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

**Ans:**

No, the Python Standard Library is not included with PyInputPlus

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

**Ans:**

import pyinputplus as pypi is just a common convention used when importing the PyInputPlus library in Python.

The import pyinputplus as pypi line allows you to import the PyInputPlus library and give it an alias or a shorter name, in this case, pypi. By using an alias, you can refer to the library using a shorter name in your code, which can make your code more concise and readable.

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

**Ans:**

inputInt() is used when you specifically want to accept an integer value from the user.

inputInt() validates that the user enters an integer value. If the user enters a non-integer value, it will display an error message and prompt for input agai

inputFloat() is used when you specifically want to accept a floating-point (decimal) value.

inputFloat() validates that the user enters a floating-point value. It allows both integer and decimal input.

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

**Ans:**

inputInt() function is used to prompt the user for an integer input between 0 and 99 (inclusive). The min argument specifies the minimum allowed value (0 in this case), and the max argument specifies the maximum allowed value.

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

**Ans:**

In PyInputPlus, the keyword arguments allowRegexes and blockRegexes are used to specify regular expressions that define patterns of text that are either allowed or blocked as user input.

The allowRegexes keyword argument accepts a list of regular expressions. If provided, PyInputPlus will allow input only if it matches any of the regular expressions in the allowRegexes list. In other words, the input must satisfy at least one of the allowed regular expressions to be considered valid.

The blockRegexes keyword argument also accepts a list of regular expressions. If provided, PyInputPlus will block input that matches any of the regular expressions in the blockRegexes list. If the input matches any of the blocked regular expressions, PyInputPlus will display an error message and prompt for input again.

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

**Ans:**

If a blank input is entered three times consecutively when using inputStr(limit=3) in PyInputPlus, the function will raise a TimeoutException.

The limit parameter in **inputStr(limit=3)** specifies the maximum number of attempts or inputs allowed before a timeout occurs. In this case, if the user enters a blank input three times in a row, PyInputPlus will raise a TimeoutException indicating that the maximum limit has been reached.

**import pyinputplus as pypi**

**try:**

**text = pypi.inputStr("Enter text: ", limit=3)**

**print("You entered:", text)**

**except pypi.TimeoutException:**

**print("Maximum limit reached. No input received.")**

the inputStr() function is used with limit=3 to prompt the user for text input. If the user fails to provide any input for three consecutive attempts (blank input), a TimeoutException will be raised, and the corresponding message will be printed.

**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 when using inputStr(limit=3, default='hello') in PyInputPlus, the function will return the default value 'hello'.

The limit parameter in inputStr(limit=3, default='hello') specifies the maximum number of attempts or inputs allowed before a timeout occurs. In this case, if the user enters a blank input three times in a row, PyInputPlus will return the default value specified, which is 'hello'.

**import pyinputplus as pypi**

**text = pypi.inputStr("Enter text: ", limit=3, default='hello')**

**print("You entered:", text)**

the inputStr() function is used with limit=3 and default='hello' to prompt the user for text input. If the user fails to provide any input for three consecutive attempts (blank input), the function will return the default value 'hello'. The value will be assigned to the text variable, and the message "You entered: hello" will be printed.