1. Is the Python Standard Library included with PyInputPlus?

ANS: PyInputPlus is not a part of the Python Standard Library, so you must install it separately using Pip

In [3]:

*#!pip install pyinputplus*

2. Why is PyInputPlus commonly imported with import pyinputplus as pypi?

ANS: pypi is alias of PyInputPlus.

The as pyip code in the import statement saves us from typing pyinputplus each time we want to call a PyInputPlus function. Instead we can use the shorter pyip name

3. How do you distinguish between inputInt() and inputFloat()?

ANS: inputInt() : Accepts an integer value, and returns int value

inputFloat() : Accepts integer/floating point value and returns float value

Both takes additional parameters ‘min’, ‘max’, ‘greaterThan’ and ‘lessThan’ for bounds

In [8]:

**import** pyinputplus **as** pyip

pyip**.**inputInt()

pyip**.**inputFloat()

3.5

'3.5' is not an integer.

2

3

Out[8]:

3.0

4. Using PyInputPlus, how do you ensure that the user enters a whole number between 0 and 99?

ANS: In the inputint function we can set the min = 0 and max =99 to ensure user enters number between 0 and 99

In [9]:

pyip**.**inputInt(min **=** 0, max **=**99)

180

Number must be at maximum 99.

3

Out[9]:

3

5. What is transferred to the keyword arguments allowRegexes and blockRegexes?

ANS: we can also use regular expressions to specify whether an input is allowed or not. The allowRegexes and blockRegexes

keyword arguments take a list of regular expression strings to determine what the PyInputPlus function will accept or

reject as valid input.

In [23]:

response **=** pyip**.**inputNum(allowRegexes**=**[r'(I|V|X|L|C|D|M)+', r'zero']) *# it allowes roman letters as numbers too*

V

In [25]:

response **=** pyip**.**inputNum(blockRegexes**=**[r'[02468]$'])*# blocks the even numbers*

44

This response is invalid.

46

This response is invalid.

68

This response is invalid.

45

6. If a blank input is entered three times, what does inputStr(limit=3) do?

ANS: it will throw RetryLimitException exception.

In [27]:

response **=** pyip**.**inputStr(limit**=**3)

Blank values are not allowed.

Blank values are not allowed.

Blank values are not allowed.

**---------------------------------------------------------------------------**

**ValidationException** Traceback (most recent call last)

**~\anaconda3\lib\site-packages\pyinputplus\\_\_init\_\_.py** in \_genericInput**(prompt, default, timeout, limit, applyFunc, validationFunc, postValidateApplyFunc, passwordMask)**

166 **try:**

**--> 167 possibleNewUserInput = validationFunc(**

168 userInput

**~\anaconda3\lib\site-packages\pyinputplus\\_\_init\_\_.py** in <lambda>**(value)**

242

**--> 243 validationFunc = lambda value: pysv.\_prevalidationCheck(**

244 value**,** blank**=**blank**,** strip**=**strip**,** allowRegexes**=**allowRegexes**,** blockRegexes**=**blockRegexes**,** excMsg**=None,**

**~\anaconda3\lib\site-packages\pysimplevalidate\\_\_init\_\_.py** in \_prevalidationCheck**(value, blank, strip, allowRegexes, blockRegexes, excMsg)**

249 **# value is blank but blanks aren't allowed.**

**--> 250** \_raiseValidationException**(**\_**("Blank values are not allowed."),** excMsg**)**

251 **elif** blank **and** value **==** **"":**

**~\anaconda3\lib\site-packages\pysimplevalidate\\_\_init\_\_.py** in \_raiseValidationException**(standardExcMsg, customExcMsg)**

221 **if** customExcMsg **is** **None:**

**--> 222 raise** ValidationException**(**str**(**standardExcMsg**))**

223 **else:**

**ValidationException**: Blank values are not allowed.

During handling of the above exception, another exception occurred:

**RetryLimitException** Traceback (most recent call last)

**<ipython-input-27-af00d69b8cd2>** in <module>

**----> 1** response **=** pyip**.**inputStr**(**limit**=3)**

**~\anaconda3\lib\site-packages\pyinputplus\\_\_init\_\_.py** in inputStr**(prompt, default, blank, timeout, limit, strip, allowRegexes, blockRegexes, applyFunc, postValidateApplyFunc)**

245 )[1]

246

**--> 247 return \_genericInput(**

248 prompt**=**prompt**,**

249 default**=**default**,**

**~\anaconda3\lib\site-packages\pyinputplus\\_\_init\_\_.py** in \_genericInput**(prompt, default, timeout, limit, applyFunc, validationFunc, postValidateApplyFunc, passwordMask)**

186 **else:**

187 **# If there is no default, then raise the timeout/limit exception.**

**--> 188 raise** limitOrTimeoutException

189 **else:**

190 **# If there was no timeout/limit exceeded, let the user enter input again.**

**RetryLimitException**:

7. If blank input is entered three times, what does inputStr(limit=3, default='hello') do?

ANS: When you use limit keyword arguments and also pass a default keyword argument, the function returns the default value instead of raising an exception

In [32]:

response **=** pyip**.**inputStr(limit**=**3,default**=**'hello')

response

Blank values are not allowed.

Blank values are not allowed.

Blank values are not allowed.

Out[32]:

'hello'