**Python Assignment - 2**

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

**Ans:** A boolean is a data type with two possible values: true (1) or false (0). The two values help represent truth conditions found in logic control structures.

A Boolean value is used to create conditions and control how a program behaves when certain things happen (e.g. if a condition is true, then do something). They can have only two possible values: either 0 or 1. You cannot add or subtract them. A Boolean *variable* is a special type of memory in a computer that can only store two values: true or false.

Generally, it is used to represent the truth values of the expressions. For example, 1==1 is True whereas 2<1 is False.

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

**Ans**: When using Booleans in a program, it’s important to understand the Boolean operators. These are used in conditions and conditional statements which control how the program will behave. Operator examples include AND (&&), OR (||) and NOT (~).

The following are examples of the Boolean value operators in programming:

* >=  True if a number is greater than or equal to another.
* <=  True if a number is less than or equal to another.
* ==  True if two values are equivalent.
* !=  True if two values are not equivalent.
* && True if both values are true.
* ||  True if either of the values are true.
* !  True if the value is false.
* ~  Reverses all of the bits in a variable (e.g.: 0000000000000000 becomes 1111111111111111). This can be useful when necessary because it allows you to change Booleans without affecting values or other types of data. It can also be used to revert changes made by a previous statement (e.g.: if (!k) then k = 1).

Boolean operators are used to make decisions in programs and indicate how the program should behave. For example, if p is true AND q is also true, then do something.

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 ).

**Ans:** List of each Boolean operator's truth tables

| **A** | **not A** |
| --- | --- |
| True | False |
| False | True |

:

| **A** | **not A** | | **Identity** | | **Yes** | **No** |
| --- | --- | --- | --- | --- | --- | --- |
| True | False | | True | | True | False |
| False | True | | False | | True | False |
| **A** | | **B** | | **A and B** | | |
| True | | True | | True | | |
| False | | True | | False | | |
| True | | False | | False | | |
| False | | False | | False | | |

| **A** | **B** | **A or B** |
| --- | --- | --- |
| True | True | True |
| False | True | True |
| True | False | True |
| False | False | False |

4. What are the values of the following expressions?

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

**Ans:** False

not (5 > 4)

**Ans:** False

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

**Ans:**True

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

**Ans:** False

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

**Ans:** False

(not False) or (not True)

**Ans:** True

5. What are the six comparison operators?

**Ans:** Comparison operators are used to compare two values:

|  |  |  |
| --- | --- | --- |
| Operator | Name | Example |
| == | Equal | x == y |
| != | Not equal | x != y |
| > | Greater than | x > y |
| < | Less than | x < y |
| >= | Greater than or equal to | x >= y |
| <= | Less than or equal to | x <= y |

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

**Ans:** The = is an assignment operator, while == and === are called equality operators.

| = | == |
| --- | --- |
| It is an[assignment operator.](https://www.geeksforgeeks.org/assignment-operators-in-c-c/) | It is a relational or[comparison operator.](https://www.geeksforgeeks.org/cpp-comparison-operators/#:~:text=Comparison%20operators%20are%20operators%20used,operand1%20is%20greater%20than%20operand2.) |
| It is used for assigning the value to a variable. | It is used for comparing two values. It returns 1 if both the values are equal otherwise returns 0. |
| Constant term cannot be placed on left hand side. Example: 1=x; is invalid. | Constant term can be placed in the left hand side. Example: 1==1 is valid and returns 1. |

Describe a condition and when you would use one.

#### Assignment Operator (=)

x=(a+b);

y=x;

Here, When first expression evaluates value of (a+b) will be assigned into x and in second expression y=x; value of variable x will be assigned into y.

#### Equal To Operator (==)

int x,y;

x=10;

y=10;

if(x==y)

printf("True");

else

printf("False");

When expression x==y evaluates, it will return 1 (it means condition is TRUE) and "TRUE" will print.

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')

**Ans:** ham

spam

spam

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.

**Ans:**

spam **=** int(input("Input a no."))

**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?

**Ans:** An infinite loop is a loop that runs indefinitely and it only stops with external intervention or when a break statement is found. You can stop an infinite loop with CTRL + C

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

**Ans:** Break:A break statement in Python alters the flow of a loop by terminating it once a specified condition is met. Continue: The continue statement in Python is used to skip the remaining code inside a loop for the current iteration only.

| **Basis for comparison** | **Break** | **Continue** |
| --- | --- | --- |
| Use | It is used for the termination of all the remaining iterations of the loop. | It is used for the termination of the only current iteration of the loop. |
| Control after using break/continue statement | The line which is just after the loop will gain control of the program. | The control will pass to the next iteration of that current loop by skipping the current iteration. |
| Causes | It performs the termination of the loop. | It performs early execution of the next loop by skipping the current one. |
| Continuation | It stops the continuation of the loop. | It stops the execution of the current iteration. |
| Other | It can be used with labels and switches. | It can't be used with labels and switches. |

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

**Ans:**

range(10) : It takes a single argument and starts at 0, increments by 1 and stops at one less than the value of “stop” argument given.

0 1 2 3 4 5 6 7 8 9

range(0, 10) : It takes two arguments: start and stop. The sequence will start from the value given at start, increment by 1 and stop at one less than the value given.

0 1 2 3 4 5 6 7 8 9

range(0, 10, 1) : It takes three arguments and lets you decide the start and stop values and the difference between each integer in that sequence. The default value of the step is 1.

0 2 4 6 8

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.

**Ans:** programme of prints the numbers 1 to 10 using a for loop

print("For Loop")

for i in range(1, 11):

print(i)

Output

For Loop

1

2

3

4

5

6

7

8

9

10

Programme of prints the numbers 1 to 10 using a for While loop

print("While Loop")

i = 1

while i <= 10:

print(i)

i = i+1

output

While Loop

1

2

3

4

5

6

7

8

9

10

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

**Ans:** This function can be called with spam.bacon().