# Control Statements

# Week 3

What is the *data-type* of the result when evaluating comparison (relational) expressions such as < and >?

*Answer:*

Boolean.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

For each of the following expressions write the result of their evaluation.

100 < 101

*Answer:*

True

100 > 99

*Answer:*

True

100 >= 100

*Answer:*

True

100 != 100

*Answer:*

False

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

For each of the following expressions write the result of their evaluation.

"abc" < "xyz"

*Answer:*

True

"abc" < "XYZ"

*Answer:*

False

"100" == 100

*Answer:*

False

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

For each of the following expressions write the result of their evaluation.

10 > 20 and 10 >= 10

*Answer:*

False

10 > 30 > 20

*Answer:*

False

40 < 20 or 20 < 30

*Answer:*

True

not True

*Answer:*

False

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

What would be the output shown following the execution of the following Python statements?

colours = [ "Blue", "Black", "Orange" ]

print("The colour black is in the list : ", "Black" in colours)

*Answer:*

The colour black is in the list : True

print("The colour orange is in the list : ", "orange" in colours)

*Answer:*

The colour orange is in the list : False

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Which of the following concepts does the Python ‘if’ statement support?

**Sequence**, **Selection** or **Iteration**?

*Answer:*

Selection

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

What would be the output shown following the execution of the following Python statements?

num1 = 100

num2 = 10

if num1 % num2 == 0:

print("num1 is divisible by num2")

else:

print("num1 is not divisible by num2")

*Answer:*

num1 is divisible by num2

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

What would be the output shown following the execution of the following Python statements?

num1 = 99

num2 = 70

if num1 < num2:

print("num1 is less than num2")

elif num1 > num2:

print("num1 is greater than num2")

else:

print("num1 is equal to num2")

*Answer:*

num1 is greater than num2

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

What is the name given to the following type of Python operator shown below?

lowest = x if x < y else y

*Answer:*

‘Conditional’ or ‘Ternary’ operator.

And, what value would be assigned to the variable ‘lowest’ when ‘x’ was equal to 10 and ‘y’ was equal to 5?

*Answer:*

The lowest variable would be assigned the value 5 when, x was equal to 10 and y was equal to 5.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Within the answer box below write a small Python program, that asks the user to enter a value between 1 and 10.Once the value has been input display a message saying whether the value was in the requested range.

Remember: values returned from the **input()** function are *strings*, and need converting before being used within expressions, i.e. you will need code such as this -

num = input("please enter a number between 1 and 10 : ")

num = int(num)

*Answer:*

num = int(input(“Please enter a number between 1 and 10 : “))

if 1 <= num <= 10:

print("The value that you entered is in the requested range.")

else:

print("The value that you entered is not in the requested range.")

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Within the answer box below write a small Python program that asks the user to enter two values. Store these in variables called x and y respectively.

If the 'x' value is larger than 'y' then print

The value 'x' is larger than the value 'y'

otherwise print

The value 'y' is larger than the value 'x'

*Answer:*

x = int(input("Enter the first value: "))

y = int(input("Enter the second value: "))

if x > y:

print("The value 'x' is larger than the value 'y'")

else:

print("The value 'y' is larger than the value 'x'")

Examine the output generated by the above program. Is the displayed text entirely accurate in all cases? If not Why?

*Answer:*

The displayed text is not entirely accurate in all cases because the program considers only two possibilities either x is larger than y or y is larger than x.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Within the answer box below write a small Python program that asks the user to enter two values.

Store these values in two variables then output a message displaying the result of dividing the first value by the second value.

Include code that prevents a run-time error being reported when the user inputs a value of '0' for the second input. *Hint:* use an ‘if’ statement

If a '0' value is input, print a message saying "division by 0 is not possible".

*Answer:*

num1 = float(input("Enter the first number: "))

num2 = float(input("Enter the second number: "))

if num2 == 0:

print("Division by 0 is not possible")

else:

result = num1 / num2

print("The result of dividing the first number by the second number is:", result)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Which of the following concepts does the Python while statement support?

**Sequence**, **Selection** or **Iteration**?

*Answer:*

Iteration

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

What would be the output shown following the execution of the following Python statements?

num = 5

while num > 0:

print(num)

num -= 1

*Answer:*

5

4

3

2

1

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Write a small Python program that prints your name to the screen 100 times, then enter the program into the answer box below. Hint: use a ‘while’ loop.

*Answer:*

count = 0

while count < 100:

print("Spandan")

count += 1

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

What would be the output shown following the execution of the following Python statements?

vals = ["A", "B", "C", "D"]

for letter in vals:

print(letter)

*Answer:*

A

B

C

D

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

What would be the output shown following the execution of each of the following Python statements?

for num in range(5):

print(num)

*Answer:*

0

1

2

3

4

5

for num in range(10,16):

print(num)

*Answer:*

10

11

12

13

14

15

for num in range(0,10,-1):

print(num)

*Answer:*

9

8

7

6

5

4

3

2

1

0

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Enter and execute the python code shown below, then show the exact output into the answer box.

for x in range(1,10):

for y in range (1,x):

print("\*")

print()

*Answer:*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

What is the term used to refer to code blocks that appear inside other code blocks as in the above program?

*Answer:*

The term used to refer to code blocks that appear inside other code blocks as in the above program is called nested code blocks.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**