

Multiple Parameters in Python Functions

Textbook

Multiple Parameters in Python Functions



Functions can accept more than one parameter. Let's explore how this is done with positional arguments.

Positional Arguments

```
1 def gifts(first, second):
2     print("Your first choice for a birthday gift would be " + first)
3     print("Your second choice for a birthday gift would be " + second)
4
5
6 gifts("bike", "basketball")
7 gifts("speaker", "tickets")
8
9
10
```

Try it!

Each item in the function call knows where to go inside the function declaration because of its position. These are considered [positional arguments](#).

Experiment to see what can happen with 3 or more parameters!

Keyword Arguments

You can also use [keyword arguments](#). Keyword arguments slot into the function based on the position of where they appear in the function call.

```
1 def gifts(first, second):
2     print("Your first choice for a birthday gift would be " + first)
3     print("Your second choice for a birthday gift would be " + second)
4
5
6 gifts(second = "ball", first = "bike")
7
```

Try it!

This allows you to be able to place your arguments in the order that you'd like.

The arguments can also go in the function declaration

```
1 def gifts(first = "bike", second = "ball"):
2     print("Your first choice for a birthday gift would be " + first)
3     print("Your second choice for a birthday gift would be " + second)
4
5
6 gifts()
7
```

Try it!

Note: If you use arguments in the function declaration AND in the function call, the program will run the items in the function call.

```
1 def gifts(first = "bike", second = "ball"):
2     print("Your first choice for a birthday gift would be " + first)
3     print("Your second choice for a birthday gift would be " + second)
4
5
6 gifts("veggies", "soap")
7
```

Try it!

Let's try a few exercises to explore different Positional and Keyword Arguments

What is the expected output of this code?

```
1 def myfunction(total, age=10, distance=6, is_present=True):
2     return([is_present, age, total, distance])
3
4 print(myfunction(2)[3])
```

Try it!

[Show answer/example](#)

[Show answer/example](#)

What is the expected output of this code?

```
1 def myfunction(total, age=10, distance=6, is_present=True):
2     return([is_present, age, total, distance])
3
4 print(myfunction(age=7, total=3)[1])
```

Try it!

[Show answer/example](#)

[Show answer/example](#)

Local and Global Scope Variables

Sometimes you want to create a variable *inside* a function. This is called a [local variable](#). Local variables can only be used inside the function.

Here is an example of a local variable.

```
1 favorite = "I love juice"
2 def myfunction():
3     fruit = "apple"
4     print(fruit)
5
6 myfunction()
7
8 print(favorite)
```

In this example, `fruit` is the local variable.

Variables that are created outside functions are called [global variables](#). Global variables can be used anywhere in the program.

In the above example, `favorite` is the global variable.

If you want a variable inside a function to be global, you can use the `global` keyword.

```
1 favorite = "I love juice"
2 def myfunction():
3     global fruit
4     fruit = "apple"
5     print(fruit)
6
7 myfunction()
8
9 print(fruit)
```

Try it!

Now the variable named `fruit` can be used in other places in the program.

Using a List as a Parameter

You can actually pass an entire list as a parameter

```
1 favorite = ["red", "green", "orange", "yellow"]
2
3 def myfunction(first):
4
5     first.append("purple")
6     print(favorite)
7
8 myfunction(favorite)
9
```

Try it!

This will print out the list with the value `purple` appended to the end.

```
['red', 'green', 'orange', 'yellow', 'purple']
```

If the **entire** list variable gets reassigned inside the function, it will not affect the original list, since it will have created a new list.

```
1 favorite = ["red", "green", "orange", "yellow"]
2
3 def myfunction(first):
4     first = [3, 4, 5, 6]
5
6 myfunction(favorite)
7
8 print(favorite)
```

Try it!

To be clear, if parts of the list are added or adjusted from inside the function, it will update the original global variable list.

If the entire list variable gets reassigned, the original global variable will not update.

Checkpoint

Multiple Parameters in Python Functions

Create a function that has **two** parameters.

Inside the function, print each parameter in a concatenated string.

Call the function **twice** with different **parameters**.

