



DU **25** AU **28**  
**OCTOBRE**  
**2014**

CONFÉRENCES SUR LE LANGAGE  
DE PROGRAMMATION PYTHON,  
POUR TOUS LES NIVEAUX

Entrée libre

samedi/dimanche : Conférences  
lundi/mardi : Sprints

Université Lyon 1  
[www.pycon.fr/2014](http://www.pycon.fr/2014)



# Table des matières

Bienvenue à PyconFR 2014.....	5
Venir à PyconFR.....	6
Plan du campus.....	7
Bâtiment Thémis.....	7
Bâtiment Nautibus.....	8
Accès en train.....	9
Accès en avion.....	10
Accès en voiture.....	10
Accès en vélo.....	11
Plan de Lyon.....	12
Plan des transports en commun.....	13
Informations générales.....	14
Accès Wifi.....	14
Déjeuner.....	14
Sandwichs.....	14
À proximité.....	15
Soirée le samedi 25 octobre.....	16
Que faire à Lyon ?.....	17
Office du tourisme de Lyon.....	17
Hôtels.....	17
Culture.....	17
Théâtres.....	17
Cinémas.....	17
Musées.....	18
Galeries d'art.....	19
Divers.....	19
Piscines.....	19
Shopping.....	19
Tourisme.....	20
Restaurants.....	20
Les chefs étoilés du guide Michelin.....	20
Les rues / quartiers de « bouchons Lyonnais » typiques.....	21
Les toques blanches Lyonnaises.....	21
Pubs.....	21
Programme de la conférence.....	23
Samedi 25 octobre 2014.....	24
Dimanche 26 octobre 2014.....	25
Sprints.....	26
Présentations et conférenciers.....	27
Amphi T7 – Samedi 25 octobre.....	27
BDD avec Behave.....	27
PostgreSQL et Python, un beau mariage.....	28
Un serveur fiable avec python 3.4.....	29
Why SCons is not slow.....	30
CubicWeb - Vos données ont du sens.....	31
Je teste mon code avec py.test.....	32
La gestion de version, ce problème tellement simple.....	33
RedBaron, une approche bottom-up au refactoring en Python.....	34
Let's do some docker.....	35
xbus.....	36
Radicale : Serveur de calendriers en Python, simple, libre et pragmatique.....	37
Amphi T8 – Samedi 25 octobre.....	38

SIG et Python : Love Story.....	38
Reactive : Document + Event + Types.....	39
Wagtail - why we built another CMS.....	40
Performance des frameworks web : Python vs the world.....	41
DEPOT, story of a file.write() gone wrong.....	42
Modoboa, le mail avec du python dedans.....	43
Anitya, tenez vous au courant des nouvelles sorties !.....	44
How The Oslo Program Reduces The Technical Debt.....	45
Les docstrings pour bien documenter son projet.....	46
Ansible, au dela du dome de YAML.....	48
Comment le packaging m'a simplifié la vie.....	49
Amphi T9 - Samedi 25 octobre - Session enseignement – recherche.....	50
Sage, un logiciel libre de mathématiques.....	50
Pattes de mouche: une modélisation de biophysique avec python scientifique.....	51
Python dans la recherche, l'enseignement, et l'innovation en biotechnologie.....	52
Retour d'expérience sur TOMUSS.....	54
De la gestion de clusters à l'examen in silico des processeurs : la polyvalence de Python.....	55
IRPF90 : a Fortran code generator for HPC.....	57
Ipython Notebook pour l'enseignement.....	59
Python et les supernovae.....	60
Du python qui ne manque pas d'air.....	61
Mathématiques expérimentales : Sage pour la combinatoire.....	62
Visualisation avec python.....	63
SymPy une bibliothèque pure Python pour le calcul symbolique.....	64
Retour d'expérience : l'école d'été python scientifique.....	65
Quelques idées d'optimisation d'un code numérique écrit en Python.....	66
Poppy - la plateforme robotique humanoïde qui s'adresse à tous.....	67
Salle 51 - Samedi 25 octobre - Session enseignement – recherche.....	68
BoF retour expérience enseignement Python.....	68
Amphi T7 – Dimanche 26 octobre.....	69
Présentation de Nao.....	69
Obscurcissement de bytecode Python.....	70
Asyncio: offrez des tulipes à vos entrées sorties asynchrones.....	71
Exploration de la boucle d'événement asyncio.....	73
Gunicorn, le problème du thundering herd et autres challenges liés à la concurrence.....	75
Bootstrapping Machine Learning.....	76
Kivy - a very short introduction.....	78
Amphi T8 – Dimanche 26 octobre.....	79
Une interface expérimentale pour lier OpenGL à Python.....	79
APIs hypermédia.....	82
Grimper en haut de la Pyramid.....	83
Pyramid, OpenERP, des métaclasses et encore plus !.....	85
Brood, une application web par des gens qui n'y connaissent pas grand chose.....	86
Brython (Python dans le navigateur).....	88
Lean data science, or how to do machine learning with what you have to hand.....	90
Ce que je n'aime pas dans Python.....	92
Posters.....	93
Project of software for research in fluid dynamics.....	93
OSRM, le routing par excellence.....	94
Science reproductible.....	95
Une interface expérimentale pour lier OpenGL à Python.....	96
Code de conduite.....	99
Charte AFPy.....	99

La version “courte” .....	99
La version “longue” .....	100
Sponsors.....	101
Gold.....	101
Bronze.....	103
Partenaires.....	104

# Bienvenue à PyconFR 2014

PyconFR, c'est le rendez-vous annuel des utilisateurs du langage de programmation Python organisé par l'AFPy, Association Francophone Python. La conférence, gratuite et libre d'accès, accueille chaque année plus de 200 développeurs venus de toute la France et d'ailleurs.

La conférence Python aura lieu du 25 au 28 Octobre à Lyon,

Les 27 et 28 Octobre seront dédiés aux sprints.

Grâce au soutien Faculté des Sciences et Technologies de l'Université Lyon 1 et de la Fédération Lyonnaise de Modélisation et Sciences Numérique (FLMSN), nous avons pu mettre en place une session enseignement / recherche et encourager les contributions du monde académique où le Python est très utilisé avec les notebook Python pour la partie enseignement et dans de nombreux domaines de recherche (calcul HPC, simulation numérique, neurosciences, robotique, web sémantique, simulation économique, ...).

Suivez @pyconfr sur twitter.

Toute l'équipe de PyconFR est ravie de vous accueillir à Lyon et vous attend nombreux pour la conférence annuelle PyconFR 2014.

## Venir à PyconFR

Le campus LyonTech-La Doua est situé au Nord de Villeurbanne, sur une zone riche en espaces verts (Parc de la Tête d'Or, Coulée verte de la Feysine), et à proximité immédiate de Lyon.



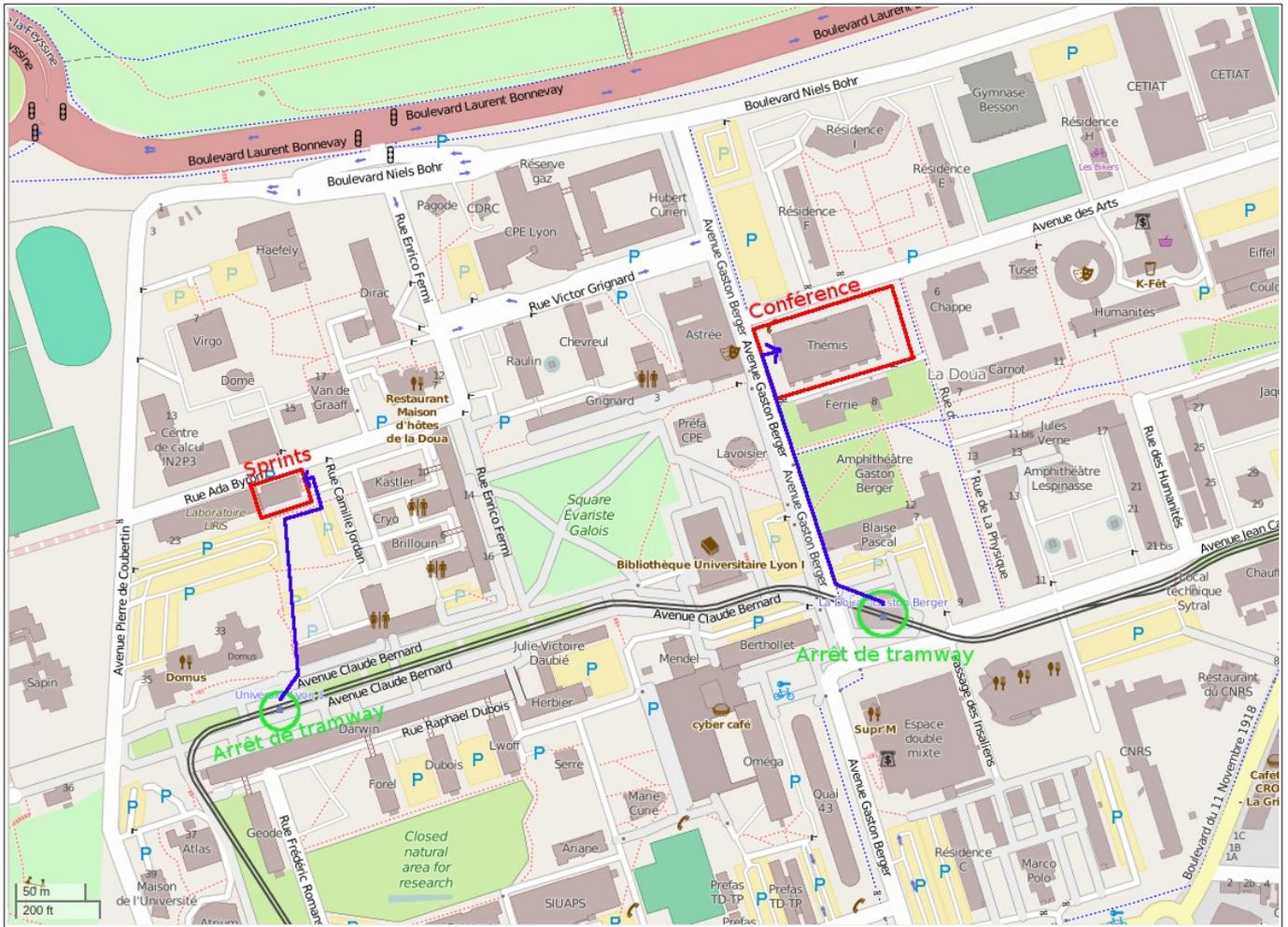
Caractéristiques du campus LyonTech-la Doua :

