

Python

What is Python?

an interpreted, dynamic, object-oriented programming language

Python

What is Python?

an interpreted, dynamic, object-oriented programming language

Why should I care?

- free software
- easy to learn
- object-oriented programming is not necessary (but possible)
- supports functional programming

Python

What is Python?

an interpreted, dynamic, object-oriented programming language

Why should I care?

- free software
- easy to learn
- object-oriented programming is not necessary (but possible)
- supports functional programming
- includes vast collection of tools:
 - email
 - re (regular expressions)
 - threading
 - xml, HTMLParser, HTTP-Server
 - GUIs: tkinter, wxPython, PyQt, ...

Python

What is Python?

an interpreted, dynamic, object-oriented programming language

Why should I care?

- free software
- easy to learn
- object-oriented programming is not necessary (but possible)
- supports functional programming
- widely used:
 - Scripting, e.g. for GIMP, Raspberry Pi, ...
 - Zope (Web Application Server)
 - Google, YouTube
 - Dropbox
 - ILM (Star Wars, Terminator 2, Jurassic Park, ...)

Why should I care as a scientist?

- numerical tools:
 - numpy: Arrays
 - scipy:
 - integrate
 - interpolate
 - linalg
 - stats (Statistics)
 - special (special functions)
 - fftpack

Why should I care as a scientist?

- numerical tools:
 - numpy: Arrays
 - scipy:
 - integrate
 - interpolate
 - linalg
 - stats (Statistics)
 - special (special functions)
 - fftpack
- plotting: matplotlib

Why should I care as a scientist?

- numerical tools:
 - numpy: Arrays
 - scipy:
 - integrate
 - interpolate
 - linalg
 - stats (Statistics)
 - special (special functions)
 - fftpack
- plotting: matplotlib
- speed:
 - Cython: Python-to-C + compilation
 - C/C++ backends
 - Parallelization: threading
 - GPU: PyCUDA

Interactive shell

use **ipython** for interactive use – much better than plain python

- ① **basic computations**
- ② **work with arrays**

Interactive shell

use **ipython** for interactive use – much better than plain python

- ① **basic computations**
- ② **work with arrays**

Documentation

- interactive help in the Python shell: `help("<command>")`
- or in IPYTHON also: `object?`
- Python Tutorial:
<https://docs.python.org/2/tutorial/index.html>
- matplotlib-website:
<http://matplotlib.sourceforge.net/gallery.html>