1. Is the Python Standard Library included with PyInputPlus?

Ans: No, PyInputPlus is a Python module used for taking inputs with additional validation features. PyInputPlus will keep asking the user for text until they enter valid input.

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

Ans. We can import the module with import pyinputplus as pyip so that we can enter a shorter name when calling the module's functions.

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

Ans: inputInt() : It helps to take integer input.

inputFloat() : Iit helps to take floating point number input.

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

Ans:

import pyinputplus as pyip

inp = pyip.inputInt(prompt = "Enter an Integer... ", min = 0, lessThan = 100 )

print(inp)

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

Ans: allowRegexesSpecify the allowed format,blockRegexesSpecify a format that is not allowed.

You 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. For example, enter the following code into the interactive shell so that inputNum() will accept Roman numerals in addition to the usual numbers:

>>> import pyinputplus as pyip

>>> response = pyip.inputNum(allowRegexes=[r'(I|V|X|L|C|D|M)+', r'zero'])

XLII

>>> response

'XLII'

>>> response = pyip.inputNum(allowRegexes=[r'(i|v|x|l|c|d|m)+', r'zero'])

xlii

>>> response

'xlii'

Of course, this regex affects only what letters the inputNum() function will accept from the user; the function will still accept Roman numerals with invalid ordering such as 'XVX' or 'MILLI' because the r'(I|V|X|L|C|D|M)+' regular expression accepts those strings.

You can also specify a list of regular expression strings that a PyInputPlus function won’t accept by using the blockRegexes keyword argument. Enter the following into the interactive shell so that inputNum() won’t accept even numbers:

>>> import pyinputplus as pyip

>>> response = pyip.inputNum(blockRegexes=[r'[02468]$'])

42

This response is invalid.

44

This response is invalid.

43

>>> response

43

If you specify both an allowRegexes and blockRegexes argument, the allow list overrides the block list. For example, enter the following into the interactive shell, which allows 'caterpillar' and 'category' but blocks anything else that has the word 'cat' in it:

>>> import pyinputplus as pyip

>>> response = pyip.inputStr(allowRegexes=[r'caterpillar', 'category'],

blockRegexes=[r'cat'])

cat

This response is invalid.

catastrophe

This response is invalid.

category

>>> response

'category'

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

Ans. If the user fails to enter valid input, these keyword arguments will cause the function to raise a RetryLimitException or TimeoutException, respectively.

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

Ans: It will take default input 'hello' as input.