

Programming

Introduction

Daniel Dörr

Faculty of Technology, Bielefeld University

```
332
333
334     extrapolate = None:
335     x = np.asarray(x)
336     x_shape, x_ndim = x.shape, x.ndim
337     x = np.ascontiguousarray(x.ravel(), dtype=np
338
339     # With periodic extrapolation we map x to the
340     # [self.t[k], self.t[n]].
341     if extrapolate == 'periodic':
342         n = self.t.size - self.k - 1
343         x = self.t[self.k] + (x - self.t[self.k]) *
344         extrapolate = False
345
346     out = np.empty((len(x), prod(self.c.shape[1:])),
347     self._ensure_c_contiguous()
348     self._evaluate(x, nu, extrapolate, out)
349     out = out.reshape(x_shape + self.c.shape[1:])
350     if self.axis != 0:
351         # transpose to move the calculated values to 0
352         l = list(range(out.ndim))
353         l = l[x_ndim:x_ndim+self.axis] + l[:x_ndim] +
354         out = out.transpose(l)
355     return out
356
357 def _evaluate(self, xp, nu, extrapolate, out):
358     _bspl.evaluate_spline(self.t, self.c.reshape(self.c
359     self.k, xp, nu, extrapolate, out)
360
361 def _ensure_c_contiguous(self):
362     """
363     c and t may be modified by the user. The Cython code
364     c and t are C contiguous.
365     """
366     if not np.iscarray(self.c, 'C') or not np.iscarray(
367         self.t, 'C'):
```

Who am I?

❖ Dr. Daniel Dörr

email: `ddoerr@cebitec.uni-bielefeld.de`

office: UHG U10-132

❖ Post-doc in the research group “Genome Data Science” headed by Prof. Dr. Alexander Schönhuth

`https://gds.techfak.uni-bielefeld.de`

**Organizational
matters**

**What is Pro-
gramming?**

**Overview of
Python**

Python Basics

Organizational matters

- ❖ Course prerequisites: *none*

Organizational matters

- ❖ Course prerequisites: *none*
- ❖ Coursework
 - ❖ Weekly exercises
 - Submission in groups of 2-3
 - Upload to corresponding assignment in the “LernraumPlus”
 - Submission deadline is every **Friday 23:59**.
 - ❖ Written exam on **Wed. July 15, 2020 14:00-16:00 in T2-205**
 - Admitted: everyone exceeding 50% of total exercise points

Organizational matters

- ❖ Course prerequisites: *none*
- ❖ Coursework
 - ❖ Weekly exercises
 - Submission in groups of 2-3
 - Upload to corresponding assignment in the “LernraumPlus”
 - Submission deadline is every **Friday 23:59**.
 - ❖ Written exam on **Wed. July 15, 2020 14:00-16:00 in T2-205**
 - Admitted: everyone exceeding 50% of total exercise points
- ❖ Lecture part of module *39-Inf-Pro “Programming”*, study program *Data Science*

Course material

- ❖ ... available on course website:
<https://gds.techfak.de/teaching/2020summer/prog>
 - ❖ Slides and pointers to literature
 - ❖ Exercise sheets
- ❖ LernraumPlus:
 - ❖ e-Learning Videos
 - ❖ Exercise sheets
 - ❖ Pointers to literature
 - ❖ Forum
 - ❖ **Weekly submission of exercise solutions**

Tutorials

- ❖ Every **Monday, 12:15-13:45**
- ❖ ZOOM meeting: <https://uni-bielefeld.zoom.us/j/91342969268?pwd=bDh0SG5YMzZ1NWpPeWl0bm1QMDY2QT09>
- ❖ Discussion of exercise solutions
- ❖ You will present solutions to your classmates

