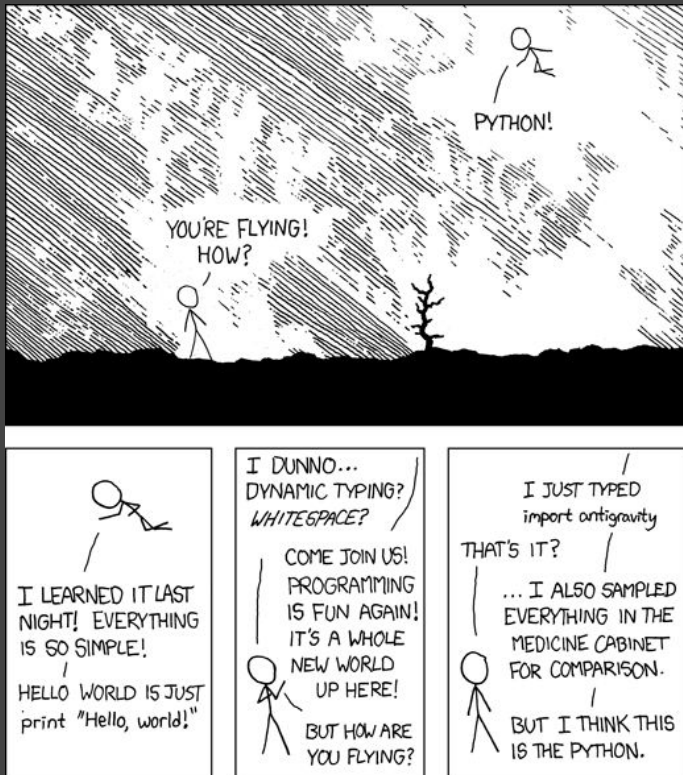

THE POWER OF PYTHON

IN SCIENCE AND EDUCATION



Simon Funke (simon@simula.no)

About me



- Researcher @ Simula Research Laboratory:
 - Numerical solution of PDEs
 - PDE-constrained optimisation
 - Applications
- Teaching intermediate-level Python-course @ UiO
- Scientific software



FENICS
PROJECT



dolfin-adjoint

OpenTidalFarm



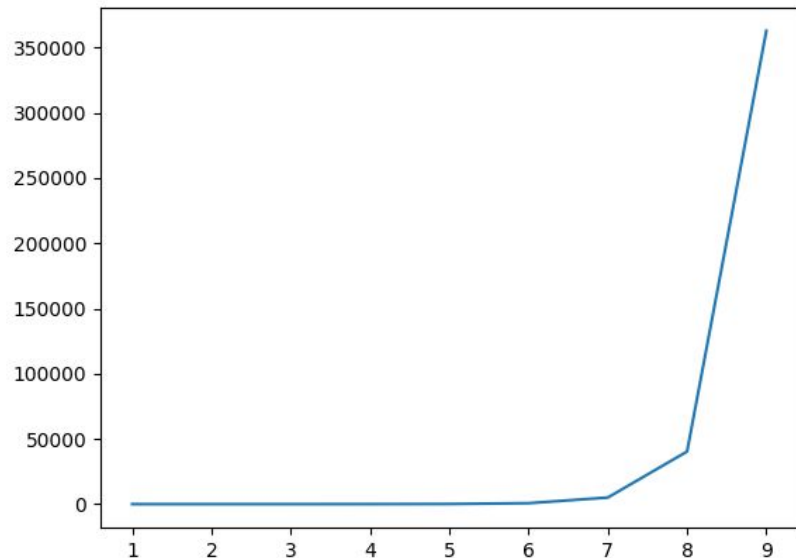
Why should we teach Python?

Programming Language	2017	2012	2007	2002	1997	1992	1987
Java	1	1	1	1	15	-	-
C	2	2	2	2	1	1	1
C++	3	3	3	3	2	2	5
C#	4	4	7	14	-	-	-
Python	5	7	6	11	27	-	-

TIOBE Index - very Long Term History

Python is (relatively) easy to learn

```
1 from matplotlib import pyplot
2
3 def factorial(n):
4     if n == 1:
5         return 1
6     else:
7         return n * factor
8
9 # Compute and plot the fa
10 x = range(1, 10)
11 y = [factorial(val) for v
12
13 pyplot.plot(x, y)
14 pyplot.show()
```