- Technopôle de 100 hectares, basé au Nord de Villeurbanne et à proximité immédiate de Lyon
- 26 000 personnes travaillent sur le site
- 40 % du potentiel scientifique lyonnais
- 1er site d'enseignement et de recherche en France pour la collaboration avec l'industrie, avec 31 M€ de contrats réalisés



# Plan du campus

Vous pouvez prendre des repères à l'aide du site de l'Université Lyon1 [http://oscar.univ-lyon1.fr/appli-externe/plan/plans/plan\\_campus\\_general.html](http://oscar.univ-lyon1.fr/appli-externe/plan/plans/plan_campus_general.html) ou avec OpenStreetMap <http://www.openstreetmap.org/>



## Bâtiment Thémis

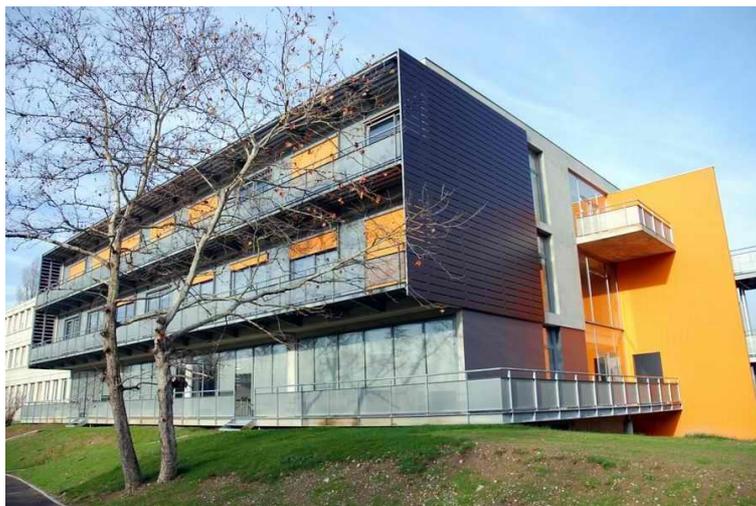
La conférence se déroulera dans les amphithéâtres 7, 8 et 9 du bâtiment Thémis de l'Université Lyon 1.



Illustration 1: ©Eric.Le Roux/Communication/UCBL

## Bâtiment Nautibus

Les sprints se dérouleront au rez-de-chaussée du bâtiment Nautibus dans les salles C4 et C5.

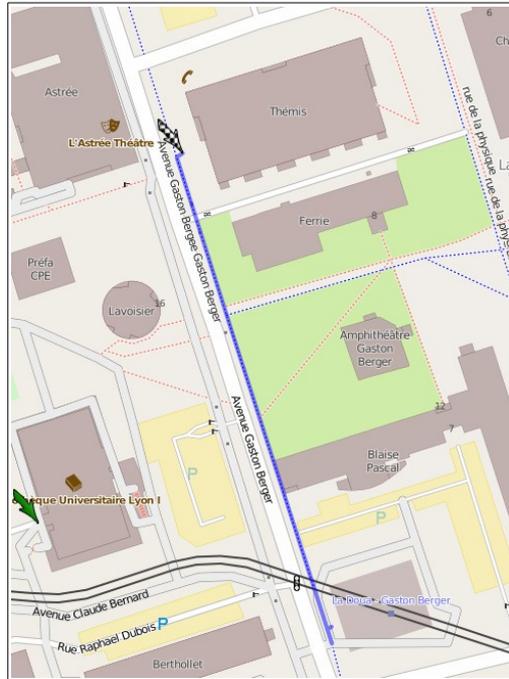


*Illustration 2: ©Eric.Le Roux/Communication/UCBL*

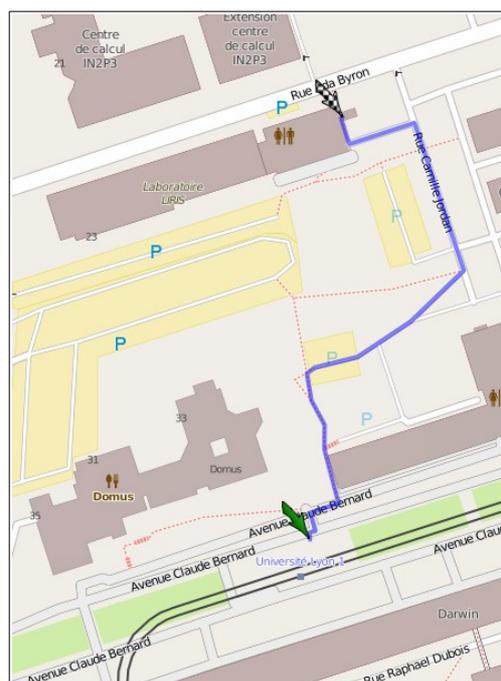
## Accès en train

La gare de Lyon Part Dieu est directement reliée à de nombreuses gares françaises et des pays limitrophes. Vous pouvez trouver le train idéal sur Voyages SNCF (<http://www.voyages-sncf.com/>) ou TGV Europe (<http://www.tgv-europe.com/>).

Depuis la gare, il faut ensuite prendre le **tramway T1** ou **T4** en direction d'**IUT Feysine**, puis descendre à **La Doua - Gaston Berger**. Ensuite, il faudra parcourir 200m vers le nord sur l'avenue Gaston Berger.



Pour se rendre aux sprints, il faudra prendre le **tramway T1** ou **T4** en direction d'**IUT Feysine**, puis descendre à **Université Lyon 1**. Ensuite, il faudra suivre le plan d'accès au bâtiment.