Literature

- ❖ VanderPlas, Jake. (2016). *Python data science handbook*. Beijing; Boston; Farnham; Sebastopol; Tokyo: O'Reilly.
- ❖ Toomey, Dan. (2017). *Jupyter for data science*. Birmingham; Mumbai: Packt.
- ❖ Ana Bell, Eric Grimson, John Guttag (2016) *MIT 6.0001 Introduction to Computer Science and Programming in Python*: <http://ocw.mit.edu/6-0001F16>
- ❖ Eric Grimson, John Guttag, Ana Bell (2016) *MIT 6.0002 Introduction to Computational Thinking and Data Science*: <http://ocw.mit.edu/6-0002F16>

Course syllabus

Part 1

- Programming basics and terminology
- Introduction to Python

Part 2

- Scientific Programming
- Data Science with Python

**Organizational
matters**

**What is Pro-
gramming?**

**Overview of
Python**

Python Basics

What is a programming language?

- ❖ Natural vs. programming language
- ❖ Human-readable vs. machine-readable

Syntax and semantics

Syntax

Symbols, words, sentences

Semantics

Meaning behind symbols, words, and sentences

Syntax and semantics

Syntax

Symbols, words, sentences

e.g. English:

- Words: He, She, It, Program, ...
- Sentence grammar rule:
Subject + Verb + Object

She loves Python ✓

The house table the cup ✗

The table reads the cup ✓

Semantics

Meaning behind symbols, words, and sentences

Syntax and semantics

Syntax

Symbols, words, sentences

e.g. English:

- Words: He, She, It, Program, ...
- Sentence grammar rule:
Subject + Verb + Object

She loves Python ✓

The house table the cup ✗

The table reads the cup ✓

Semantics

Meaning behind symbols, words, and sentences

She loves Python ✓

The table reads the cup ✗

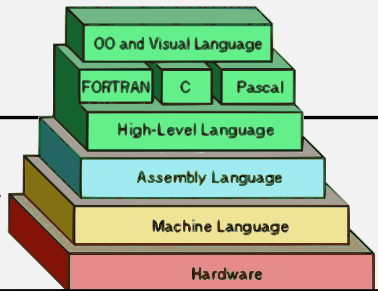
Programming languages ...

- are formal languages with unambiguous context-free grammars,

syntactic ambiguity:
“Tom hit the man with a stick.”
- offer different levels of abstraction,

High level language

- Easy to understand



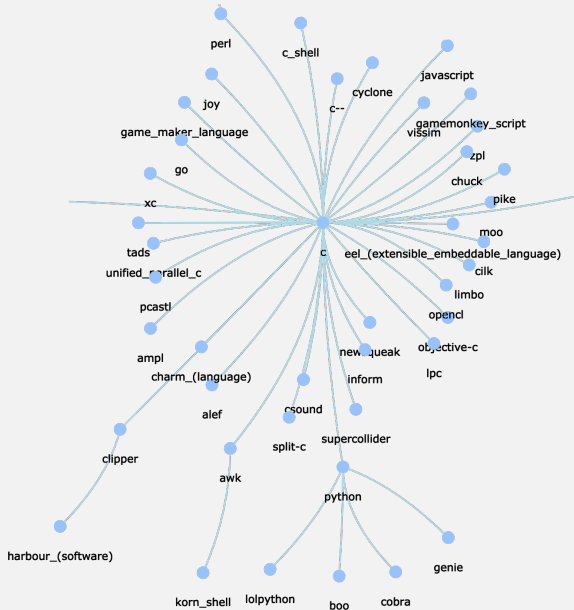
Low level language

- Machine code
(processor instructions
and binary code)

Source: <https://thebittheories.com>

Programming languages ...

- are formal languages with unambiguous context-free grammars,
- syntactic ambiguity:
“Tom hit the man with a stick.”
- offer different levels of abstraction,
- change over time,
- inspire new generations of languages.



