

# Formations ASARD

Vous trouverez ici les éléments concernant les formations réalisées par des membres de ASARD (anciens P2I ou SADL). Les identifiants classiques LISN permettent de se logger aux repositories Gitlab le cas échéant.

date	orateurs	formation
11/2022	Anne-Catherine Letournel	<a href="#">présentation ASARD aux nouveaux entrants</a>
07/2021	Erik Bray	<a href="#">Configurable parameters of complex software</a>
04/2020	Erik Bray	<a href="#">Gitlab Continuous Integration tutorial</a>
01/2020 et 02/2020	Erik Bray	<a href="#">Introduction puis atelier Docker</a>
11/2019	Jonas Renault	<a href="#">Python Data Formats (Cartolabe)</a>
12/2018 et 02/2019	Diviyan Kalainathan & Corentin Tallec	<a href="#">ateliers administrateurs et utilisateurs pour le séquenceur slurm /cluster GPU</a>
03/2018	Laurent Pointal	<a href="#">Introduction sur les bases du langage python</a>
03/2018	Sylvain Caillou	<a href="#">Python pour la programmation scientifique</a>
	Jalel Chergui	<a href="#">Calcul sur grappe</a>

## Configurable parameters of complex software

**Abstract:** A common problem faced in researchers' code is how to manage the many dozens, or even hundreds of parameters that can effect the outcomes of their experiments. While this is a multi-faceted problem with no one-size-fits-all solution, we will look at a couple best practices from the software engineering community for reasoning about the configurable parameters of complex software.

First we will take a brief look at popular hierarchical data formats that are commonly used today to configure software, including:

- JSON
- YAML
- (mention of others including TOML, XML, ...)

These formats share common features, in that they represent data types commonly found in most modern programming languages (scalar types such as ints, strings, etc. and containers such as arrays/lists and dictionaries/associative arrays) and so can be used as an interchange format between software written in different languages.

Then we will look at how we can organize the *structure* of our software's configuration using a *schema*, specifically in the JSON Schema format (which despite its name is not restricted to use with JSON).

Time permitting we will also look at some examples of how JSON Schema can be extended for application-specific semantic validation for your configuration, or to generate code (e.g. how to generate a command-line interface from a declarative schema).

The language used for all examples will be Python, though none of these concepts or tools are unique

to Python.

[Repo gitlab 2021-07-01-JSON-Schema-Tutorial.](#)

## Gitlab CI

A partir d'un retour d'expérience sous gitlab (celui du LRI), présentation par Erik Bray de la mise en oeuvre de l'intégration continue offerte en natif par la plateforme Gitlab.

Dans la version présentée, l'utilisateur fournit et configure ses propres 'runners' qui sont les unités matérielles d'exécution des tests. En l'occurrence, les 'runners' pris en exemple seront TitanV et Kepler: les 2 serveurs à GPU qui sont en libre service pour les tests et développements au LRI.

[Repo gitlab 2020-04-07-GitLab-CI-Tutorial.](#)

## Docker

Présentation par Erik Bray des principes de Docker, indépendamment des systèmes d'exploitation et des langages de programmation utilisés:

[Repo gitlab 20-01-14-Docker.](#)

Dans la continuité de l'introduction à Docker présentée le 14 janvier par Erik, une deuxième séance interactive à partir d'exercices est proposée autour de Docker. Il n'est pas nécessaire d'avoir assisté à la séance du 14 janvier pour profiter de celle du 5 février.

[Repo gitlab 20-02-05-Docker-Tutorial.](#)

## Python Data Formats

Retour d'expérience de Jonas Renault, co-auteur de [Cartolabe](#), autour des formats de manipulation de données volumineuses en python (stockage, échange, utilisation). Cartolabe est une application web qui utilise des grandes quantités de données et qui se doit d'être hautement 'responsive'. Présentation sous forme de jupyter Notebook.

[Repo gitlab pandas-dataformats-benchmark](#)

## Séquenceur Slurm

Contexte (décembre 2018):

- Le programme Saclay-IA finance 1,5M€ de calculateurs GPU dans les mois qui viennent. Cette somme sera répartie en 2 projets: Factory-IA et Lab-IA..
- Le premier projet est lab-IA: c'est l'évolution du cluster actuel 'dsp@saclay'. Accessible à la communauté du LRI, il comportera à terme une trentaine de GPUs, dont 23 nvidia V100 à 32Go

de mémoire.

