

## Assignment - 8

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

```
# PyInputPlus is not the part of Python Standard Library, so one need to install it using pip
```

```
pip install pyinputplus
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting pyinputplus
  Downloading PyInputPlus-0.2.12.tar.gz (20 kB)
  Installing build dependencies: started
  Installing build dependencies: finished with status 'done'
  Getting requirements to build wheel: started
  Getting requirements to build wheel: finished with status 'done'
  Preparing metadata (pyproject.toml): started
  Preparing metadata (pyproject.toml): finished with status 'done'
Collecting stdiomask>=0.0.3
  Downloading stdiomask-0.0.6.tar.gz (3.6 kB)
  Installing build dependencies: started
  Installing build dependencies: finished with status 'done'
  Getting requirements to build wheel: started
  Getting requirements to build wheel: finished with status 'done'
  Preparing metadata (pyproject.toml): started
  Preparing metadata (pyproject.toml): finished with status 'done'
Collecting pysimplevalidate>=0.2.7
  Downloading PySimpleValidate-0.2.12.tar.gz (22 kB)
  Installing build dependencies: started
  Installing build dependencies: finished with status 'done'
  Getting requirements to build wheel: started
  Getting requirements to build wheel: finished with status 'done'
  Preparing metadata (pyproject.toml): started
  Preparing metadata (pyproject.toml): finished with status 'done'
Building wheels for collected packages: pyinputplus, pysimplevalidate, stdiomask
  Building wheel for pyinputplus (pyproject.toml): started
  Building wheel for pyinputplus (pyproject.toml): finished with status 'done'
  Created wheel for pyinputplus: filename=PyInputPlus-0.2.12-py3-none-any.whl size=11322 sha256=79e3f8179a4cdd8119709816d58c3c8
  13d7e19b375df185e35c4d21cb6edfcde
  Stored in directory: c:\users\admin\appdata\local\pip\cache\wheels\2d\72\d9\818396cecb072b3d697a47b007d9bbad74859a83a635e47d4
  d
  Building wheel for pysimplevalidate (pyproject.toml): started
  Building wheel for pysimplevalidate (pyproject.toml): finished with status 'done'
  Created wheel for pysimplevalidate: filename=PySimpleValidate-0.2.12-py3-none-any.whl size=16204 sha256=91c0879afcedf82cab81e
  dab6f92e3092b8afc0fe03dc39c835dbc8d32dfebd
  Stored in directory: c:\users\admin\appdata\local\pip\cache\wheels\05\ad\4d\7289d8b9a519e88167728584876a1a1073a3f71666d875aba
  7
  Building wheel for stdiomask (pyproject.toml): started
  Building wheel for stdiomask (pyproject.toml): finished with status 'done'
  Created wheel for stdiomask: filename=stdiomask-0.0.6-py3-none-any.whl size=3321 sha256=9f644873d191de0386454b0ea26ac6913d77b
  d547a3ce025b03eb7987fa302d5
  Stored in directory: c:\users\admin\appdata\local\pip\cache\wheels\41\ee\55\6e4b70f6248b280caae3d938bc543206bb1a75988b739f123
  8
Successfully built pyinputplus pysimplevalidate stdiomask
Installing collected packages: stdiomask, pysimplevalidate, pyinputplus
Successfully installed pyinputplus-0.2.12 pysimplevalidate-0.2.12 stdiomask-0.0.6
Note: you may need to restart the kernel to use updated packages.
```

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

```
In [1]: # pypi is acronym of PyInputPlus. Pypi saves us from typing pyinputplus each time we want to call a PyInputPlus function.
```

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

```
In [2]: #inputInt() : Accepts an integer value, and returns int value
        #inputFloat() : Accepts integer/floating point value and returns float value
```

```
In [3]: import pyinputplus as pyip
        pyip.inputInt()
        pyip.inputFloat()
```

```
1.5
'1.5' is not an integer.
10
7
```

```
Out[3]: 7.0
```

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

```
In [4]: #In the inputInt function we can set the min = 0 and max =99 to ensure user enters number between 0 and 99
```

```
In [5]: pyip.inputInt(min = 0, max =99)
```

```
123
Number must be at maximum 99.
50
```

```
Out[5]: 50
```

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

```
In [6]: #We can also use regular expressions to specify whether an input is allowed or not.
```

```
In [7]: response = pyip.inputNum(allowRegexes=[r'(I|V|X|L|C|D|M)+', r'zero']) # it allows roman letters as numbers too
V
```

```
In [8]: response = pyip.inputNum(blockRegexes=[r'[02468]$'])# blocks the even numbers
```

```
8
This response is invalid.
4
This response is invalid.
7
```

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

```
In [9]: #It will throw RetryLimitException exception.
```

```
In [10]: response = pyip.inputStr(limit=3)
```

```
Blank values are not allowed.
```

```
Blank values are not allowed.
```

```
Blank values are not allowed.
```

```
-----
ValidationException                                Traceback (most recent call last)
File ~\AppData\Roaming\Python\Python310\site-packages\pyinputplus\_init_.py:167, in _genericInput(prompt, default, timeout, limit, applyFunc, validationFunc, postValidateApplyFunc, passwordMask)
    166 try:
--> 167     possibleNewUserInput = validationFunc(
    168         userInput
    169     ) # If validation fails, this function will raise an exception. Returns an updated value to use as user input (e.g. stripped of whitespace, etc.)
    170     if possibleNewUserInput is not None:

File ~\AppData\Roaming\Python\Python310\site-packages\pyinputplus\_init_.py:243, in inputStr.<locals>.<lambda>(value)
    241 pysv._validateGenericParameters(blank, strip, allowRegexes, blockRegexes)
--> 243 validationFunc = lambda value: pysv._prevalidationCheck(
    244     value, blank=blank, strip=strip, allowRegexes=allowRegexes, blockRegexes=blockRegexes, excMsg=None,
    245 )[1]
    247 return _genericInput(
```

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

```
In [11]: # When limit keyword arguments are used.
```

```
In [12]: response = pyip.inputStr(limit=3,default='hello')
response
```

```
Blank values are not allowed.
```

```
Blank values are not allowed.
```

```
Blank values are not allowed.
```

```
Out[12]: 'hello'
```