**1.Is the Python Standard Library included with PyInputPlus?**

Ans. No, the Python Standard Library is separate from PyInputPlus. PyInputPlus is a third-party library that provides additional functionality for input handling in Python. It is not part of the Python Standard Library and needs to be installed separately.The Python Standard Library, on the other hand, is a collection of modules that come bundled with the Python programming language. It includes modules for various purposes such as file handling, networking, data manipulation, and more. The Python Standard Library is available by default when you install Python, and you don't need to install it separately.

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

Ans. PyInputPlus is commonly imported with the alias pypi to make it easier to reference and use the library in your code. The alias pypi is just a convention and can be any valid Python identifier you choose.

The purpose of importing with an alias is to provide a shorter or more convenient name to refer to the library throughout your code. This can be especially useful if the library name is long or if there is a chance of name conflicts with other imported modules.

By importing **PyInputPlus** as **pypi**, you can use pypi as a shorthand when calling functions or using classes from the PyInputPlus library. For example, instead of writing **pyinputplus**.**inputStr**(), you can write **pypi**.**inputStr()**. This can save typing and make your code more readable.

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

Ans. **inputInt():**This term seems to refer to a custom function that you might have defined to specifically accept integer input. Typically, you can use the input() function to read user input as a string and then convert it to an integer using the int() function. e.g. below how to use

**num = int(input("Enter an integer: "))**

**inputFloat():**

Similarly, inputFloat() appears to be a custom function that you might have defined to specifically accept floating-point numbers as input.Using the input() function, you can read user input as a string and then convert it to a float using the float() function.Here's an example:

**num = float(input("Enter a float: "))**

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

Ans. In order to ensure that the user enters a whole number between 0 and 99 using the PyInputPlus library in Python, you can utilize the inputInt() function with the min and max parameters set accordingly.Here's an example:

**import pyinputplus as pyip**

**user\_input = pyip.inputInt(prompt="Enter a number between 0 and 99: ", min=0, max=99) print("You entered:", user\_input)**

In the example above, the inputInt() function is used to prompt the user for input and ensure that the entered value is an integer. The min parameter is set to 0 and the max parameter is set to 99, which restricts the acceptable range of values.

If the user enters a value outside the specified range or a non-integer value, PyInputPlus will reprompt the user until a valid input within the specified range is provided.

Note that you'll need to have the PyInputPlus library installed in your Python environment in order to use it. You can install it using pip install pyinputplus.

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

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

Ans. The inputStr(limit=3) function call with limit=3 in Python using the PyInputPlus library provides a limit on the number of attempts the user has to enter a non-blank input. If the user enters a blank input three times consecutively, the function will raise a ValidationException

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

**Ans.** If a blank input is entered three times consecutively, the **inputStr(limit=3, default='hello')** function call in Python using the PyInputPlus library will return the default value 'hello'.