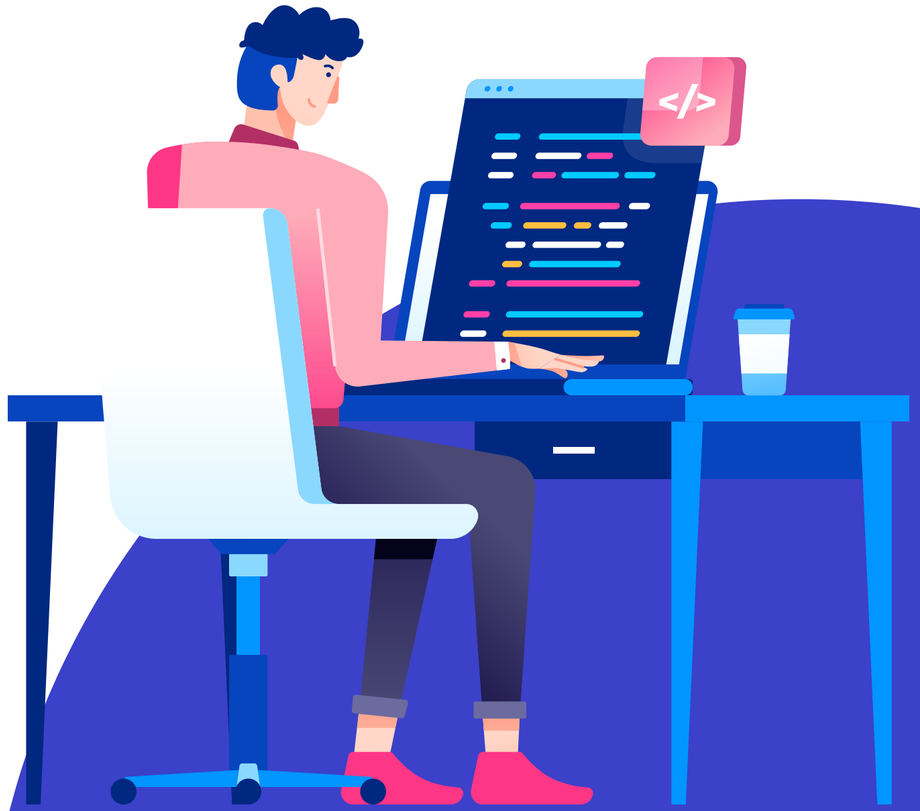
SIT325 การเขียนโปรแกรมคอมพิวเตอร์

ครูน้อยดอทคอม

PYTHON

ONLINE COURSE

บทที่ 9: ฟังก์ชัน (Functions)



ฟังก์ชัน (Function) คือ บล็อกของโค้ดที่ถูกออกแบบมาเพื่อทำงานเฉพาะอย่าง เมื่อเราเรียกใช้ฟังก์ชัน โค้ดภายในฟังก์ชันจะถูกดำเนินการ ฟังก์ชันช่วยให้เราสามารถจัดระเบียบโค้ดได้ดีขึ้น ทำให้โค้ดอ่านง่ายและนำกลับมาใช้ใหม่ได้

