Python Assignment - 2

**1.What are the two values of the Boolean data type? How do you write them?**

=> The Boolean data type typically represents two values: “True” and “False”.

Is\_odd = True

Is\_even = False

**2. What are the three different types of Boolean operators?**

=> The three different types of Boolean operators are.

1. AND operator(||)
2. OR operator (|)
3. NOT operator (!)

**3. Make a list of each Boolean operator's truth tables (i.e. every possible combination of Boolean values for the operator and what it evaluate ).**

=> 1. “AND” operator Truth table.

|  |  |  |
| --- | --- | --- |
| **Operand 1** | **Operand 2** | **Result** |
| True | True | True |
| True | False | False |
| False | True | False |
| False | False | False |

2.”OR” operator Truth table.

|  |  |  |
| --- | --- | --- |
| **Operand 1** | **Operand 2** | **Result** |
| True | True | True |
| True | False | True |
| False | True | True |
| False | False | False |

3. “NOT” operator Truth table.

|  |  |
| --- | --- |
| **Operand** | **Result** |
| True | False |
| False | True |

**4. What are the values of the following expressions?**

(5 > 4) and (3 == 5) => False

not (5 > 4) => False

(5 > 4) or (3 == 5) => True

not ((5 > 4) or (3 == 5)) => False

(True and True) and (True == False) => False

(not False) or (not True) => True

**5. What are the six comparison operators?**

=>

1.Equal to (==): Compares if two operands are equal and returns True if they are, and False

Otherwise.

Example: 2 == 2 returns True.

2.Not equal to (!=): Compares if two operands are not equal and returns True if they are not, and False otherwise.

Example: 3 != 4 returns True.

3.Greater than (>): Compares if the left operand is greater than the right operand and returns True if it is, and False otherwise.

Example: 5 > 3 returns True.

4.Less than (<): Compares if the left operand is less than the right operand and returns True if it is, and False otherwise.

Example: 2 < 7 returns True.

5.Greater than or equal to (>=): Compares if the left operand is greater than or equal to the right operand and returns True if it is, and False otherwise.

Example: 5 >= 5 returns True.

6.Less than or equal to (<=): Compares if the left operand is less than or equal to the right operand and returns True if it is, and False otherwise.

Example: 3 <= 4 returns True.

**6. How do you tell the difference between the equal to and assignment operators?Describe a condition and when you would use one.**

=> In Python, the equal to operator (==) and the assignment operator (=) have different purposes and behaviours. Here's how you can tell the difference between them.

1.Equal to Operator (==): The equal to operator is used to compare whether two values are equal or not. It returns a Boolean value of True if the operands are equal, and False otherwise. It is used for comparison within conditional statements or to check for equality in expressions.

Example:

x = 5

y = 7

if x == y:

print("x and y are equal")

else:

print("x and y are not equal"

2.Assignment Operator (=): The assignment operator is used to assign a value to a variable. It assigns the value on the right side of the operator to the variable on the left side. It does not perform any comparison; it simply assigns the value.

Example:

X = 5

Y = X + 2

Print(Y)

In above Example value 5 will be assigned to X and summation of X with 2 will be assigned to Y.

**7. Identify the three blocks in this code:**

**spam = 0**

**if spam == 10:**

**print('eggs')**

**if spam > 5:**

**print('bacon')**

**else:**

**print('ham')**

**print('spam')**

**print('spam')**

=> Block 1:

spam = 0

Block 2:

if spam == 10:

print(‘eggs’)

Block 3:

if spam > 5:

print(‘bacon’)

else:

print(‘ham’)

**8. Write code that prints Hello if 1 is stored in spam, prints Howdy if 2 is stored in spam, and prints Greetings! if anything else is stored in spam.**

=>

Input1 = input(“Enter your choice between 1 and 2”)

spam = int(input1)

if spam == 1:

print('Hello')

elif spam == 2:

print('Howdy')

else:

print('Greetings!')

**9.If your programme is stuck in an endless loop, what keys you’ll press?**

=> If program stuck in an endless loop we press “ctrl+c” in keyboard.

**10. How can you tell the difference between break and continue?**

=>

* The ‘break’ statement is used to immediately terminate the execution of a loop. When encountered within a loop, the break statement exits the loop entirely, regardless of any remaining iterations.
* The ‘continue’ statement is used to skip the rest of the current iteration and move to the next iteration of the loop. When encountered within a loop, the continue statement skips any remaining statements within the loop's block and proceeds to the next iteration of the loop.

**11. In a for loop, what is the difference between range(10), range(0, 10), and range(0, 10, 1)?**

=> 1. Range(10):-

* It generates a sequence of numbers starting from 0 up to the specified end value.
* In these case it will generate numbers from 0 to 9.

2.Range(0,10):-

* It takes two arguments, the start and end values of the sequence.
* It generates numbers from starting specified value to the end specified value.

3.Range(0, 10, 1):-

* It also take two arguments start and end, additionally, it allows us to specify the step size.
* The sequence will start from the specified start value and increment by the specified step value until it reaches the specified end value.

**12. Write a short program that prints the numbers 1 to 10 using a for loop. Then write an equivalent program that prints the numbers 1 to 10 using a while loop.**

=> 1. Using for loop.

for num in range(1,11):

print(num)

2. Using while loop.

num = 1

while num <= 10:

print(num)

num += 1

**13. If you had a function named bacon() inside a module named spam, how would you call it after importing spam?**

=> import spam

spam.bacon ()