1. Is the Python Standard Library included with PyInputPlus?

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

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

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

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

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

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

ANSWER

1. No, the Python Standard Library is not included with PyInputPlus. PyInputPlus is a separate third-party library for input validation and handling, and it is not a part of the Python Standard Library.

2. PyInputPlus is commonly imported with the alias `pypi` (or another alias of your choice) to make the code more concise and readable. Using `import pyinputplus as pypi`, you can use shorter function calls like `pypi.inputInt()` or `pypi.inputChoice()`, which is often more convenient.

3. `inputInt()` and `inputFloat()` are both functions provided by PyInputPlus for input validation. The primary distinction between them is the type of value they expect from the user:

- `inputInt()` expects the user to enter an integer value, and it will raise an exception if the input is not a valid integer.

- `inputFloat()` expects the user to enter a floating-point number, and it will raise an exception if the input is not a valid floating-point number.

4. To ensure that the user enters a whole number between 0 and 99 using PyInputPlus, you can use the `inputInt()` function with the `min` and `max` keyword arguments as follows:

```python

import pyinputplus as pypi

number = pypi.inputInt(prompt='Enter a number: ', min=0, max=99)

```

This code will prompt the user to enter a number and enforce that the input is an integer between 0 and 99.

5. The `allowRegexes` and `blockRegexes` keyword arguments in PyInputPlus are used to specify regular expressions that define patterns for allowed and blocked input. When you pass regular expressions to these arguments, PyInputPlus will allow or block input based on whether it matches these regular expressions.

6. If a blank input is entered three times using `inputStr(limit=3)`, and the user does not provide valid input within those three attempts, PyInputPlus will raise a `pyinputplus.RetryLimitException`. This exception indicates that the user has exceeded the allowed number of retries without providing valid input.

7. If blank input is entered three times using `inputStr(limit=3, default='hello')`, and the user does not provide valid input within those three attempts, PyInputPlus will return the default value 'hello' after the third attempt. Instead of raising an exception, it uses the provided default value to handle the situation where the user exceeds the retry limit.