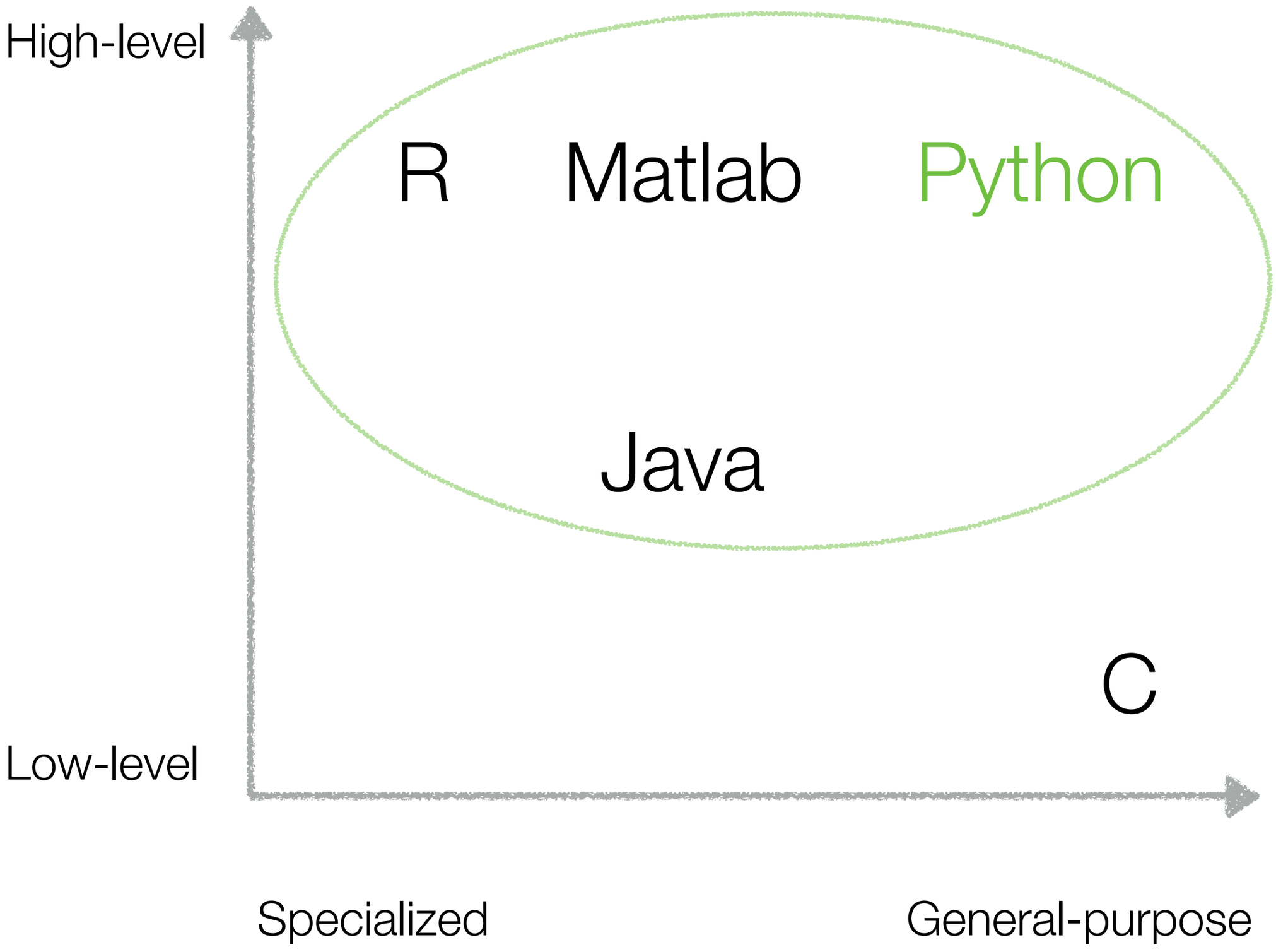
Brief Introduction to Python

Stefan Pfenninger

23 June 2014, SFI Complex Systems Summer School


High-level
R
Matlab
Python
Java
C
Low-level
Specialized
General-purpose

① How to run Python

Code → Compiler → Executable

Code → Interpreter

① Create a script file (.py)

② Use the interpreter interactively

② Dynamic typing

Variables do not need to be declared

Variables do not need to stick to a specific type

③ Whitespace has meaning

C

```
if (x > 1)
{
    y = 10
    z = 0
} else {
    y = 5
    z = 5
}
```

Python

```
if x > 1:
····y = 10
····z = 0
else:
····y = 5
····z = 5

print y
```

④ Everything in Python is an object

`0.25` is a `Float` object
= instance of the Class `Float`.