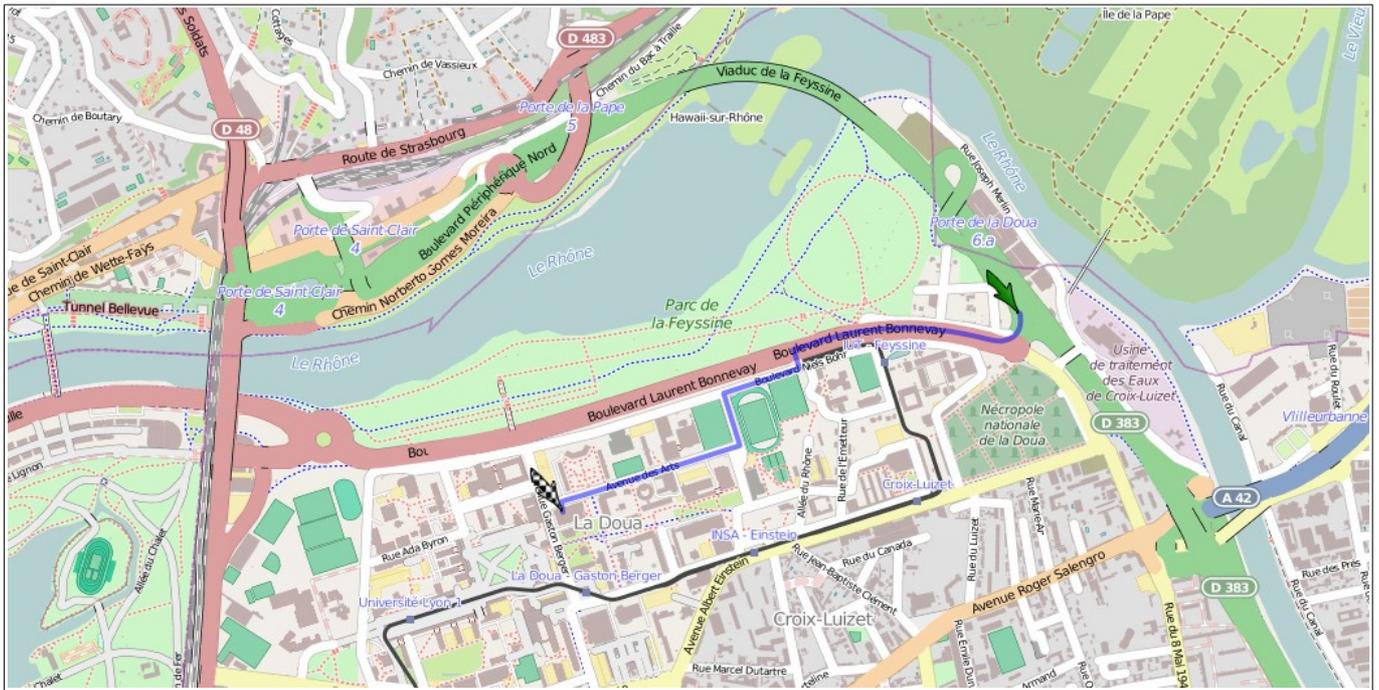
## Accès en avion

L'aéroport de Lyon Saint Exupéry est relié à la gare de Lyon Part Dieu par le tramway Rhône Express., <http://www.rhonexpress.fr/>

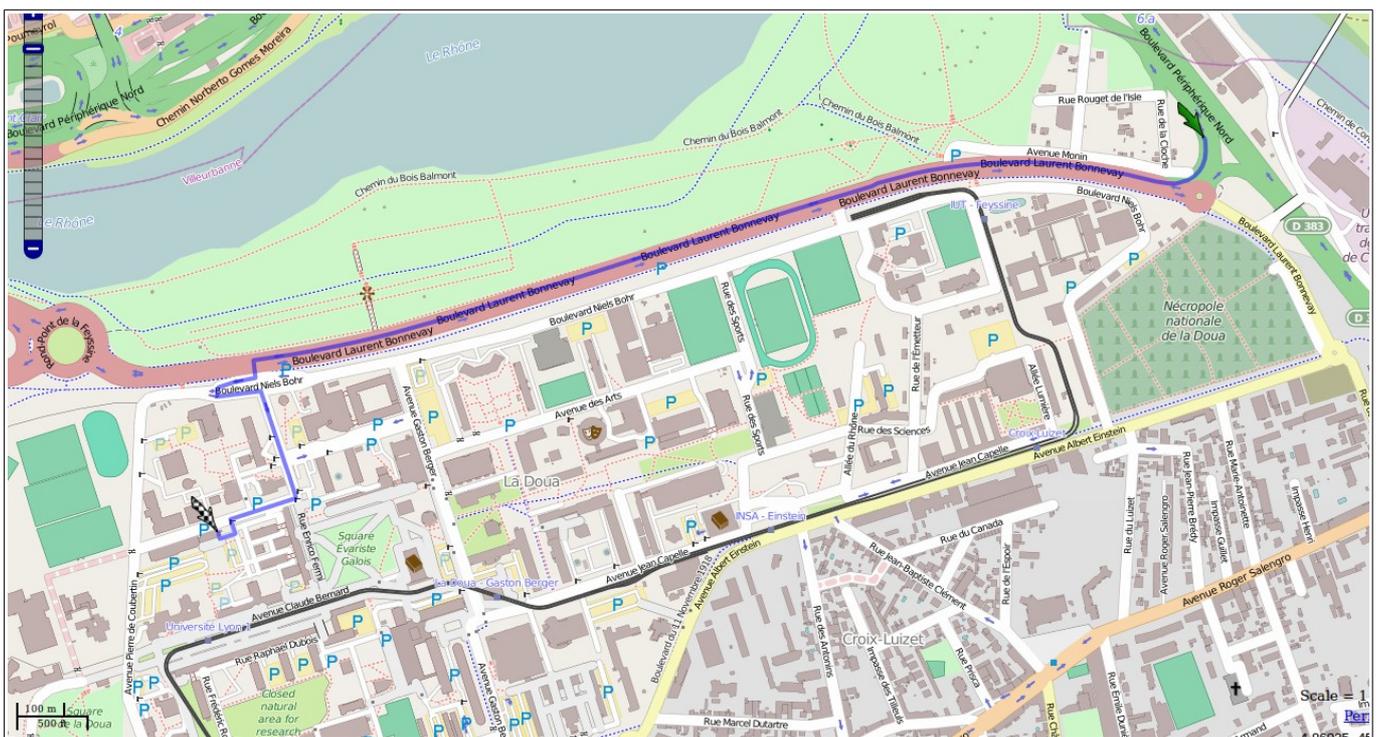
Une fois arrivé à la gare, suivez les instructions ci dessus.

## Accès en voiture

Prendre en direction du **périphérique Nord**, sortir à la **porte de La Doua**. Suivez ce plan pour vous rendre aux conférences :



ou celui-ci pour vous rendre aux sprints.



De nombreuses places de parking sont disponible autour des bâtiments.

## **Accès en vélo**

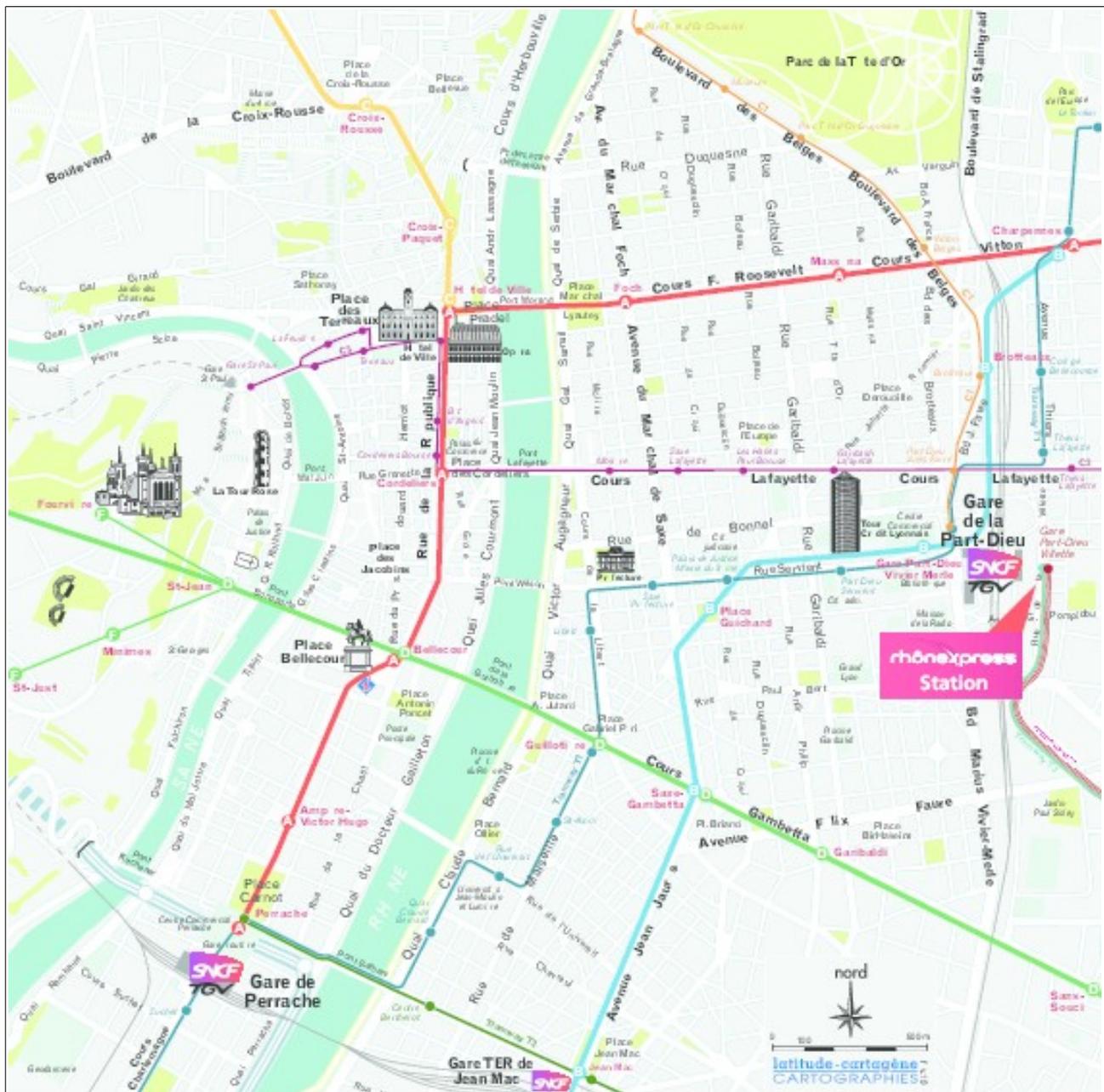
Lyon dispose d'un bon réseau cyclable et de vélos en libre service, les vélo'v  
<http://www.velov.grandlyon.com/>

Les stations vélo'v à proximité sont :

- station La Doua / Avenue des Arts (au pied du bâtiment Thémis)
- DOUA / Gaston Berger (proche de l'arrêt de tramway "La Doua - Gaston Berger")
- Condorcet/ 11 Nov. 1918 (proche de l'arrêt de tramway "Université Lyon 1")

# Plan de Lyon

Vous pouvez consulter le plan interactif de la ville de Lyon : <http://cartes.lyon.fr/plan/>



# Plan des transports en commun

Voici un mini plan des principales lignes de transport en commun (métro et tramway) des TCL (Transports en Commun Lyonnais).

