Viviane Pons

Maîtresse de conférences, Univ. Paris-Sud Orsay
viviane.pons@lri.fr – @PyViv

Teaching with Jupyter

at Université Paris-Sud

Context: who am I? What do I teach?

- Associate professor (Maîtresse de conférences) at University Paris-Sud since 2014
- My research is in Combinatorics
- I teach Computer Science at both undergraduate and graduate levels (from "L1" to "M2")

Example: last year

- Introduction to programming (C++, year 1)
- Interdisciplinary project Math and Computer science (SageMath, year 1)
- Algorithmic (Python, Engineering school, year 3)
- Advanced Algorithmic (Pyhton, year 4)
- Recursive generation of combinatorial objects (Python, year 5)

Example: last year

- Introduction to programming (C++, year 1)
- Interdisciplinary project Math and Computer science (SageMath, year 1)
- Algorithmic (Python, Engineering school, year 3)
- Advanced Algorithmic (Pyhton, year 4)
- Recursive generation of combinatorial objects (Python, year 5)

I used Jupyter for all of them!

Example 1: teaching "Interdisciplinary project" to first year students

- Started in 2014
- 1st year student in "Math, Physics, and Computer Science"
- Optional class in the second semester
- Around 30 students in 2014

Context: me

What I had

- About 6 months experience at University Paris-Sud
- Not much knowledge about the University computer rooms (how to install a software? Who to ask *this*? How to do *that*?)
- Total freedom

Context: me

What I had

- About 6 months experience at University Paris-Sud
- Not much knowledge about the University computer rooms (how to install a software? Who to ask *this*? How to do *that*?)
- Total freedom

What I didn't have

time...

So what did I do?

I decided to use **SageMath**.

So what did I do?

I decided to use **SageMath**.

What is SageMath?

SageMath is a free open-source mathematics software system licensed under the GPL. It builds on top of many existing open-source packages: NumPy, SciPy, matplotlib, Sympy, Maxima, GAP, FLINT, R and many more. Access their combined power through a common, Python-based language or directly via interfaces or wrappers.

How to use Sage with students?

How to use Sage with students?

Here came **Cocalc** (used to be called SageMathCloud). It is an online open-source platform that allows users to create shared projects (basically virtual linux machines) with a bunch of pre installed scientific software, including Sage.

demo

How to use Sage with students?

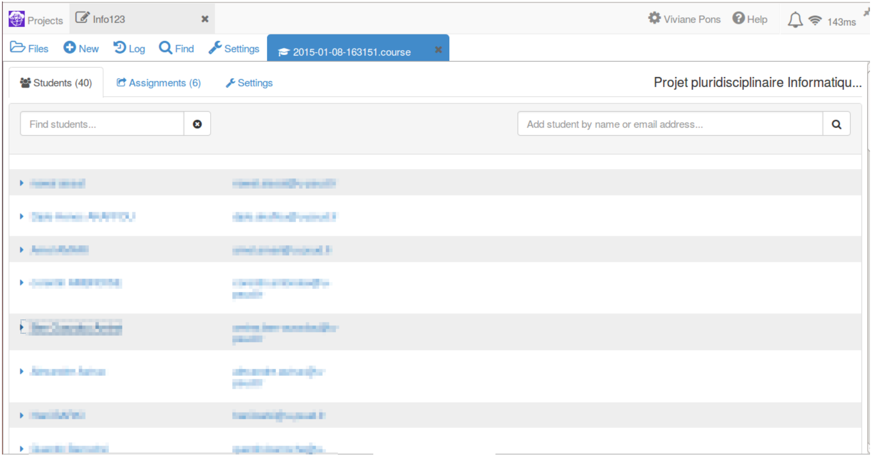
Here came **Cocalc** (used to be called SageMathCloud). It is an online open-source platform that allows users to create shared projects (basically virtual linux machines) with a bunch of pre installed scientific software, including Sage.

demo

And...

