1. Is the Python Standard Library included with PyInputPlus?

ANSWER.

No

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

ANSWER.

Can be more concise and readable, especially in larger projects.

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

ANSWER.

`inputInt()` is used to get integer input from the user, while `inputFloat()` is used to get floating-point input. They both provide input validation and reprompting functionality to ensure that the user enters a valid value of the respective type.

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

ANSWER.

You can use PyInputPlus to ensure that the user enters a whole number between 0 and 99 using the `inputInt()` function along with the `min`, `max`, and `greaterThan` parameters.

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

ANSWER.

In PyInputPlus, the keyword arguments `allowRegexes` and `blockRegexes` are used to specify regular expressions that define patterns to allow or block input values, respectively. These regular expressions can be used to filter and validate user input according to custom criteria.

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

ANSWER.

If `inputStr(limit=3)` is called and the user enters a blank input three times consecutively, PyInputPlus will raise a `pyinputplus.RetryLimitException`. This exception is raised when the user exceeds the maximum number of retries specified by the `limit` parameter.

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

ANSWER.

If `inputStr(limit=3, default='hello')` is called and the user enters a blank input three times consecutively, PyInputPlus will return the default value `'hello'` instead of raising a `pyinputplus.RetryLimitException`.