The class defines methods, such as:

- `Float.is_integer()`
- `Float.real()`
- `Float.imag()`
- …

④ Objects

```
>>> x = 0.25
```

Declaring a variable: equivalent to creating an object

Objects have a **class**, which defines **methods** that operate on the object

```
>>> x.is_integer()
False
```

⑤ Packages and namespaces

```
>>> import math

>>> math.floor(1.99)
1.0
```

- Namespaces mean that it’s always exactly clear where a certain function came from, which keeps code clear and organized

⑤ 3rd-party packages

- About 45,000 packages available
- For example: packages for reading almost any imaginable data format with ease
- “import” to use installed packages:

```
import package_name
```

⑤ Key scientific packages

NumPy	n-dimensional arrays/matrices
SciPy	basic science/stats, numerical analysis
Sympy	symbolic maths
matplotlib	plotting
pandas	time series and tabular data
IPython	interactive Python “lab”

The IPython notebook

- Combines interactive interpreter with the ability to save a file for later use

- Very useful for explorative analysis

- Organize code, comments, and results (e.g. plots) together in one file

IPython Notebook: Intro to Python

http://nbviewer.ipython.org/gist/sjpfenninger/0b96957f27e2a61123ce

IPython Notebook: NetworkX + IPython

http://nbviewer.ipython.org/gist/sjpfenninger/034053040ab23fcbb95b

Additional notes

Other important scientific packages

- Statistical modeling: statsmodels, seaborn
- Machine learning: scikit-learn
- Bayesian statistics: pymc
- GIS: geopandas, shapely, pysal, and others
- Mathematical optimization: pyomo
- Networks: networkx

Which Python version to use?

Free Anaconda distribution: includes all common scientific packages, makes updating easy

https://store.continuum.io/cshop/anaconda/

Documentation is here:

http://docs.continuum.io/anaconda/index.html

What editor to use?

- My choice: a simple text editor (Sublime Text) in combination with IPython notebook

- Spyder is a Matlab-like IDE that is included in the Anaconda distribution: https://pythonhosted.org/spyder/

- PyCharm is a full-featured IDE with a free community edition: https://www.jetbrains.com/pycharm/

How to learn: Documentation

- Interactive learning environments:
  http://www.learnpython.org/
  and http://www.codecademy.com/tracks/python

- Official Tutorial & Dos and Dont's:
  http://docs.python.org/2.7/tutorial/index.html
  http://docs.python.org/2/howto/doanddont.html

- The Hitchhiker's Guide to Python:
  http://docs.python-guide.org/en/latest/

How to learn

- Follow code style guidelines: docs.python-guide.org/en/latest/writing/style/

- Search for answers on stackoverflow.com — many questions are already answered

- Start learning about advanced language features: e.g. https://stackoverflow.com/questions/101268/hidden-features-of-python

How to learn: Scientific Python

- Lots of material, primarily on numpy/scipy: http://scipy-lectures.github.io/

- Lectures with IPython notebooks: http://www.astro.washington.edu/users/vanderplas/Astr599/schedule

- Blog with many examples (searchable): http://glowingpython.blogspot.com/

How to find packages

- Google: python + the thing you want to do
- Search the Python Package Index, PyPI: http://pypi.python.org/

How to install packages

- With anaconda, try “conda install <package>” in a terminal window
- If that doesn’t work, try “pip install <package>”
- If that doesn’t work, read the package docs

Python 3 vs 2

- Python 3 is still *in the process of* replacing 2
- Major scientific packages support Python 3
- But many tutorials, websites, books, etc, still based on Python 2
- Anaconda installs Python 2 by default but can easily switch to Python 3
- Easiest to start learning with Python 2 and switch to 3 later when it has become more completely established