Source: <http://svalver.github.io/Proglang/paradigms.html>

Programming paradigms

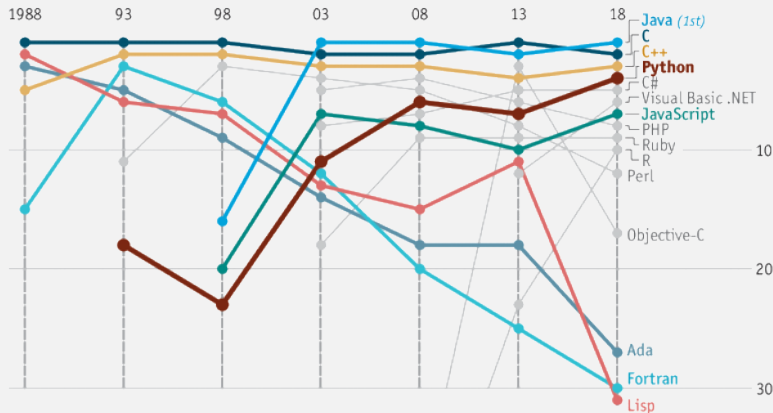
Many different general paradigms (notable excerpts):

- ❖ Imperative – *Do this, then do that!*
 - ❖ Procedural (C)
 - ❖ Object-oriented (C++, C#, Java)
- ❖ Declarative – *I want this, I want that!*
 - ❖ Logic (Prolog)
 - ❖ Functional (Haskell, Lisp)
- ❖ Mixed (Python, R)

There are also special-purpose languages (not necessarily considered “programming” languages), e.g. *LaTeX*, *HTML*, *XML*.

Language popularity

Ranking of programming languages*

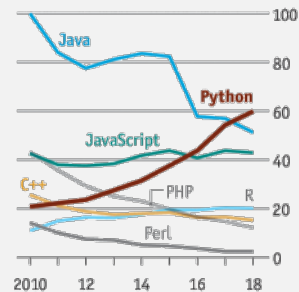


The Economist

*Ranked by global search-engine popularity

Source: Python is becoming the world's most popular coding language - The Economist (2018)

US, Google searches for coding languages
100 = highest annual traffic for any language



Source: TIOBE, Google Trends

Quiz

❖ *Syntactic or semantic ambiguity?*

- ❖ “Milk drinkers are turning to powder.”
- ❖ “Stolen painting found by tree.”
- ❖ “She went to her house, and so did Jane.”

❖ *True or false?*

- ❖ “All context-free grammars are unambiguous.”
- ❖ “Assembly language is a low level language.”
- ❖ “Functional programming is a form of imperative programming.”

Quiz

❖ *Syntactic or semantic ambiguity?*

- ❖ “Milk drinkers are turning to powder.” syntactic
- ❖ “Stolen painting found by tree.” syntactic
- ❖ “She went to her house, and so did Jane.” semantic

❖ *True or false?*

- ❖ “All context-free grammars are unambiguous.” false
- ❖ “Assembly language is a low level language.” true
- ❖ “Functional programming is a form of imperative programming.” false

**Organizational
matters**

**What is Pro-
gramming?**

**Overview of
Python**

Python Basics

Python



Originally developed by Guido van Rossum in the late 1980s.

- ❖ Open-source and actively maintained
- ❖ Applicable to a wide range of applications
- ❖ Extremely popular in the data science community

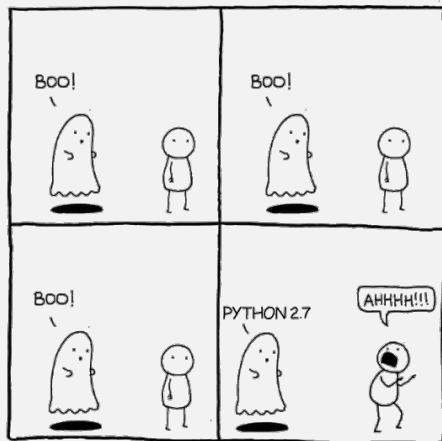
But: There are alternative programming languages.
Make sure to use the right one for the task.



Guido van Rossum, source:
<https://gvanrossum.github.io>, ©Michael Cavotta,
license: CC BY-NC-ND 4.0

Which Python version?

