Calcul et programmation avec SageMath/Python et meilleures pratiques

Cours de l'EDMI
25 février, 4, 11, 18 mars 2021

Sébastien Labbé
CNRS, LaBRI, Université de Bordeaux

Titre du cours

Calcul et programmation avec SageMath/Python et meilleures pratiques

Le titre se décline comme ceci :

- **Calcul et programmation** avec Python
  Structures de données de base, les bases de la programmation, les différentes interfaces
- Apprentissage des **meilleures pratiques**
  Wilson et al., Best Practices for Scientific Computing,
  PLoS Biology 12 (2014) 1, `https://doi.org/10.1371/journal.pbio.1001745`
- Basé sur une expérience acquise via le développement de **SageMath**
  logiciel de mathématiques développé depuis 2006

Tout ça en 4 cours de 3h.

Best Practices for Scientific Computing

Wilson et al., Best Practices for Scientific Computing,
PLoS Biology 12 (2014) 1, `https://doi.org/10.1371/journal.pbio.1001745`

Box 1. Summary of Best Practices

1. Write programs for people, not computers.
   a. A program should not require its readers to hold more than a handful of facts in memory at once.
   b. Make names consistent, distinctive, and meaningful.
   c. Make code style and formatting consistent.
2. Let the computer do the work.
   a. Make the computer repeat tasks.
   b. Save recent commands in a file for re-use.
   c. Use a build tool to automate workflows.
3. Make incremental changes.
   a. Work in small steps with frequent feedback and course correction.
   b. Use a version control system.
   c. Put everything that has been created manually in version control.
4. Don't repeat yourself (or others).
   a. Every piece of data must have a single authoritative representation in the system.
   b. Modularize code rather than copying and pasting.
   c. Re-use code instead of rewriting it.
5. Plan for mistakes.
   a. Add assertions to programs to check their operation.
   b. Use an off-the-shelf unit testing library.
   c. Turn bugs into test cases.
   d. Use a symbolic debugger.
6. Optimize software only after it works correctly.
   a. Use a profiler to identify bottlenecks.
   b. Write code in the highest-level language possible.
7. Document design and purpose, not mechanics.
   a. Document interfaces and reasons, not implementations.
   b. Refactor code in preference to explaining how it works.
   c. Embed the documentation for a piece of software in that software.
8. Collaborate.
   a. Use pre-merge code reviews.
   b. Use pair programming when bringing someone new up to speed and when tackling particularly tricky problems.
   c. Use an issue tracking tool.

Développeurs de SageMath au LaBRI et à l'IMB

- Bill **Allombert**, CNRS, IMB (Debian, PARI/GP, GAP)
  `https://www.math.u-bordeaux.fr/~ballombe/`
  `https://www.cnrs.fr/fr/personne/medailles-de-cristal-2020`
- Xavier **Caruso**, CNRS, IMB
  `http://xavier.toonywood.org/softwares/`
- Vincent **Delecroix**, CNRS, LaBRI
  `https://www.labri.fr/perso/vdelecro/programming.html`,
  `https://pypi.org/project/surface-dynamics/`,
  `https://pypi.org/project/sage-flatsurf/`, ...
- Sébastien **Labbé**, CNRS, LaBRI
  `https://pypi.org/project/slabbe/`
  `https://gitlab.com/seblabbe/slabbe/-/pipelines`
  `https://arxiv.org/abs/1808.07768` avec Jupyter notebook associé
- Pascal **Weil**, CNRS, LaBRI
  `https://pypi.org/project/stallings-graphs/`
  `https://www.labri.fr/perso/weil/software/version-0-2/html/`

Rejoignez-nous pendant les jeudis Sage du LaBRI (10h à 12h) :
`https://cea.labri.fr/pmwiki.php/Groupe/Sage`

Présentation des participants et Sondage

Les participantEs se présentent. Prise des présences.
Les laboratoires des personnes inscrites :

LaBRI	Laboratoire Bordelais de Recherche en Informatique
IMB	Institut de Mathématiques de Bordeaux
IMS	Intégration du Matériau au Système
LCPO	Laboratoire de Chimie des Polymères Organiques
LCTS	Laboratoire des Composites Thermostructuraux
LP2N	Laboratoire Photonique, Numérique et Nanosciences
ICMCB	Institut de Chimie de la Matière Condensée de Bordeaux
LAB	Laboratoire d'Astrophysique de Bordeaux

- Sondage : Linux, Windows, OS X.
- Sondage installation : tout est installé (python + jupyter + jupyterlab), problème d'installation, pas essayé.

Les 4 séances

Séance 1 (25 février 2021) : Python

- Calculatrice Python et structures de données de base
- Notebook Jupyter et programmation en Python
- JupyterLab et les classes en Python

Séance 2 (4 mars 2021) : `git`

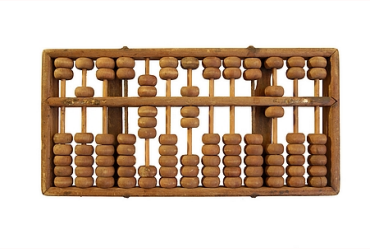
- La GPL, le terminal (bash), logiciel de contrôle de version (git), serveur git, premiers commits dans un projet, création d'une clé SSH.

