1. The Two values of Boolean data types are:
   * True
   * False

In means of maintaining the readability and understanding of the code; lower case (‘true’ and ‘false’) for Boolean values is the conventional method in programming languages.

1. Three different types of Boolean operators:
   * AND
   * OR
   * NOT
2. Truth Table for Boolean Operators

AND

|  |  |  |
| --- | --- | --- |
| Input A | Input B | Output C |
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 0 | 0 |
| 0 | 1 | 0 |

OR

|  |  |  |
| --- | --- | --- |
| Input A | Input B | Output C |
| 1 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 0 | 0 |
| 0 | 1 | 1 |

NOT

|  |  |
| --- | --- |
| Input A | Output C |
| 1 | 0 |
| 0 | 1 |

1. (5>4) and (3==5)

**True**

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**

1. Six comparison operators are:
   * Equal to: ==
   * Not equal to: !=
   * Greater than: >
   * Greater than or equal to: >=
   * Less than: <
   * Less than or equal to: <=
2. The equal ‘=’ sign can be used to assign the values to the specific variable and equality ‘==’ sign can be used to compare or check whether variable expressions on both side of the operators are equal or not.

Condition example:

a = 7

if a == 0:

print (“a is zero.!”)

else:

print (“a is not zero.!”)

Based on the code, the variable ‘a’ has an assigned value of 7. There is a condition to check whether the variable ‘a’ is equal (‘==’) to zero (‘0’) or not. Whether it is equality to zero, then it should print ‘a is zero.!’; else it would print ‘a is not zero.!’

1. spam = 0 # Block 1: Value Assignment

if spam == 10: # Block 2: If condition

print (‘eggs’)

if spam > 5: # Block 3: If condition

print (‘bacon’)

else:

print (‘ham’)

print (‘spam’)

print (‘spam’)

1. spam = int(input("Enter the value:"))

if spam == 1:

print("Hello")

elif spam == 2:

print("Howdy")

else:

print("Greetings!")

1. If the programme stuck in an endless loop, “Ctrl + C” keys have to press. It leads to raising the Keyboard Interrupt exception, which can handle and catch the loop accordingly.
2. Break statement in python refers to terminating the flow of the loop, once the specified condition is met.

Continue statement in python refers to skip the remaining code inside a loop for current iteration only.

1. Python range (10): The user will get series of numbers that start from 0 and ends before the provided number as a stop. [range (stop)]

Python range (0, 10): The user now decides not only where the series of numbers stops, but also where it starts. [range (start, stop)]

Python range (0, 10, 1): The user can choose not only where the series of numbers starts and stops, but also how big the difference between the one number and the next.

1. Short program by for loop:

for i in range(1,11):

print(i)

Short program by while loop:

i = 0

while i < 11:

print(i)

i += 1

1. If a function named bacon() inside a module named spam, after importing spam it may called as spam.bacon()