🡪

True+ False ### 1+0 =1

True + True ### 1+1=2

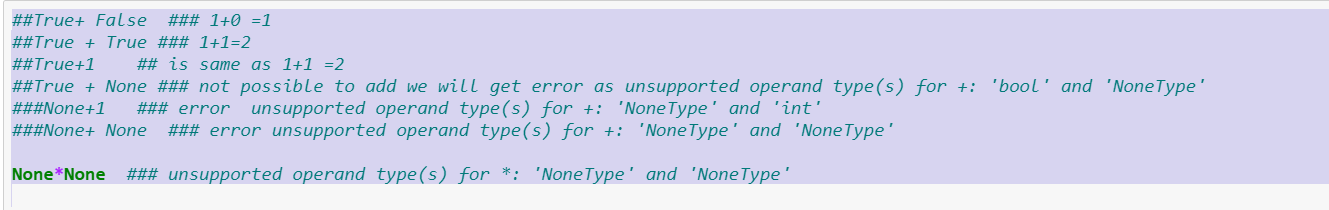
True+1 ## is same as 1+1 =2

True + None ### not possible to add we will get error as unsupported operand type(s) for +: 'bool' and 'NoneType'

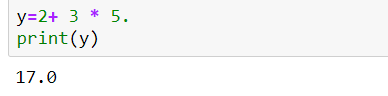
None+1 ### error unsupported operand type(s) for +: 'NoneType' and 'int'

None+ None ### error unsupported operand type(s) for +: 'NoneType' and 'NoneType'

None\*None ### unsupported operand type(s) for \*: 'NoneType' and 'NoneType'



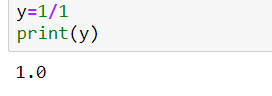
🡪



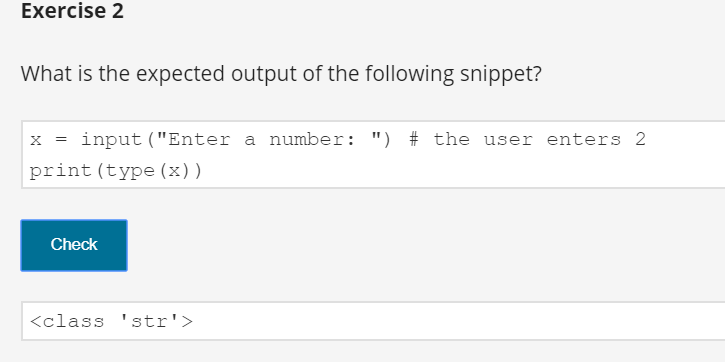
Python will consider 5. As 5.0

After 5 one dot is there but no fraction value defined still it is valid in python