Requirements:

- Create a function that has two parameters.
- Inside the function, print each parameter in a concatenated string.
- Call the function twice with different parameters.

Questions (10)

1. True or False: Python functions cannot accept multiple parameters.

MULTIPLE CHOICE

Choose the correct answer:

- A. True
- B. False

2. Edit the text box below to debug (fix) the code:

DEBUG CODE

Code to Debug:

```
1 def eggs(4, second):  
2     print(first * second)  
3  
4 eggs(4, 5)
```

3. True or False: You can have 3 or more parameters in a Python function.

MULTIPLE CHOICE

Choose the correct answer:

- A. True
- B. False

4. What will the following code print out?

```
def breads(first, second): print("I will have to go buy some " + first + " and " + second) breads("rolls", "baguettes")
```

Choose the correct answer:

- A. I will have to go buy some rolls and second
- B. I will have to go buy some rolls and baguettes
- C. I will have to go buy some baguettes and rolls
- D. I will have to go buy some first and second

5. What will the following code print out?

```
def breads(first, second): print("I will have to go buy some " + first + " and " + "second") breads("rolls", "baguettes")
```

Choose the correct answer:

- A. I will have to go buy some rolls and second
- B. I will have to go buy some rolls and baguettes
- C. I will have to go buy some first and second
- D. I will have to go buy some first and baguettes

6. Edit the text box below to debug (fix) the code:**Code to Debug:**

```
1 def breads(first, second):
2     print("I will have to go buy some " + first + " and " + second)
3
4
5 breads("rolls" "baguettes")
```

7. Edit the text box below to debug (fix) the code:**Code to Debug:**

```
1 deff eggs(first, second):
2     print(first * second)
3
4 eggs(4, 5)
```

8. What is a variable called if it's created inside a function?**Choose the correct answer:**

- A. local variable
- B. global variable
- C. function variable
- D. personal variable

9. In the code example, what kind of variable is the variable named class?

```
class = "I have chemistry class" def myfunction(): homework = "essay" print(homework) myfunction() print(favorite)
```

Choose the correct answer:

- A. local variable
- B. global variable
- C. personal variable
- D. function variable

10. What are variables called if they are created outside the function?**Choose the correct answer:**

- A. local variable
- B. global variable
- C. function variable
- D. personal variable

Challenges (6)

1. Hungry for Apples 3

1. Despite our best efforts the user is now asking for **3 different kinds of fruits!** But just like last time we can't ask them for an input inside the function!
2. This time around ask the user for three different fruits they want to eat!
3. Have the user input them outside the function.
4. Three SEPARATE inputs are required.
5. Then declare the function, passing in all three of those data members as parameters.
6. Print them all in one statement within the function and the user's hunger will finally be satisfied.
7. Be sure to call the function!

Keep your eyes out for Hungry for Apples 1, and 2! Unless you already finished them, then good job!

For example:

Inputs: `Apple` , `Orange` , `Banana`

Output: `AppleOrangeBanana`

Another example:

Inputs: `mango` , `papaya` , `coconut`

Output: `mangopapayacoconut`

2. Find the Smallest Integer Function

Write a function that takes in two integers as inputs.

These **inputs** will be the function parameters.

The function will then print the smallest number.

For example:

Inputs: `10` , `3`

Output: `3`

Another example:

Inputs: `4` , `5`

Output: `4`

3. How Many are Equal Function

1. Write a function that takes three integers as inputs.
2. These inputs become your parameters.
3. Your function will see which integers are the same and print out how many are the same.
4. If all the numbers are the same, print `3`.
5. If two are the same, print `2`.
6. If they're all different print `0`.

For example:

Inputs: `7` , `5` , `7`

Output: `2`

Another example:

Inputs: `8` , `8` , `8` ,

Output: `3`

4. Find the Smallest of Five Function

Write a function that takes in five integers as inputs.

These inputs will become parameters for a function.

The function will then print the **smallest of the five numbers**.

For example:

Inputs: `5` , `2` , `9` , `7` , `6`

Output: `2`

Another example:

Inputs: `2` , `1` , `10` , `9` , `18`

Output: `1`

Hint: try using a replacing variable to simplify your code.

