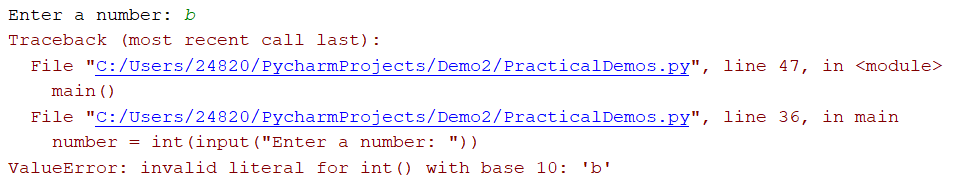
A picture containing graphical user interface

Description automatically generated

5.8: Validating Input Using Nested Control Structures

Run your program again (make sure that you uncomment the counter updating line!). However, this time, when it asks for a number, enter a letter. You should get the following error:



This error has occurred as we are trying to convert a letter to an integer, which is not possible. It would actually be better for us to validate our data before trying to convert it. This way, we can protect our users from errors. In this situation, we can use a while loop to keep iterating until we get the data type we want.

Add the following function to your file:

def get\_number\_input():

invalid\_entry = True

while(invalid\_entry):

input\_number = input("Please enter a number: ")

if(input\_number.isdigit()):

invalid\_entry = False

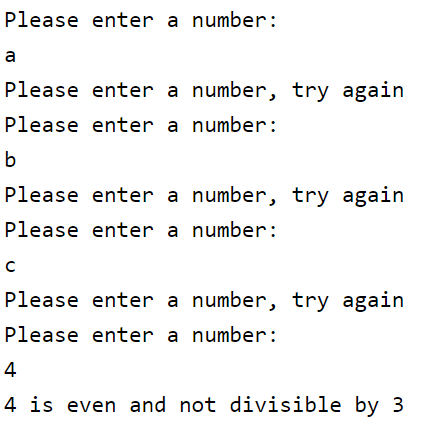
else:

print("Please enter an number, try again")

return int(input\_number)

In the code, you will notice that we have used a new function which is called isdigit(). This will test whether the string which has been entered is a number – returning True when the string is a number, and False when it is not.

Adapt your main() and run the function. You should be able to use letters indefinitely until you enter a number:



This demonstrates how useful the while loop would be – you wouldn’t be able to do a similar loop using the for structure. The for loop requires you to have a specified number of iterations, which is not possible when it comes to validating user input.