1. No, the Python Standard Library is not included with PyInputPlus. PyInputPlus is a separate library that provides additional functionality for taking user input in a robust manner.
2. PyInputPlus is commonly imported with **import pyinputplus as pypi** to simplify the code and make it more concise. This alias allows developers to use a shorter name (**pypi**) when referring to PyInputPlus functions, making the code more readable and reducing the likelihood of naming conflicts.
3. **inputInt()** and **inputFloat()** are PyInputPlus functions used to specifically validate and return integers and floating-point numbers, respectively. **inputInt()** ensures that the user enters an integer, while **inputFloat()** ensures that the user enters a floating-point number.
4. To ensure that the user enters a whole number between 0 and 99 using PyInputPlus, you can use the **min** and **max** parameters. Here's an example:

import pyinputplus as pypi user\_input = pypi.inputInt(prompt='Enter a whole number between 0 and 99: ', min=0, max=99)

This code will keep prompting the user until a valid whole number within the specified range is entered.

1. The **allowRegexes** and **blockRegexes** keyword arguments in PyInputPlus are used for regex-based pattern matching to allow or block certain input values. These arguments take a list of regular expression patterns. **allowRegexes** allows input that matches any of the specified patterns, while **blockRegexes** blocks input that matches any of the specified patterns.
2. If a blank input is entered three times with **inputStr(limit=3)**, PyInputPlus raises a **pyinputplus.RetryLimitException** after three consecutive blank inputs. The **limit** parameter specifies the maximum number of retries allowed before raising the exception.
3. If a blank input is entered three times with **inputStr(limit=3, default='hello')**, PyInputPlus does not raise an exception. Instead, it returns the default value **'hello'** after reaching the specified limit of three retries with blank input.