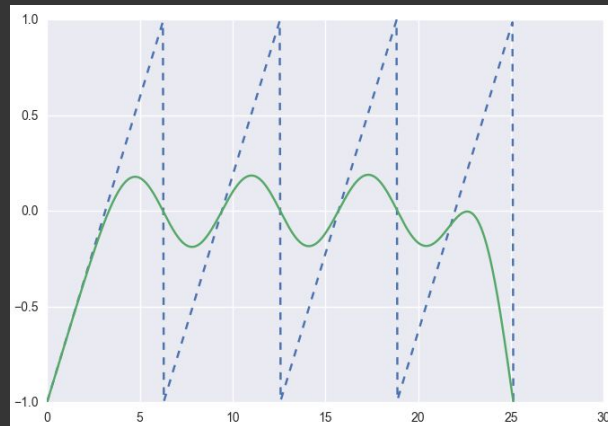
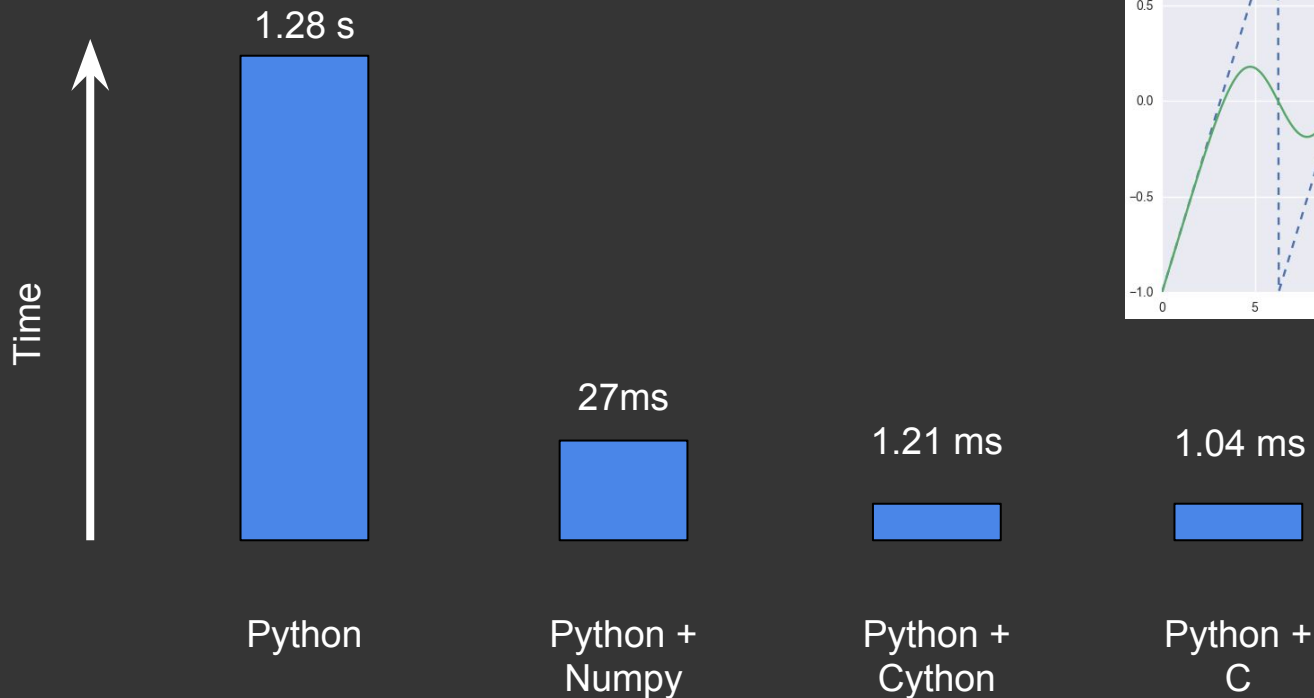