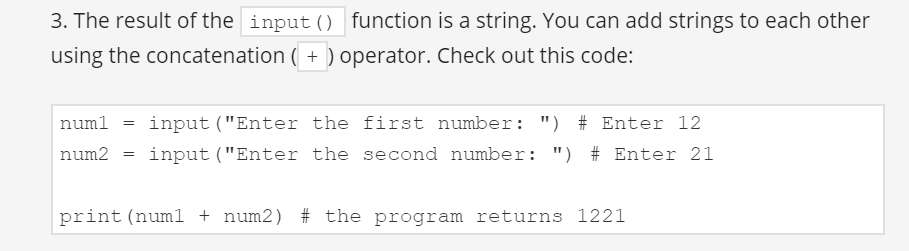
🡪



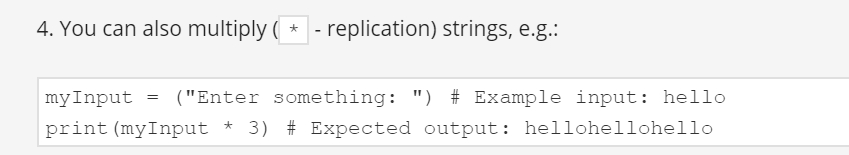
🡪



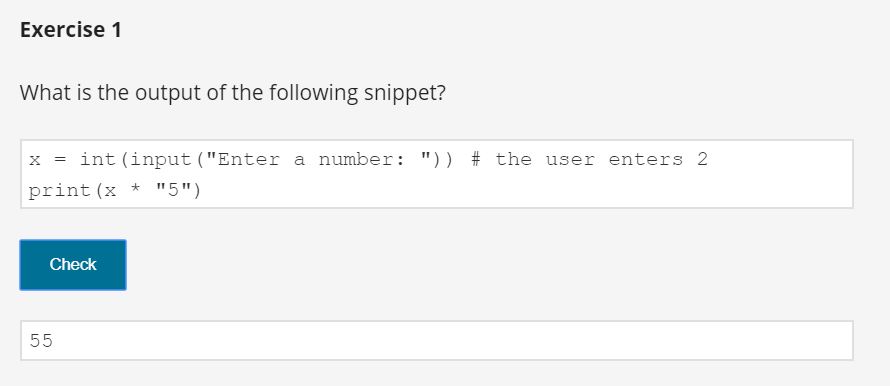
🡪



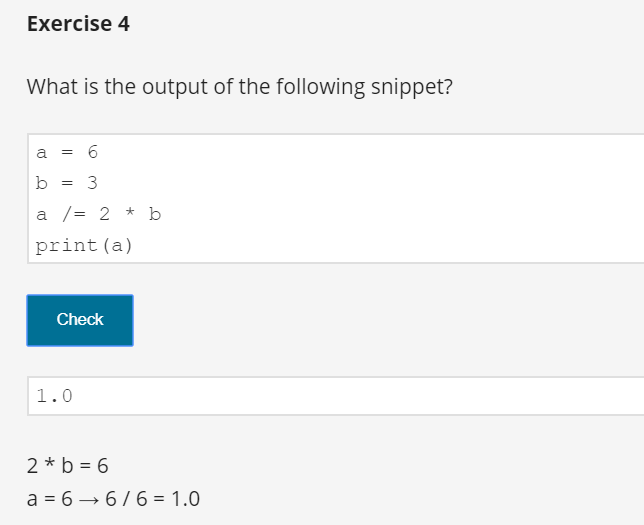
🡪



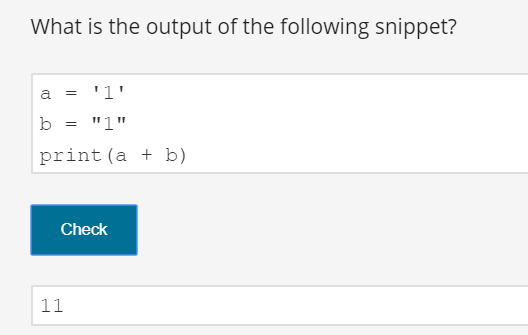
-->



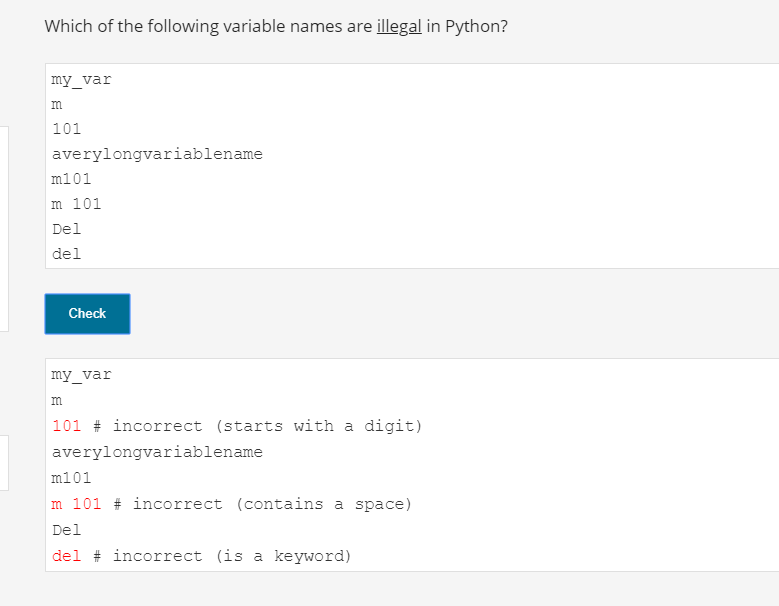
🡪



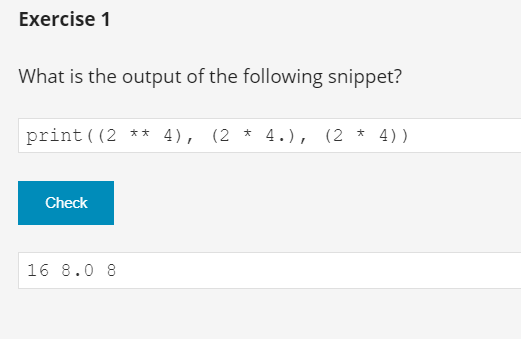
🡪



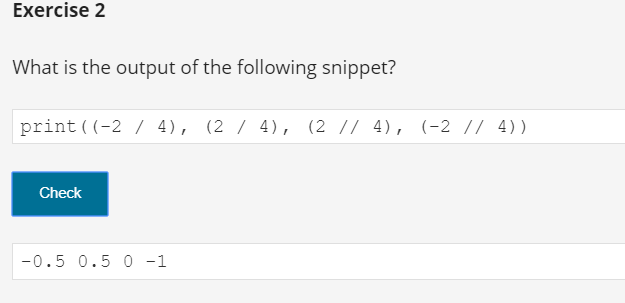
🡪



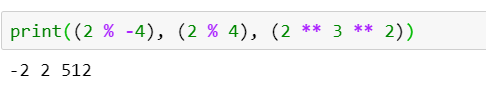
🡪



🡪



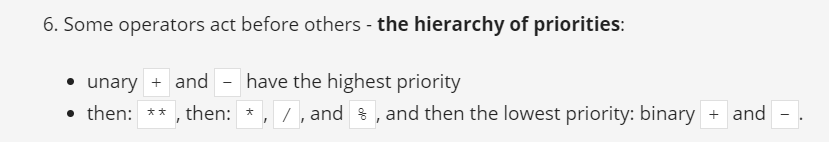
🡪



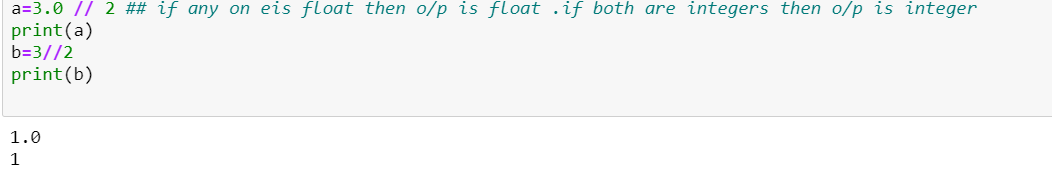