Séance 3 (11 mars 2021) et Séance 4 (18 mars 2021) :

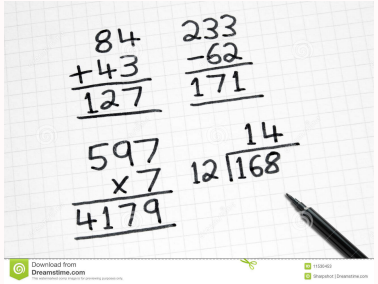
- Structures de données de base : `tuple, dict, set, frozenset`
- Éditeur de texte, logiciel de création de fichiers (make)
- Documentation et tests (`doctest`), compilation de la documentation (`ReStructuredText`, `sphinx`), création d'un paquet Python et reproductibilité.
- Librairies Python (Cython, Pandas, Matplotlib, SageMath, argparse, ...) selon les goûts.

Si c'est trop ambitieux, on adaptera le rythme.

Instruments d'aide aux calculs

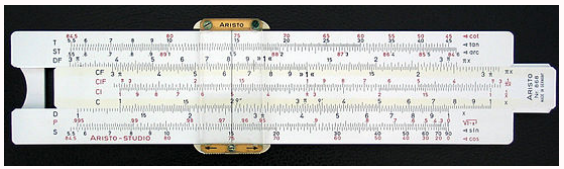


Boulier



Chiffres arabes (-IIIe s.)

La Pascaline (1645)



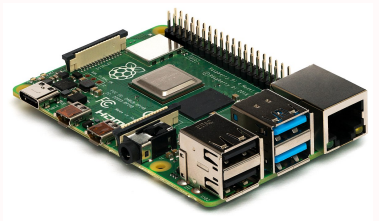
Règle à calcul (XVIIe s.)

HP-35 (1972)

TI-89 (1998)

IMB PC 5150 (1981)

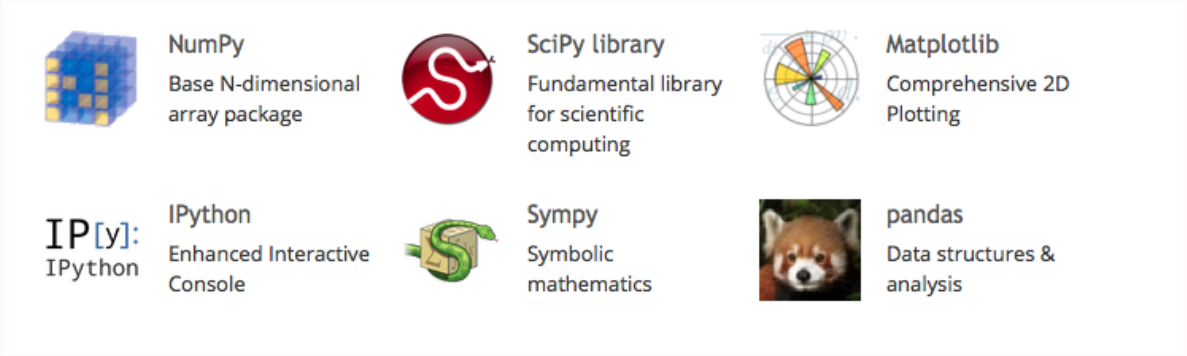
Jade (2010)



Raspberry Pi (2012)

Source des images : wikipédia.

Python scientifique



- 1991 : première version de **Python**
- 2000-2001 : Matplotlib, **IPython**, SciPy
- 2006-2008 : NumPy, **SageMath**, SymPy, Pandas, Cython
- 2009 : Python 3.0

W. Stein, *Math. Software and Me:A Very Personal Recollection*, Dec. 2009.
F. Perez, *The IPython notebook: a historical retrospective*, 8 jan 2012.

- 2012-2014 : Julia, **Jupyter**
- 2015 : 70 000 librairies Python dans le *Python Package Index (PyPI)*
- 2018 : Project Jupyter receives the 2017 ACM Software System Award
- 2018 : **JupyterLab**
- 2020 : SageMath 9.0 basé sur Python 3

Références

- https://docs.python.org/3.8/tutorial/
- How to Think Like a Computer Scientist - Learning with Python, http://openbookproject.net/thinkcs/python/english3e/
- Cours *Logiciels mathématiques*, U. de Liège, 2016 (10h + 20h TP) (Chapitre 11 à 17 : initiation à la programmation en Python) http://www.slabbe.org/Enseignements/MATH2010/notesdecours/ http://www.slabbe.org/Enseignements/MATH2010/exercices.pdf
- *Programming with Python*, Software Carpentry, (2019) https://swcarpentry.github.io/python-novice-inflammation/
- *Calcul mathématique avec Sage*, 468 pages (dispo en français, English, or German) http://sagebook.gforge.inria.fr/
- Livres de Nicolas Rougier, https://www.labri.fr/perso/nrougier/
  - Scientific Visualization - Python & Matplotlib (2020)
  - Towards Reproducible Research (2019)
  - From Python to Numpy(2017)

Aujourd'hui – 25 février 2021 – Python

- 9h00 à 9h20 : Présentation d'introduction
- 9h20 à 9h50 : Division en groupes Windows, OSX, Linux (discussions installation)
- 9h50 à 10h20 : Calculatrice Python, et structures de données de base `int, float, +, -, *, /, //, math, bool, help(), type, print variables, affectation, bool, str, list, range`
- 10h20 à 11h30 : Pause
- 10h30 à 11h30 : Notebook Jupyter et programmation en Python : User Interface Tour, `if, for, while, def`
- 11h30 à 12h00 : Du temps pour faire des exercices (séparés en 8 groupes).

Les contenus des cours seront disponibles ici :

`http://www.slabbe.org/blogue-index/`