Intro To Python

NOSERYOUNG

Python

What is it?

- Python is a **interpreted, multi paradigm, high-level, dynamically-typed** programming language
- Designed to be easy to read and simple to implement ➔ rapid development & “glue language”
- Open source
- Built on C

NOSERYOUNG

Python

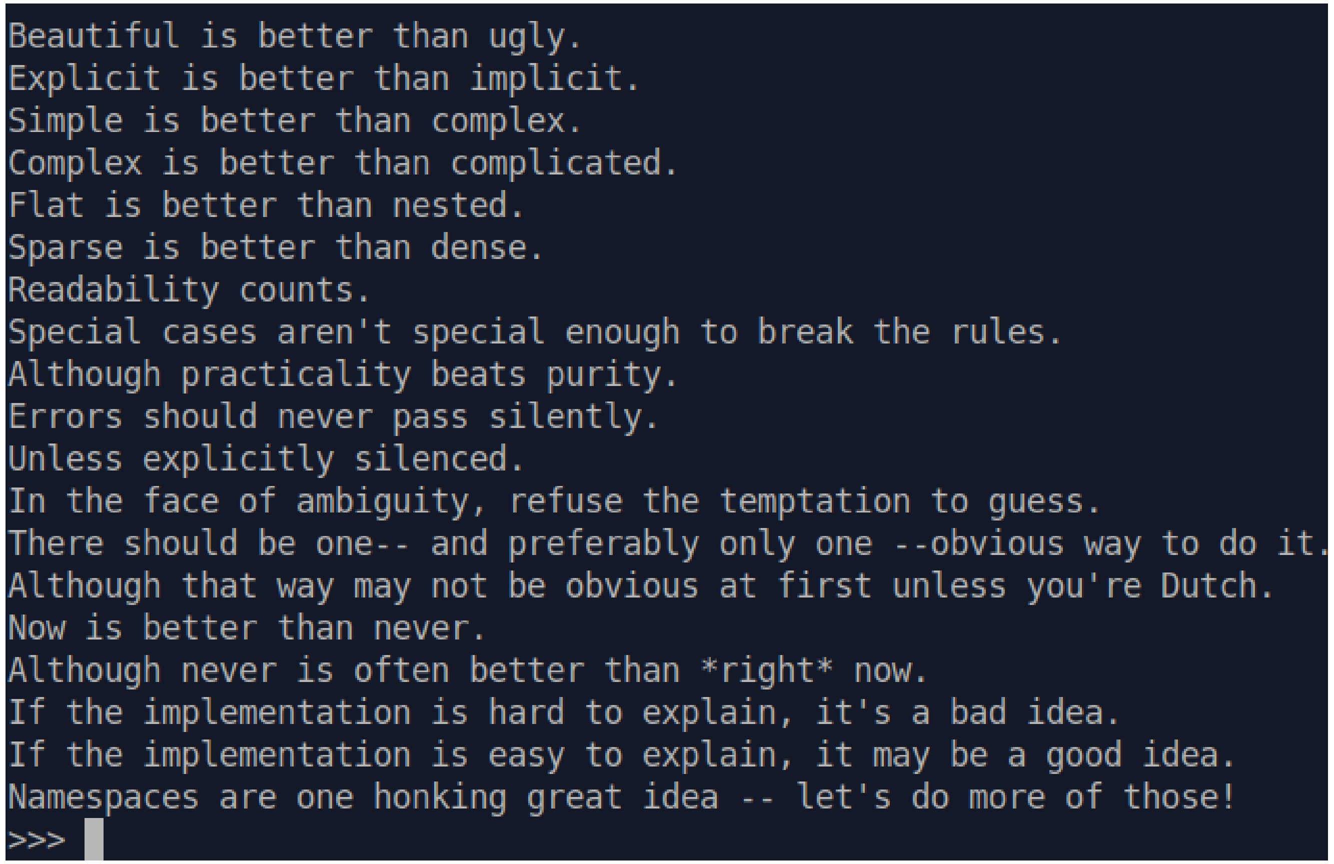
History

- Conceived in 1989
- 2000: “modern” python 2.0
- 2012: Anaconda package manager and distribution
- Today: Python 3.x, some legacy code in 2.7
  - Scientific and numeric computing
  - Machine Learning applications
  - Image processing

Python

The Zen of Python

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Readability counts.



```
Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>>
```

NOSERYOUNG

Anaconda

- A distribution of Python and R programming languages for scientific computing.
- Many typical packages preinstalled (miniconda for minimal install)
- Packages are managed in **virtual environments**