Pour plus de détails sur un trajet, vous pouvez consulter le plan interactif en ligne : <http://plan-interactif.tcl.fr/> et pour les différents types de tickets : <http://www.tcl.fr/Tarifs/Tickets>



# Informations générales

## Accès Wifi

Suite à votre inscription sur Yurplan , vous avez reçu un identifiant et un mot de passe par mail.

Muni de ces identifiants, connectez-vous sur le réseau de Lyon1 "**UCBL-Portail**".

Ouvrez un navigateur et saisissez votre identifiant, votre mot de passe et sélectionnez « **univ-lyon1.fr** » comme FDQN.

Attention, le réseau UCBL-Portail n'est pas sécurisé, le trafic entre votre ordinateur et le point d'accès n'est pas chiffré et peut donc facilement être intercepté.



The image shows a web browser window displaying the Lyon 1 captive portal. The header features the Lyon 1 logo and the text "Portail captif du réseau sans fil". Below the header, there is a login form on the left and a welcome message on the right. The login form includes fields for "Nom d'utilisateur", "Mot de passe", and "FDQN" (set to "univ-lyon1.fr"), along with a "Log In" button. A disclaimer at the bottom of the form states: "En vous connectant, vous vous engagez à respecter la charte informatique de l'Université". The welcome message on the right reads: "Bienvenue sur le portail captif du réseau sans fil de l'Université Claude Bernard Lyon 1".

Si vous avez un compte dans un établissement supérieur d'enseignement et de recherche, vous pourrez vous connecter au réseau **EDUROAM**.

## Déjeuner

Plusieurs options sont possibles pour déjeuner.

## Sandwichs

L'organisation prévoit une cinquantaine de sandwichs qui seront distribués en priorité aux participants de la session "**retour d'expérience enseignement Python**" le samedi midi, et de l'AG de l'AFPy le dimanche midi.

Le lot de sandwich sera composé de 20 jambon/beurre, 20 poulet et 10 végétarien.

Une participation aux frais d'un montant de 3€ sera demandé pour chacun des sandwiches.

## À proximité

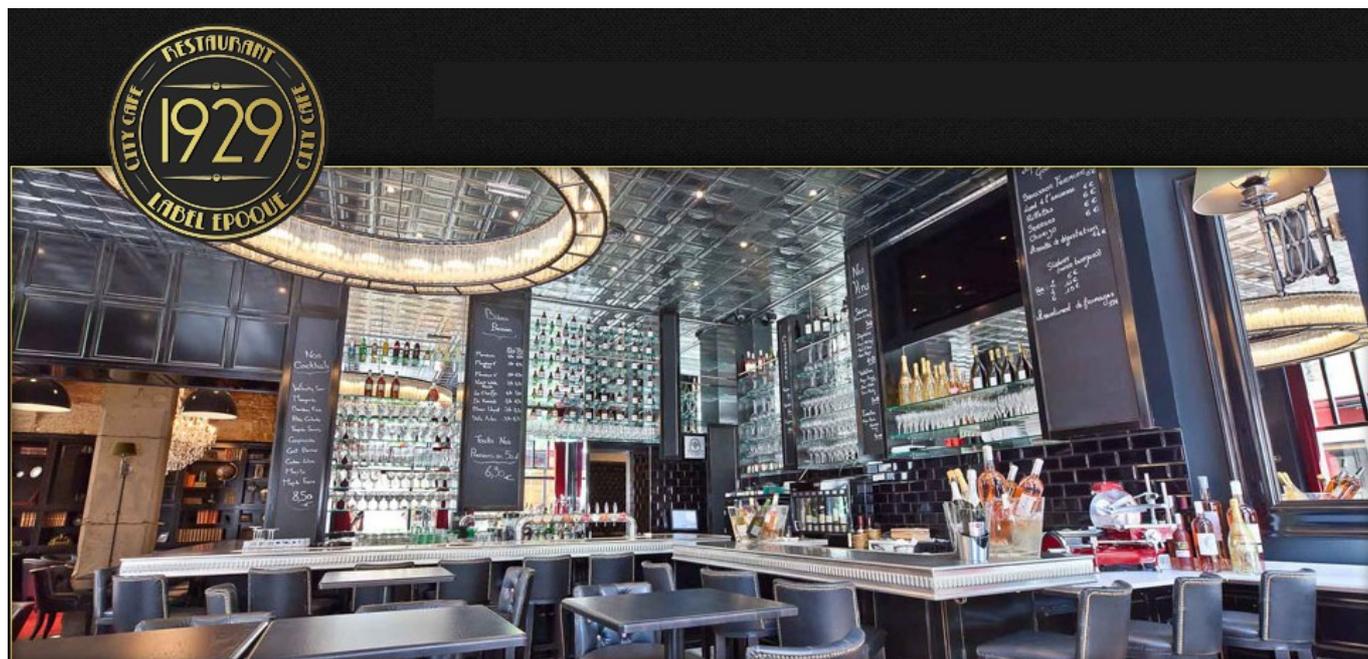
Plusieurs restaurants se situent à proximité de la conférence, et seront ouverts les deux midis :

- le Supr'm café : plat du jour, pizzas, burgers, salades. Environ 80 places. À 200m de la conférence, près de l'arrêt de tram. Avenue Gaston Berger/Avenue Jean Capelle, Villeurbanne.
- l'Univers Café : pizzas, salades, kebabs. Environ 80 places. À 400m de la conférence, continuer tout droit après l'arrêt de tram, puis prendre à gauche au feu. 126 boulevard du 11 novembre, Villeurbanne.
- Mosaïc Kebab : kebabs. Environ 80 places. À 400m de la conférence, continuer tout droit après l'arrêt de tram. 6 rue Léon Fabre, Villeurbanne.
- Snack Ambiance : kebabs. Environ 50 places. À 400m de la conférence, continuer tout droit après l'arrêt de tram. 128 boulevard du 11 novembre, Villeurbanne.

Consulter la page : <http://www.pycon.fr/2014/dejeuner/>

## Soirée le samedi 25 octobre

Une soirée est organisée le samedi 25 octobre dans le restaurant **1929** situé au 10, rue de la République à Lyon (<http://www.1929-restaurant.com/>).



Pour y accéder depuis la conférence, prenez le **tramway T1** ou **T4** jusqu'à **Charpennes**, puis le **métro A** jusqu'à **Hôtel de ville - Louis Pradel**.

Le nombre de places est limité, pensez à réserver sur Framadate, <http://framadate.org/1ulr4iot61rzvles#bas> !

Le menu coûtera **22 €** (à régler sur place) et est composé de :

- **Entrée** : Bruschetta à l'italienne ou Perles de saumon Gravlax
- **Plat** : Épaule d'agneau rôtie ou Filet de daurade royale
- **Dessert** : Île flottante au caramel ou Fondant au chocolat
- **1 verre au choix** : rouge (côté du Rhône "Belleruche", maison Chapoutier), rosé (Côtes de Provence Château Mistral) ou blanc (Macon "la roche vineuse")

En cas de grande affluence, des plans B, C, D et E sont prévus dans la même rue.

# Que faire à Lyon ?

## Office du tourisme de Lyon

Le Pavillon d'Accueil de l'office du tourisme place Bellecour est ouvert tous les jours de 9h à 18h non stop, 363 jours par an (fermé le jour de Noël et du Nouvel An).