🡪



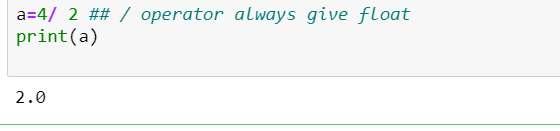
🡪



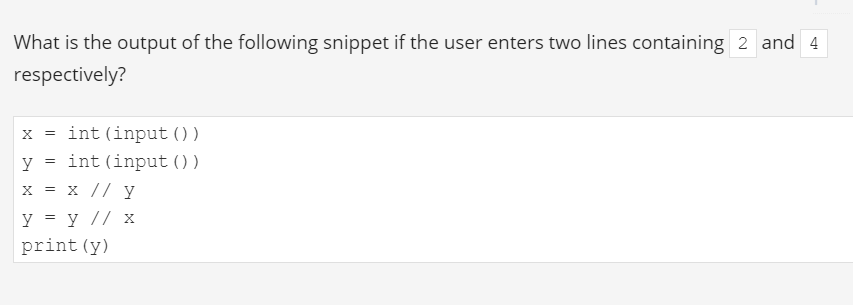
🡪 imp on // operator



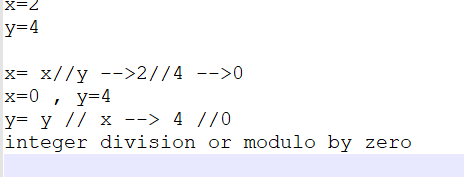
Imp on / operator



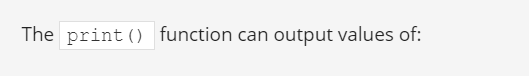
🡪

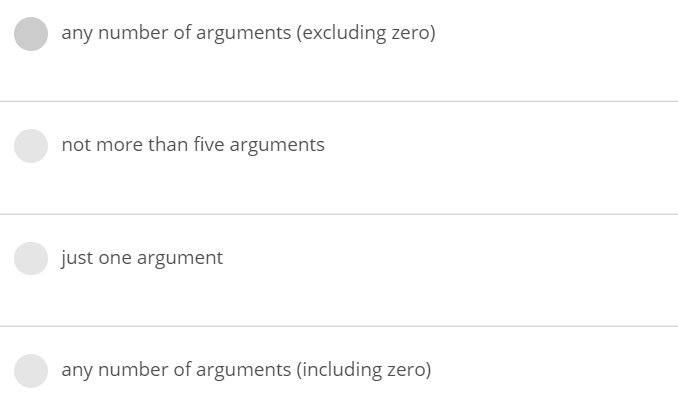


Ans: zero division error we will get



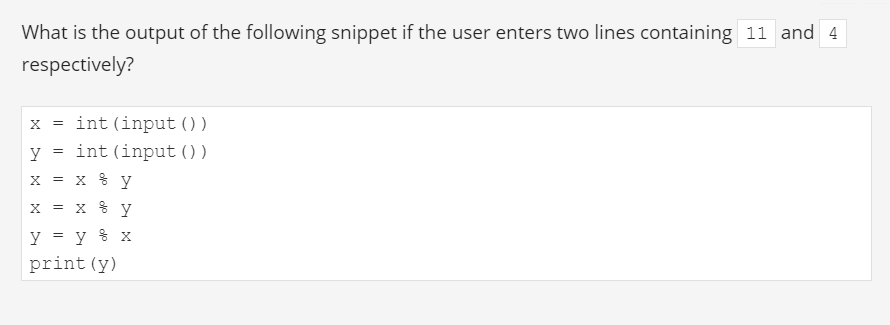
🡪

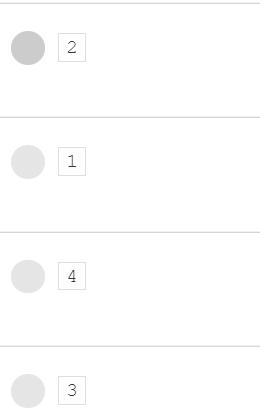




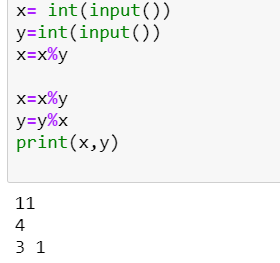
Ans: any number of arguments (including zero)