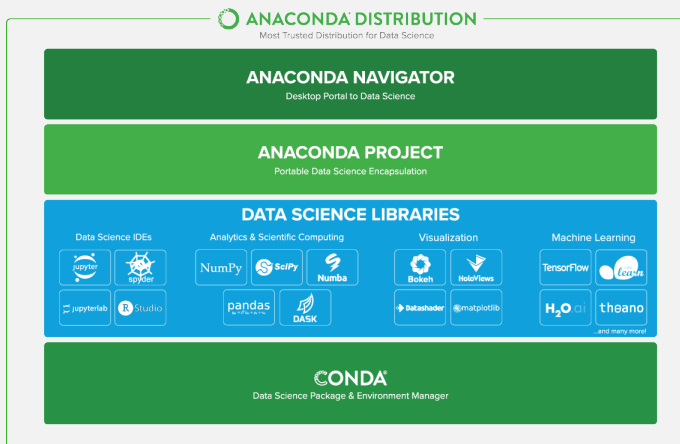
- ❖ Python 2: still common, although no longer maintained
- ❖ Python 3: modernized, *backwards-incompatible* version of the language



source: https://www.reddit.com/r/ProgrammerHumor/comments/91vtas/python_27/

Development environment: Anaconda

Python Data Science Distribution



Download Anaconda



Python 3.7 version

Download

64-Bit Graphical Installer 466 MB

32-Bit Graphical Installer 433 MB

<https://www.anaconda.com/distribution#download-section>

Anaconda navigator

The screenshot displays the Anaconda Navigator desktop application. The interface includes a sidebar on the left with navigation options: Home, Environments, Learning, and Community. The main area shows a grid of application cards for the 'base (root)' environment. Each card contains an icon, the application name, version number, a brief description, and a button to either 'Launch' or 'Install' the application. A 'Sign in to Anaconda Cloud' button is visible in the top right corner.

Applications on base (root)

Application	Version	Description	Action
JupyterLab	1.2.6	An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.	Launch
Jupyter Notebook	6.0.3	Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.	Launch
Qt Console	4.6.0	PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.	Launch
Spyder	4.0.1	Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features.	Launch
Glueviz	0.15.2	Multidimensional data visualization across files. Explore relationships within and among related datasets.	Install
Orange 3	3.23.1	Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.	Install

Quiz

❖ *True or false?*

- ❖ “Python has been developed for data science analysis.”
- ❖ “Python is the only language used in data science analysis.”
- ❖ “The university has bought Python licenses for this course.”

Quiz

❖ *True or false?*

- ❖ “Python has been developed for data science analysis.” false
- ❖ “Python is the only language used in data science analysis.” false
- ❖ “The university has bought Python licenses for this course.” false

**Organizational
matters**

**What is Pro-
gramming?**

**Overview of
Python**

Python Basics

Launching Python

The screenshot displays the Anaconda Navigator desktop application. The interface is divided into a left sidebar and a main content area. The sidebar contains navigation options: Home, Environments, Learning, and Community, along with links to Documentation and Developer Blog, and social media icons for Twitter, YouTube, and GitHub. The main content area shows a grid of application cards under the heading 'Applications on base (root)'. The cards are: JupyterLab (1.2.6), Jupyter Notebook (6.0.3), Qt Console (4.6.0), Spyder (4.0.1), Glueviz (0.15.2), and Orange 3 (3.23.1). Each card includes a description and a button to either 'Launch' or 'Install' the application. The Qt Console card is highlighted in red.