<http://www.lyon-france.com/>

## Hôtels

Consulter la page : <http://www.pycon.fr/2014/se-loger/>

## Culture

### Théâtres

- Théâtre des Célestins: <http://www.celestins-lyon.org>
- Les Subsistances: <http://www.les-subs.com/>
- Théâtre de la Croix-Rousse: <http://www.croix-rousse.com>
- Théâtre du Point du jour: <http://www.lepointdujour.fr/>
- New Generation Theatre: <http://www.tng-lyon.fr/>

### Cinémas

- UGC Ciné Cité Lyon Quai Charles de Gaulle 69006 Lyon
- Institut-Lumière 25 rue du Premier-Film 69008 Lyon
- Cinéma Comoedia 13 avenue Berthelot 69007 Lyon
- UGC Lyon Part-Dieu 2-4 Centre commercial La Part-Dieu 69003 Lyon
- Pathé Vaise 43 rue des docks 69009 Lyon
- UGC Astoria 31 cours Vitton 69006 Lyon
- Le Cinéma-Opéra 6 rue Joseph-Serlin 69001 Lyon
- Le Cinéma 18 rue Saint-Polycarpe 69001 Lyon
- Pathé Cordeliers 20 rue Thomassin 69002 Lyon
- Pathé Bellecour 79 rue de la République 69002 Lyon
- CNP Terreaux 40 rue du Président-Edouard-Herriot 69001 Lyon
- La Fourmi 68 rue Pierre-Corneille 69003 Lyon
- Ciné-Duchère avenue Andreï-Sakharov 69009 Lyon

## Musées

### *Musées dans Lyon*

- Musée des Beaux-Arts de Lyon: <http://www.mba-lyon.fr>
- Musée d'Art Contemporain: <http://www.moca-lyon.org>
- Institut Lumière: <http://www.institut-lumiere.org>
- Musée de l'Imprimerie: <http://www.imprimerie.lyon.fr>
- Musée Gadagne: <http://www.museegadagne.com>
- Musées gallo-romain (Lyon-Fourvière fermé pour travaux et Saint-Romain-en-Gal): <http://www.musees-gallo-romains.com/>
- Musée des Tissus et des Arts Décoratifs: <http://www.musee-des-tissus.com>
- Musée des confluences (ouverture le 20 décembre 2014): <http://www.museedesconfluences.fr/>
- Institut d'Art Contemporain: <http://www.i-art-c.org>
- Musée Africain de Lyon: <http://www.musee-africain-lyon.org>
- Musée d'Art Sacré de Fourvière: <http://www.fourviere.org/>
- Musée des Automates EMA: <http://www.automates-ema.com>
- Musée des Hospices Civils de Lyon (fermé actuellement) : [http://www.chu-lyon.fr/web/Histoire%20et%20Patrimoine\\_684\\_700.html](http://www.chu-lyon.fr/web/Histoire%20et%20Patrimoine_684_700.html)
- Musée des sapeurs-pompiers du Grand Lyon: <http://museepompiers.com/>
- Musée international de la miniature: <http://www.mimlyon.com>
- Musée Testut Latarjet d'Anatomie: <http://museetl.univ-lyon1.fr>
- Centre d'Histoire de la Résistance et de la Déportation: <http://www.chrd.lyon.fr/>
- Maison des Canuts: <http://www.maisondescanuts.com/>
- Musée urbain Tony Garnier: <http://www.museeurbaintonygarnier.com/>

### *Musées autour de Lyon*

- Musée de la poupée: <http://www.lacroix-laval.com/>
- Monastère royal de Brou: <http://brou.monuments-nationaux.fr/>
- Historial du Saint Curée d'Ars: <http://www.musee-ars.org>
- Musée Claude Bernard (fermé pour travaux) : <http://www2.ac-lyon.fr/enseigne/biologie/spip.php?article21>
- Musée du Chapeau: <http://www.museeduchapeau.com>
- Musée du Vieux Saint-Etienne: <http://www.vieux-saint-etienne.com>

- Musée Théâtre Guignol des Vallons du Lyonnais: <http://guignol.ccvl.fr/>

## Galleries d'art

- Galerie du Vieux Lyon: <http://www.galerie-vieuxlyon.com>
- Galerie Laurencin: <http://www.laurencin.net>
- Galerie Olivier Houg: <http://www.olivierhoug.com>
- L'Atelier des Ombres: <http://www.atelierdesombres.com/>
- Le Soleil sur la Place: <http://www.lesoleilsurlaplace.com>
- Galerie L'Ebauchoir: <http://www.ebauchoir.com>
- Galerie de l'Olympe: <http://www.galerie-olympie.com>
- Galerie Grégory Chesne: <http://www.gregorychesne.com>
- Galerie Henri Chartier: <http://www.henrichartier.com>
- Galerie Le Réverbère: <http://www.galerielereverbere.com>
- Galerie OOBLIK: <http://www.ooblik.com>

## Divers

- Jardin botanique de Lyon: <http://www.jardin-botanique-lyon.com/>
- Parc de la Tête d'or: [http://www.lyon.fr/vdl/sections/en/environnement/parcs\\_jardins/tete\\_or\\_1](http://www.lyon.fr/vdl/sections/en/environnement/parcs_jardins/tete_or_1)
- Planétarium: <http://www.planetariumvv.com>
- Adventure Park by Fourvière Aventures Park: <http://www.fourviere-aventures.com/>
- Aquarium Grand Lyon: <http://www.aquariumlyon.fr/>

## Piscines

- La piscine du Rhône, 8 Quai Claude Bernard 69007 Lyon, +33 (0)4 78 72 04 50
- Piscine Jean Mermoz, 12 Place André Latarget 69008 Lyon, +33 (0)4 78 74 33 09
- Centre Nautique Etienne Gagnaire - Piscine de Cusset, 59 avenue Marcel Cerdan 69100 Villeurbanne, +33 (0)4 72 37 72 024

Pour plus d'adresses : <http://www.lyon.fr/page/nagez-toute-lannee-.html>

## Shopping

Vous trouverez un plus grand nombre d'adresses dans le guide de l'Office du tourisme de Lyon.

- Pentes de la Croix-Rousse pour trouver des vêtements et des accessoires originaux  
<http://www.lyon-france.com/Decouvrir-Lyon/SHopping/Shopping-sur-les-pentes-de-la-Croix-Rousse>

- Centre commercial de la Part-Dieu, <http://www.centrecommercial-partdieu.com/W/do/centre/accueil>
- Vieux-Lyon (centre ville)
- Centre commercial Confluence, <http://www.confluence.fr/>
- Centre commercial Carré de Soie, <http://www.carredesoie.com/>

## Tourisme

Lyon est une ville culturelle renommée. Parmi les principaux sites à découvrir :

- Le théâtre antique de Lyon sur la colline de Fourvière et le musée gallo-romain qui sera malheureusement en travaux au moment de la conférence (pour ceux qui le peuvent vous pouvez visiter le musée de Saint-Romain-en-Gal), [http://www.rhone.fr/gallo\\_roman](http://www.rhone.fr/gallo_roman)
- Le vieux Lyon quartier médiéval et Renaissance et ses traboules <http://www.lyon-france.com/Visites-guidees/Incontournables/Le-Vieux-Lyon-Renaissance-Cours-et-Traboules>
- La basilique de Fourvière, richement ornée où l'on a une vue panoramique magnifique sur la ville et parfois jusqu'aux Alpes, <http://www.lyon-france.com/Que-faire/Culture-loisirs/Sites-monuments-historiques/Patrimoine-religieux/Basilique-de-Fourviere>
- L'immeuble de l'Opéra près de la place des Terreaux

D'autres sites incontournables : <http://www.lyon-france.com/Que-faire/Culture-loisirs/Sites-monuments-historiques/Incontournable/?page=2>

## Restaurants

« Lyon est une ville qui donne faim » Jean-François Mespîède ancien directeur du guide Michelin.

A Lyon, il y a de nombreux restaurants, depuis les fameux « bouchons » jusqu'au restaurants étoilés, il y a plus de 1500 adresses à découvrir.

### Les chefs étoilés du guide Michelin

#### Paul Bocuse

Restaurant principal :

L'Auberge du Pont de Collonges

40 Rue de la Plage - 69660 Collonges au Mont d'Or

Tel.: +33 (0)4 72 42 90 90

<http://www.bocuse.fr/>

Brasseries - <http://www.nordsudbrasseries.com/> :

- « Le Nord » 18 Rue Neuve - 69002 Lyon (métro ligne A, arrêt « Hôtel de Ville »)
- « Le Sud » 11 Place Antonin - 69002 Lyon (métro lignes A/D, arrêt « Bellecour »)

- « L'Est » 14 Place Jules Ferry - 69006 Lyon (métro ligne B, arrêt « Brotteaux »)
- « L'Ouest » 1 Quai du Commerce - 69009 Lyon (métro ligne D, arrêt « Gare de Vaise »)
- « Argenson » 44 allée Pierre de Coubertin - 69007 Lyon (métro ligne B , arrêt « Stade de Gerland »)

### **Pierre Orsi**

3 Place Kléber - 69006 Lyon

Tel: +33 (0)4 78 89 57 68

<http://www.pierreorsi.com/>

### **Philippe Gauvreau**

200, Avenue Casino 69890 La Tour de Salvagny

Tel: +33 (0)4 78 87 29 91

<http://www.pavillon-rotonde.com/>

### **Les rues / quartiers de « bouchons Lyonnais » typiques**

- « La Meunière » 11 rue Neuve - 69001 LYON (métro ligne A, arrêt « Cordeliers »)
- « Notre Maison » 2 rue de Gadagne - 69005 Lyon
- Rue Mercière (métro ligne A, arrêt « Cordeliers »)
- Quartier Saint-Jean (métro ligne A, arrêt « Vieux Lyon »)
- Rue de Marronniers (métro lignes A/D, arrêt « Bellecour »)

