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

**Answer: No, PyInputPlus is not part of the Python Standard Library. It is a third-party module that needs to be installed separately.**

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

**Answer: PyInputPlus is commonly imported with the alias "pypi" to make it easier to type and read the code, especially when multiple PyInputPlus functions are used in the same script.**

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

**Answer: inputInt() function is used to take integer input from the user, whereas inputFloat() function is used to take floating-point input from the user.**

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

**Answer: You can use inputInt() function with the min and max parameters set to 0 and 99 respectively. Here is an example code:**

**import pyinputplus as pypi**

**number = pypi.inputInt(prompt="Enter a number between 0 and 99: ", min=0, max=99)**

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

**Answer: The allowRegexes and blockRegexes keyword arguments in PyInputPlus functions take in regular expressions as input. The allowRegexes argument specifies a list of regular expressions that are allowed as input, while the blockRegexes argument specifies a list of regular expressions that are not allowed as input.**

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

**Answer: If a blank input is entered three times, inputStr(limit=3) will raise a RetryLimitException, which indicates that the user has exceeded the limit of retries specified by the limit parameter.**

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

**Answer: If blank input is entered three times, inputStr(limit=3, default='hello') will return the string "hello" as the default value, since the user has exceeded the limit of retries specified by the limit parameter.**