Python has extensive set of scientific libraries

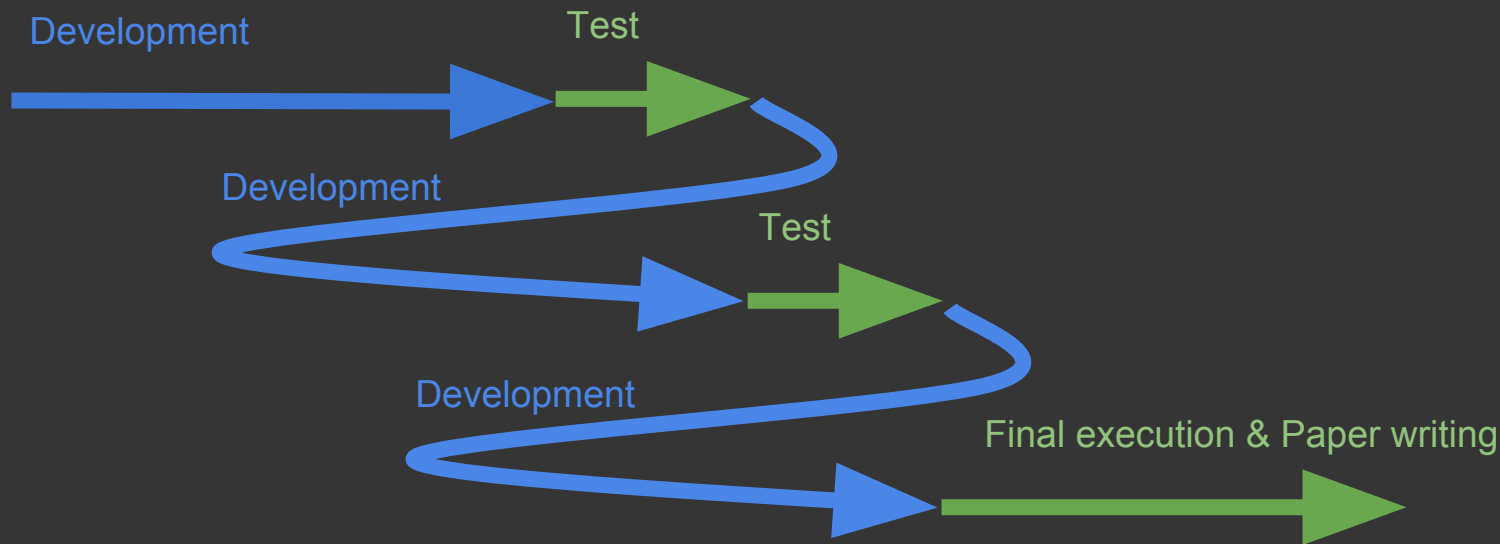
- Python
 - PyPy
- 
- Numpy
 - Cython
 - Numba
 - py.test
- 
- Matplotlib
 - Pandas
 - Scipy
 - xarray
 - IPython
 - Jupyter Notebook
 - Pytables
 - yt
- 
- FEniCS
 - Scikit-learn
 - PyMC3
 - SymPy
 - networkX
 - Sunpy
 - ...

Python can be accelerated with mixed programming

Performance study of a 1D diffusion process



Python accelerates development by avoiding common “low-level” issues



Python accelerates development by avoiding common “low-level” issues

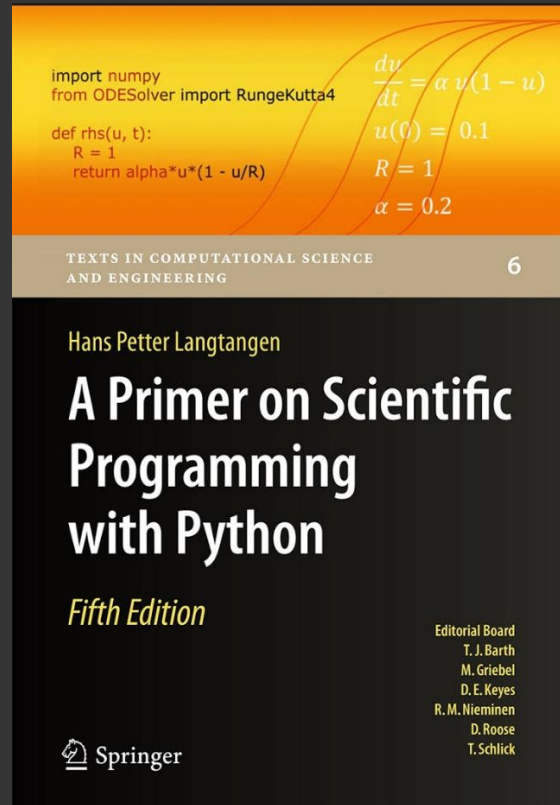


Teaching Python

My experience

- “Higher-level programming” bachelor/master course at University of Oslo
- Initiated by Hans-Petter Langtangen
- Students from CS/Physics/Maths/Chemistry

“Learning by doing and reviewing”