การสร้างฟังก์ชัน

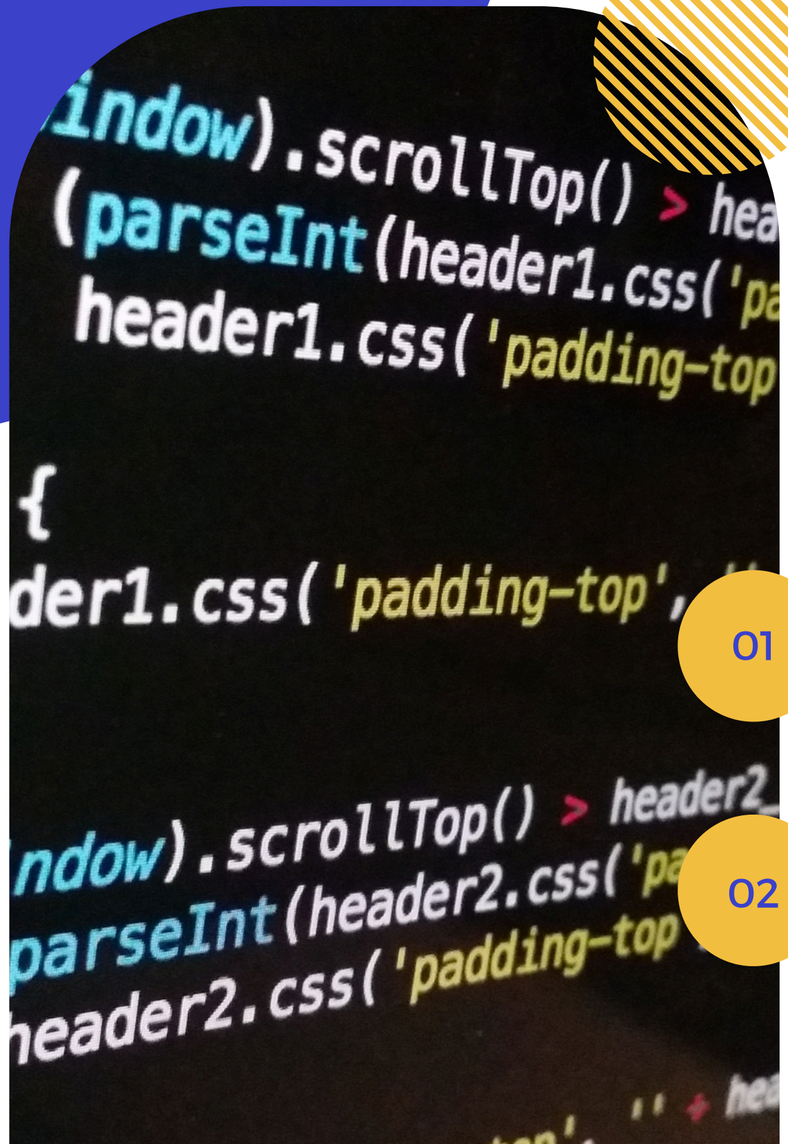
เราสามารถสร้างฟังก์ชันได้โดยใช้คำสั่ง def ตามด้วยชื่อฟังก์ชัน วงเล็บ และเครื่องหมาย colon (:)

```
def function_name(parameters):
    """เอกสารประกอบฟังก์ชัน (docstring)"""
    # คำสั่งภายในฟังก์ชัน return value #
    คืนค่า (ถ้ามี)
```

พารามิเตอร์และอาร์กิวเมนต์

- **พารามิเตอร์ (Parameter):** ตัวแปรที่อยู่ในนิยามของฟังก์ชัน ใช้สำหรับรับค่าจากภายนอกฟังก์ชัน
- **อาร์กิวเมนต์ (Argument):** ค่าที่ส่งเข้าไปในฟังก์ชันเมื่อเรียกใช้

```
def greet(name): # name เป็นพารามิเตอร์
"""ฟังก์ชันนี้ทักทายผู้ใช้"""
 print("Hello,", name)
greet("Alice") # "Alice" เป็นอาร์กิวเมนต์
```



ค่าที่ส่งกลับจากฟังก์ชัน

ฟังก์ชันสามารถส่งค่ากลับไปยังจุดที่เรียกใช้ได้โดยใช้คำสั่ง return

01

02

```
def add(x, y):
"""ฟังก์ชันนี้บวกเลขสองจำนวน"""
result = x + y
return result
sum = add(5, 3) # sum จะมีค่าเป็น 8
```

