**FSDS MAY BATCH 2022(Python Basics 8)**

**Submitted by: Shubham Tiwari**

Q1. Is the Python Standard Library included with PyInputPlus?

Ans: No, PyInputPlus is not part of the Python Standard Library. It is an external library that provides additional input validation functions for Python.

Q2. Why is PyInputPlus commonly imported with import pyinputplus as pypi?

Ans: The commonly used import statement **import pyinputplus as pypi** is used for convenience and readability purposes. The "as pypi" part of the import statement creates an alias for the library name. This allows the user to reference the library in their code using the shortened alias, "pypi", instead of the full library name, "pyinputplus". This makes the code easier to read and write, as well as reducing the risk of name collisions with other libraries that may have similar names.

Q3. How do you distinguish between inputInt() and inputFloat()?

Ans: The **inputInt()** and **inputFloat()** functions in PyInputPlus are used to validate user input and return either an integer or a floating-point number, respectively.

The **inputInt()** function takes in user input as a string and returns an integer if the input is a valid integer. If the input is not a valid integer, it will raise an exception.

The **inputFloat()** function takes in user input as a string and returns a floating-point number if the input is a valid floating-point number. If the input is not a valid floating-point number, it will raise an exception.

So, to distinguish between the two, you need to determine whether you expect the user to input an integer or a floating-point number, and then use the appropriate function to validate and return the input.

Q4. Using PyInputPlus, how do you ensure that the user enters a whole number between 0 and 99?

Ans: You can use the **inputInt()** function of the PyInputPlus library to ensure that the user enters a whole number between 0 and 99. Here's an example:

**import pyinputplus as pyip**

**num = pyip.inputInt(prompt='Enter a number between 0 and 99: ', min=0, max=99)**

The **inputInt()** function takes the following parameters:

* **prompt**: The message displayed to the user asking for input
* **min**: The minimum acceptable value for the input
* **max**: The maximum acceptable value for the input

In this case, **min** is set to **0** and **max** is set to **99** to ensure that the user enters a whole number between 0 and 99.

Q5. What is transferred to the keyword arguments allowRegexes and blockRegexes?

Ans: The values passed to the **allowRegexes** and **blockRegexes** parameters in a function call are transferred to the keyword arguments of the same name. These keyword arguments are used to specify a list of regular expressions that are used to allow or block certain strings or patterns. The actual implementation of how these regular expressions are used depends on the function definition.

Q6. If a blank input is entered three times, what does inputStr(limit=3) do?

Ans: The **inputStr(limit=3)** function is used to get a string input from the user, and the **limit** argument specifies the maximum number of attempts that the user can make to enter a valid string.

If a blank input is entered three times, and **limit=3**, then the function will return an error after the third attempt, as the maximum limit of attempts has been reached without a valid string being entered. The exact error message will depend on the implementation of the **inputStr** function.

Q7. If blank input is entered three times, what does inputStr(limit=3, default=’hello’) do?

Ans: ‘**inputStr(limit=3, default='hello')** is a function in an unknown programming language that allows a user to input a string, with a maximum of 3 attempts. If the user fails to input a string in the 3 attempts, the function returns the default value **'hello'**.