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

No, PyInputPlus is not included in the Python Standard Library. PyInputPlus is a third-party module that provides additional functionality for validating and manipulating user input in Python programs.

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

The "as pypi" part of the import statement is an alias, a short name that is used to refer to the imported module. This is a common convention when importing modules in Python, and it makes the code easier to read and less verbose. By using "pypi" as an alias for PyInputPlus, the code can refer to the module with a shorter and more intuitive name.

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

inputInt() and inputFloat() are functions provided by the PyInputPlus module in Python. They are used to retrieve user input and convert it into either an integer or a floating-point number, respectively.

The difference between the two functions is the type of data they return. inputInt() returns an integer value, while inputFloat() returns a floating-point value. This means that if the user inputs a decimal number, inputFloat() will preserve the decimal component, while inputInt() will round down to the nearest integer.

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

You can use the inputInt() function from the PyInputPlus module to ensure that the user enters a whole number between 0 and 99. The function takes several optional arguments that allow you to specify constraints on the input.

Here's an example of how you can use inputInt() to ensure that the user enters a whole number between 0 and 99:

import pyinputplus as pypi

age = pypi.inputInt("What's your age? (0-99): ", min=0, max=99)

In this example, the min argument specifies the minimum allowed value (0), and the max argument specifies the maximum allowed value (99). If the user enters a value outside of this range, the function will keep prompting the user until a valid value is entered.

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

In PyInputPlus, the allowRegexes and blockRegexes keyword arguments are used to specify the allowed and disallowed inputs, respectively, for a function like inputStr(). These arguments take a list of regular expressions as values.

allowRegexes is used to specify a list of regular expressions that define the allowed inputs. If the user's input matches one of the regular expressions in the list, it is considered valid and accepted.

blockRegexes is used to specify a list of regular expressions that define the disallowed inputs. If the user's input matches one of the regular expressions in the list, it is considered invalid and the function will prompt the user again until a valid input is entered.

Here's an example of how you could use allowRegexes and blockRegexes:

import pyinputplus as pypi

# Only accept inputs that are either 'yes' or 'no'

response = pypi.inputStr("Do you want to continue? (yes/no): ",

allowRegexes=['yes', 'no'],

blockRegexes=[r'(?i)[^yesno]'])

In this example, the allowRegexes argument is set to ['yes', 'no'], so the input must match either of these two strings exactly. The blockRegexes argument is set to a regular expression that matches anything other than 'yes' or 'no', regardless of case, so any input that doesn't match 'yes' or 'no' will be rejected.

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

If a blank input is entered three times, the inputStr(limit=3) function from PyInputPlus will raise a RetryLimitException after the third attempt. The limit argument is used to specify the maximum number of times the function will prompt the user for input.

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

If a blank input is entered three times, the inputStr(limit=3, default='hello') function from PyInputPlus will return the default value 'hello' after the third attempt. The limit argument is used to specify the maximum number of times the function will prompt the user for input, and the default argument is used to specify a default value to return if the limit is reached.

Here's an example of how you could use inputStr(limit=3, default='hello'):

import pyinputplus as pypi

name = pypi.inputStr("What's your name? ", limit=3, default='hello')

print(name)

In this example, the inputStr function is used to prompt the user for their name. The limit argument is set to 3, so the function will prompt the user a maximum of three times. If the user enters a blank input three times, the function will return the default value 'hello'.