Congratulations! You passed!

Grade received 100%

Latest Submission Grade 100%

To pass 80% or higher

Go to next item

1. Which of the following describes one true quality about Python variables? **1 / 1 point**

- Python does not have a concept of a variable scope, which means that variables can be accessed from any part of the code.
- Variables in Python don't need to be declared before they are used.
- Variables in Python are not typed, which means that they can store any type of data.
- Variables in Python are not case-sensitive.

Correct
Correct! Variables in Python aren't typed, that is you aren't required to declare or know what the type is for the value you want to assign

2. What is a valid statement about Python lists? **1 / 1 point**

- Lists are indexed starting from 0
- Lists are inmutable, that is - they can't be changed after they are created
- Lists cannot contain any arbitrary object. Just integers and strings.

Correct
Correct! In Python the first item in a list has an index of 0.

3. What is a valid statement about Python dictionaries? **1 / 1 point**

- ○ A dictionary can only contain string values
- ○ A dictionary cannot contain nested dictionaries
- ◉ Python dictionaries cannot have a dictionary as a key

Correct
Correct! Keys have to be a type that is *hashable*, and Python dictionaries aren't *hashable* and will result in an error if you try to use them as a key.

4. What is a valid statement for Python tuples? **1 / 1 point**

- ○ Tuples are mutable. They can be modified once they are created.
- ○ Tuples are created by placing comma-separated values within curly brackets
- ◉ Tuples are immutable. They cannot be modified once they are created.

Correct
Correct! Tuples are like read-only lists. Once defined (created) you cannot modify them.

5. What are three valid methods for Python lists? **1 / 1 point**

- ○ add, index, sort
- ◉ pop, append, extend
- ○ clear, remove, describe

Correct
Correct! These are three valid methods you can use in Python lists

6. What are three valid methods for Python dictionaries? **1 / 1 point**

- ○ values, extend, get
- ○ copy, update, append

- (●) get, items, values

Correct
Correct! These are three valid methods you can use with Python dictionaries.

7. What would you use to retrieve a value in a Python dictionary from a key named `"item"` that might not exist, preventing an exception and using `True` as a fallback? **1 / 1 point**

- ( ) `obj.get(True, "item")`
- ( ) `obj.safe_get("item", True)`
- (●) `obj.get("item", True)`

Correct
Correct! That is how you would retrieve the value for `"item"` even if it doesn't exist and fallback to `True`

8. What is the right syntax to loop over keys and values from a dictionary named `obj`? **1 / 1 point**

- ( ) `for key, value in obj()`
- ( ) `for key, value in obj`
- (●) `for key, value in obj.items()`

Correct
Correct! You can use `.items()` to provide both the keys and values when looping over a dictionary.

9. Can you add two lists like `[1,2,3] + [4,5,6]` ? What would happen if you do? **1 / 1 point**

- (●) Yes. You would get `[1,2,3,4,5,6]`

- ○ No. You would get a **`TypeError`**
- ○ Yes. You would get **`[[1,2,3], [4,5,6]]`**

✓ **Correct**
Correct! You can add two lists together in Python.

10. What is one useful quality of Python sets? **1 / 1 point**

- ○ That sets can use indexes just like lists
- ○ That it is an immutable data structure
- ◉ That its contents are guaranteed to be unique.

✓ **Correct**
Correct! Python sets are a useful data structure that ensures that the items it has are unique.