# Assignment 2 – Python Basic Assignment

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

**True** and **False**

We have to write them with only first letter capital.

Example for Boolean: Check the number ‘A’ is greater than ‘B’ or not.

print(10>9) 🡪 Result 🡪 **True**

print(8>10) 🡪 Result 🡪 **False**

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

Boolean Operators: **“Or,” “And,” and “Not”**

1. 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).

T -> True

F -> False

|  |  |  |
| --- | --- | --- |
| **AND** | | |
| **p** | **q** | **p and q** |
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | F |

|  |  |  |
| --- | --- | --- |
| **OR** | | |
| **p** | **q** | **p or q** |
| T | T | T |
| T | F | T |
| F | T | T |
| F | F | F |

|  |  |
| --- | --- |
| **NOT** | |
| **p** | **NOT p** |
| T | F |
| F | T |

1. What are the values of the following expressions?

(5 &gt; 4) and (3 == 5) 🡪 **False**

not (5 &gt; 4) 🡪 **False**

(5 &gt; 4) or (3 == 5) 🡪 **True**

not ((5 &gt; 4) or (3 == 5)) 🡪 **False**

(True and True) and (True == False) 🡪 **False**

(not False) or (not True) 🡪 **True**

1. What are the six comparison operators?
2. **equal to**
3. **not equal to**
4. **greater than**
5. **greater than or equal to**
6. **less than**
7. **less than or equal to**.
8. How do you tell the difference between the equal to and assignment operators? Describe a condition and when you would use one.

**== (Equal to Operator)** 🡪 The ‘==’ operator checks whether the two given operands are equal or not. If so, it returns true. Otherwise it returns false.

**= (Assignment Operator)**🡪 The ‘=’ operator is used for assigning the value to a variable.

Assignment Operator – In simple words, it assigns value of right side of expression to left side operand.

Syntax: x **=** y + z

1. Identify the three blocks in this code:

spam = 0

if spam == 10:

print(‘eggs’) ------- 1st block (Indentation after if)

if spam &gt; 5:

print(‘bacon’) ------- 2nd block (Indentation after if)

else:

print(‘ham’) -------- 3rd block (Indentation after else)

print(‘spam’)

print(‘spam’)

After removing blocks; actual code could be as follows:

**spam = 0**

**if spam == 10:**

**print(‘eggs’)**

**if spam &gt; 5:**

**print(‘bacon’)**

**else:**

**print(‘ham’)**

**print(‘spam’)**

**print(‘spam’)**

1. 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.

Code:

**spam = input(“Enter a number”) #Note: If only integer input is allowed, then int(input())**

**if spam == 1:**

**print(“Hello”)**

**elif spam == 2:**

**print(“Howdy”)**

**else:**

**print(“Greetings!”)**

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

**To exit out of infinite loops on the command line, press CTRL + C**

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

**Break statement resumes the control of the program to the end of loop and made executional flow outside that loop.**

**Continue statement resumes the control of the program to the next iteration of that loop enclosing 'continue' and made executional flow inside the loop again.**

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

**range(stop) takes one argument.**

**range(start, stop) takes two arguments.**

**range(start, stop, step) takes three arguments.**

If start is not mentioned – then it defaults to 0.

If step is not mentioned, then it defaults to 1.

So, in the given example; all will results the same.

1. 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.

Using for loop:

**for i in range(1, 11):**

**print(i)**

Using while loop:

**i = 1**

**while(i <= 10):**

**print(i)**

**i += 1**

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

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