### **Les toques blanches Lyonnaises**

<http://www.visiterlyon.com/-Toques-Blanches-Lyonnaises-.html>

### **Pubs**

- Le Palais de la Bière 1 Rue Terme - 69001 Lyon
- Le Phoebus 22 Rue Pouteau - 69001 Lyon
- The Ayers Rock 2 Rue Désirée - 69001 Lyon
- Ninkasi Opéra 27 Rue de l'Arbre Sec - 69001 Lyon
- St James Pub 19 Rue Saint-Jean - 69005 Lyon
- The Fleming's 2 Rue de la Loge - 69005 Lyon
- The Smoking Dog 16 Rue Lainerie - 69005 Lyon
- The Johnny Walsh 56 Rue Saint Georges - 69005 Lyon
- The Melting Pub 9 Rue du Doyenné - 69005 Lyon

- The Antidote Pub 108 Rue Saint Georges - 69005 Lyon
- The James Joyce Pub 68 Rue Saint-Jean - 69005 Lyon
- Wallace 2 Rue Octavio Mey - 69005 Lyon

## **Programme de la conférence**

## Samedi 25 octobre 2014

	Amphi T7	Amphi T8	Amphi T9 Session Enseignement Recherche
09h00 - 09h30	<b>Accueil et petit déjeuner</b>		
09h30 - 09h50	BDD avec Behave Christophe Brun	SIG et Python : Love Story Vincent Picavet, Vincent Mora	Sage, un logiciel libre de mathématiques. Thierry Dumont
09h50 - 10h00			Pattes de mouche: une modélisation de biophysique avec python scientifique Guillaume Gay
10h00 - 10h10	PostgreSQL et Python, un beau mariage. Stéphane Wirtel	Reactive : Document + Event + Types Lionel Barret de Nazaris	Python dans la recherche, l'enseignement, et l'innovation en biotechnologie. Marc-André Delsuc
10h10 - 10h30			
10h30 - 11h00	<b>Pause</b>		
11h00 - 11h20	Un serveur fiable avec python 3.4 Martin Richard	Wagtail - why we built another CMS Tom Dyson	Retour d'expérience sur TOMUSS Thierry Excoffier
11h20 - 11h30			De la gestion de clusters à l'examen in silico des processeurs : la polyvalence de Python Emmanuel Quemener
11h30 - 11h40	Why SCons is not slow Dirk Bächle	Performance des frameworks web : Python vs the world Ronan Amicel	IRPF90 : a Fortran code generator for HPC Anthony Scemama
11h40 - 12h00			
12h00 - 12h30	CubicWeb - Vos données ont du sens Christophe de Vienne	DEPOT, story of a file.write() gone wrong Alessandro Molina	Ipython Notebook pour l'enseignement Marc Buffat
12h30 - 13h30	<b>Pause</b>		BoF retour expérience enseignement Python Olivier Berger
13h30 - 14h00	<b>Pause</b>		
14h00 - 14h20	Je teste mon code avec py.test Ronan Amicel	Modoboa, le mail avec du python dedans Antoine Nguyen	Python et les supernovae Yannick Copin
14h20 - 14h30			Du python qui ne manque pas d'air Romaric David, Bernard BARROIS
14h30 - 14h40	La gestion de version, ce problème tellement simple ... Pierre-Yves David	Anitya, tenez vous au courant des nouvelles sorties ! Pierre-Yves Chibon	Mathématiques expérimentales : Sage pour la combinatoire Viviane Pons
14h40 - 15h00			
15h00 - 15h30	RedBaron, une approche bottom-up au refactoring en Python Laurent Peuch	How The Oslo Program Reduces The Technical Debt Julien Danjou	Visualisation avec python Nicolas Rougier
15h30 - 16h00	<b>Pause</b>		
16h00 - 16h20	Let's do some docker Florent Pigout	Les docstrings pour bien documenter son projet. Adel Daouzli	SymPy une bibliothèque pure Python pour le calcul symbolique Kamel Ibn Aziz Derouiche
16h20 - 16h30			Retour d'expérience : l'école d'été python scientifique Nelle Varoquaux
16h30 - 16h40	Xbus Florent Aide	Ansible, au delà du dome de YAML Michael Scherer	Quelques idées d'optimisation d'un code numérique écrit en Python Loic Gouarin
16h40 - 17h00			

	Amphi T7	Amphi T8	Amphi T9 Session Enseignement Recherche
17h00 - 17h30	Radicale : Serveur de calendriers en Python, simple, libre et pragmatique Guillaume Ayoub	Comment le packaging m'a simplifié la vie Xavier Ordoquy	Poppy - la plateforme robotique humanoïde qui s'adresse à tous Ribas Stephane
17h30 - 18h30	<b>Posters</b>		
18h30 - 19h00			

## Dimanche 26 octobre 2014

	Amphi T7	Amphi T8
09h30 - 10h00	Présentation de Nao Nao Sur-Seine	Une interface expérimentale pour lier OpenGL à Python Fabrice Salvaire
10h00 - 10h30	Obscurcissement de bytecode Python Nicolas Szlifierski	APIs hypermédias Olivier Hervieu
10h30 - 11h00	<b>Pause</b>	
11h00 - 11h30	Asyncio: offrez des tulipes à vos entrées sorties asynchrones Thierry Chappuis	Grimper en haut de la Pyramid Gaston TJEJBES
11h30 - 12h00	Exploration de la boucle d'événement asyncio Victor Stinner	Pyramid, OpenERP, des métaclasses et encore plus ! Houzéfa Abbasbhay
12h00 - 12h30	Gunicorn, le problème du thundering herd et autres challenges liés à la concurrence Benoit Chesneau	Brood, une application web par des gens qui n'y connaissent pas grand-chose David Cournapeau
12h30 - 14h00	<b>Assemblée générale de l'AFPy</b>	<b>Pause</b>
14h00 - 14h30	<b>Pause</b>	
14h30 - 15h00	Bootstrapping Machine Learning Louis Dorard	Brython (Python dans le navigateur) Pierre Quentel
15h00 - 15h30	Kivy - a very short introduction Gabriel Pettier	Lean data science, or how to do machine learning with what you have to hand Christophe Bourguignat
15h30 - 16h00	<b>Pause</b>	
16h00 - 17h00	<b>Lightning talks</b>	
17h00 - 17h30	Ce que je n'aime pas dans Python Sébastien Douche	

# Sprints

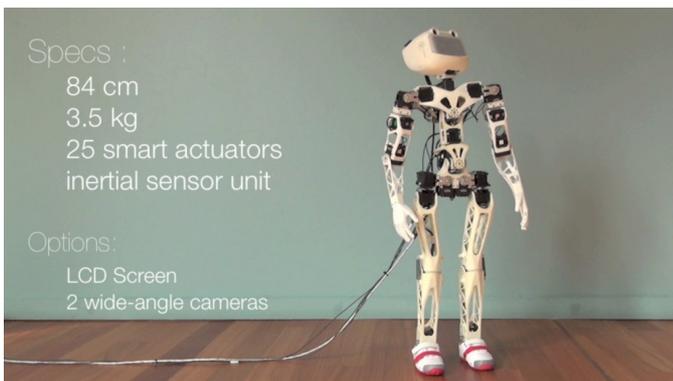
PyConFr 2014 est aussi l'occasion de travailler ensemble sur des projets en Python.

Cette année, les sprints auront lieu les 27 et 28 octobre. Ils auront lieu au rez de chaussée du bâtiment Nautibus.

Le principe des sprints est de se regrouper autour d'un projet (libre ou pas...) pour travailler dessus. On peut proposer les projets en avance (ou le faire à la dernière minute). L'important est de partager un maximum avec les autres, surtout ceux avec qui vous n'avez pas l'habitude de travailler.

Les projets déjà proposés :

- Sylvain Le Gal propose de travailler sur **OpenUpgrade**, un projet libre pour réaliser ses mises à jour OpenERP/odoo
- Akretion propose de travailler sur les projets de l'**Odoo Community Association**
- Pierre Rouanet, Clément Moulin-Frier de l'INRIA et Patrick Guillaud proposent un sprint mardi 28 octobre sur **Poppy**, le **robot humanoïde Open Source** et viendront avec des **ergots-robots**



*Illustration 3: Robot humanoïde Poppy*



*Illustration 4: Ergo-Robots*

# Présentations et conférenciers

## Amphi T7 – Samedi 25 octobre

### BDD avec Behave

**Conférencier** : Christophe Brun

**Niveau de l'audience** : Novice

### Description

Mise en oeuvre de Behave (BDD) sur un projet web (django 1.7)

### Résumé

Behave est un outil de développement agile pour faire du BDD : Behavior Driven Development (Développement piloté par le comportement). Lors de la sessions, nous verrons : - une introduction au BDD - Présentation de Behave et exemples simples de scénarios - une mise en oeuvre avec Django 1.7

### Conférencier

Responsable d'un pôle de développement PHP, mais fait du python à titre perso et pro dans le cadre de dev d'outils (Zope, Blender, Django) comme : - analyse de log (avec lib mavlink) - outils console - outils de suivi.

