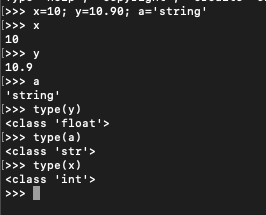
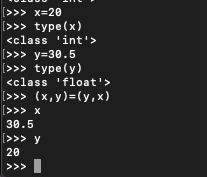
**TASK ONE: NUMBERS AND VARIABLES**

1. Create three variables in a single line and assign different values to them and make sure their data types are different. Like one is int, another one is float and the last one is a string.

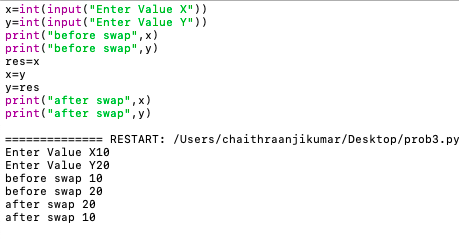


1. Create a variable of value type complex and swap it with another variable whose value is an integer.

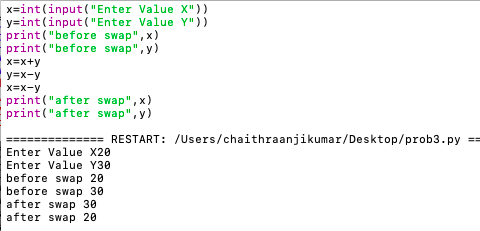


1. Swap two numbers using the third variable as the result name and do the same task without using any third variable.

With third variabel:



Without third Variable:



1. Write a program to print the value given by the user by using both Python 2.x and Python 3.x Version.
2. x=input(“Enter the value:”)

print(x)

y=raw\_input(‘Enter the value:”)

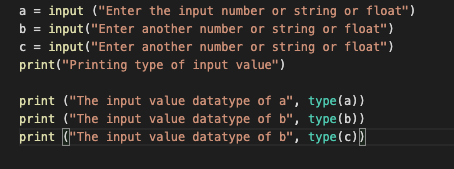
print(y)

5. Write a program to complete the task given below:

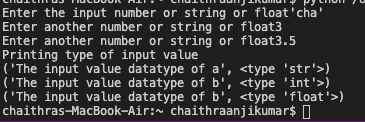
* Ask the user to enter any 2 numbers in between 1-10 and add both of them to another variable call z.
* Use z for adding 30 into it and print the final result by using variable results.



6.Write a program to check the data type of the entered values. HINT: Printed output should say - The input value data type is: int/float/string/etc



Output:



7.Create Variable using Upper CamelCase, Lower CamelCase, SnakeCase and UPPERCASE. (Refer: <https://capitalizemytitle.com/camel-case/>) - Variable Conventions to write:

|  |  |  |
| --- | --- | --- |
| lowerCamel | UpperCamel (Pascal Case) | snake\_case |
| numberOfStudents | NumberOfStudents | number\_of\_students |

8.If one data type value is assigned to ‘a’ variable and then a different data type value is assigned to ‘a’ again. Will it change the value. If Yes then Why?

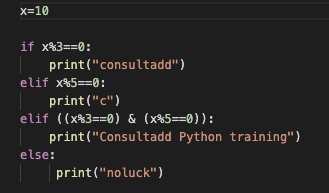
1. Yes, the variable ‘a’ changes the type depends on the last usage of value, I.e the initial value of ‘a’ is lost once the second assignment to the value of ‘a’ was evaluated, and remains same vale for the duration of session unless otherwise assigned a new value later.

**---Task 2 Continues---**

**TASK TWO: OPERATORS AND DECISION-MAKING STATEMENT**

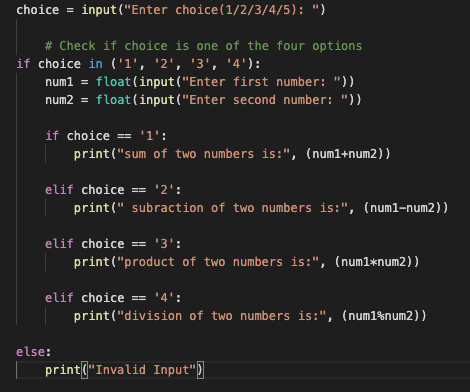
1. Write a program in Python to perform the following operation:

* If a number is divisible by 3 it should print “Consultadd” as a string
* If a number is divisible by 5 it should print “c” as a string
* If a number is divisible by both 3 and 5 its should print “Consultadd Python Training” as a string.



