

# Table of contents

<b>1</b>		<b>2</b>
	.....	2
1.1	.....	2
1.2	.....	2
1.3	.....	3
1.3.1	.....	3
1.3.2	.....	3
1.4	.....	4
1.4.1	.....	4
1.4.2	.....	4
1.5	.....	4
1.6	.....	5
1.7	.....	5
1.7.1	*args .....	5
1.7.2	**kwargs .....	5
1.8	.....	6
1.8.1	.....	6
1.8.2	global .....	6
1.9	return .....	6
1.10	.....	7
1.11	.....	7
	.....	8
1.11.1	1 .....	8
1.11.2	2 .....	9
1.11.3	3 .....	9
	.....	9
1.11.1	1 .....	9
1.11.2	2 .....	10
1.11.3	3 .....	10
1.12	.....	10
1.13	.....	10
1.14	.....	11
	.....	11

# 1

- 
- 
- 
- 
- 

## 1.1

- 
- 
- 
- 

```
#  
def greet(name):  
    return f"    {name}  "  
  
message = greet(" ")  
print(message)  #  :
```

## 1.2

```
def  ( ):
    """ """
    #
    return  #
```

- def
- Python
- 
- :
- 
- return

## 1.3

### 1.3.1

```
def say_hello():
    print("Hello, World!")

say_hello()  #
```

### 1.3.2

```
def get_pi():
    return 3.14159

pi_value = get_pi()
print(f" {pi_value} ")
```

## 1.4

### 1.4.1

```
def square(number):  
    return number * number  
  
result = square(5)  
print(result)  # : 25
```

### 1.4.2

```
def add_numbers(a, b):  
    return a + b  
  
sum_result = add_numbers(10, 20)  
print(sum_result)  # : 30
```

## 1.5

```
def greet(name, greeting=" "):  
    return f"{greeting} {name} "  
  
print(greet(" "))  #  
print(greet(" ", " "))  #
```

```
def create_profile(name, age=25, city=" "):  
    return f"{name} {age} {city} "  
  
print(create_profile(" "))  
print(create_profile(" ", 30))  
print(create_profile(" ", 28, " "))
```

## 1.6

```
def describe_pet(name, animal_type, age):
    return f"{name} {age} {animal_type} "

#
print(describe_pet(" ", " ", 3))

#
print(describe_pet(name=" ", animal_type=" ", age=2))
print(describe_pet(age=1, name=" ", animal_type=" "))
```

## 1.7

### 1.7.1 \*args

```
def sum_all(*numbers):
    total = 0
    for num in numbers:
        total += num
    return total

print(sum_all(1, 2, 3))          # 6
print(sum_all(1, 2, 3, 4, 5))   # 15
```

### 1.7.2 \*\*kwargs

```
def create_student(**info):
    for key, value in info.items():
        print(f"{key}: {value}")

create_student(name=" ", age=20, major=" ")
```

## 1.8

### 1.8.1

```
x = 10 #

def test_scope():
    x = 20 #
    print(f" : {x}")

test_scope()          # : 20
print(f" : {x}")      # : 10
```

### 1.8.2 global

```
counter = 0 #

def increment():
    global counter
    counter += 1

increment()
print(counter) # : 1
```

## 1.9 return

```
def get_name_length(name):
    return len(name)

def get_user_info():
    return " ", 25, " " #

def is_even(number):
    return number % 2 == 0 #
```

```
name, age, job = get_user_info()
```

```
def calculate_grade(score):  
    if score >= 90:  
        return "A"  
    elif score >= 80:  
        return "B"  
    elif score >= 70:  
        return "C"  
    else:  
        return "D"
```

## 1.10

```
def calculate_area(length, width):  
    """  
  
    Args:  
        length (float):  
        width (float):  
  
    Returns:  
        float:  
    """  
    return length * width  
  
#  
print(calculate_area.__doc__)
```

## 1.11

```

def add(a, b):
    """    """
    return a + b

def subtract(a, b):
    """    """
    return a - b

def multiply(a, b):
    """    """
    return a * b

def divide(a, b):
    """    """
    if b != 0:
        return a / b
    else:
        return " : "

#
result1 = add(10, 5)      # 15
result2 = subtract(10, 5) # 5
result3 = multiply(10, 5) # 50
result4 = divide(10, 5)   # 2.0

```

### 1.11.1 1

```

def celsius_to_fahrenheit(celsius):
    #
    pass

def fahrenheit_to_celsius(fahrenheit):
    #
    pass

#

```



```
print(celsius_to_fahrenheit(0))    # 32
print(fahrenheit_to_celsius(32))  # 0
```

### 1.11.2 2

```
def is_valid_password(password):
    # 8
    #
    pass

#
print(is_valid_password("abc123"))    # False ( )
print(is_valid_password("abcdefgh"))  # False ( )
print(is_valid_password("abc12345"))  # True
```

### 1.11.3 3

```
def calculate_stats(numbers):
    #
    pass

#
stats = calculate_stats([1, 2, 3, 4, 5])
print(stats)  # {'min': 1, 'max': 5, 'average': 3.0}
```

### 1.11.1 1

```
def celsius_to_fahrenheit(celsius):
    return (celsius * 9/5) + 32

def fahrenheit_to_celsius(fahrenheit):
    return (fahrenheit - 32) * 5/9
```

### 1.11.2 2

```
def is_valid_password(password):
    if len(password) < 8:
        return False

    has_letter = any(char.isalpha() for char in password)
    has_number = any(char.isdigit() for char in password)

    return has_letter and has_number
```

### 1.11.3 3

```
def calculate_stats(numbers):
    if not numbers:
        return None

    return {
        'min': min(numbers),
        'max': max(numbers),
        'average': sum(numbers) / len(numbers)
    }
```

## 1.12

1. : calculate\_tax() calc()
2. :
3. :
4. :
5. :
6. :

## 1.13

- 
- 
-

•  
•  
•  
•

## 1.14

- 1.
- 2.
3. `*args **kwargs`
- 4.
5. 3