**INEURON ASSIGNMENTS**

**Python Basics : Assignment 2**

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

**Answer**:

* True, False

## **Question 2:** What are the three different types of Boolean operators

**Answer**:

* and, or, not

## **Question 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 ).

**Answer**:

1. AND Truth Table

|  |  |  |
| --- | --- | --- |
| a | b | a AND b |
| False | False | False |
| False | True | True |
| True | False | True |
| True | True | True |

2. OR Truth Table

|  |  |  |
| --- | --- | --- |
| a | b | a OR b |
| False | False | False |
| False | True | True |
| True | False | True |
| True | True | False |

3. NOT Truth Table

|  |  |  |
| --- | --- | --- |
| a | b | NOT (a AND b) |
| False | False | False |
| False | True | True |
| True | False | True |
| True | True | False |

|  |  |  |
| --- | --- | --- |
| a | b | NOT (a OR b) |
| False | False | True |
| False | True | False |
| True | False | False |
| True | True | False |

## **Question 4:** What are the values of the following expressions?

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

not (5 > 4)

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

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

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

(not False) or (not True)

**Answer**:

|  |  |
| --- | --- |
| Expressions | Results |
| (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 |

## **Question 5:** What are the six comparison operators

**Answer**:

* >, <, >=, <=, ==, !=

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

**Answer**:

* Equal to (=) is a simple assignment operator which used to store a value in a variable.
* Whereas, other assignment operators like +=, -=, \*=, /=, %=, //= are used to do arithmetic operation and assignment in same line.
* Eg., a += 10 => a = a + 10

## **Question 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')

**Answer**:

spam = 0

if spam == 10:

print('eggs') ==> block 1

if spam > 5:

print('bacon') ==> block 2

else:

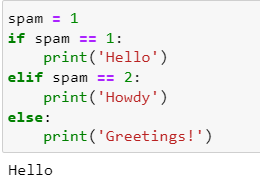
print('ham') ==> block 3

print('spam')

print('spam')

## **Question 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?

**Answer**:



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

**Answer**:

* CTRL + C

**Question 10:** How can you tell the difference between break and continue

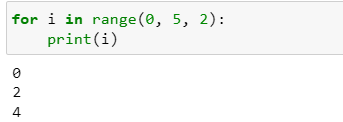
**Answer**:

* Break: It will terminate the loop and proceed with next code after loop
* Continue: It will skip the next lines in loop block and moves control back to top of the loop

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

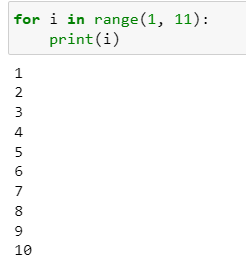
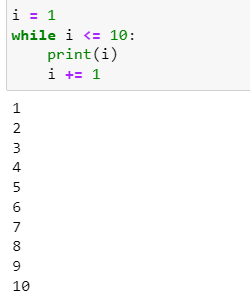
**Answer**:

* range(10) : Program fix the range from 0th to 9th with count of 10. Starting index is counted as 0 in python.
* Range(0 , 10) : It is same as above. 0th is the index number and 10 is the count of positions from index 0
* Range(0, 10, 1) : It also same as above, starts with index 0th and finish at 9th index (total of 10 counts) and 1 indicates the steps to choose the index.
  + Eg.,
    - range (0, 5, 1) -> 0, 1, 2, 3, 4
    - range (0, 5, 2) -> 0, 2, 4



**Question 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?

**Answer**:

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

**Answer**:

spam.bacon()