



[Curso](#) > [Week 2...](#) > [4. Func...](#) > [Exercis...](#)

Audit Access Expires 5 de ago de 2020

You lose all access to this course, including your progress, on 5 de ago de 2020.

Exercise 1

Finger Exercises due Aug 5, 2020 20:30 -03 *Completo*

Exercise 1

6/6 points (graded)

ESTIMATED TIME TO COMPLETE: 12 minutes

Note that you will have to answer all questions before you can click the Check button.

Part 1: Function Types

For each of the following functions, specify the type of its **return**. You can assume each function is called with an appropriate argument, as specified by its docstring.

If the output can be either an int or a float, select num, which isn't a real Python type, but which we'll use to indicate that either basic numeric type is legal.

In fact, in Python, booleans True and False can be operated on as if they were the integers 1 and 0; but it is ugly and confusing to take advantage of this fact, and we will resolutely pretend that it isn't true.

What are those lines under the function definitions?

In this and future problems, you'll see function definitions that look like this:

```
def a(x):  
    ...  
    x: int or float.  
    ...  
    return x + 1
```



What are those three lines between `def a(x):` and `return x + 1`? These lines are called the *docstring* of the function. A docstring is a special type of comment that is used to document what your function is doing. Typically, docstrings will explain what the function expects the type(s) of the argument(s) to be, and what the function is returning.

In Python, docstrings appear immediately after the `def` line of a function, before the body. Docstrings start and end with triple quotes - this can be triple single quotes or triple double quotes, it doesn't matter as long as they match. To sum up this general form:

```
def my_function(argument):  
    """  
    Docstring goes here. Explain what type argument(s) should have, and what  
    your function  
    is going to return.  
    """  
    < Code for your function (the body of the function) goes here >
```

As you begin coding your own functions, we strongly encourage you to document all your functions by using properly-formatted docstrings!

1.

```
def a(x):  
    '''  
    x: int or float.  
    '''  
    return x + 1
```

Indicate the type of the output that the function `a` will yield.

num

✓ Answer: num

2.

```
def b(x):  
    '''  
    x: int or float.  
    '''  
    return x + 1.0
```

Indicate the type of the output that the function `b` will yield.

float

✓ Answer: float

3.



```
def c(x, y):  
    '''  
    x: int or float.  
    y: int or float.  
    '''  
    return x + y
```

Indicate the type of the output that the function `c` will yield.

num

✓ Answer: num

4.

```
def d(x, y):  
    '''  
    x: Can be int or float.  
    y: Can be int or float.  
    '''  
    return x > y
```

Indicate the type of the output that the function `d` will yield.

boolean

✓ Answer: boolean

5.

```
def e(x, y, z):  
    '''  
    x: Can be int or float.  
    y: Can be int or float.  
    z: Can be int or float.  
    '''  
    return x >= y and x <= z
```

Indicate the type of the output that the function `e` will yield.

boolean

✓ Answer: boolean

6.

```
def f(x, y):  
    '''  
    x: int or float.  
    y: int or float  
    '''  
    x + y - 2
```



Indicate the type of the output that the function `f` will yield.

NoneType ▼

✓ Answer: NoneType

Explanation:

The last function, unlike the other ones, does not have a return statement. It only does an operation (the operation is `x+y-2`). Since it does not explicitly return anything, Python by default returns the value `None` whose type is `NoneType`. So this function and any other that does not have a return statement can be rewritten as:

```
def f(x, y):  
    x + y - 2  
    return None
```

Enviar

i Answers are displayed within the problem

Exercise 1

Ocultar discussão

Topic: Lecture 4 / Exercise 1

Show all posts ▼

por atividade recente ▼

- ? [number 6](#) 4
[Could I trouble someone to explain why number 6 is a NoneType?](#)
- ? [Can someone explain how to know whether the answer will be an int/float/etc?](#) 3
[I forgot the rules that determine whether it will be an int or float, could someone send a link or expla...](#)
- 💬 [True/False 1/0](#) 3
[So we don't want to ever analyse the 1/0 outputs of Booleans? It seems like other languages utilise t...](#)
- ✓ [Question about #6](#) 9 new_
[Is #6 NoneType because there is no return at the end? And if there is no return, what exactly does th...](#)
- ? [num vs. int vs. float](#) 3
[I don't quite understand what a num is? I thought ints and floats covered all numbers?](#)
- ? [Does a function have to have a "return" statement inside?](#) 3
[Does a function have to have a "return" statement inside? For example in Question 6, the function re...](#)

💬 [A reminder for myself](#)



#question2 Either an int or a float add/minus/times/divided by a float, the result is a float!

3

© All Rights Reserved