Python

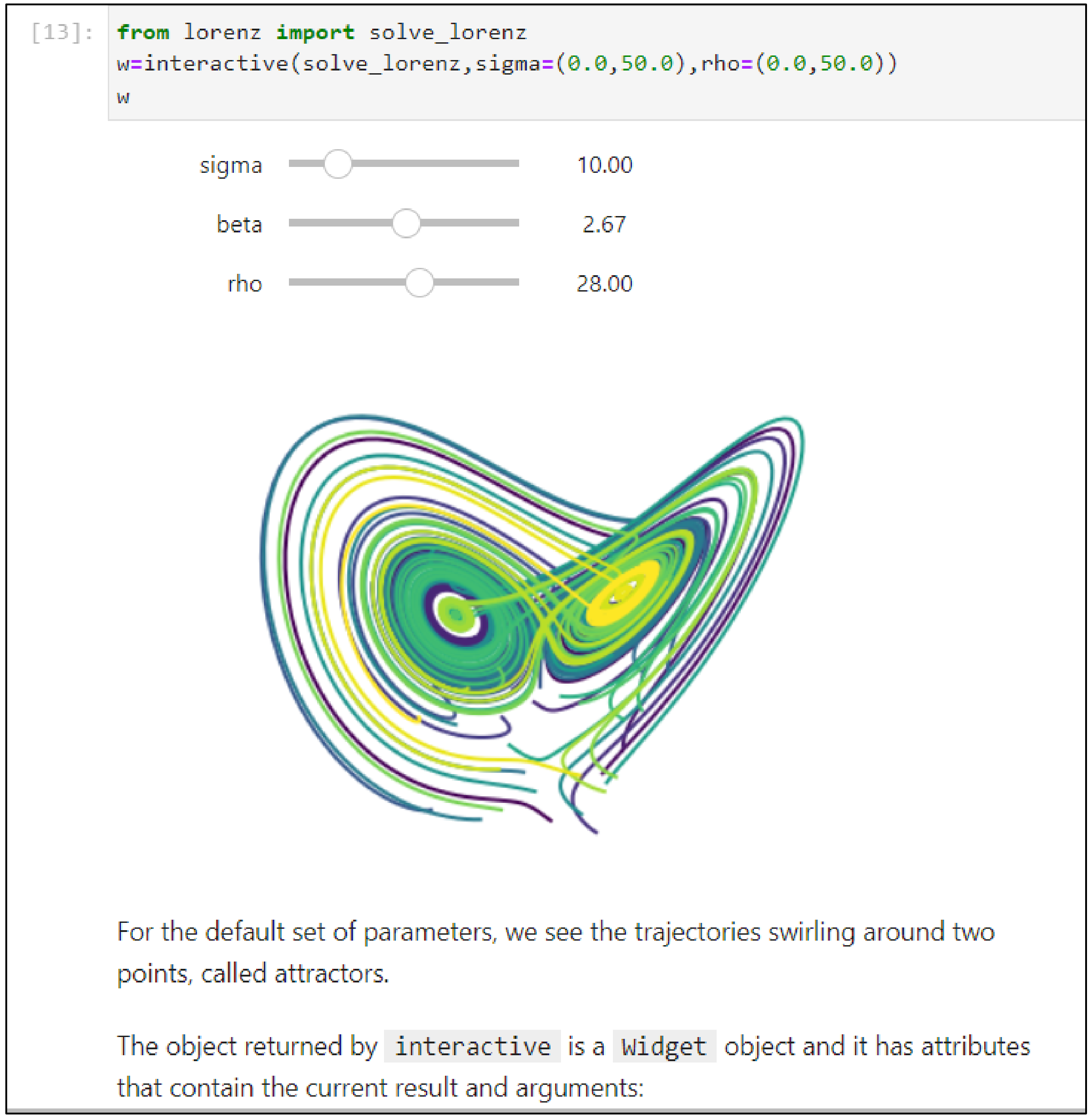
Jupyter Notebooks

- A web-based **interactive computational environment**

Python

Jupyter Notebooks

- A web-based **interactive computational environment**
- A list of input/output cells which can contain **code, text (using Markdown), mathematics, plots and media.**
- Try it yourself: https://mybinder.org/v2/gh/jupyterlab/jupyterlab-demo/HEAD?urlpath=lab/tree/demo



```
[13]: from lorenz import solve_lorenz
      w=interactive(solve_lorenz,sigma=(0.0,50.0),rho=(0.0,50.0))
      w
```

```
sigma   10.00
beta     2.67
rho     28.00
```

For the default set of parameters, we see the trajectories swirling around two points, called attractors.

The object returned by `interactive` is a `Widget` object and it has attributes that contain the current result and arguments:

NOSERYOUNG

Python Syntax

Python

Syntax

- Dynamically typed

```python
# Declaring variables
x, y = 12, 10
isTrue = True
greeting = "Welcome!"
```

Python

Syntax

- Dynamically typed
- Code blocks by indentation
- Lists and arrays

```
for i in range(1,3):
    for j in range(4,6):
        print(i+j)
```

Python

Syntax

- Dynamically typed
- Code blocks by indentation
- Lists and arrays

```python
# Working with Lists
countries = [ "Portugal",
"England", "Brazil"]
numbers = [12, 14, 9, 10, 9]
# Sorting a List
countries.sort()
# Looping a List
for country in countries:
    print(country)
```

NOSERYOUNG

Python

Syntax

- Dynamically typed
- Code blocks by indentation
- Lists and Arrays
- Classes & Functions

```
# Defining a class with constructor and a
method
class Person:
      def __init__(self, name,
                    nationality, age):
             self.name = name
             self.nationality = nationality
             self.age = age
# Declaring a method with if/else statement #
and returning a boolean
      def isMinor(self):
             if self.age >= 18:
                    return False
             else:
                    return True
```

Python

Syntax

- Dynamically typed
- Code blocks by indentation
- Lists and Arrays
- Classes & Functions

```
# Instantiate an Object
from class Person
p1 = Person("John Doe",
countries[0], 19)

print(p1.isMinor())
```

Hands-On

Part 1

1. Get Python and Juptyer working
   - Setup a virtual environment as explained in the Documentation
2. Explore the Numpy and Pandas example-noteboks