**Question 1: Boolean Values**

**What are the two Boolean values in Python?**

Hint: The Python Boolean type can represent one of two values.

**Question 2: Boolean Operations**

**Perform the AND operation on the Boolean values True and False. What is the result?**

Hint: Use the **and** keyword to perform the logical AND operation.

**Question 3: Boolean OR Operation**

**What is the result of the OR operation when both operands are False?**

Hint: Use the **or** keyword to perform the logical OR operation.

**Question 4: NOT Operation**

**What is the result of applying the NOT operation to True?**

Hint: Use the **not** keyword to invert a Boolean value.

**Question 5: Comparison Operators**

**Use a comparison operator to determine if 10 is greater than 2. What is the Boolean result?**

Hint: Use the **>** operator to compare the two numbers.

**Question 6: Equality and Identity**

**Explain the difference between == and is when used in a condition. Provide examples where the results differ.**

Hint: **==** checks for equality of value, whereas **is** checks for identity (i.e., if two variables point to the same object).

**Question 7: Boolean Conversion**

**Convert the integer 0 to a Boolean. What is the result?**

Hint: Use the **bool()** function to convert numeric values to Boolean values, noting how Python treats 0.

**Question 8: Using Booleans in Conditional Statements**

**Write a simple if statement that prints "Yes" if a = True, and "No" otherwise.**

Hint: An **if** statement can directly evaluate a Boolean expression.

**Question 9: Truthiness of Non-Boolean Values**

**List some examples of non-Boolean values that are treated as True in Python when evaluated in a Boolean context.**

Hint: Most values are treated as **True** except for a few specific cases, such as 0, **None**, and empty collections.

**Question 10: Short-circuiting in Logical Operations**

**Explain the concept of short-circuiting in the context of the AND and OR Boolean operations. Provide examples.**

Hint: Short-circuiting occurs when Python stops evaluating a logical expression as soon as the overall truth value is determined.