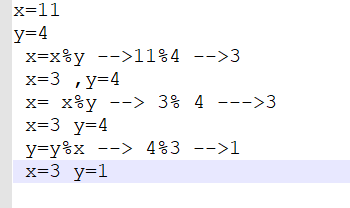
🡪



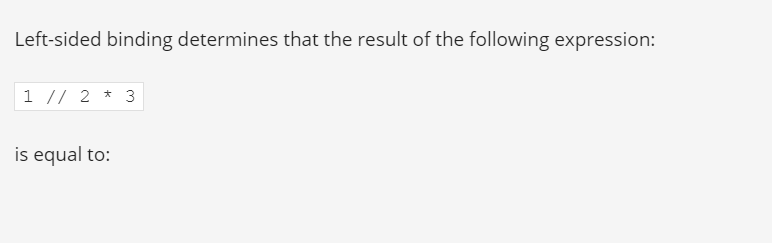


ans : 1





🡪



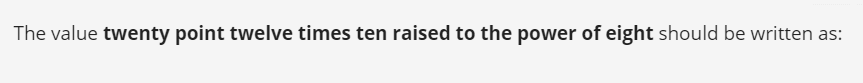


ANS:0

1//2 means 0 then

0\*3 is zero

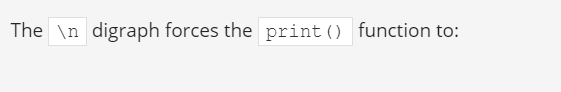
🡪

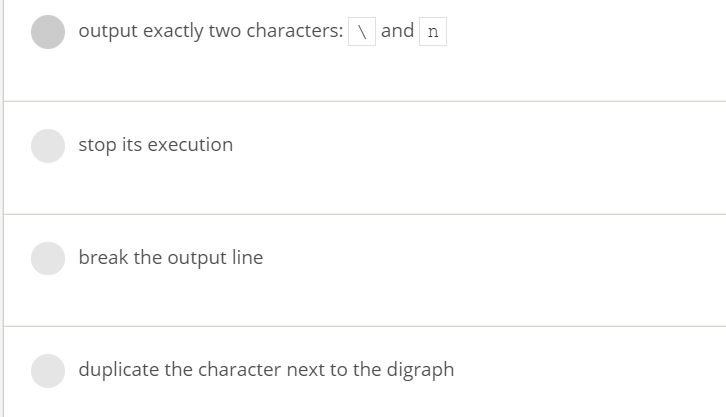




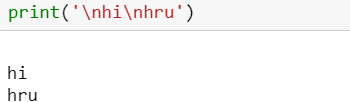
ANS : 20.12E8

🡪

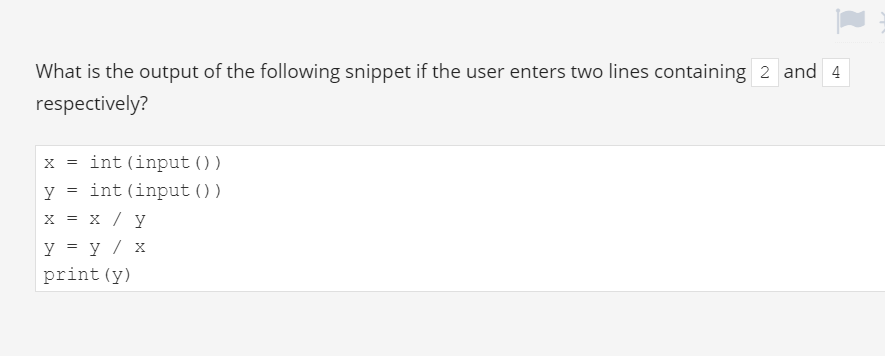




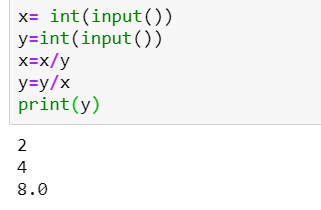
ans : break the output line

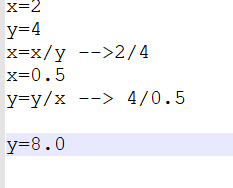


🡪

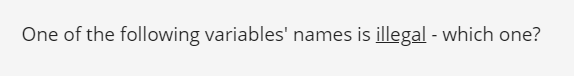


Ans:8.0





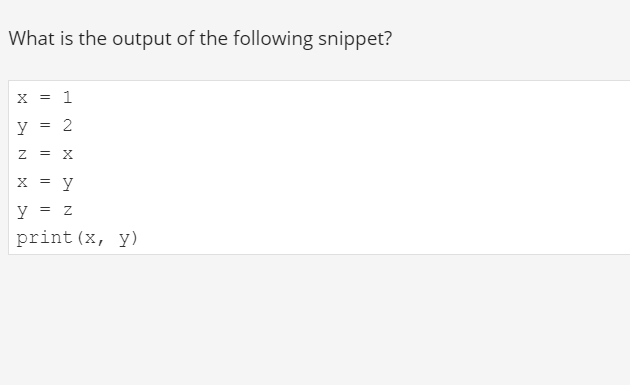
🡪





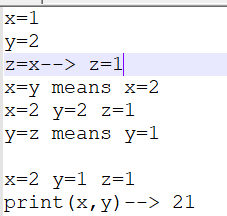
Ans: True

🡪

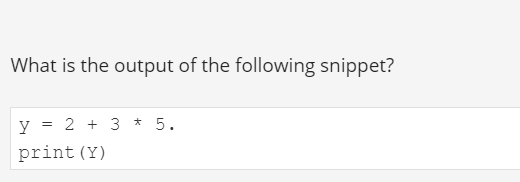


Ans: 2 1

Explanation:



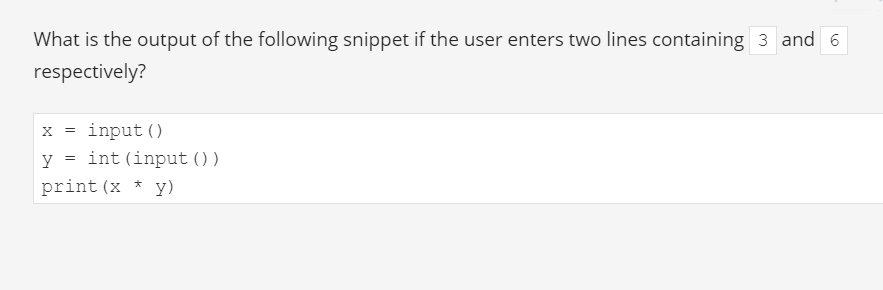
🡪



Ans: 17

Explanation: 2 + 3 \* 5 means 2+15= 17

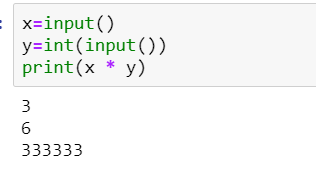
🡪



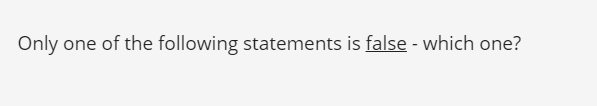
Ans: 333333

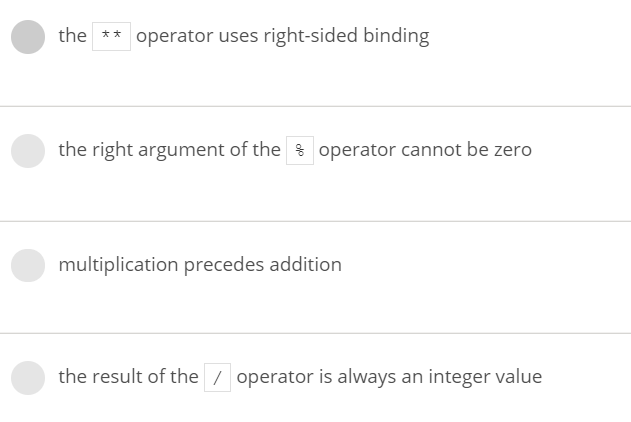
print(x\*y ) means same as print('3' \*6) so it will print character 3 as 6 times

Explanation:



🡪





Ans : the result of the / operation is always an integer values – it is false

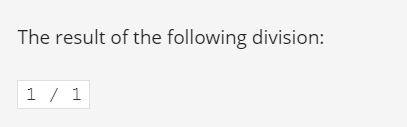
Note: (1)the operator \*\* uses right –sided binding means we have to evaluate from right to left is --True

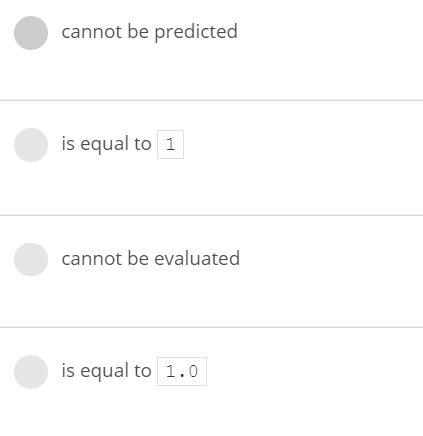
(2)the right argument of the % operator cannot be zero – True

(3)multiplication precedes addition ( means multiplication has high priority)—True

(4)the Result of the / operator is always an integer values -- is false because result / operator is always float