ตัวอย่างการใช้งาน

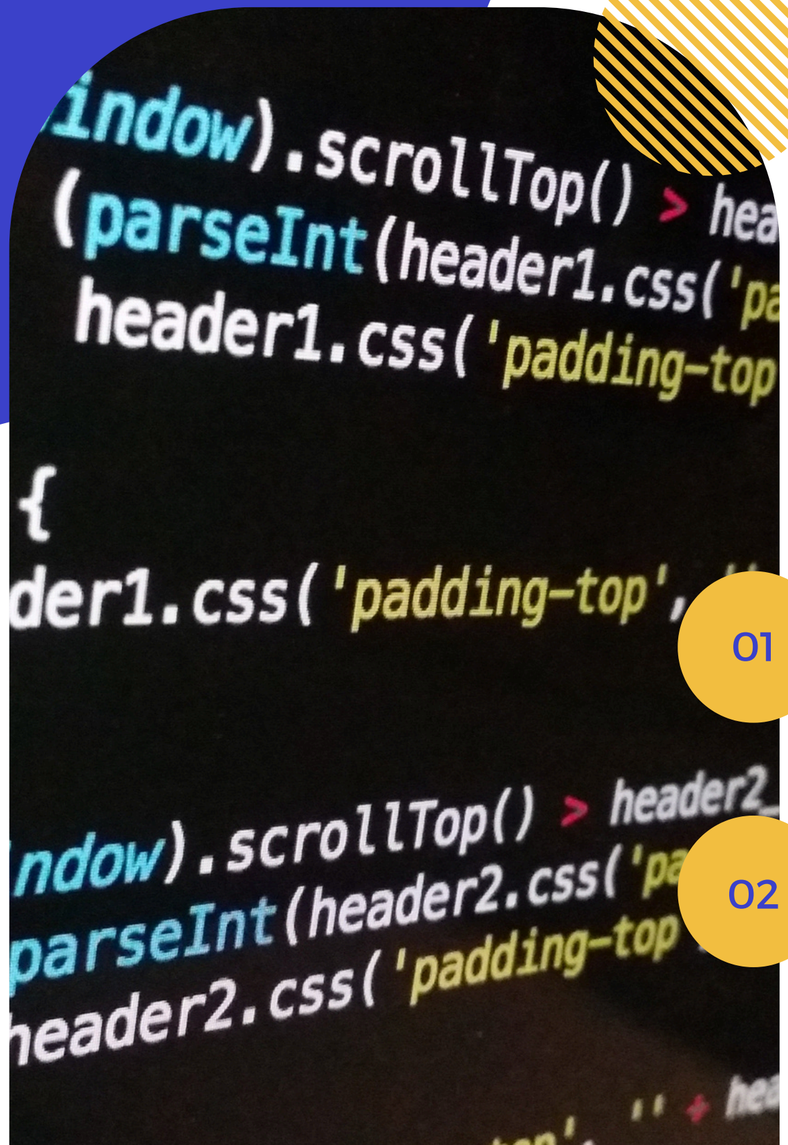
```
def calculate_area(width, height):
"""ฟังก์ชันนี้คำนวณพื้นที่สี่เหลี่ยม"""
area = width * height
return area
width = 10
height = 5
area = calculate_area(width, height)
print("Area:", area) # Output: Area: 50
```

Example

สร้างฟังก์ชัน is_even() ที่รับตัวเลขเป็นพารามิเตอร์ และคืนค่า True ถ้าเลขนั้นเป็นเลขคู่ มิฉะนั้นคืนค่า False

```
def is_even(num):
"""ฟังก์ชันนี้ตรวจสอบว่าเลขเป็นเลขคู่หรือไม่"""
 if num % 2 == 0:
   return True
else:
   return False
```

```
# เรียกใช้ฟังก์ชัน
 number = 10
if is_even(number):
    print(number, "is even")
else:
    print(number, "is odd")
```



Example

สร้างฟังก์ชัน calculate_area() ที่รับความกว้างและความสูงของสี่เหลี่ยมเป็นพารามิเตอร์ และคืนค่าพื้นที่ของสี่เหลี่ยม

```
def calculate_area(width, height):
    """ฟังก์ชันนี้คำนวณพื้นที่สี่เหลี่ยม"""
    area = width * height
return area
```

```
# เรียกใช้ฟังก์ชัน
width = 5
height = 8
area = calculate_area(width, height)
print("The area of the rectangle is:", area)
```

Example

สร้างฟังก์ชัน factorial() ที่รับจำนวนเต็มบวกเป็นพารามิเตอร์ และคืนค่าแฟกทอเรียลของจำนวนนั้น

```
def factorial(n):
    """ฟังก์ชันนี้คำนวณแฟกทอเรียล"""
    if n == 0:
return 1
else:
return n * factorial(n - 1)
# เรียกใช้ฟังก์ชัน
number = 5
result = factorial(number)
print("The factorial of", number, "is:", result)
```

Lambda Function ใน Python

Lambda function หรือ anonymous function คือ ฟังก์ชันขนาดเล็กที่ไม่มีชื่อ **ถูกนิยามในบรรทัดเดียว** และมักใช้สำหรับงานที่ไม่ซับซ้อนหรือต้องการใช้งานฟังก์ชันเพียงครั้งเดียว

lambda arguments: expression

Lambda Function ใน Python

- lambda: คีย์เวิร์ดที่ใช้บอกว่านี่คือ lambda function
- arguments: รายการของพารามิเตอร์ที่ฟังก์ชันรับ (สามารถมีได้หลายตัว หรือไม่มีก็ได้)
- expression: นิพจน์ที่ฟังก์ชันจะคำนวณและคืนค่ากลับ (ต้องมีเพียงนิพจน์เดียว)

Lambda Function ใน Python

```
# ฟังก์ชันปกติที่บวกเลขสองจำนวน
def add(x, y):
    return x + y
# lambda function ที่ทำหน้าที่เดียวกัน
add_lambda = lambda x, y: x + y
# เรียกใช้ lambda function
result = add_lambda(5, 3)
print(result) # Output: 8
```