5. Spot the Difference Function

1. Write a function that takes in three integers as input.
2. These inputs will become the parameters.
3. Two numbers will be the same, and one will be different.
4. Print the order **(1, 2 or 3)** of the different number.
5. For example, if the input is `7` , `5` , `7` your program should print `2` because the different number is in position 2.

For example:

Inputs: `7` , `5` , `7`

Output: `2`

Another example:

Inputs: `12` , `12` , `8`

Output: `3`

6. What Day is it Tomorrow Function

1. Write a function that takes in two inputs - a month (1-12) and a date (1 - 31). These inputs will become the function's parameters.
2. It will print the day after the inputted date for the year 2021.
3. Your function should have **2 print** statements.
4. One print statement will print the month.
5. The other print statement will print the date.

If the input is 6, 30 (June 30) your output should be 7, 1 (July 1).

For example:

Inputs: `6` , `30`

Outputs: `7` `1`

Another example:

Inputs: `2` , `28`

Outputs: `3` `1`

Note: Months that have 31 days: January, March, May, July, August, October, December

February had 28 days in 2021

Hint: This challenge uses many if/elif statements. Some nested within others.

Answer Keys & Solutions

Checkpoint Solutions

Multiple Parameters in Python Functions

```
1 def name (first, last) :  
2     print("The first name is " + first)  
3     print("The last name is " + last)  
4  
5  
6 name("Peter", "Wilson")  
7 name("Lucy", "Williams")
```

Questions

1. True or False: Python functions cannot accept multiple parameters.

MULTIPLE CHOICE

Correct Answer:

A. True

✗ Incorrect

B. False

✓ Correct

Explanation:

Python functions can have many parameters

2. Edit the text box below to debug (fix) the code:

DEBUG CODE

Incorrect Code:

```
1 def eggs(4, second):  
2     print(first * second)  
3  
4 eggs(4, 5)
```

Correct Solution:

```
1 def eggs(first, second):  
2     print(first * second)  
3  
4 eggs(4, 5)
```

Explanation:

The number 4 needs to say something else

3. True or False: You can have 3 or more parameters in a Python function.

MULTIPLE CHOICE

Correct Answer:

A. True ✓ Correct

B. False ✗ Incorrect

Explanation:

Python functions can have many parameters

4. What will the following code print out?

MULTIPLE CHOICE

Correct Answer:

A. I will have to go buy some rolls and second ✗ Incorrect

B. I will have to go buy some rolls and baguettes ✓ Correct

C. I will have to go buy some baguettes and rolls ✗ Incorrect

D. I will have to go buy some first and second ✗ Incorrect

Explanation:

The strings "rolls" and "baguettes" will go into the print statement

5. What will the following code print out?

MULTIPLE CHOICE

Correct Answer:

A. I will have to go buy some rolls and second ✓ Correct

B. I will have to go buy some rolls and baguettes ✗ Incorrect

C. I will have to go buy some first and second ✗ Incorrect

D. I will have to go buy some first and baguettes

✖ Incorrect

Explanation:

The last word in the print statement is actually a string and will print exactly.

6. Edit the text box below to debug (fix) the code:

DEBUG CODE

Incorrect Code:

```
1 def breads(first, second):
2     print("I will have to go buy some " + first + " and " + second)
3
4
5 breads("rolls" "baguettes")
```

Correct Solution:

```
1 def breads(first, second):
2     print("I will have to go buy some " + first + " and " + second)
3
4
5 breads("rolls", "baguettes")
```

Explanation:

There's a missing comma

7. Edit the text box below to debug (fix) the code:

DEBUG CODE

Incorrect Code:

```
1 deff eggs(first, second):
2     print(first * second)
3
4 eggs(4, 5)
```

Correct Solution:

```
1 def eggs(first, second):
2     print(first * second)
3
4 eggs(4, 5)
```

Explanation:

deff doesn't have 2 f's

8. What is a variable called if it's created inside a function?

MULTIPLE CHOICE

Correct Answer:

