Quiz - Python Training - Part 27

1. How is Logitech pronounced?
2. What happened in this video?
3. What acronym helps us to remember the order of operations?
4. What does it mean for subtraction to appear later on in the list of operations?
5. Can you name all the functions that appeared in this video?
6. How can we write a function that takes in more than one parameter? [03:25]
7. In the video, what did we find interesting "straightaway" about the example of calling a function? [04:06]
8. We were shown a function called `trip_welcome`. Can you remember what the two parameters were in this? [04:15]
9. Is the ordering of your parameters important? [05:04]
10. At one point in the video, I state "maybe that's the rule". Can you explain what I am talking about? [06:00]
11. In the example, we are shown some example output. It begins "Welcome to Tripcademy. Looks like…". What does it say? [06:55]
12. Where is Prospect Park?
13. Can you remember all the arguments that our `calculate_expenses` function took? [08:05]
14. We might suppose that in Python, there are three different types of arguments. What are they? [20:40]
15. Consider our attempt to build the function `trip_planner`. Why was the word `None` printed in the output? [33:00]

16. Does the order you specify parameters in the function declaration determine the order you can mention the parameters in the function body?

17. What are the advantages of using positional arguments?

18. What are the advantages of using keyword arguments?

19. If you list a number of parameters in a function declaration, do you have to specify all of them if you call the function?

20. What are the disadvantages of using keyword arguments?

ANSWERS

1. How is Logitech pronounced?

   According to Wikipedia, it is pronounced with a short "o" sound, similar to the sound in the word "body".

   The pronunciation is:

   /ˈlɒdʒɪtɛk/

   Logitech International S.A. is a Swiss multinational manufacturer of computer peripherals and software. Headquartered in Lausanne, Switzerland, the company has offices throughout Europe, Asia, Oceania, and the Americas.

   They are one of the world's leading manufacturers of input and interface devices for personal computers (PCs) and other digital products.

   It is a component of the flagship Swiss Market Index, and listed on the Nasdaq.

2. What happened in this video?

   We wrote a function called `calculate_expenses`. When writing this function, we had to think about the price for the plane ticket, hotel, and car. The function definition itself did not have any content. For example, there were no actual ticket prices (e.g. £280).

   When writing the function `calculate_expenses`, we had to think about the order of operations. We considered the acronym PEDMAS. We were misled by AI at one point in this video. If we have:

   A * B - C

   Without any further modifications, A will be multiplied by B before we subtract C. The fact that the multiplication occurs *first*, and the subtraction *later*, is implied by PEDMAS.

   At 11:31 I asked "do I need to add any parentheses here?" In reality, I did not need to add any parentheses. However, I ended up believing that I did need to add some.

   At [20:39] we started to reflect on the fact that in Python there are **three different types of arguments** we can give a function. There are positional argument, keyword arguments, and default argument.

   We looked at a function called `calculate_taxi_price` and this enabled us to explore positional arguments. The order in which you define parameters in the function definition important.

   At [25:38] we looked at an example of providing a default value to an argument. To demonstrate this, we gave the `discount` parameter a default value of 10. Instead of just writing `discount`, you write `discount = 10`.

In the final part of the video, we wrote a function called `trip_planner`. The function printed a string which was an itinerary. This gave us the opportunity to experiment with the order of arguments. We called the function and passed in France, Germany, and Denmark.

We read **Step 5** at [35:33]. Step 5 asked us to call the function `trip_planner` again, but this time using keyword arguments. We were to state `first_destination="Iceland", final_destination="Germany", second_destination="India"`. We made an interesting observation here. Using keywords to specify arguments, the order in which we specify arguments need not match the order of arguments in the function definition. Even though `final_destination` is defined last in the function definition, using keyword arguments, we do not have to specify the `final_destination` argument last. (You'll notice that we actually wrote `final_destination="Germany"` before we typed `second_destination="India"`.)

We made a **section observation**: the order in which we pass in arguments does not need to match the order in which the arguments are printed. At 37:02 I noted that "even though I specified India last, we don't get that printed in the sentence last".

