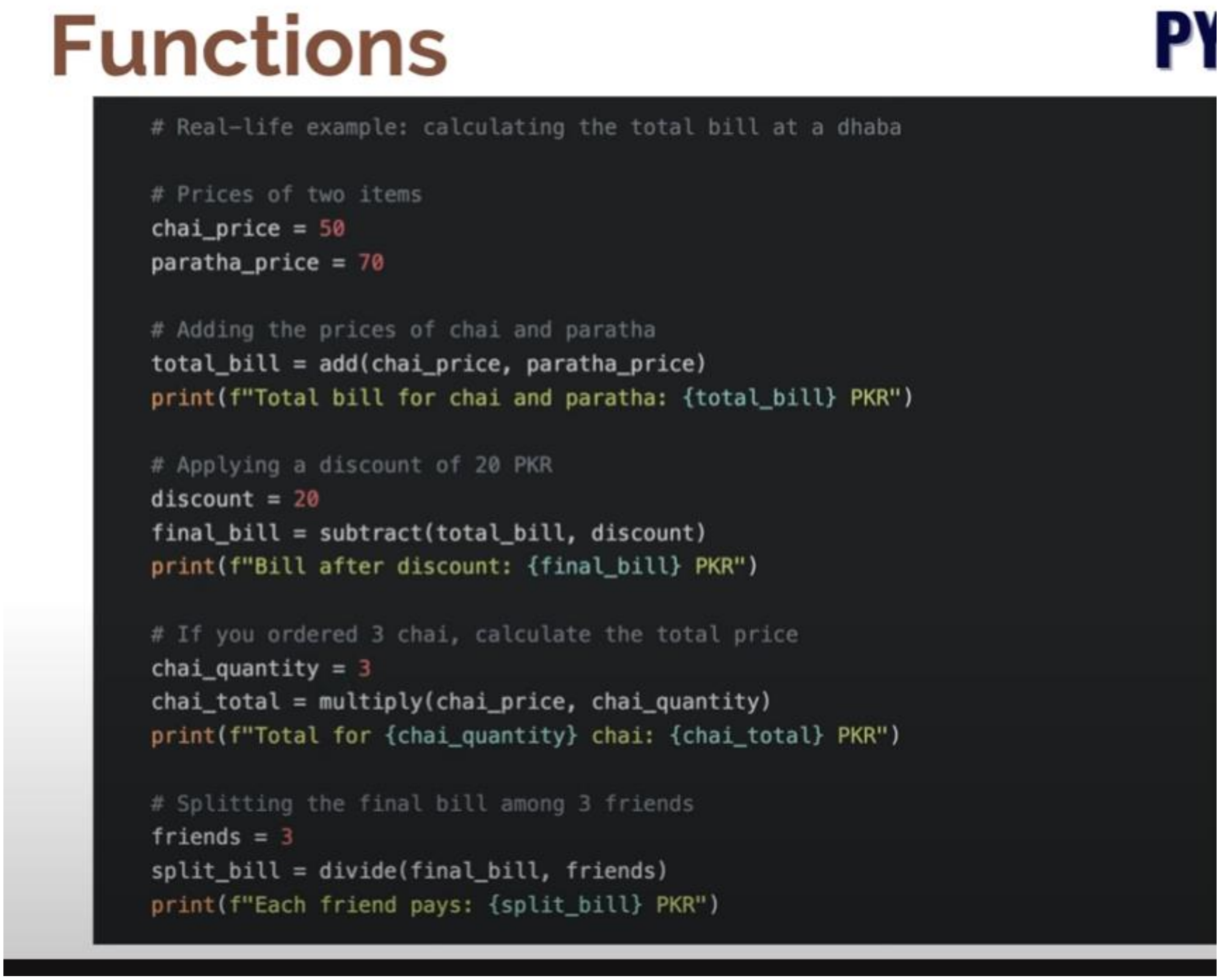


```
# Function for addition (e.g., adding the price of two items)
def add(item1, item2):
    return item1 + item2

# Function for subtraction (e.g., applying a discount to the total bill)
def subtract(total, discount):
    return total - discount

# Function for multiplication (e.g., calculating the total price for multiple items
def multiply(price, quantity):
    return price * quantity

# Function for division (e.g., splitting the bill among friends)
def divide(total, friends):
    return total / friends
```



Functions

```
# Real-life example: calculating the total bill at a dhaba

# Prices of two items
chai_price = 50
paratha_price = 70

# Adding the prices of chai and paratha
total_bill = add(chai_price, paratha_price)
print(f"Total bill for chai and paratha: {total_bill} PKR")

# Applying a discount of 20 PKR
discount = 20
final_bill = subtract(total_bill, discount)
print(f"Bill after discount: {final_bill} PKR")

# If you ordered 3 chai, calculate the total price
chai_quantity = 3
chai_total = multiply(chai_price, chai_quantity)
print(f"Total for {chai_quantity} chai: {chai_total} PKR")

# Splitting the final bill among 3 friends
friends = 3
split_bill = divide(final_bill, friends)
print(f"Each friend pays: {split_bill} PKR")
```