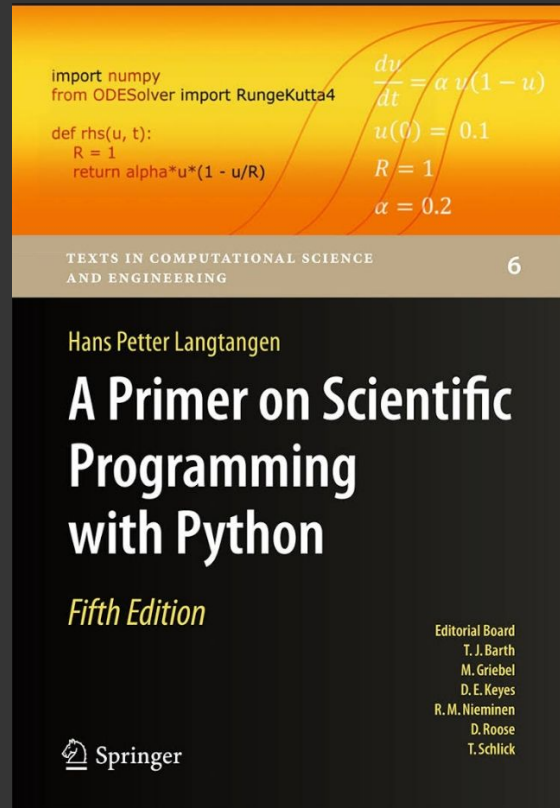


- The goal of the course is to teach tools that make computational science more effective

Syllabus

- The Python language (3 week crash-course)
- Numerical Python with numpy/matplotlib
- Accelerating Python with mixed programming
- Pandas, flask, regular expressions
- **Best practices:** Git, documentation, testing

Coding assignments are the core of the course

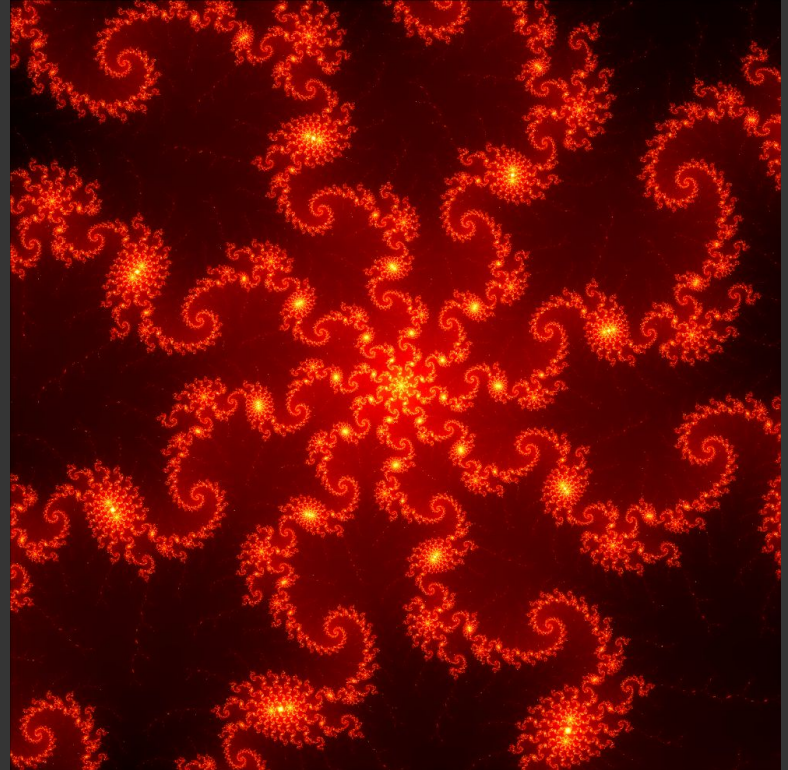


Assignments example: Fractal generator

Task:

- Implement a fractal generator
- Accelerate and compare timings between different versions:
 - Python
 - Python + Numpy
 - Cython
 - Pure C

Contest: best picture award



More assignments examples

File syntax highlighter

Core techniques taught:

- Regular Expressions

```
from my_unit_testing import UnitTest

def better_addition(a, b, num_rechecks=2):
    """Returns sum of a, b, but double checks answer several times."""
    sum_computations = [a + b for n in range(num_rechecks)]

    for n in range(num_rechecks):
        if sum_computations[n] != sum_computations[n-1]:
            print("Hang on, let me recheck that")
            return better_addition(a, b, num_rechecks)

    return sum_computations[0] # if all computations match, return whichever
```

Jupyter Notebook Clone

Core techniques taught:

- Graphical user interface (web)
- Advanced Python

range allows us to iterate over a sequence of numbers:

```
In [19]: for i in range(3):
          print(i)
```

```
0
1
2
```

Temperature/CO2 level analysis and predictions

Core techniques taught:

- Data analysis with Pandas

Good coding practices are enforced through test assignments and points on documentation

4.6 Testing the code (5 pt)

Write appropriate tests for 4.1 through 4.5. Depending on the problem choose either Pydoc-test or Py.test. At least one of each test must be included.