- Le deuxième projet, Factory-IA, est géré par le CEA et comporte environ 150 GPU.
- De son côté, l'Ildris investit aussi par l'intermédiaire de GENCI, dans 1000 GPU V100.
- Pour sa part, le LRI vient de faire l'acquisition d'un modeste noeud avec 2 V100.

Ces 4 clusters/noeuds seront gérés avec l'ordonnanceur de tâches SLURM.

Dans ce contexte, Diviyan R. et Corentin T., doctorants de l'équipe A&O et experts en administration de clusters à GPU avec Slurm, proposent 2 types d'ateliers autour de cette technologie.

- un premier atelier orienté 'administrateurs' pour aider les administrateurs à déployer slurm et à maintenir l'infrastructure du cluster (avant les congés de Noël);
- un ou plusieurs ateliers ultérieurs, à partir de mi-janvier 2019, destiné(s) aux utilisateurs, pour les aider à organiser leurs soumissions et leur faire gagner du temps au démarrage de l'utilisation de ces nouveaux clusters.

NB: ces 2 ateliers seront inscrits au catalogue des formations ADUM de l'école doctorale dans la catégorie 'formations transverses'.

[Repo gitlab sequenceur-slurm](#)

## Python

### Introduction sur les bases du langage python — L.Pointal

- Installation / Module / Lancement / Modèle d'exécution /
- Types de base / Opérations / Conteneurs / Manipulations / Affectation / Espaces de noms / Imports
- Instructions composées If For While / Blocs d'instructions / Générateurs / Fonctions / Portée variables / Gestion des erreurs
- Lecture fichier texte (+structuré CSV) / Écriture fichier texte (+structuré CSV)
- Extras (lecture/écriture binaire, formatage de chaînes, programmation objet...)

Présentation répétée sur 3 sessions, début mars 2018.

Documents:

- [Support de présentation](#)
- [Slides et notes en ligne.](#)
- [Mémento Python 3](#)
- [Conda cheat sheet](#)

## Python pour la programmation scientifique

Préambule (Notes de A. Reberac (LATMOS/CNRS,) et L. Falletti (LATMOS/UPMC) issues de la formation dispensée par leur soin sur la programmation scientifique avec Python - 12.2017)

- [Notes sur Anaconda](#)

- [Notes sur les Modules, installation et règles d'importation](#)
- [Notes sur Jupyter et les notebooks](#)

Intégrale de la formation:

Python pour la programmation scientifique

## Numpy: outil de calcul numérique en python — S.Caillou

- Introduction: Qu'est-ce que Numpy? / Pourquoi Numpy ? / Différences entre les tableaux Numpy et les listes Python / Configuration, Installation
- Représentation de tableaux: L'objet ndarray / Organisation des tableaux en mémoire / Différentes méthodes pour construire des ndarray / Performances
- Typage des données: Définition du type des données / Définition de son propre type de données / Conversion de chaînes, de listes et de dictionnaire de caractères en data-type
- Accès et manipulation des données d'un ndarray: Indexation et Slicing / Manipulation des données / Copie de tableau / Comparaisons et masques
- Opérations sur les tableaux: Opérations de bases / Fonctions mathématiques
- Manipulation de fichiers: Fichiers textes / Fichiers binaires

Présentation répétée sur 2 sessions, mars 2018.

Documents et références:

- [Numpy cheat sheet](#)
- <https://docs.scipy.org/doc/numpy/user/index.html>
- <https://docs.scipy.org/doc/numpy/reference/>
- <http://www.scipy-lectures.org/intro/numpy/numpy.html>
- <http://perso.univ-perp.fr/langlois/images/pdf/mp/www.mathprepa.fr-une-petite-reference-numpy.pdf>

## La préparation des données avec Pandas — S.Caillou

- Introduction: Qu'est-ce que Pandas? / Fonctionnalités de Pandas / Comparaisons entre Pandas et R / Installation et import
- Structures de données spécifiques de Pandas: Series / DataFrame
- Lecture et écriture de fichiers: csv / excel / hdf5 et trafic séquentiel de gros fichier
- Manipulation des tables de données: Regroupement par facteur, jointure, concaténation / Tri et rangs / Permutation et échantillonnage aléatoire / Transformations, opérations
- Statistiques de bases et visualisation: Description univarié / Description bivariée / Gestion données manquantes

Présentation répétée sur 2 sessions, mars 2018.