3. What acronym helps us to remember the order of operations?

   The acronym **PEDMAS** helps us to remember the order of operations. We discussed this in the video from [13:28] onwards.

   P - for parentheses

   E - for exponents

   M - for multiplication

   D - for division

   A - for addition

   S - for subtraction

4. What does it mean for subtraction to appear later on in the list of operations?

   It means, firstly, that subtraction occurs **later** rather than **earlier**.

   Take this calculation as an example:

   ```
   hotel_total = hotel_rate * trip_time - 10
   ```

   The line contains a multiplication operation and a subtraction operation.

   If we consider PEMDAS to be telling us something about **ordering in time**, then M occurs before S. We perform the multiplication *before* the subtraction. In this case, we perform `hotel_rate * trip_time` and allow this operation to output a single value. Then, we subtract 10 from this value.

   However, there is something else to consider. **Ordering in time** is not the only thing to consider. Even if a person knows that multiplication occurs before subtraction, they might also suppose that the multiplication operation has wide scope. The things multiplied together are not simply

`hotel_rate` and `trip_time` but `hotel_rate` and `trip_time-10`. We might refer to this variable as the **scope** of the operation.

Well, PEDMAS has nothing to do with the scope of the operations. Or rather, it assumes that the scope of all the operations is the same. A multiplication operator only considers the operands immediately next to it. A subtraction operator only considers the operands immediately next to it.

5. Can you name all the functions that appeared in this video?

We considered lots of different functions in this video.

- At [04:15], we looked at an example function called `trip_welcome`.
- At [08:02] we began building `calculate_expenses`.
- At [21:16], we looked at `calculate_taxi_price`.
- At [28:00], we began building a function called `trip_planner`.

6. How can we write a function that takes in more than one parameter? [03:25]

"We can write a function that takes in more than one parameter by using **commas**."

7. In the video, what did we find interesting "straightaway" about the example of calling a function?

"Straightaway, we can notice that we haven't actually included `argument3`.

So maybe we don't have to include all of [the arguments]."

8. We were shown a function called `trip_welcome`. Can you remember what the two parameters were in this? [04:15]

`origin` and `destination`

9. Is the ordering of your parameters important? [05:04]

    Yes.

    "The ordering of your parameters is important, as their position will map to the position of the arguments, and will determine their assigned value in the function body."

10. At one point in the video, I state "maybe that's the rule". Can you explain what I am talking about? [06:00]

    Maybe the rule is:

    > you have to have some statement using `origin`, and only *then* can you have some statement using `destination`.

    Where `origin` and `destination` are the

    We need to think carefully about why, exactly, the order of parameters matters. We know that:

    > the ordering of your parameters is important as their position will map to the position of the arguments and will determine their assigned value in the function body

    However, this merely tells us that when *calling* a function, we need to pass in arguments in a certain order. It does not state that if we define parameters in a certain order, we are constrained to use them in a certain order in the body of the function definition.

11. In the example, we are shown some example output. It begins "Welcome to Tripcademy. Looks like…". What does it say? [06:55]

    Welcome to Tripcademy.

    Looks like you are traveling from Prospect Park

    And you are heading to Atlantic Terminal.

12. Where is Prospect Park?

    Prospect Park is an urban park in the NYC borough of Brooklyn.

    Prospect Park is also a public park in Reading, in the UK.

    The park in Brooklyn is 526 acres. The park in Reading is 120 acres.

13. Can you remember all the arguments that our `calculate_expenses` function took?

    `plane_ticket_price`

    `car_rental_price`

    `hotel_rate`

    `trip_time`

14. We might suppose that in Python, there are three different types of arguments. What are they? [20:40]

    Positional arguments - arguments that can be called by their position in the function definition.

    Keyword arguments - arguments that can be called by their name.

    Default arguments - arguments that can be given default values.

15. Consider our attempt to build the function `trip_planner`. Why was the word `None` printed in the output?

    If a function does not explicitly return a value, it implicitly returns the word `None`.