Python Basics

Score: 10/12

1. What is the output of the following code? x = 10 y = 5 print(x//y)

- 2
- 2.0
- 3
- 3.0

Explanation

The '//' operator performs floor division, which returns the largest integer that is less than or equal to the result of the division. So, the output of the code will be 2.

2. Which of the following statements is true about mutable objects in Python?

- Lists and dictionaries are mutable
- Tuples are mutable

Sets are mutable

Strings are mutable

Explanation

Mutable objects in Python can be changed or modified after creation. This means that the value of the object can be altered.

3. What will be the value of x after the following code is executed? x = [1, 2, 3] y = x y.append(4)

[1, 2, 3]

[1, 2, 3, 4]

[1, 2, 4]

[1, 2, 3, 4, 4]

Explanation

In Python, when a list is assigned to a new variable, it points to the same list object in memory. So, any changes made to 'y' will also affect 'x'. Therefore, the value of 'x' will be [1, 2, 3, 4] after the code is executed.

4. In Python, what is the purpose of the 'and' operator?

- It is a bitwise operator
- It is used for string concatenation
- It is used for logical conjunction
- It is used to check object mutability

Explanation

The 'and' operator in Python is used for logical conjunction. It returns True if both the operands are True; otherwise, it returns False.

5. What will be the output of the following code? x = 5 y = 8 z = 10 print(x < y and y < z)

- True

False

Explanation

The 'and' operator checks the truth value of both operands. In this case, x is less than y (True) and y is less than z (True), so the result of the 'and' operation will be True.

6. Which of the following is a valid Python comment?

// This is a comment

This is a comment

/* This is a comment */

Explanation

In Python, comments are created using the '#' symbol, and anything following the '#' on that line is considered a comment and is ignored by the interpreter.

7. **What is the value of 'result' after the following code is executed? x = 10 result = 'Pass' if x > 10 else 'Fail'**

- Pass
- Fail
- Error
- None

Explanation

The code uses a conditional expression (ternary operator) to assign the value of 'result'. Since x is not greater than 10 (x = 10), the expression 'x > 10' evaluates to False, so the value of 'result' will be 'Fail'.

8. **What is the output of the following code? x = True y = False print(not x or y)**

- True
- False

Explanation

The 'not' operator negates the boolean value of x (True), resulting in False. The 'or' operator returns True if at least one of the operands is True. In this case, the result will be False.

9. Which of the following is NOT a valid comparison operator in Python?

- =
- ==
- !=
- >=

Explanation

In Python, '=' is an assignment operator, '==' is used for equality comparison, '!=' is used to check for inequality, '<' for less than, '>' for greater than, '<=' for less than or equal to, and '>=' for greater than or equal to. However, '=' alone is not a comparison operator.

10. What will be the output of the following code? x = 15 if x < 10: print('Less than 10') else: print('Greater than or equal to 10')

- Less than 10
- Greater than 10

Greater than or equal to 10

Error

Explanation

Since the value of x is 15, the condition 'x < 10' is False. Therefore, the 'else' block will be executed, and the output will be 'Greater than or equal to 10'.

11. What is the value of 'result' after the following code is executed? x = 7 result = 'Even' if x % 2 == 0 else 'Odd'

Even

Odd

Error

None

Explanation

The code uses a conditional expression (ternary operator) to assign the value of 'result' based on whether x is even or odd. Since 7 is not even (7 % 2 != 0), the value of 'result' will be 'Odd'.

12. In Python, which of the following is NOT a valid boolean value?

- True
- False
- 0
- 1

Explanation

In Python, the boolean values are 'True' and 'False'. Any other values, such as 0 and 1, are interpreted as False and True, respectively.