🡪

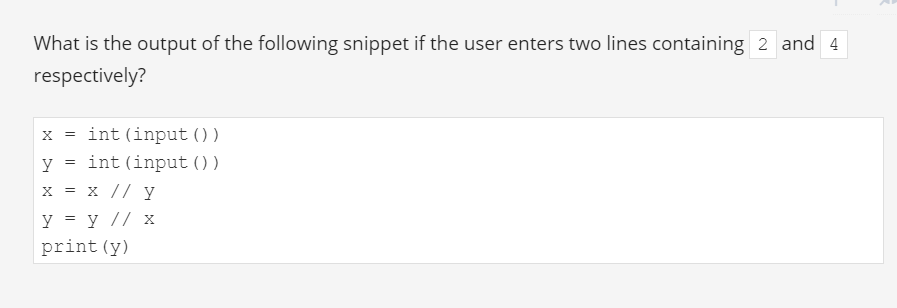




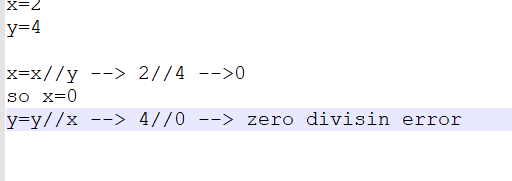
Ans : 1.0

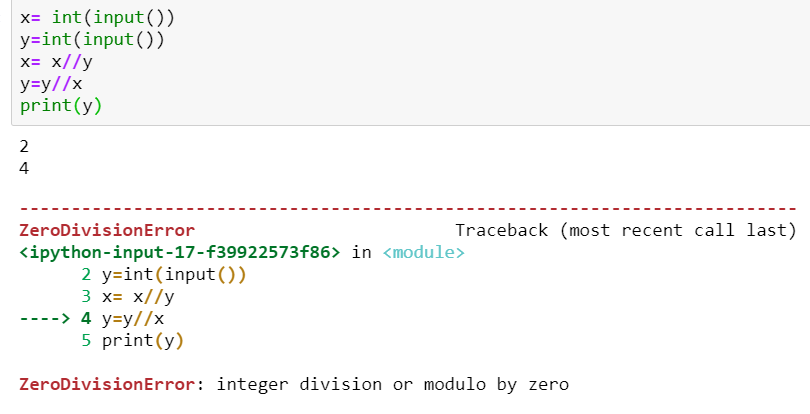
In python 1/1 gives 1.0

🡪

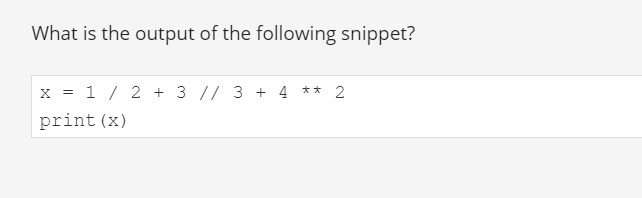


Ans: code will cause runtime error.