ANACONDA NAVIGATOR

Sign in to Anaconda Cloud

Applications on base (root) Channels Refresh

lab
JupyterLab
1.2.6
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
Launch

jupyter
Notebook
6.0.3
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Launch

IP[ty]
Qt Console
4.6.0
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Launch

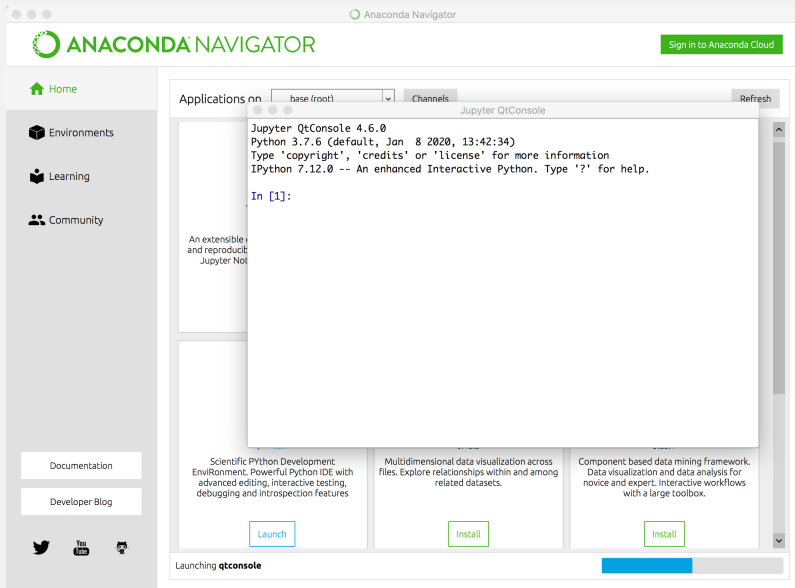
Spyder
4.0.1
Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features.
Launch

Glueviz
0.15.2
Multidimensional data visualization across files. Explore relationships within and among related datasets.
Install

Orange 3
3.23.1
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.
Install

Documentation
Developer Blog
Twitter YouTube GitHub

Launching Python



The screenshot displays the Anaconda Navigator application window. The title bar reads "Anaconda Navigator". The main header features the "ANACONDA NAVIGATOR" logo and a "Sign in to Anaconda Cloud" button. A left-hand sidebar contains navigation options: Home, Environments, Learning, and Community. Below these are buttons for "Documentation" and "Developer Blog", and social media icons for Twitter, YouTube, and GitHub. The main content area is titled "Applications on base (root)" and includes a "Channels" dropdown and a "Refresh" button. A "Jupyter QtConsole" window is open, showing the following text:

```
Jupyter QtConsole 4.6.0
Python 3.7.6 (default, Jan 8 2020, 13:42:34)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.12.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]:
```

Below the console, there are three cards for different Python environments:

- Scientific Python Development Environment:** Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features. Includes a "Launch" button.
- Multidimensional data visualization across files:** Explore relationships within and among related datasets. Includes an "Install" button.
- Component based data mining framework:** Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox. Includes an "Install" button.

At the bottom of the interface, a status bar shows "Launching qtconsole" and a blue progress bar.

Launching Python

The screenshot displays the Anaconda Navigator application window. The title bar reads "Anaconda Navigator". The main header features the "ANACONDA NAVIGATOR" logo and a "Sign in to Anaconda Cloud" button. A left sidebar contains navigation options: Home, Environments, Learning, and Community. Below these are buttons for "Documentation" and "Developer Blog", and social media icons for Twitter, YouTube, and GitHub. The main content area is titled "Applications on" and shows a dropdown menu set to "base (root)" and a "Channels" tab. A "Jupyter QtConsole" window is open, displaying the following text:

```
Jupyter QtConsole 4.6.0
Python 3.7.6 (default, Jan 8 2020, 13:42:34)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.12.0 -- An enhanced Interactive Python. Type '?' for help.

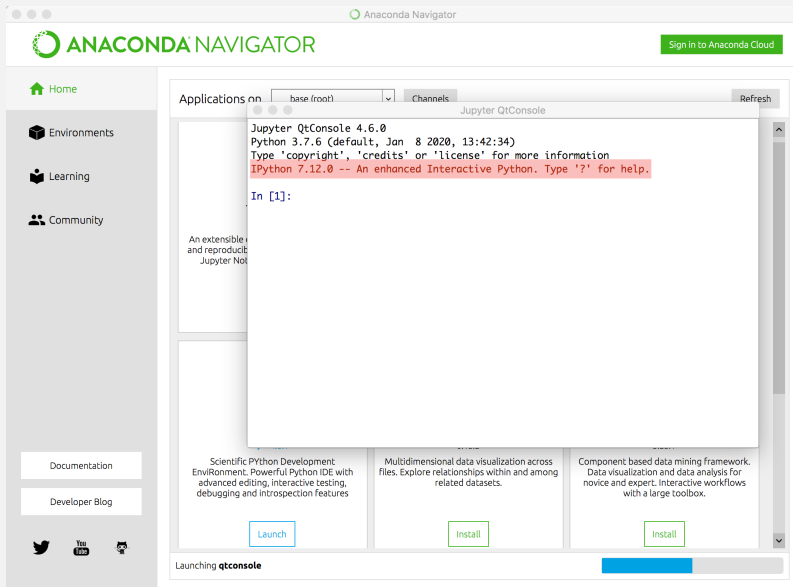
In [1]:
```

