Python Assignment - 8

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

=> No, the Python Standard Library is not included with PyInputPlus. PyInputPlus is a third-party library that provides additional functionality for taking user input in Python, such as input validation, type conversion, and other input handling features.

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

=> PyInputPlus is commonly imported with import pyinputplus as pypi to provide a shorter and more convenient alias for the module name. This alias allows you to use a shorter reference to the PyInputPlus module throughout your code, making it easier to read and write.

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

=> In PyInputPlus, you can distinguish between inputInt() and inputFloat() based on the expected type of the input value.

The inputInt() function is used to accept integer input from the user. It validates the input to ensure that it is a valid integer and re-prompts the user if the input is not an integer. If the input is valid, it returns the entered value as an integer.

On the other hand, the inputFloat() function is used to accept floating-point (decimal) input from the user. It validates the input to ensure that it is a valid floating-point number and re-prompts the user if the input is not a float. If the input is valid, it returns the entered value as a float.

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

=> To ensure that the user enters a whole number between 0 and 99 using PyInputPlus, you can use the inputInt() function with additional arguments to specify the minimum and maximum allowed values.

Example:

import pyinputplus as pypi

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

print(you entered: {number}")

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

=> In PyInputPlus, the keyword arguments allowRegexes and blockRegexes are used to specify regular expressions that control which inputs are allowed and blocked, respectively.

* The allowRegexes argument accepts a list of regular expressions. When provided, PyInputPlus will only accept inputs that match at least one of the regular expressions in the allowRegexes list. If an input does not match any of the allowed regular expressions, PyInputPlus will re-prompt the user for valid input.
* On the other hand, the blockRegexes argument accepts a list of regular expressions. If provided, PyInputPlus will block any input that matches at least one of the regular expressions in the blockRegexes list. If a blocked input is entered, PyInputPlus will re-prompt the user until a non-blocked input is provided.

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

=> When using inputStr(limit=3) in PyInputPlus and a blank input is entered three times consecutively, it will raise a ValidationException.

The limit argument in inputStr() specifies the maximum number of times PyInputPlus will allow re-prompts for invalid input before raising an exception. By default, limit is set to ‘3’.

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

=> When using inputStr(limit=3, default='hello') in PyInputPlus and a blank input is entered three times consecutively, it returns the default value provided ('hello') instead of raising a ValidationException.