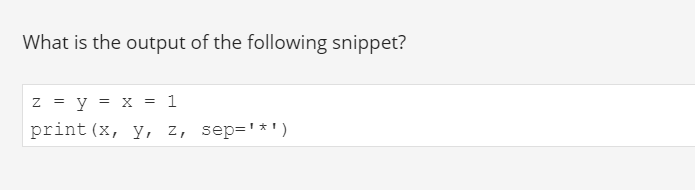
🡪

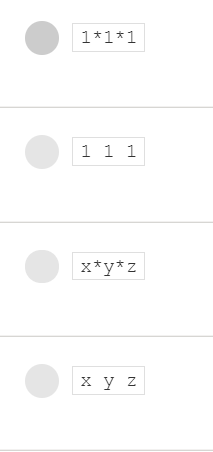


Ans:17.5

Explanation: 0.5+ 1+ 16 = 17.5

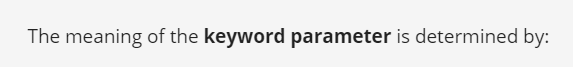
🡪

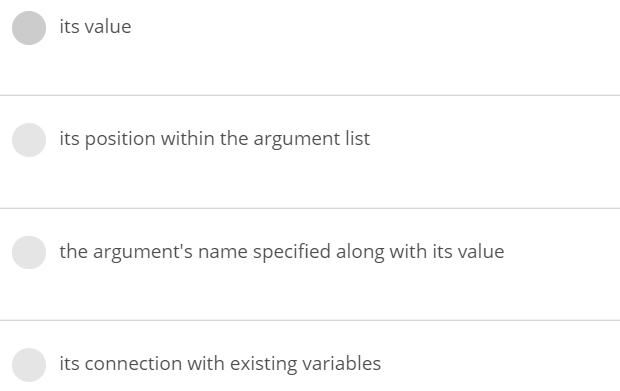




Ans:1\*1\*1

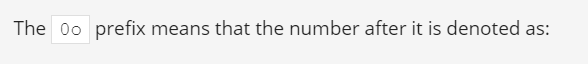
🡪

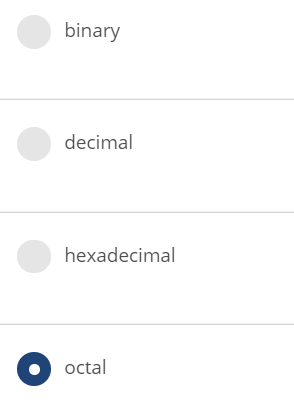




Ans: The argument’s name specified along with it’s value

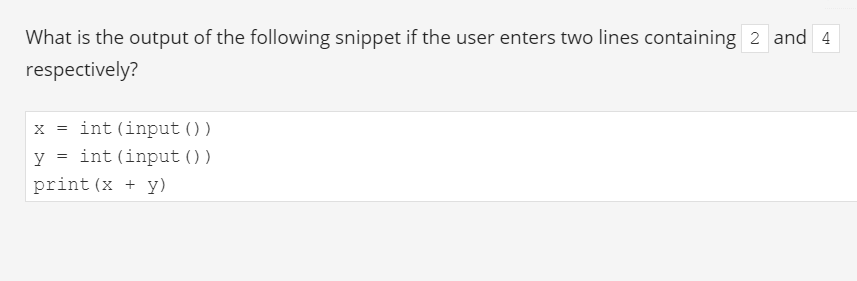
🡪





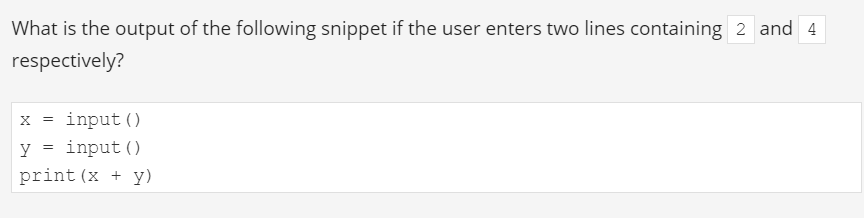
Ans: Octal

🡪



Ans: 6

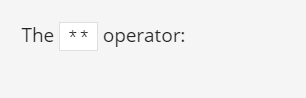
🡪

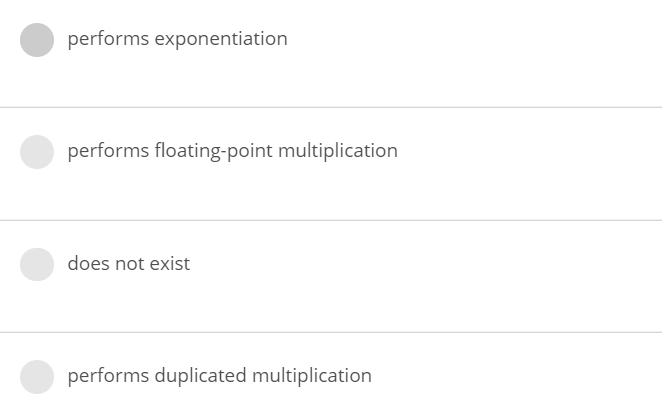




Ans: 24

🡪

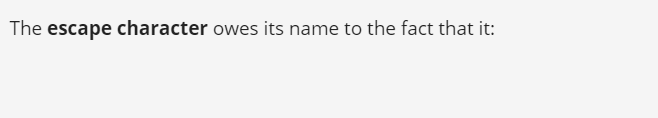




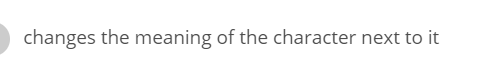
Ans:

performs exponentiation

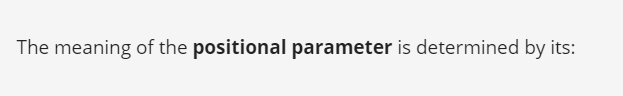
🡪

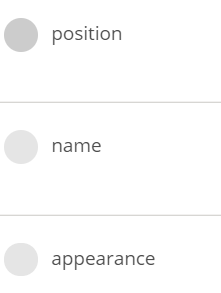


ANS:



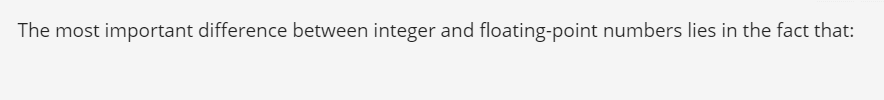
🡪

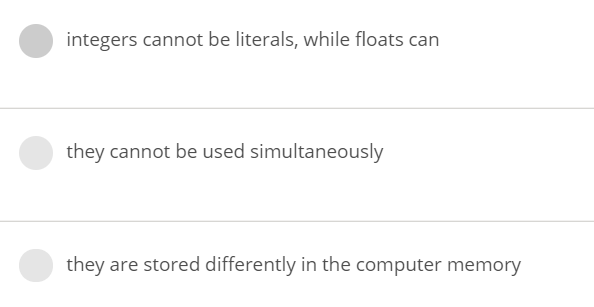




Ans: Position

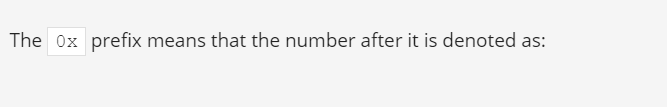
🡪

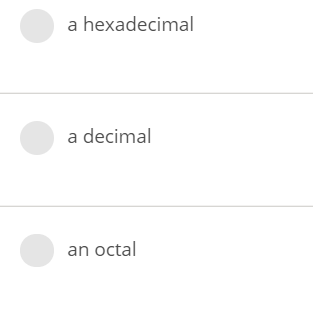




Ans: They are stored differently in the computer memory

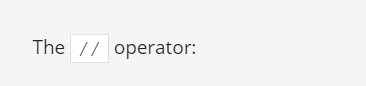
🡪

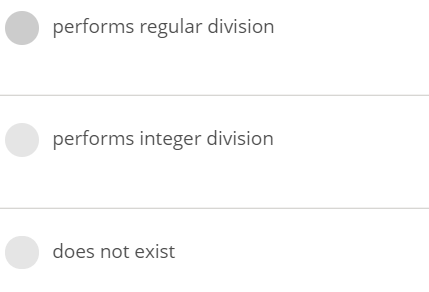




Ans: hexadecimal

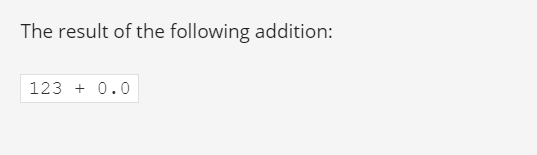
🡪

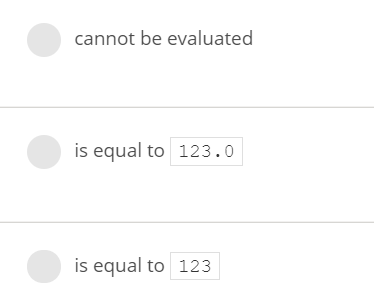




Ans: performs integer division

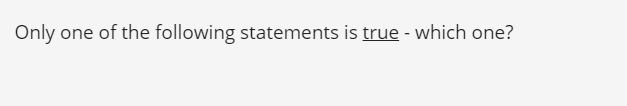
🡪

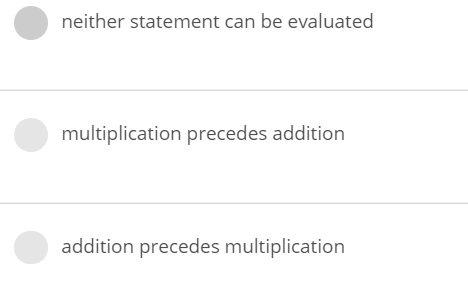




Ans: 123.0

🡪

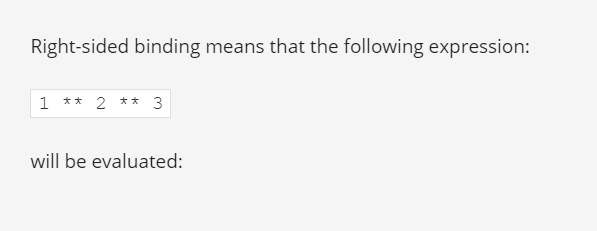


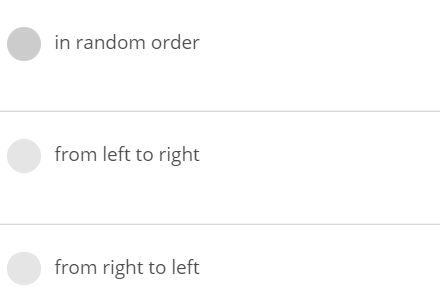


Ans:

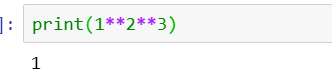
Multiplication precedes addition ( means multiplication comes before addition)

🡪



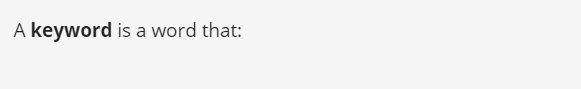


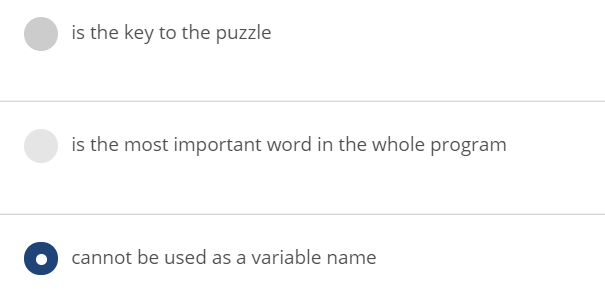
Ans: from right to left



Means 1\*\*8 is 1

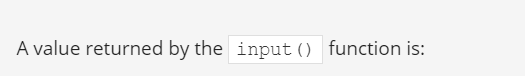
🡪

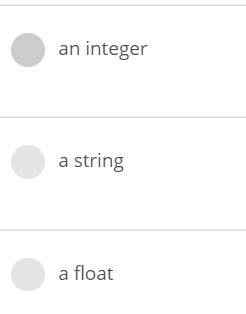




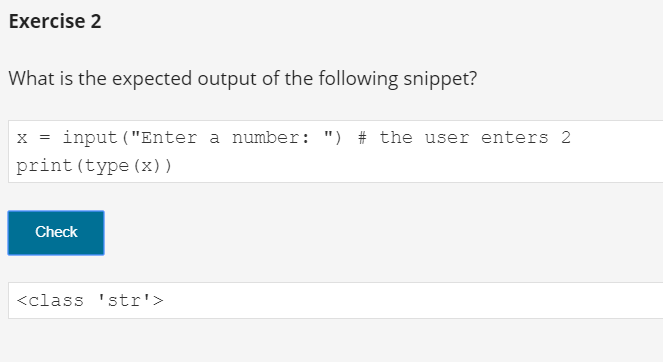
Ans: cannot be used as a variable name

🡪





Ans: a string



Important points

**Key takeaways**

1. The print() function **sends data to the console**, while the input() function **gets data from the console**.

2. The input() function comes with an optional parameter: **the prompt string**. It allows you to write a message before the user input, e.g.:

name = input("Enter your name: ")

print("Hello, " + name + ". Nice to meet you!")

3. When the input() function is called, the program's flow is stopped, the prompt symbol keeps blinking (it prompts the user to take action when the console is switched to input mode) until the user has entered an input and/or pressed the *Enter* key.

3. The result of the input() function is a string. You can add strings to each other using the concatenation (+) operator. Check out this code:

num1 = input("Enter the first number: ") # Enter 12

num2 = input("Enter the second number: ") # Enter 21

print(num1 + num2) # the program returns 1221

4. You can also multiply (\* - replication) strings, e.g.:

myInput = ("Enter something: ") # Example input: hello

print(myInput \* 3) # Expected output: hellohellohello

note:

print('hi' \*3) is same as print(3\*'hi')

example:

x = int(input("Enter a number: ")) # the user enters 2

print(x \* "5")

0/p is : 55 bcz 2\*”5” means it will print 5 2 times