2. Write a program in Python to perform the following operator based task:

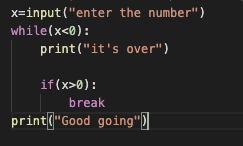
* Ask the user to choose the following option first:
  + If User Enter 1 - Addition
  + If User Enter 2 - Subtraction
  + If User Enter 3 - Division
  + If USer Enter 4 - Multiplication
  + If User Enter 5 - Average
* Ask the user to enter the 2 numbers in a variable for first and second for the first 4 options mentioned above.
* Ask the user to enter two more numbers as first1 and second2 for calculating the average as soon as the user chooses an option 5.
* In the end, if the answer of any operation is Negative print a statement saying “Zsa”
* NOTE: At a time users can perform one action at a time.



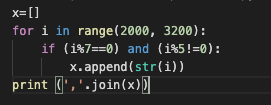
3. Write a program in Python to implement the given flowchart:

4. Write a program in Python to break and continue if the following cases occur:

* If a user enters a negative number just break the loop and print “It’s Over”
* If a user enters a positive number just continue in the loop and print “Good Going”



5. Write a program in Python which will find all such numbers which are divisible by 7 but are not a multiple of 5, between 2000 and 3200.



6. What is the output of the following code examples?

* x=123

for i in x:

print(i)

>>Error

* i = 0

while i < 5:

print(i)

i += 1

if i == 3:

break

else:

print(“error”)

>>SyntaxError

* count = 0

while True:

print(count)

count += 1

if count >= 5:

Break

>>

0

1

2

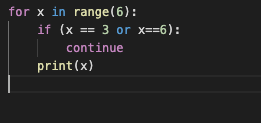
3

4

7. Write a program that prints all the numbers from 0 to 6 except 3 and 6.

Expected output: 0 1 2 4 5

Note: Use the ‘continue’ statement

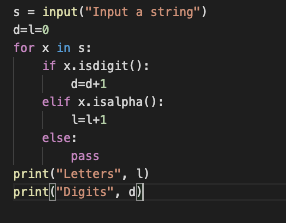


8. Write a program that accepts a string as an input from the user and calculates the number of digits and letters.

Expected output: consul72

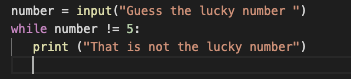
Letters 6

Digits 2

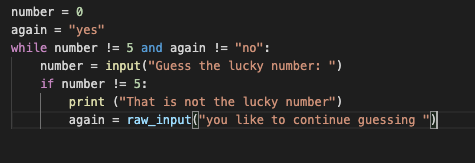


9. Read the two parts of the question below:

* Write a program such that it asks users to “guess the lucky number”. If the correct number is guessed the program stops, otherwise it continues forever.



* Modify the program so that it asks users whether they want to guess again each time. Use two variables, ‘number’ for the number and ‘answer’ for the answer to the question of whether they want to continue guessing. The program stops if the user guesses the correct number or answers “no”. ( The program continues as long as a user has not answered “no” and has not guessed the correct number)



10. Write a program that asks five times to guess the lucky number. Use a while loop and a counter, such as

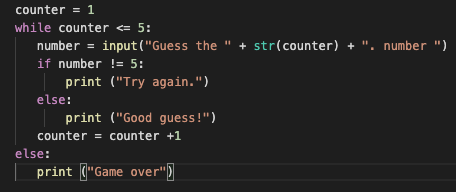
counter=1

While counter <= 5:

print(“Type in the”, counter, “number”

counter=counter+1

The program asks for five guesses (no matter whether the correct number was guessed or not). If the correct number is guessed, the program outputs “Good guess!”, otherwise it outputs “Try again!”. After the fifth guess, it stops and prints “Game over!”.



11. In the previous question, insert “break” after the “Good guess!” print statement. “break” will terminate the while loop so that users do not have to continue guessing after they found the number. If the user does not guess the number at all, print “Sorry but that was not very successful”.