Below the console, there are three cards with descriptions and buttons:

- Scientific Python Development Environment:** Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features. [Launch](#)
- Multidimensional data visualization across files:** Explore relationships within and among related datasets. [Install](#)
- Component based data mining framework:** Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox. [Install](#)

At the bottom of the main area, it says "Launching qtconsole" with a blue progress bar.

Launching Python



The screenshot displays the Anaconda Navigator application window. The title bar reads "Anaconda Navigator". The main header features the "ANACONDA NAVIGATOR" logo and a "Sign in to Anaconda Cloud" button. A left-hand sidebar contains navigation options: Home, Environments, Learning, and Community. Below these are buttons for "Documentation" and "Developer Blog", and social media icons for Twitter, YouTube, and GitHub. The main content area is titled "Applications on base (root)" and shows a "Channels" tab. A "Jupyter QtConsole" window is open, displaying the following text:

```
Jupyter QtConsole 4.6.0
Python 3.7.6 (default, Jan 8 2020, 13:42:34)
Type 'copyright', 'credits' or 'license' for more information
Python 7.12.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]:
```

Below the console, there are three cards for different Python environments:

- Scientific Python Development Environment:** Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features. Includes a "Launch" button.
- Multidimensional data visualization across files:** Explore relationships within and among related datasets. Includes an "Install" button.
- Component based data mining framework:** Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox. Includes an "Install" button.

At the bottom of the interface, a status bar indicates "Launching qtconsole" with a progress indicator.

Arithmetic in Python

Numeric types:

- Integer: `int` 42
- Real valued numbers: `float` 42.0
- Complex numbers: `complex` 42+0j

Operators

- Addition and subtraction + -
- Multiplication and division * / // %
- Exponentiation **

Variables

Variable assignment

❏ `a = 42`

❏ `b = a - 6.0`

`type(«name of the variable»)`: returns type of variable

Quiz

❖ What are types numeric types of the following calculations?

- ❖ `type(42 / 3)`
- ❖ `type(42 // 3)`
- ❖ `type(3.14 + 2.71+8j)`
- ❖ `type(42 // 3.14)`
- ❖ `a = 1`
`a - 10 * 1.0`
`type(a)`

Quiz

❖ What are types numeric types of the following calculations?

- ❖ `type(42 / 3)` float
- ❖ `type(42 // 3)` int
- ❖ `type(3.14 + 2.71+8j)` complex
- ❖ `type(42 // 3.14)` float
- ❖ `a = 1`
`a - 10 * 1.0`
`type(a)` int

Recap

Summary

- ❖ Course logistics
- ❖ Introduction to Programming
- ❖ First steps in Python

What comes next?

- ❖ Go to the course website
(<https://gds.techfak.de/teaching/2020summer/prog>) or
Lernraum+ (<https://lernraumplus.uni-bielefeld.de/course/view.php?id=4688>) and download this week's exercise sheet
- ❖ Group yourselves into pairs or (less recommended) triples
- ❖ Due date for this week's exercises is **Friday, April 24, 2020.**

Next lecture: Programming & Python basics continued ...