4.1 Download the page for reference:

`http://www.islostarepeat.com/`

Check if your program creates the same document.

4.2 Check if Hannestad creates the link



Students deliver assignments digitally



virtual-classroom

Peer-review of assignments

“Learning by doing and reviewing”
> 300 students signed up for the course last year

Lectures are available
as Jupyter notebooks
and on Youtube



Guest lecturers



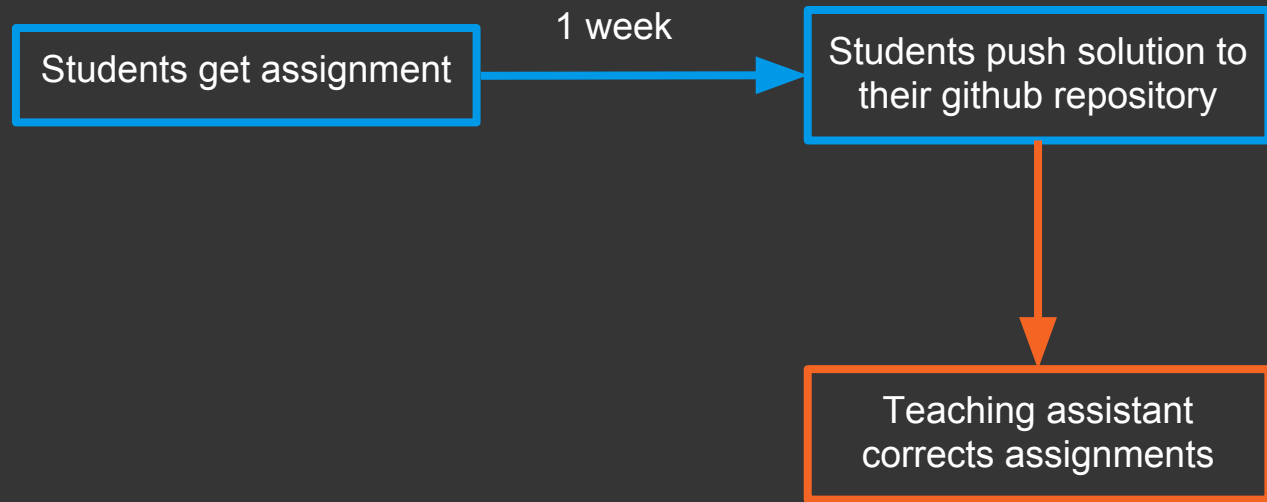
Students help each other



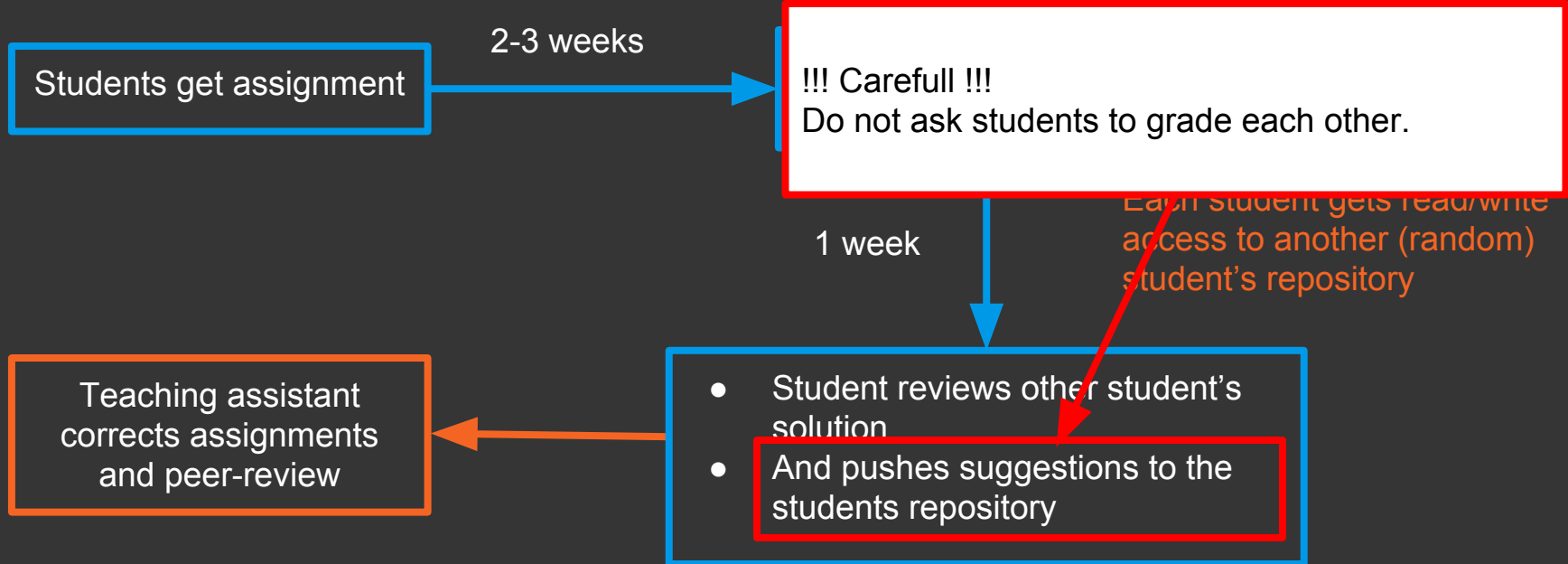
— The coding assignments form the core of the course

- Two type of assignments:
 - Short (1 week)
 - Long (2-3 weeks), **peer-reviewed**
- Each student has their own private github repository
- Points are given for correctness and quality of **code and documentation**

Non peer-reviewed assignment



Peer-reviewed assignment



The peer-review is delivered as a Latex report or as a git pull-request

Peer-review of assignment 5 for *INF3331-harad*

November 12, 2016

1 Introduction

1.1 Goal

The review should provide feedback on the solution to the student. The main goal is to *give constructive feedback and advice* on how to improve the solution. You, the peer-review team, can decide how you organise the peer-review work between you.

1.2 Guidelines

For each (coding) exercise, one should review the following points:

- Is the code **working as expected**? For non-internal functions (in particular for scripts that are run from the command-line), does the program handle invalid inputs sensibly?
- Is the code **well documented**? Are there docstrings and are they useful?
- Is the code written in **Pythonic way**¹? Is the code easy to read? Are the variable/class/function names sensible? Do you find overuse of classes or not sufficient use of functions or classes? Are there parts of the program that are hard to understand?

UIO-INF3331 / INF3331-harad Private

Unwatch 5 Star 0 Fork 2

Code Issues Pull requests 0 Projects 0 Wiki Settings Insights

Small changes here and there #1

Merged hardah merged 3 commits into uio-inf3331:master from sigmunsl:master on Dec 23, 2016

Conversation 0 Commits 3 Files changed 5 +129 -13

sigmunsl commented on Dec 7, 2016

The program works more or less as it is supposed to, but I added some things here and there which I thought needed to be done. It is summarized in the commits

sigmunsl added some commits on Dec 7, 2016

- Pydoc.html created and will now be visualized when pressing documenta... d3c4b3a
- Added support for titles. All plots will now be visualized with title 8be1cc4
- Now possible to change month from flask server 1e2ba8c