- | | |
|----------------------|-------------|
| A. local variable | ✓ Correct |
| B. global variable | ✗ Incorrect |
| C. function variable | ✗ Incorrect |
| D. personal variable | ✗ Incorrect |

Explanation:

It's used locally, in that function

9. In the code example, what kind of variable is the variable named class?

MULTIPLE CHOICE

Correct Answer:

- | | |
|----------------------|-------------|
| A. local variable | ✗ Incorrect |
| B. global variable | ✓ Correct |
| C. personal variable | ✗ Incorrect |
| D. function variable | ✗ Incorrect |

Explanation:

Global variables can be used anywhere

10. What are variables called if they are created outside the function?

MULTIPLE CHOICE

Correct Answer:

- | | |
|--------------------|-------------|
| A. local variable | ✗ Incorrect |
| B. global variable | ✓ Correct |

C. function variable

✗ Incorrect

D. personal variable

✗ Incorrect

Explanation:

These variables can be used anywhere

Challenges

1. Hungry for Apples 3

Solution:

```
1 fruit1 = input("Please enter the name of the fruit you would like: ")
2 fruit2 = input("Please enter the name of the fruit you would like: ")
3 fruit3 = input("Please enter the name of the fruit you would like: ")
4
5 def hungryForApples(fruit1,fruit2,fruit3):
6     print(fruit1 + fruit2 + fruit3)
7
8 hungryForApples(fruit1,fruit2,fruit3)
```

2. Find the Smallest Integer Function

Solution:

```
1 choice1 = int(input("What is the first number?"))
2 choice2 = int(input("What is the second number?"))
3
4 def my_function(first, second):
5     if first < second:
6         print(first)
7     else:
8         print(second)
9
10
11 my_function(choice1, choice2)
```

3. How Many are Equal Function

Solution:

```
1 choice1 = int(input("What is the first number?"))
2 choice2 = int(input("What is the second number?"))
3 choice3 = int(input("What is the third number?"))
4
```

```

5
6 def my_function(first, second, third):
7     if first == second == third:
8         print(3)
9     elif first == second or second == third or first == third :
10        print(2)
11    else:
12        print(0)
13
14
15 my_function(choice1, choice2, choice3)

```

4. Find the Smallest of Five Function

Solution:

```

1 first = int(input("Choose a number"))
2 second = int(input("Choose a number"))
3 third = int(input("Choose a number"))
4 fourth = int(input("Choose a number"))
5 fifth = int(input("Choose a number"))
6
7
8 def my_function(choice1, choice2, choice3, choice4, choice5):
9     smallest = choice1
10    if choice2 < choice1:
11        smallest = choice2
12    if choice3 < choice2:
13        smallest = choice3
14    if choice4 < choice3:
15        smallest = choice4
16    if choice5 < choice4:
17        smallest = choice5
18    print(smallest)
19
20
21
22 my_function(first, second, third, fourth, fifth)

```

5. Spot the Difference Function

Solution:

```

1 choice1 = int(input("What is the first number?"))
2 choice2 = int(input("What is the second number?"))
3 choice3 = int(input("What is the third number?"))
4
5
6 def my_function(first, second, third):
7     if first == second:
8         print(3)
9     elif second == third:

```



```
10     print(1)
11 else:
12     print(2)
13
14
15 my_function(choice1, choice2, choice3)
```

6. What Day is it Tomorrow Function

Solution:

```
1 first = int(input("Which month? 1-12"))
2 second = int(input("What date? 1-31"))
3
4
5 def my_function(month, date):
6     if month == 4 or month == 6 or month == 9 or month == 11:
7         if date == 30:
8             next_month = month + 1
9             print(next_month)
10            print(1)
11        else:
12            next_day = date + 1
13            print(month)
14            print(next_day)
15    elif month == 2:
16        if date == 28:
17            next_month = month + 1
18            print(next_month)
19            print(1)
20
21        else:
22            next_day = date + 1
23            print(month)
24            print(next_day)
25
26    else:
27        if date == 31:
28            next_month = month + 1
29            print(next_month)
30            print(1)
31        else:
32            next_day = date + 1
33            print(month)
34            print(next_day)
35
36
37 my_function(first, second)
```