It has a simple (but neat) course interface.

Interdisciplinary project



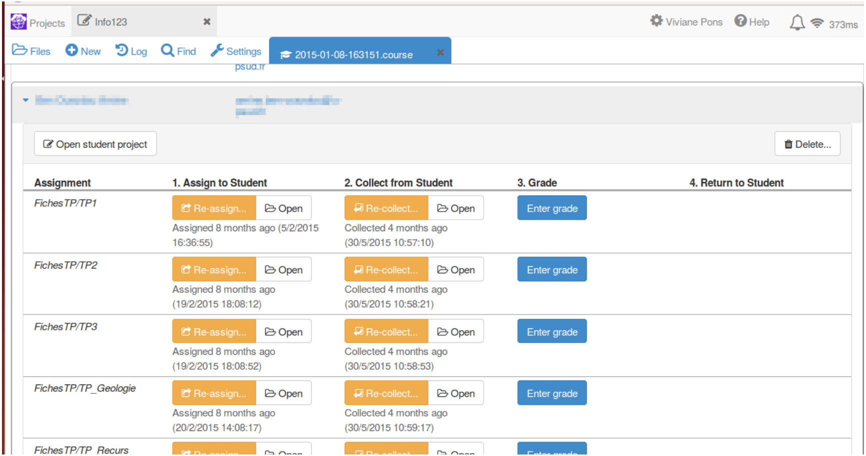
Projects | Info123 ✖ | Viviane Pons | Help | 143ms

Files | New | Log | Find | Settings | 2015-01-08-163151.course ✖

Students (40) | Assignments (6) | Settings — Projet pluridisciplinaire Informatiqu...

Find students... | Add student by name or email address...

[redacted]



psud.fr

[redacted] [redacted]

Assignment	1. Assign to Student	2. Collect from Student	3. Grade	4. Return to Student
FichesTP/TP1	Re-assign... Open Assigned 8 months ago (5/2/2015 16:36:55)	Re-collect... Open Collected 4 months ago (30/5/2015 10:57:10)	Enter grade	
FichesTP/TP2	Re-assign... Open Assigned 8 months ago (19/2/2015 18:08:12)	Re-collect... Open Collected 4 months ago (30/5/2015 10:58:21)	Enter grade	
FichesTP/TP3	Re-assign... Open Assigned 8 months ago (19/2/2015 18:08:52)	Re-collect... Open Collected 4 months ago (30/5/2015 10:58:53)	Enter grade	
FichesTP/TP_Geologie	Re-assign... Open Assigned 8 months ago (20/2/2015 14:08:17)	Re-collect... Open Collected 4 months ago (30/5/2015 10:59:17)	Enter grade	
FichesTP/TP_Recurs	Re-assign... Open	Re-collect... Open	Enter grade	

How did the class work?

Remember, first year students:

- don't know Sage
- don't know much about python
- don't know much about anything, actually
- often not very autonomous
- often lack motivation

I decided...

- to ask difficult things
- with little explanations (still, some)
- and lots of freedom

I decided...

- to ask difficult things
- with little explanations (still, some)
- and lots of freedom

But...

- Using Cocalc and Jupyter breaks the technical barriers
- Using interactive worksheets help to guide students through exercises while pushing them to experiment
- I gave them fun projects (at I least, I find them fun)

In practice

First half of the semester

Students work on Jupyter worksheets to familiarize themselves with sage and Python. The worksheets include programming exercises and small math problems to solve with programming.

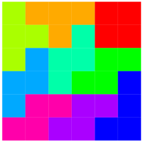
Second half of the semester

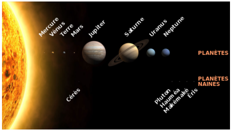
Students work in groups of 2 or 3 on difficult mathematical research projects that require computer exploration. The teachers are here to guide and help them.

Final evaluation

A 10 minutes group presentation on their research problem and findings.