hardah merged commit 3e64354 into uio-inf3331:master on Dec 23, 2016

Revert

Reviewers: No reviews—request one

Assignees: No one—assign yourself

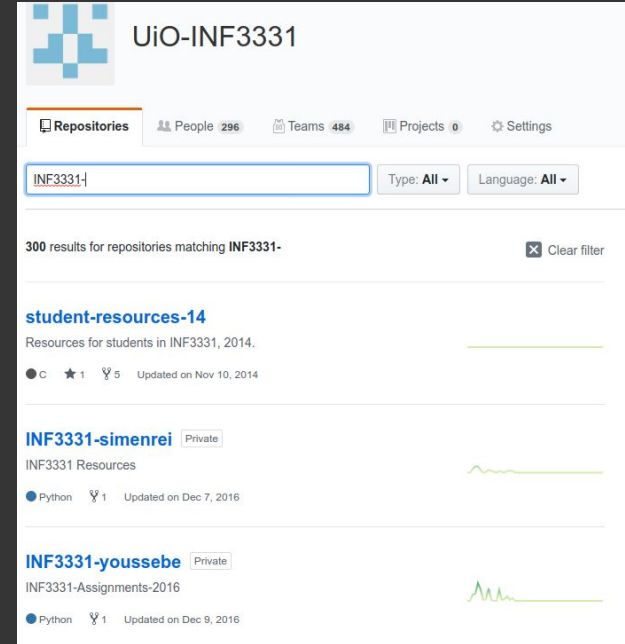
Labels: None yet

Projects: None yet

Milestone: No milestone

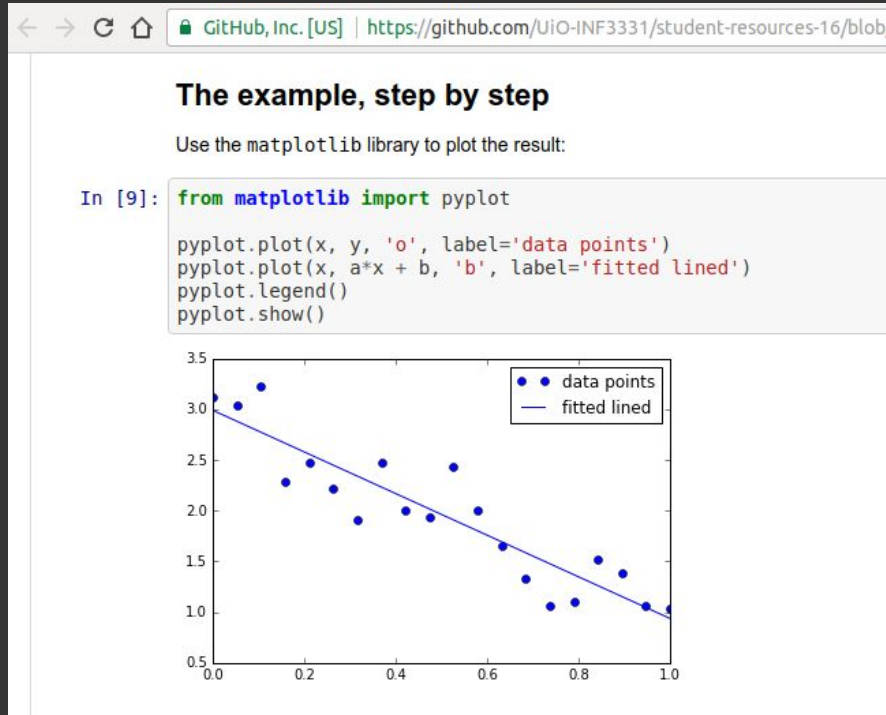
virtual-classroom manages the peer-review process

- Creates private repositories for every student in a course.
- Divides students into groups for exercise assessment.
- Opens up private repositories temporarily for assessment groups.



Scripts are available at <https://github.com/hplgit/virtual-classroom>

Jupyter notebook allows students to experiment with the slides



- Students can download slides, run and change them
- Lecturer can test that slides still work

Tipp: Use the RISE plugin to convert a Jupyter notebook to a slideshow

Guest lectures bring domain-experts into the classroom



Benjamin Ragan-Kelley

Core developer of Jupyter Notebook

“Introduction to Jupyter Notebooks”



