

Annotated follow-along guide_ Functions and conditional statements

October 30, 2023

1 Annotated follow-along guide: Functions and conditional statements

This notebook contains the code used in the instructional videos from [Module 2: Functions and conditional statements](#).

1.1 Introduction

This follow-along guide is an annotated Jupyter Notebook organized to match the content from each module. It contains the same code shown in the videos for the module. In addition to content that is identical to what is covered in the videos, you'll find additional information throughout the guide to explain the purpose of each concept covered, why the code is written in a certain way, and tips for running the code.

As you watch each of the following videos, an in-video message will appear to advise you that the video you are viewing contains coding instruction and examples. The in-video message will direct you to the relevant section in the notebook for the specific video you are viewing. Follow along in the notebook as the instructor discusses the code.

To skip directly to the code for a particular video, use the following links:

1. **Section ??**
2. **Section ??**
3. **Section ??**
4. **Section ??**
5. **Section ??**

1. [Define functions and returning values](#)

```
[27]: # The print() function can print text to the screen.  
print('Black dove, where will you go?')
```

Black dove, where will you go?

```
[28]: # The type() function returns an object's data type.  
number = 15
```

```
type(number)
```

[28]: int

```
[29]: # The str() function converts an object into a string.
number = str(number)

type(number)
```

[29]: str

```
[30]: # Define a function.
def greeting(name):

    print('Welcome, ' + name + '!')
    print('You are part of the team!')

greeting('Rebecca')
```

Welcome, Rebecca!
You are part of the team!

```
[31]: # Define a function to calculate the area of a triangle.
def area_triangle(base, height):
    return base * height / 2
```

```
[32]: # Use the function to assign new variables and perform calculations.
area_a = area_triangle(5, 4)
area_b = area_triangle(7, 3)
total_area = area_a + area_b
total_area
```

[32]: 20.5

```
[33]: # Define a function that converts hours, minutes, and seconds to total seconds.
def get_seconds(hours, minutes, seconds):
    total_seconds = 3600*hours + 60*minutes + seconds
    return total_seconds
```

```
[34]: # Use the function to return a result.
get_seconds(16, 45, 20)
```

[34]: 60320

2. Write clean code

```
[35]: # This code does the same thing for two different people.
name = "Marisol"
number = len(name)*9
print("Hello " + name + ". Your lucky number is " + str(number))

name = "Ardashir"
number = len(name)*9
print("Hello " + name + ". Your lucky number is " + str(number))
```

Hello Marisol. Your lucky number is 63
Hello Ardashir. Your lucky number is 72

```
[36]: # Define a function that simplifies the above code and makes it reusable.
def lucky_number(name):
    number = len(name)*9
    print("Hello " + name + ". Your lucky number is " + str(number))

lucky_number("Marisol")
lucky_number("Ardashir")
```

Hello Marisol. Your lucky number is 63
Hello Ardashir. Your lucky number is 72

```
[37]: # This code requests a number from the user and returns its factorial,
# printing each iteration of the multiplication.
a = input(); y = 1

for i in range(int(a)):
    y = y*(i+1)
    print(y)
```

8
1
2
6
24
120
720
5040
40320

```
[47]: # This function takes an integer as an input and returns its factorial.
def factorial(n):
    # Exclude 0 as product, start with 1
    y = 1
    for i in range(int(n)):
        y = y*(i+1)
    return y
```

```
# Enter a numerical value between 1-9 in the command line that appears.
input_num = input()
# Apply factorial function to an integer input
print(factorial(input_num))
```

2

2

3. Use comments to scaffold your code

```
[48]: def seed_calculator(fountain_side, grass_width):
    """
    Calculate number of kilograms of grass seed needed for
    a border around a square fountain.

    Parameters:
        fountain_side (num): length of 1 side of fountain in meters
        grass_width (num): width of grass border in meters

    Returns:
        seed (float): amount of seed (kg) needed for grass border
    """
    # Area of fountain
    fountain_area = fountain_side**2
    # Total area
    total_area = (fountain_side + 2 * grass_width)**2
    # Area of grass border
    grass_area = total_area - fountain_area
    # Amount of seed needed (35 g/sq.m)
    seed = grass_area * 35
    # Convert to kg
    seed = seed / 1000

    return seed
```

```
[49]: seed_calculator(12, 2)
```

```
[49]: 3.92
```

4. Make comparisons using operators

```
[50]: # > checks for greater than
print(10>1)
```

True

```
[51]: # == checks for equality
print("cat" == "dog")
```

False

```
[52]: # != checks for inequality
print(1 != 2)
```

True

```
[53]: # Some operators cannot be used between different data types.
print(1 < "1")
```

```

      □
↳-----

      TypeError                                Traceback (most recent call↳
↳last)

      <ipython-input-53-46cc2171c791> in <module>
          1 # Some operators cannot be used between different data types.
----> 2 print(1 < "1")
```

TypeError: '<' not supported between instances of 'int' and 'str'

```
[54]: # Letters that occur earlier in the alphabet evaluate to less than letters from↳
↳later in the alphabet.
# BOTH sides of an `and` statement must be true to return True.
print("Yellow" > "Cyan" and "Brown" > "Magenta")
```

False

```
[55]: # An `or` statement will return True if EITHER side evaluates to True.
print(25 > 50 or 1 != 2)
```

True

```
[56]: # `not` reverses Boolean evaluation of what follows it.
print(not 42 == "Answer")
```

True

5. Use if/elif/else statements to make decisions

```
[57]: # Define a function that checks validity of username based on length.
def hint_username(username):
    if len(username) < 8:
        print("Invalid username. Must be at least 8 characters long.")
    else:
        print("Valid username.")
```

```
[58]: # Define a function that uses modulo to check if a number is even.
def is_even(number):
    if number % 2 == 0:
        return True
    return False
```

```
[59]: is_even(19)
```

```
[59]: False
```

```
[60]: # Define a function that checks validity of username based on length.
def hint_username(username):
    if len(username) < 8:
        print("Invalid username. Must be at least 8 characters long.")
    elif len(username) > 15:
        print("Invalid username. Cannot exceed 15 characters.")
    else:
        print("Valid username.")
```

```
[61]: hint_username("ljñkljffñklasdjflkñadjglk{a}")
```

Invalid username. Cannot exceed 15 characters.

Congratulations! You've completed this lab. However, you may not notice a green check mark next to this item on Coursera's platform. Please continue your progress regardless of the check mark. Just click on the "save" icon at the top of this notebook to ensure your work has been logged.