Assignment\_2

1. Two values of Boolean Data type are: True and False

2. AND, OR, NOT

3.

|  |  |  |
| --- | --- | --- |
| And | **True** | **False** |
| **True** | True | False |
| **False** | False | False |

|  |  |  |
| --- | --- | --- |
| Or | **True** | **False** |
| **True** | True | True |
| **False** | True | False |

|  |  |
| --- | --- |
| Not |  |
| **True** | False |
| **False** | True |

4.

|  |  |
| --- | --- |
| Expression | Output |
| (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. Six comparison operators : ==, !=, >, <, >=, <=

6. Equal to operator uses double equal sign (==) where assignment operator uses only one (=).

Example:

A = 5 # this is assignment

B = 6 # this is assignment

A == B # this is checking if A is equal to B. The output will be False.

7. The three blocks are shaded in different colors as below:

spam = 0

if spam == 10:

print(‘eggs’)

if spam > 5:

print(‘bacon’)

else:

print(‘ham’)

print(‘spam’)

print(‘spam’)

8.

spam = int(input("enter a number"))

if spam == 1:

print('Hello')

elif spam == 2:

print('Howdy')

else:

print('Greetings!')

9. I think it’s Ctrl+C

10. Break : once the loop reaches the break statement, it will stop doing any operation inside the loop and exit the loop.

Continue : once the loop reaches the continue statement, it will go back up to the top of the loop and begins the next iteration.

11.

range(10) 🡪 it will loop through 0 to 9

range(0,10) 🡪 the same as above (except that here, we explicitly mention the starting point 0)

range(0,10,1) 🡪 the same as the first expression. Here, we mention starting point (0), ending point (10) and step size (1).

So, all three expressions give the same functionality.

12.

#Using for loop

for i in range(1,11):

print(i)

#Using while loop

j = 0

while j < 10:

j += 1

print(j)

13.

We have a function bacon() inside a module named spam.

#Example code

Import spam as sp

sp.bacon()