

Python Assignment 2

June 19, 2023

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

[]: ANS: The two values of the boolean data `type` are `True` & `False`. Boolean `type` is `one of the built-in` data types provided by Python.It `is` used to represent the truth value of the `expression`. We write them by using capital T `and` F, `while` the rest `in` lower case.
For eg:
`1 == 1 is True while 2 < 1 is False.`

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

[]: ANS:The 3 different types of boolean operators are: `and`, `or` , `not`.

[]: 3. Make a `list` of each Boolean operators truth tables (i.e. every possible `combination of Boolean` values `for` the operator `and` what it evaluate).

[]: ANS:`True and True is True.`
`True and False is False.`
`False and True is False.`
`False and False is False.`
`True or True is True.`
`True or False is True.`
`False or True is True.`
`False or False is False.`
`not True is False.`
`not False is True.`

[]: ANS:

[3]: `(5 > 4) and (3 == 5)`

[3]: `False`

[4]: `not (5 > 4)`

[4]: False

```
[5]: (5 > 4) or (3 == 5)
```

[5]: True

```
[6]: not ((5 > 4) or (3 == 5))
```

[6]: False

```
[8]: (True and True) and (True == False)
```

[8]: False

```
[9]: (not False) or (not True)
```

[9]: True

```
[ ]: 5. What are the six comparison operators?
```

```
[ ]: ANS:Equal-to operator(==)  
      not equal-to operator(!=)  
      Greater than operator(>)  
      lesser than opeartor(<)  
      Greater than equal-to operator( > = )  
      lesser than equal-to operator( < = )
```

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

```
[ ]: ANS: '=' is the equal to operator that compares two values and evaluates to a  
      ↪ boolean, while '='  
      is the assignment opeartor that stores a value in a variable.  
      Assigning value 10 to a using '=' opeartor, a = 10  
      so the value of a is 10.  
      The '=' operator checks whether the two given operands are equal or not. If  
      ↪ so, it returns true.  
      Otherwise it returns false.  
      5==5  
      This will return true.  
      A condition is an expression used in a flow control statement that evaluates to  
      ↪ a boolean value.
```

```
[ ]: 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:The three blocks in this code are if statement and the line print('spam')
→ print('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.

```
[12]: def myfun(spam):  
        if spam == 1:  
            print("Hello")  
        elif spam == 2:  
            print("Howdy")  
        else:  
            print("Greetings!")  
  
myfun(1)  
myfun(2)  
myfun(3)
```

```
Hello  
Howdy  
Greetings!
```

[]: 9.If your programme is stuck in an endless loop, what keys you'll press?

[]: ANS:Press CTRL-C to stop a program stuck in an infinite loop.

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

[]: ANS:A)The break statement will move the execution outside and just after a loop.
→ The continue
statement will move the execution to the start of the loop.

B)whenever break statement executed then else block will not be executed
→ whereas In continue ,all
the time else block is executed.

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

[]: ANS:They all do the same thing. The range(10) call ranges from 0 up to (but not
→ including) 10,
range(0, 10) explicitly tells the loop to start at 0, and range(0, 10, 1)
→ explicitly tells the loop to
increase the variable by 1 on each iteration

[]: 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.

[13]: `for i in range(1, 11):`
 `print(i)`

1
2
3
4
5
6
7
8
9
10

[14]: `i = 1`
 `while i <= 10:`
 `print(i)`
 `i = i + 1`

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()`.