## PostgreSQL et Python, un beau mariage

**Conférencier** : Stéphane Wirtel

**Niveau de l'audience** : Intermediare

### Description

Vous utilisez PostgreSQL, mais savez-vous que vous pouvez utiliser et étendre PostgreSQL avec Python et PL/Python?

### Résumé

En tant normal, nous utilisons PostgreSQL via psycopg2, mais dans cette présentation, nous verrons comment travailler avec PostgreSQL via SQLAlchemy et Peewee, ou comment réaliser des migrations avec Alembic. De plus, je vous ferai la visite guidée de PL/Python qui vous permettra d'étendre PostgreSQL en Python.

### Conférencier

Stéphane est un développeur utilisant Python depuis les années 2000, fort d'une expérience dans le développement d'Odoo (anciennement OpenERP), il est aime partager des connaissances avec vous.

Ses intérêts sont vastes car tout est intéressant dans notre domaine, les systèmes distribués, la compilation sont quelques exemples.

A la fois membre de l'AFPy, de la PSF et de l'EuroPython Society, il organise avec ses amis, le Python FOSDEM qui se déroule en février à Bruxelles.

## Un serveur fiable avec python 3.4

**Conférencier** : Martin Richard

**Niveau de l'audience** : Intermediare

### Description

De nombreuses fonctionnalités de python, certaines apparues dans la version 3.4, sont très pratiques pour écrire un serveur fiable et prêt pour être utilisé en production. Cette présentation propose un retour d'expérience sur la conception et le développement d'un serveur en tirant partie des fonctionnalités du langage et de sa bibliothèque standard, notamment asyncio.

### Résumé

Cette présentation est un retour d'expérience sur la conception et le développement d'un serveur basé sur asyncio et exploitant d'autres fonctionnalités de Python très pratiques.

Je présenterai une architecture en couches qui permet d'implémenter facilement un serveur capable de gérer de nombreuses connexions simultanées rapidement et efficacement. On verra plus en détail comment rendre l'ensemble scalable: où et comment optimiser son code, gérer efficacement la mémoire et se protéger contre les utilisateurs malveillants ou les attaques DOS (Deny of Service).

Python offre de nombreuses fonctionnalités très pratiques pour écrire un serveur, et on verra qu'il n'est pas toujours nécessaire d'écrire du code bas niveau pour être performant !

### Conférencier

J'ai passé plusieurs années à faire du développement web et de l'administration de serveurs. Aujourd'hui, je suis ingénieur système chez l'hébergeur alwaysdata. Je m'intéresse particulièrement à Linux, aux systèmes distribués et à la haute disponibilité. J'écris du code en Python, et parfois en Javascript ou en C. J'ai un page perso à l'adresse <http://marti.us>

## Why SCons is not slow

**Conférencier** : Dirk Bächle

**Niveau de l'audience** : Intermediate

### Description

This talk presents the results of analysing and profiling the runtime of SCons, a build system that is rumored to be slow. We found that this isn't true, and can tell you which trap we fell into...so that you possibly don't with your own Python application

### Résumé

SCons ([www.scons.org](http://www.scons.org)) is a build system, entirely written in Python. It offers cross-platform support for languages like C/C++, D and Fortran, and uses content-based change detection (MD5) over the traditional timestamp approach. Despite all these obvious advantages over the classic "make", many people reject SCons as a build system because it is said to get very slow in large projects (10000+ files). That's why the SCons team stepped up and closely investigated some of the benchmarks on the internet that helped to spread the bad rumors. The results we found by very intensive and large-array profilings, boil down to the following point: It's actually not SCons that is slow, but the underlying process spawning mechanism (fork) that's causing the problems. The major slowdown of the running time can affect any other application (Python, or even a plain C program) that spawns single processes sequentially, while the memory usage of the parent process grows.

Against this unwanted effect, we developed a simple wrapper which redirects the standard calls of the subprocess module to `posix_spawn()` directly. It's going to be integrated to our source base in the upcoming release (Posix systems only), and will reduce the runtime for clean builds by a significant amount.

With this presentation we'd like to make our results and findings publicly available to a broader audience. We'd also like to get in touch with users and projects that experienced similar problems and discuss our and their solutions with them.

### Notes

I have given another talk about SCons at the PyConDE 2013 in Cologne (<http://pyvideo.org/video/2409/scons-software-bauen-in-python>).

### Conférencier

Software developer at daytime...regular contributor, researcher and mentor (user mailing list) for the SCons project in the night.

## **CubicWeb - Vos données ont du sens**

**Conférencier** : Christophe de Vienne

**Niveau de l'audience** : Intermediare

### **Description**

CubicWeb est un framework pour le web des données qui permet la construction d'applications par assemblage de composants (appelés cubes) et s'appuie sur un langage de requête proche de SPARQL.

### **Résumé**

Cette présentation générale montrera la facilité de construction d'une application à partir d'un modèle de données et de composants existants.

Elle sera également l'occasion de voir que CubicWeb, avec l'intégration récente de pyramid et bootstrapcss, évolue et se rend plus accessible à la communauté des développeurs web en Python.

### **Conférencier**

Développeur Python, CTO & co-fondateur de UNLISH.

## Je teste mon code avec py.test

**Conférencier** : Ronan Amicel

**Niveau de l'audience** : Intermediare

### Description

py.test est un outil pour écrire des tests en Python. Cette session présentera les bases de py.test en mettant l'accent sur ses fonctionnalités les plus pratiques / puissantes / originales, et sur l'intérêt à passer à py.test si vous utilisez actuellement unittest ou nosetests.

### Résumé

L'utilisation de py.test (1) pour écrire et lancer les tests de son projet est devenue de plus en plus populaire ces derniers temps dans la communauté Python.

Cette session présentera les bases de py.test en mettant l'accent sur ses fonctionnalités les plus pratiques, les plus puissantes, ou les plus originales :

- les assertions légères (plus besoin de assertEqual !)
- les fixtures modulaires
- quelques options et plugins bien pratiques

Cette session s'adresse principalement aux développeurs de niveau intermédiaire, qui ont déjà utilisé le module standard unittest ou bien nosetests, et qui souhaitent aller plus loin dans leur pratique des tests, ou bien sont juste curieux de découvrir cet outil.

(1) <http://pytest.org/>

**Conférencier**

/

## La gestion de version, ce problème tellement simple

**Conférencier** : Pierre-Yves David

**Niveau de l'audience** : Novice

### Description

Tout le monde utilise un gestionnaire de version. D'ailleurs, ces outils n'ont pas l'air bien compliqué... Mais le diable est dans le détail. Lors de ce talk, nous tourneront votre curiosité vers certains aspects de la gestion de version bien plus complexe qu'ils n'y paraissent à première vue. Ces problèmes seront abordés de manière générale, sans lien fort avec un outil particulier.

### Résumé

De nos jours, tout le monde utilise un gestionnaire de version. D'ailleurs, ces outils n'ont pas l'air bien compliqué, il suffit d'enregistrer des fichiers, de les fusionner et de les restaurer. Facile non ?

Mais comme d'habitude le diable est dans le détail. Lors de ce talk, nous tourneront votre curiosité vers certains aspects de la gestion de version bien plus complexe qu'ils n'y paraissent à première vue. Ces Problèmes (et leur solution... quand elles existent) seront abordés de manière générale, sans lien fort avec un outil particulier.

Les sujets traités seront entre autre: La gestion de fichier multi plateforme (encodages, types, ...), les fusions (merges) et l'utilité des méta-données (date, utilisateur, permission)

### Notes

Le but de ce talk est simplement d'assouvir la curiosité de l'assistance en présentant les problèmes intéressant d'un domaine avec lequel chacun est familier mais peu sont intimes. Au programme: plusieurs discussions autour de l'encodage de caractère et ses implications cross plateforme, Des graphes de révisions rigolos qui rendent les merges compliqués et l'absence totale de fiabilité des méta-donnés comme les dates (pour l'ordre) ou les utilisateurs (Pour l'authentification) dans les systèmes distribués d'aujourd'hui.

### Conférencier

Après plusieurs années chez Logilab (cubiweb, pylint,...) à travailler, en python, sur le framework Cubicweb et le gestionnaire de version Mercurial, je suis maintenant employé par Facebook pour poursuivre mes travaux sur Mercurial. Je m'intéresse particulièrement aux problématiques de réécriture d'historique (rebase, amend, ...) et aux partages de celles çi.

# RedBaron, une approche bottom-up au refactoring en Python

**Conférencier** : Laurent Peuch

**Niveau de l'audience** : Intermediare

## Description

Écrire du code qui modifie du code reste réservé à une petite poignée de développeurs téméraires. Construit sur les fondations d'un arbre syntaxique (AST) sans perte, RedBaron a pour but de rendre le refactoring accessible au plus grand nombre, en offrant une abstraction qui permet de se concentrer uniquement sur le résultat voulu en faisant fi des détails peu importants.

