**FULL STACK ASSIGNMENT – 2**

1. A variable of the primitive data type boolean can have two values: **true and false** (Boolean literals). or off. Boolean expressions use relational and logical operators. The result of a Boolean expression is either true or false.
2. **There are three basic Boolean search commands: AND, OR and NOT.**

AND searches find all of the search terms. For example, searching on dengue AND malaria AND zika returns only results that contain all three search terms. ...OR searches find one term or the other. ...NOT eliminates items that contain the specified term.

**3.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **condition 1 (e.g., X)** | **condition 2 (e.g., Y)** | **NOT X ( ~ X )** | **X AND Y ( X && Y )** | **X OR Y ( X || Y )** |
| false | false | true | false | false |
| false | true | true | false | true |
| true | false | false | false | true |
| true | true | false | true | true |
|  |  |  |  |  |

**4.**

* False
* False
* True
* False
* False
* True

**5. Python has six comparison operators, which are as follows:**

* Less than ( < )
* Less than or equal to ( <= )
* Greater than ( > )
* Greater than or equal to ( >= )
* Equal to ( == )
* Not equal to ( != )

**6.** The “=” is an assignment operator is used to assign the value on the right to the variable on the left. **The '==' operator checks whether the two given operands are equal or not**. If so, it returns true. Otherwise it returns false.

**7.** ham

spam

spam

**8.** spam=1

If spam ==0:

print(‘hello’)

if spam>0:

print(‘howdy’)

else:

print(‘greetings’)

**output:-**

hello

greetings

**9.** 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.** The main difference between both the statements is that when break keyword comes, it terminates the execution of the current loop and passes the control over the next loop or main body, whereas when continue keyword is encountered, it skips the current iteration and executes the very next iteration in the loop.

**11.** range(10) :- 0,1,2,3,4,5,6,7,8,9,

range(0,10):- 0,1,2,3,4,5,6,7,8,9,

range(0,10,1):-0,1,2,3,4,5,6,7,8,9,

**12. for loop program:-**

**i=0**

**for i in range(10):**

**print(i+1,end=',')**

**output:- 0,1,2,3,4,5,6,7,8,9,10,**

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