Dichi Academy
Data Science Module 1 - Data with Python

Function

Table of Content

DICHI ACADEMY

What is function?

DICHI ACADEMY

Find the sum of integers from
“1 to 10”, “20 to 37”, and “35 to 49”

```python
sum = 0
for i in range(1, 11):
    sum += i
print("Sum from 1 to 10 is", sum)
```

```python
sum = 0
for i in range(20, 38):
    sum += i
print("Sum from 20 to 37 is", sum)
```

```python
sum = 0
for i in range(35, 50):
    sum += i
print("Sum from 35 to 49 is", sum)
```

```python
def sum(i1, i2):
    result = 0
    for i in range(i1, i2 + 1):
        result += i

    return result
```

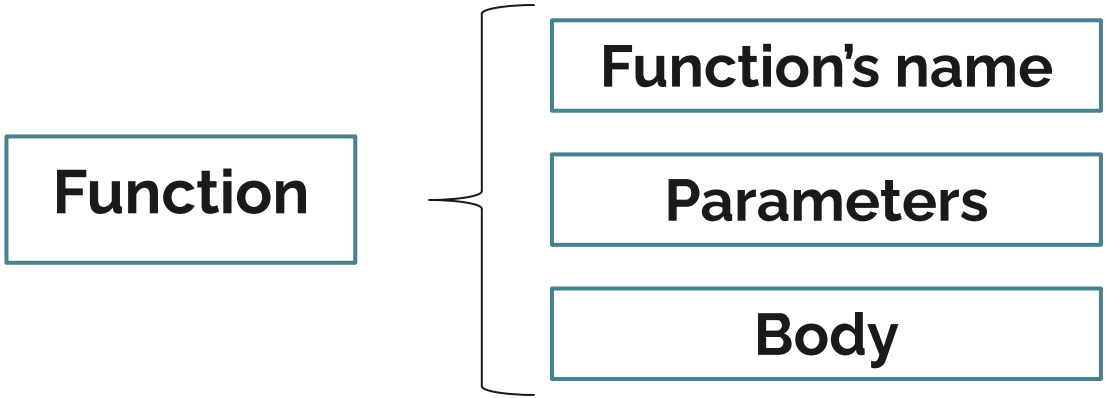
```python
def main():
    print("Sum from 1 to 10 is", sum(1, 10))
    print("Sum from 20 to 37 is", sum(20, 37))
    print("Sum from 35 to 49 is", sum(35, 49))

main() # Call the main function
```

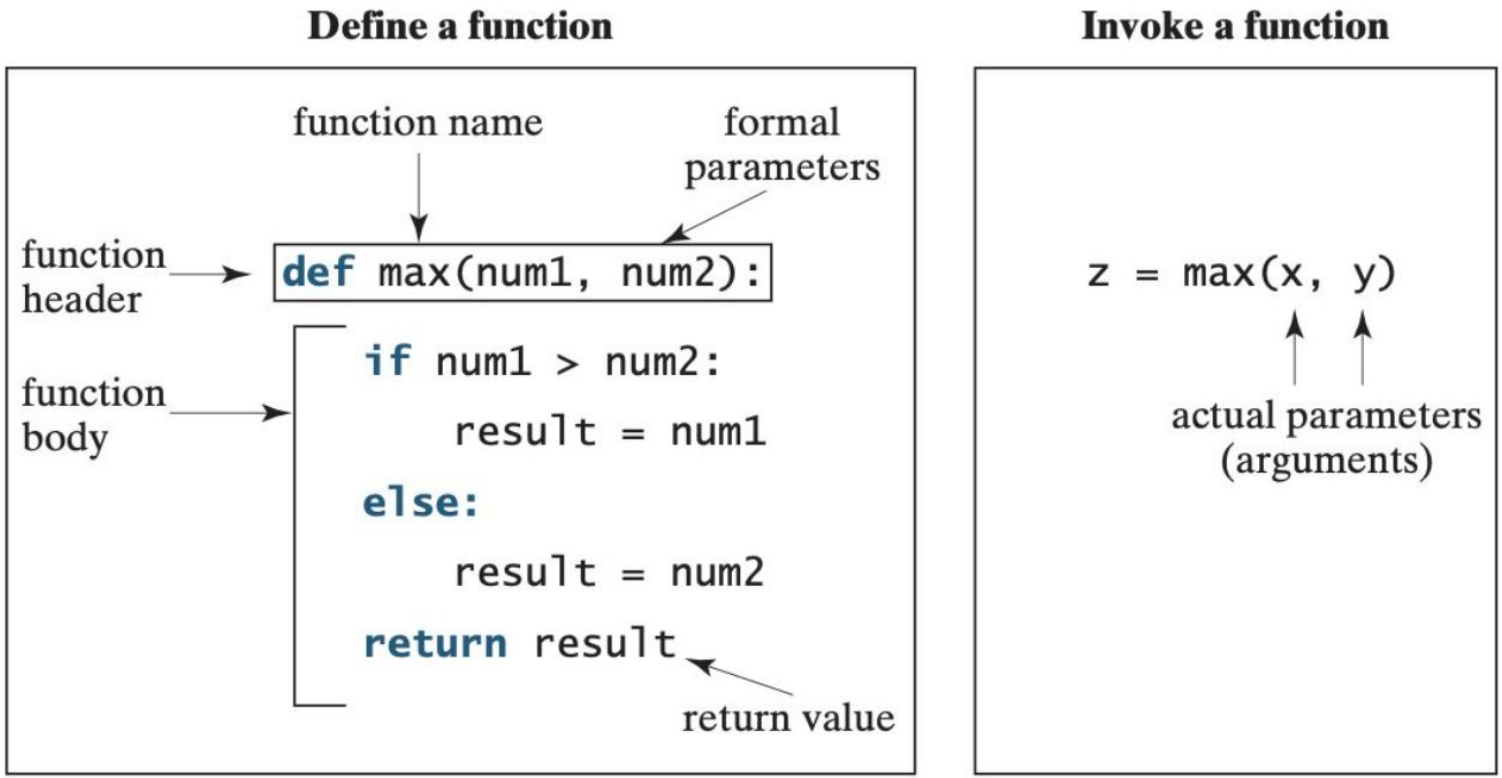
What is function?

