

1. Is the Python Standard Library included with PyInputPlus?

Answer: - No, the Python Standard Library is not included with PyInputPlus. PyInputPlus is a separate third-party library that provides additional functionality for taking user input.

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

Answer: - PyInputPlus is commonly imported with `import pyinputplus as pypi` to provide a shorter alias ('pypi') for the module name. This allows you to use shorter and more convenient code when calling PyInputPlus functions.

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

Answer: - `inputInt()` is used to take input from the user and ensures that the input is an integer value. If the user enters a value that cannot be converted to an integer, an error is raised. On the other hand, `inputFloat()` is used to take input from the user and ensures that the input is a floating-point number. It allows the user to enter decimal values.

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

Answer: - `inputInt()` is used to take input from the user and ensures that the input is an integer value. If the user enters a value that cannot be converted to an integer, an error is raised. On the other hand, `inputFloat()` is used to take input from the user and ensures that the input is a floating-point number. It allows the user to enter decimal values.

```
import pyinputplus as pypi
```

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

The `min` argument specifies the minimum allowed value, and the `max` argument specifies the maximum allowed value. If the user enters a number outside this range, PyInputPlus will ask them to enter a valid number.

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

Answer: - The `allowRegexes` and `blockRegexes` keyword arguments in PyInputPlus are used to specify regular expressions that define patterns of text that are either allowed or blocked from being entered by the user. `allowRegexes` contains a list of regular expressions for allowed patterns, and `blockRegexes` contains a list of regular expressions for blocked patterns. PyInputPlus will match the input against these regular expressions to determine if it is allowed or blocked.

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

Answer: - If a blank input is entered three times when using `inputStr(limit=3)`, PyInputPlus will raise a `RetryLimitException` and display an error message. It means that the user has exceeded the maximum number of retries allowed (in this case, 3) for entering a non-blank input. You can catch the exception and handle it accordingly in your code.

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

Answer: - If blank input is entered three times when using `inputStr(limit=3, default='hello')`, `PyInputPlus` will return the default value 'hello'. In this case, when the user exceeds the maximum number of retries for entering a non-blank input, the function will return the default value instead of raising an exception.