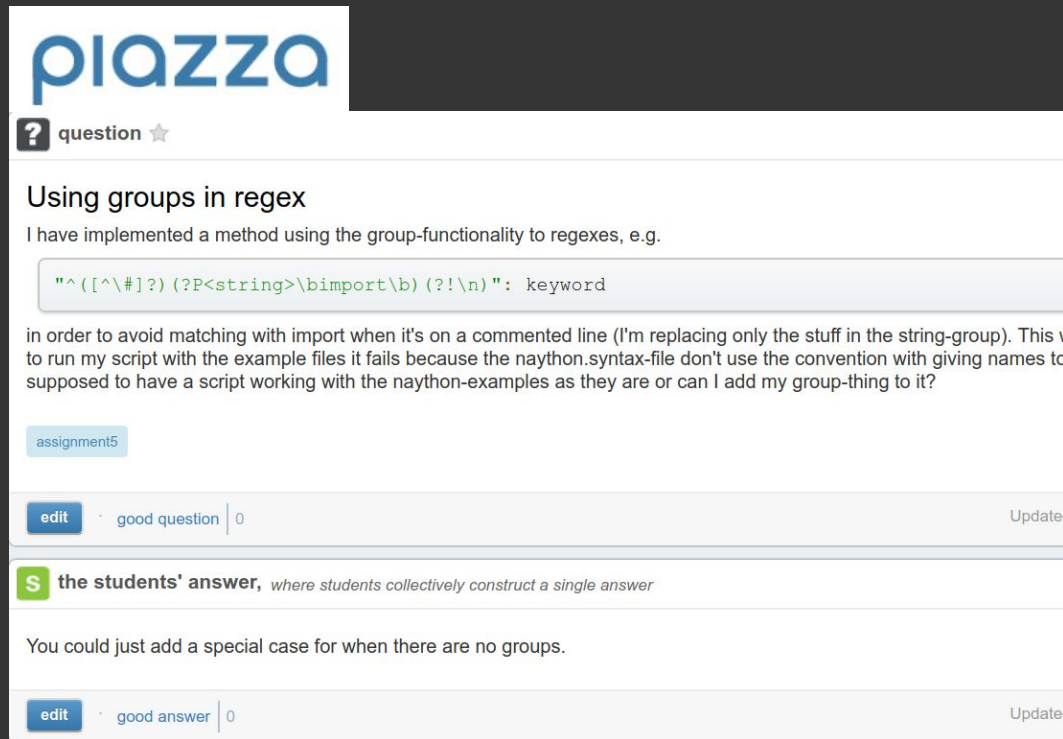
Jonathan Feinberg

Data scientist at Expert Analytics

“Introduction to Regular Expression”

Piazza is a Q&A for classes that allows students to help each other

- Typically high-quality answers and fast response time
- Reduced load of teaching assistants
- Difficult topics become visible to teaching staff
- Allows students to suggest improvements/raise problems quickly



The screenshot shows a Piazza interface. At the top is the 'piazza' logo. Below it is a 'question' section with a title 'Using groups in regex' and a star icon. The question text reads: 'I have implemented a method using the group-functionality to regexes, e.g.' followed by a code block containing a regex: `"^([\^#]?) (?P<string>\bimport\b) (?!\n)": keyword`. The question continues: 'in order to avoid matching with import when it's on a commented line (I'm replacing only the stuff in the string-group). This v to run my script with the example files it fails because the naython.syntax-file don't use the convention with giving names to supposed to have a script working with the naython-examples as they are or can I add my group-thing to it?'. Below the question is a tag 'assignment5'. At the bottom of the question section are buttons for 'edit', 'good question' (with a count of 0), and 'Update'. Below the question section is an 'answer' section with a green 'S' icon and the text 'the students' answer, where students collectively construct a single answer'. The answer text reads: 'You could just add a special case for when there are no groups.' At the bottom of the answer section are buttons for 'edit', 'good answer' (with a count of 0), and 'Update'.

piazza

? question ★

Using groups in regex

I have implemented a method using the group-functionality to regexes, e.g.

```
"^([\^#]?) (?P<string>\bimport\b) (?!\n)": keyword
```

in order to avoid matching with import when it's on a commented line (I'm replacing only the stuff in the string-group). This v to run my script with the example files it fails because the naython.syntax-file don't use the convention with giving names to supposed to have a script working with the naython-examples as they are or can I add my group-thing to it?

assignment5

edit · good question | 0 Update

S the students' answer, where students collectively construct a single answer

You could just add a special case for when there are no groups.

edit · good answer | 0 Update

The Anaconda Python distribution is a good choice installation

- Works on Windows, Mac, Linux
- Can be installed locally or globally
- Provides a consistent Python installation across all students



Anaconda: Surprisingly few problems - except Cython on Windows machines

Summary

- Python is easy-to-learn, flexible and can be made performant
- Teaching Python tips:
 - Code peer-reviews allow to learn from others
 - Best practices can be taught “on the side”
 - Tools like github, virtual-classroom and piazza allow to scale the course

“Learning by doing and reviewing”

Course material (lecture slides, videos, assignments) are available on goo.gl/SEfw1u
Email me (simon@simula.no) if you have any questions.



Python is (relatively) easy to learn

Crucial for student projects with students without a computer science background.

Less time coding/fixing code means more time doing research



Python for high-performance computing

Chris N. Richardson, Garth N. Wells,
Parallel scaling of DOLFIN on ARCHER, 2015