DICHI ACADEMY

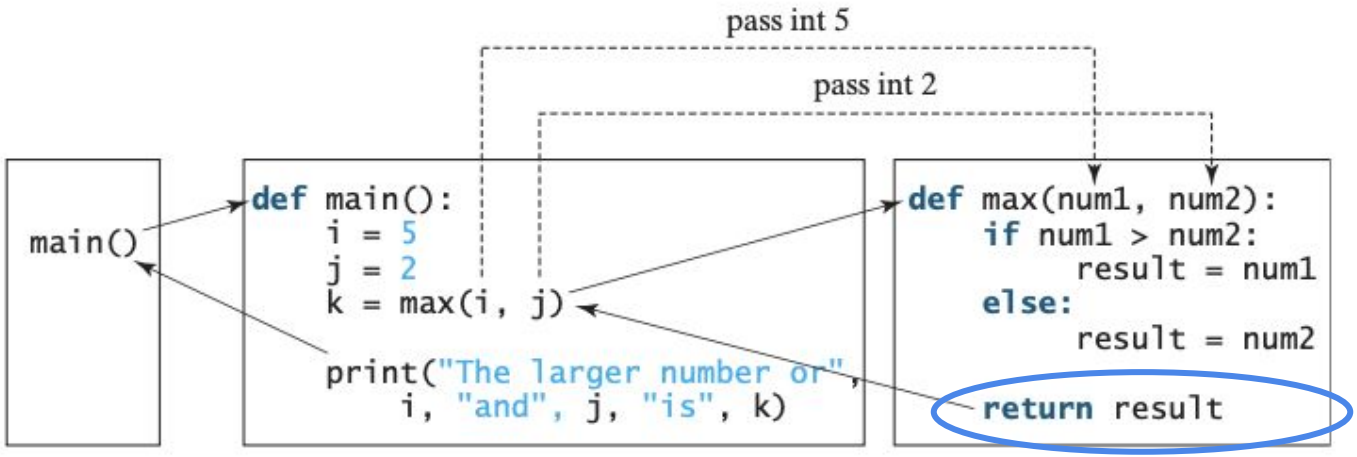
Functions can be used to define **reusable code** and organize and simplify code.



Define and Invoke Function



How function works?



??

return a **value**

How function works?

DICHI ACADEMY

```python
# Print grade for the score
def printGrade(score):
    if score >= 90.0:
        print('A')
    elif score >= 80.0:
        print('B')
    elif score >= 70.0:
        print('C')
    elif score >= 60.0:
        print('D')
    else:
        print('F')

def main():
    score = eval(input("Enter a score: "))
    print("The grade is ", end = " ")
    printGrade(score)

main() # Call the main function
```

```python
# Return the grade for the score
def getGrade(score):
    if score >= 90.0:
        return 'A'
    elif score >= 80.0:
        return 'B'
    elif score >= 70.0:
        return 'C'
    elif score >= 60.0:
        return 'D'
    else:
        return 'F'

def main():
    score = eval(input("Enter a score: "))
    print("The grade is", getGrade(score))

main() # Call the main function
```

