

1. Is the Python Standard Library included with PyInputPlus?

Ans: PyInputPlus is a Python module used for taking inputs with additional validation features. PyInputPlus will keep asking the user for text until they enter valid input.

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

Ans: The 'as pypi' code in the import statement saves us from typing `pyinputplus` each time we want to call a PyInputPlus function.

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

Ans:

➔ "`inputInt()`" : Accepts an integer value. This also takes additional parameters 'min', 'max', 'greaterThan' and 'lessThan' for bounds. Returns an int.

➔ "`inputFloat()`" : Accepts a floating-point numeric value. Also takes additional 'min', 'max', 'greaterThan' and 'lessThan' parameters. Returns a float.

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

Ans:

```
import pyinputplus as pypi

# integer input with
# specific bounds
inp = pypi.inputInt(prompt = "Enter an Integer... ", min = 0, max = 99 )

print(inp)
```

```
Enter an Integer... 5.5
'5.5' is not an integer.
Enter an Integer... -9
Number must be at minimum 0.
Enter an Integer... 600
Number must be at maximum 99.
Enter an Integer... 95
95
```

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

Ans: The `allowRegexes` and `blockRegexes` keyword arguments take a list of regular expression strings to determine what the PyInputPlus function will accept or reject as valid input.

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

Ans: It will raise `RetryLimitException`.

```
[12]: inp = pypi.inputStr(limit=3)

Blank values are not allowed.

Blank values are not allowed.

Blank values are not allowed.
-----
ValidationException                                Traceback (most recent call last)
D:\Anaconda\lib\site-packages\pyinputplus\__init__.py in _genericInput(prompt, default, timeout, limit, applyFunc, validationFunc, postValidateApplyFunc, passwordMask)
    166         try:
--> 167             possibleNewUserInput = validationFunc(
    168                 userInput

D:\Anaconda\lib\site-packages\pyinputplus\__init__.py in (value)
    242
--> 243     validationFunc = lambda value: pysv._prevalidationCheck(
    244         value, blank=blank, strip=strip, allowRegexes=allowRegexes, blockRegexes=blockRegexes, excMsg=None,

D:\Anaconda\lib\site-packages\pysimplevalidate\__init__.py in _prevalidationCheck(value, blank, strip, allowRegexes, blockRegexes, excMsg)
    249     # value is blank but blanks aren't allowed.
--> 250     _raiseValidationException_("Blank values are not allowed.", excMsg)
    251     elif blank and value == "":

D:\Anaconda\lib\site-packages\pysimplevalidate\__init__.py in _raiseValidationException(standardExcMsg, customExcMsg)
    221     if customExcMsg is None:
--> 222         raise ValidationException(str(standardExcMsg))
    223     else:

ValidationException: Blank values are not allowed.

During handling of the above exception, another exception occurred:

RetryLimitException                                Traceback (most recent call last)
```

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

Ans: The function returns the default value instead of raising an exception.

```
: inp = pypi.inputStr(limit=3, default='hello')
```

Blank values are not allowed.

Blank values are not allowed.

Blank values are not allowed.

```
: inp
```

```
: 'hello'
```