Documents et références:

- [Pandas](#)
- [Pandas Cheat Sheet](#)
- [Pandas Cheat Sheet \(un autre\)](#)
- <http://pandas.pydata.org/pandas-docs/stable/overview.html>
- <http://www.webpages.uidaho.edu/%7Esteval/504/Pandas%20DataFrame%20Notes.pdf>

- <https://www.python-course.eu/pandas.php>
- <http://python-simple.com/python-pandas/panda-intro.php>
- <https://www.dataquest.io/blog/pandas-python-tutorial/>

## Introduction à Scikit-learn, bibliothèque pour l'apprentissage automatique — S.Caillou

- Introduction
- Représentation et visualisation des données
- Données d'apprentissage et de test
- Régression
- Classification
- Clustering
- Évaluation et sélection de modèle
- Sauvegarde de modèle

Présentation répétée sur 2 sessions, mars 2018.

Documents et références:

- <http://scikit-learn.org/stable/tutorial/index.html>
- <https://www.coursera.org/learn/machine-learning>
- <http://ml-cheatsheet.readthedocs.io/en/latest/index.html>
- <https://ai.stanford.edu/~nilsson/MLBOOK.pdf> (un peu daté mais loin d'être obsolète)
- <https://www.youtube.com/watch?v=vdqu6fvjc5c> (un talk du brillant Chris Olah sur les réseaux de neurones)
- <https://www.youtube.com/watch?v=UzxYlbK2c7E> (premier des cours de la série des lecture d'Andrew Ng, professeur à Stanford)

## Visualisation de données en Python — S.Caillou

- Matplotlib: Plot de Données en 2D
- Bokeh: Visualisation de données interactive et diffusion web
- Mayavi : Visualisation scientifique 3D

Présentation répétée sur 2 sessions, mars 2018.

Documents et références:

- <http://www.aosabook.org/en/matplotlib.html>
- <http://www.labri.fr/perso/nrougier/teaching/matplotlib/>
- <https://python.developpez.com/tutoriels/graphique-2d/matplotlib/#LI>
- [https://github.com/cea-irfu-sap/CEAPythonWorkshopForAstronomers/tree/master/06-plotting-Fa bio](https://github.com/cea-irfu-sap/CEAPythonWorkshopForAstronomers/tree/master/06-plotting-Fa%20bio)
- <http://nbviewer.jupyter.org/github/jrjohansson/scientific-python-lectures/blob/master/Lecture-4-Matplotlib.ipynb>
- <https://github.com/jakevdp/PythonDataScienceHandbook/blob/master/notebooks/04.00-Introduction-To-Matplotlib.ipynb>
- <http://www.python-simple.com/python-matplotlib/matplotlib-intro.php>
- <https://bokeh.pydata.org/en/latest>

- <https://github.com/bokeh/bokeh-notebooks> (notebooks support pour la présentation de Bryan Van de Ven : Bokeh for Data Applications and Visualization Intermediate | SciPy 2016)
- [http://www.scipy-lectures.org/packages/3d\\_plotting/index.html](http://www.scipy-lectures.org/packages/3d_plotting/index.html) (Notes de lecture de Gaël Varoquaux)

## Calcul sur grappe

Utilisation du cluster grappe pour le calcul - J. Chergui

- [Présentation](#)
- [Démonstrations](#)

## Appel python→C/Fortran

Exemples commentés lors de la présentation:

- [Python→C avec ctypes](#) ([ctypes](#) - a foreign function library for Python)
- [Python→C avec swig](#) ([SWIG](#) - Simplified Wrapper and Interface Generator)
- [Python→C avec f2py](#) ([f2py](#) - Fortran to Python interface generator)

Installations préalables: création d'un environnement virtuel, activation de celui-ci, et installation de paquets Python dans cet environnement.

```
conda create -n formation1 python=3.6
conda activate formation1
conda install numpy requests
```

+ installer (ou faire installer) le paquet swig.

## liens vers cette page:

**Plugin Backlinks: Pas de réponse.**

From:

<https://asard.lisn.upsaclay.fr/> - **Wiki ASARD**

Permanent link:

<https://asard.lisn.upsaclay.fr/doku.php?id=fr:general:formation:start>

Last update: **2023/02/16 12:10**