Return Multiple Values

DICHI ACADEMY

```
def sort(number1, number2):
    if number1 < number2:
        return number1, number2
    else:
        return number2, number1

n1, n2 = sort(3, 2)
print("n1 is", n1)
print("n2 is", n2)
```

Practice 1: Add Function

Write a program with a function to add two numbers

add(num1, num2)

The program allows users to input two values. Then, the program calls the function in the same file to perform the addition using the two input values from users.

```
add(10, 5) # Output: 15
add(20, 2) # Output: 22
```

Scope of variables

```
def main():
    x = 1
    print("Before the call, x is", x)
    increment(x)
    print("After the call, x is", x)

def increment(n):
    n += 1
    print("\tn inside the function is", n)

main() # Call the main function
```

```
x = 1
def increase():
    global x
    x =  x + 1
    print(x) # Displays 2

increase()
print(x) # Displays 2
```

Code Modularization

DICHI ACADEMY

- Modularizing makes code easy to maintain and debug, and enables the code to be reused.
- Functions can be used to reduce redundant code and enable code reuse.
- Functions can also be used to modularize code and improve a program's quality.

Code Modularization

DICHI ACADEMY

Function (increment.py)

```python
x = 1
print("Before the call, x is", x)
increment(x)
print("After the call, x is", x)
```

```python
x=0
print("x = ", x)
y = x
z = y+1
increment(x)
z = y+2
print("z = ", z)
```

```python
def increment(n):
    n += 1
    print("\tn inside the function is", n)
```

Code Modularization

DICHI ACADEMY

main.py

```python
from addition import sum

x = 1
y = 2

res = sum(x,y)
print("Result is", res)
```

addition.py

```python
def sum(a,b):
  y = a + b
  return y
```

Practice 2: Multiply Function

Write a program with a function to multiply two numbers

multiply(num1, num2)

The program allows users to input two values. Then, the program calls the function module (multiplication.py) to perform the addition using the two input values from users.

```
multiply(10, 2) # Ouput: 20
multiply(5, 3) # Ouput: 15
```

Common confusions in function

```
def main():
    x = 1
    print("Before the call, x is", x)
    increment(x)
    print("After the call, x is", x)

def increment(n):
    n += 1
    print("\tn inside the function is", n)

main() # Call the main function
```

Common confusions in function

```python
globalVar = 1

def f1():
  localVar = 2
  print(globalVar)
  print(localVar)

f1()
print(globalVar)
print(localVar)
```

Out of Scope Error

Common confusions in function

```
def print_text():
  x = 4
  y = 2
  print(x)
  print(y)

print_text()
print(x)
print(y)
```

Common confusions in function

```
def printy(x):
  if x < 0:
    y = -1
  else:
    y = 1


z = printy(10)
print('y is', z)
```

Default Arguments

DICHI ACADEMY

```
def printArea(width = 1, height = 2):
    area = width * height
    print("width:", width, "\theight:", height, "\tarea:", area)
```

```
printArea() # Default arguments width = 1 and height = 2
printArea(4, 2.5) # Positional arguments width = 4 and height = 2.5
printArea(height = 5, width = 3) # Keyword arguments width
printArea(width = 1.2) # Default height = 2
printArea(height = 6.2) # Default width = 1
```

Practice 3: Calculate the Area of a Triangle

Write a Python function **triangle_area( )** to calculate the area of a triangle using its base and height.

```
triangle_area(10, 5)  # Output: 25.0
triangle_area(7, 3)   # Output: 10.5
```

The formula to calculate the area of a triangle is:

$$\text{Area} = \frac{1}{2} \times \text{base} \times \text{height}$$

Practice 4: Convert Celsius to Fahrenheit

The formula to convert the temperature from Celsius to Fahrenheit is:

$$\text{Fahrenheit} = (\text{Celsius} \times \frac{9}{5}) + 32$$

Write a Python function **celsius_to_fahrenheit( )** to convert a temperature from Celsius to Fahrenheit.

```
celsius_to_fahrenheit(0)    # Output: 0°C is equal to 32.0°F
celsius_to_fahrenheit(100)  # Output: 100°C is equal to 212.0°F
celsius_to_fahrenheit(37)   # Output: 37°C is equal to 98.6°F
```

Practice 5: Find the Greatest of Three Numbers

Write a Python function to find the largest number among three given numbers.

```
find_largest(5, 10, 3)   # Output: The largest number among 5, 10, 3 is 10
find_largest(-4, -1, -7) # Output: The largest number among -4, -1, -7 is -1
find_largest(8, 8, 2)    # Output: The largest number among 8, 8, 2 is 8
```

Thank You
for
Your Attention !

DICHI

ACADEMY