ch01-intro

May 5, 2019

1 Chapter 1: Computing with Python

1.0.1 Overview: a typical Python-based scientific computing stack.

- Resources:
- SciPy
- Python Numeric & Scientific topics

1.1 Interpreter

- The easist way to execute Python code: run the program directly.
- Use Jupyter magic command to write Python source file to disk:

```
In [1]: %%writefile hello.py
        print("Hello from Python!")
```

```
Overwriting hello.py
```

- Use the ! system shell command (included in the Python Jupyter kernel) to interactively run Python with hello.py as its argument.

```
In [2]: !python hello.py
```

```
Hello from Python!
```

```
In [3]: !python --version
```

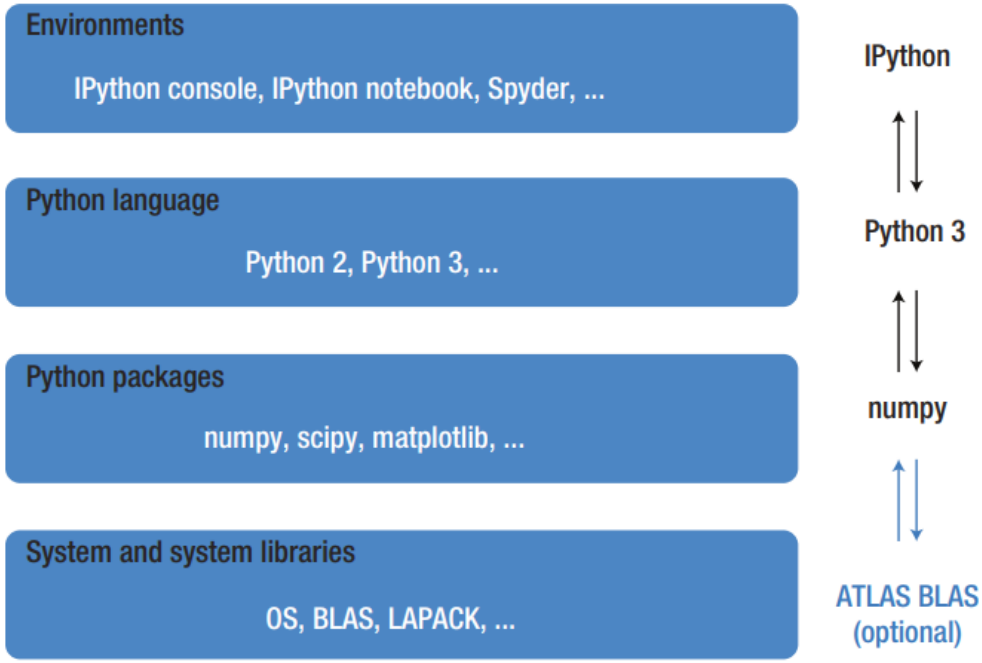
```
Python 3.6.5 :: Anaconda, Inc.
```

1.2 Input and output caching

- Input & output history can be accessed using **In** (a list) & **Out** (a dictionary). Both can be indexed with a cell number.

```
In [4]: 3 * 3
```

```
Out[4]: 9
```



software stack

```
In [5]: In[1]

Out[5]: 'get_ipython().run_cell_magic(\'writefile\', \'hello.py\', \'print("Hello from Python!")
```

- A single underscore = the most recent output;
- A double underscore = the *next* most recent output.

```
In [6]: 1+1

Out[6]: 2

In [7]: 2+2

Out[7]: 4

In [8]: _, __

Out[8]: (4, 2)

In [9]: # In = a list
        In

Out[9]: ['',
         'get_ipython().run_cell_magic(\'writefile\', \'hello.py\', \'print("Hello from Python!"
         "get_ipython().system('python hello.py')",
         "get_ipython().system('python --version')",
         '3 * 3',
```