Projects




Mercure
Vénus
Terre
Mars
Jupiter
Saturne
Uranus
Neptune
PLANÈTES
PLANÈTES NAINES
Cérès
Pluton
Hauméa
Makémaké
Éris

UEFA
CHAMPIONS
LEAGUE

dreamstime

Context

Between 30 and 40 students, year 1, using Sage

Technical solution

Using Jupyter worksheets on CoCalc with SageMath kernel and the CoCalc course interface

Technical problems

Very few

Cost in time

Not much: all time spent on pedagogy, not technique.

Cost in Money

2014 – 2016: Used the free version (with its limitations)
Since 2017: paying cocalc subscriptions (Course packages, from 500 dollars a year to 3500)

Example 2: teaching Algorithmic to engineers

- 20 students, year 3 level
- Computer sessions come in complement to lectures and exercise sessions
- Started the class in 2014, stared to use Jupyter in 2015.

Why use Jupyter?

- It allows me to ask simple algorithmic questions without worrying about a complete program
- It is easier to evaluate
- The students concentrate more on the algorithmic than on the technicalities.

show worksheet

Technichal solution

I also used Cocalc, for the simplicity for its course management

Technichal solution

I also used Cocalc, for the simplicity for its course management

Future improvements?

I should use **nbgrader** to make my life even easier but I haven't had the time to look into it so far.

Example 3: teaching C++ to first year students

- Around 400 students, year 1 level

Example 3: teaching C++ to first year students

- Around 400 students, year 1 level
- Divided into 13 groups

Example 3: teaching C++ to first year students

- Around 400 students, year 1 level
- Divided into 13 groups
- Teaching team: 1 professor + 2 associate professor (inc. me) + around 10 TA

Example 3: teaching C++ to first year students

- Around 400 students, year 1 level
- Divided into 13 groups
- Teaching team: 1 professor + 2 associate professor (inc. me) + around 10 TA
- In C++

Example 3: teaching C++ to first year students

- Around 400 students, year 1 level
- Divided into 13 groups
- Teaching team: 1 professor + 2 associate professor (inc. me) + around 10 TA
- In C++
- Teaching it since 2014, using Jupyter since 2016

Why use Jupyter?

- so that students can see the results of their computation right away
- to postpone learning about compilation
- to break the initial technical barriers
- to push the students to test more and experiment

Technical challenges

- Which kernel?
- How to install it? (On our machines, on the student machines)
- How can the student work outside of the university?

Solutions...

We use the kernel **xseus-cling** developed by Sylvain Corlay based on the cling C++ interpreter.

Install on teacher machines

globally ok (through Conda)

Install on university computer rooms

not so simple...

Solution

We made a local install on a public university account and created a custom command line for students to start the program from there.

Work from home

We also have a JupyterHub installed on a university server.

The challenges

- We use many different (new) technologies
- When you put 400 students on a system, it makes bugs appear!
- Our server would break sometime
- No good synchronization between server and local machines
- No convenient way for teachers to access / grade student work

We have to deal with the bugs on a very tight schedule

Still, it worked!

show notebook

In practice

We work on Jupyter only for around 4-5 weeks, then we have a gentle transition to compiled C++.

From a pedagogical point of view

The +

- More interactivity
- We can work directly on programming issues (variables, loops, conditions...) before compilation
- an online environment with everything installed (students can work from home right away)

The -

- A big technology stack that can be confusing for students
- they tend to call the system "buggy"

Going further...

- Getting more stable
- Using nbgrader
- Using Jupyterlab

The 2 solutions

Home made solution	Cocalc
Cost lots of (your) time	Cost almost no time
Cost university resources (people, servers...)	Cost money
More freedom on your environment	Rely on professional for bug fixing

Thank you!

All teaching material is available online with an open license.

https://www.lri.fr/~pons/

http://nicolas.thiery.name/Enseignement/Info111/