## Résumé

RedBaron a pour but de rendre l'écriture de code qui modifie du code suffisamment facile (et fun) et amusante pour que cela devienne une tâche réaliste, voire même enviable. Pour ce faire, RedBaron se comporte comme BeautifulSoup/JQuery sur votre code source Python avec pour objectif d'être suffisamment puissant et expressif pour pouvoir être directement utilisé dans un shell tel que ipython.

RedBaron est construit sur Baron, un arbre syntaxique pour Python qui a la particularité de ne perdre aucune information, de telle sorte que l'opération: [code source -> arbre -> code source] donne un code source identique, y compris d'un point de vue mis en forme. Baron est relativement stable, comporte plus de 1000 tests unitaires, a été testé sur le top 100 des packages de pypi et est documenté. RedBaron est en court de développement mais est déjà stable et utilisable, le focus est sur le rendre agréable. Fin août, il comportait pas loin de 450 tests unitaires et est entièrement documenté avec de très nombreux exemples.

Cette présentation expliquera la raison d'être de ce projet en vous donnant une vue de l'ensemble des fonctionnalités avec chaque fois une petite démo.

Code source et documentation:

- RedBaron: <https://github.com/psycojoker/redbaron>, <http://redbaron.rtfld.org>
- Baron: <https://github.com/psycojoker/baron>, <http://baron.rtfld.org>

L'objectif de RedBaron est un changement de paradigme dans lequel écrire du code qui modifie du code devient une opération réaliste, normale (et amusante) de la vie d'un développeur.

## Conférencier

Bram, passionné du python, de django, d'haskell et du code en générale. Code la nuit, fais de l'associatif (FAI associatifs, FFDN, Gitoyen ...) et du militantisme (la quadrature du net, revenue de base ...) le jours quand il n'est pas dans un hackerspace.

## Let's do some docker

**Conférencier** : Florent Pigout

**Niveau de l'audience** : Intermediare

### Description

Présentation d'une pratique de développement python / django avec docker.

### Résumé

Docker devient de plus en plus incontournable dans le monde du développement. Effectivement l'outil offre de nouvelles perspectives pour les développeurs, QA et devOps.

Dans cette présentation, nous décrivons étape par étape ([aperçu][<http://tech.novapost.fr/docker-django-demo-en.html>]):

- quelques commandes docker utiles
- créer un container DB
- installer un projet django dans un container
- faire marcher le tout

Cet environnement de développement prêt à l'emploi, nous observerons les différents atouts de la démarche:

- reproductibilité
- scalabilité

Nous proposerons une pratique pour le versioning et le déploiement de nos containers. Pour finir nous listerons quelques 'use case', notamment [DotCi][<https://github.com/groupon/DotCi>], comme sources d'inspirations pour nos prochains travaux.

### Conférencier

Développeur python, django, open-source depuis 2008. Durant ces années j'ai eu l'occasion de pratiquer le python dans plusieurs domaines (db, front, script), et dans divers contextes (service, freelance, cdi).

Aujourd'hui, je travaille à Novapost.

Voici quelques uns de mes travaux:

- <https://github.com/toopy>
- <https://bitbucket.org/florent>

## **xbus**

**Conférenciers** : Florent Aide , Jeremie Gavrel

**Niveau de l'audience** : Intermediare

### **Description**

Python + Go + Docker = la recette pour faire parler vos applications

### **Résumé**

Présentation du projet Xbus, un bus d'évènement libre en Python et Go, packagé en Docker.

En urbanisation d'entreprise il est souvent difficile de débrancher une application legacy ou même de lancer une nouvelle application car il y a beaucoup de code intermédiaire, souvent ad-hoc à remplacer.

Xbus propose d'unifier les moyens de communication entre des applications qui n'ont pas vocation à se connaître mais que l'administrateur système veut faire discuter.

Il permet aussi d'avoir une vision centralisé de toutes les "interfaces" de votre infrastructure.

### **Notes**

Après 15 ans à patauger dans les égouts des infrastructures IT des grandes entreprises, j'ai pu constater que les logiciels propriétaires étaient encore bien accrochés dans le segment des ESB (entreprise service bus) et des outils de supervision (autres que pure infra). Il était temps de changer la donne.

Xbus est une réécriture from scratch d'un outil développé (en python) pendant 7 ans chez un très gros client.

Il s'agit donc de présenter un nouveau projet OpenSource mais aussi de partager un retour d'expérience sur l'urbanisation IT dans l'entreprise.

### **Conférencier**

/

## **Radicale : Serveur de calendriers en Python, simple, libre et pragmatique**

**Conférencier** : Guillaume Ayoub

**Niveau de l'audience** : Novice

### **Description**

Radicale est né il y a 6 ans. D'abord simple serveur de calendriers écrit pour un exercice scolaire, il s'est petit à petit transformé en projet libre un peu plus important. Malgré la volonté de garder ses idées originales de simplicité et de pragmatisme, il a fallu faire face aux évolutions importantes du langage et de ses interpréteurs (Python 3.x, Pypy), à l'ajout de fonctionnalités (Carnet d'adresses, support de clients) et l'arrivée de contributeurs plus ou moins sérieux.

Le but de cette présentation est, au-delà d'une rapide présentation des fonctionnalités de l'outil, de partager l'évolution d'un projet écrit en Python et de son rapport au langage, aux utilisateurs, aux contributeurs et à ses idées fondatrices.

### **Conférencier**

Je suis cogérant de Kozea, une société lyonnaise d'ingénierie informatique implantée particulièrement dans le monde de la santé et active dans le développement de logiciels libres. Je développe très majoritairement en Python, sur des projets touchant le Web de près ou de loin.

# Amphi T8 – Samedi 25 octobre

## SIG et Python : Love Story

**Conférenciers** : Vincent Picavet , Vincent Mora

**Niveau de l'audience** : Intermediare

### Description

Les systèmes d'information géographiques n'ont cessé de progresser dernièrement. Python connaît un remarquable succès pour ce qui est des SIG. Cette présentation donne quelques éléments d'explication de l'engouement des SIGistes pour Python. Puis on fera un aperçu de l'étendue fonctionnelle des outils Python pour le géospatial, avec quelques exemples concrets d'utilisation.

### Résumé

Loin de la télé-réalité, le domaine des systèmes d'information géographiques n'a cessé d'évoluer et de prendre de l'importance ces dernières années, et la valorisation de la dimension spatiale des données touche tous les domaines. L'essor des solutions opensource pour ces technologies est également très rapide.

Les solutions, produits, frameworks, bibliothèques destinées aux SIG sont multiples et sont implémentées dans divers langages.

Cependant, Python connaît un remarquable succès pour ce qui est des systèmes d'informations géographiques, et on ne compte plus les Pythonistas dans les conférences dédiées au spatial.

Après une rapide introduction aux SIG, cette présentation essaie tout d'abord de donner quelques éléments d'explication de l'engouement des SIGistes pour Python. Puis on donnera un aperçu de l'étendue fonctionnelle des outils Python pour le géospatial, avant de montrer quelques exemples concrets d'utilisation.

### Conférencier

Vincent Picavet est ingénieur en mathématiques appliquées. Après quelques années passées à travailler en imagerie satellitaire, il se consacre aux SIG, concevant et implémentant des infrastructures de données spatiales avec des outils opensource. Il crée Oslandia en 2009 avec Olivier Courtin, fournissant des services en SIG opensource. Oslandia est grande utilisatrice de Python pour nombre des composants qu'elle utilise et qu'elle développe.

## Reactive : Document + Event + Types

**Conférencier** : Lionel Barret de Nazaris

**Niveau de l'audience** : Intermediare

### Description

Le Reactive Programming (RP) est un nouveau type d'architecture qui permet de gérer la complexité liée à de très nombreuses interactions avec un utilisateur. Mais les articles disponibles sont souvent abstraits. Au contraire ce talk se veut pragmatique et présente quand utiliser le RP via les besoins d'une application (un modeler 3d) et les concepts de base via une implémentation spécifique

### Résumé

Organiser une application vaste est toujours compliqué, maîtriser la complexité induite aussi. Pour cela, MVC est sans doute le mode d'architecture logiciel le plus connu mais il montre des faiblesses dans le cas d'interactions complexes. Le Reactive Programming est un nouveau type d'architecture qui adresse ce problème. Il se concentre sur la gestion d'événement comme principe fondateur d'une application.

L'organisation autour des événements permet de gérer naturellement tous les aspects d'une application : chargement&sauvegarde, interface utilisateur, undo/redo, etc.

C'est une voie prometteuse mais les présentations sur le sujet sont souvent trop abstraites, entourées de hype et de jargon, et le passage à la pratique semble difficile. Aussi ce talk va s'appuyer sur le cas concret de deux applications pour à la fois illustrer les besoins que satisfait le Reactive Programming, montrer un type d'implémentation et enfin présenter les limites de cette approche.

Le plan est donc le suivant :

- présenter une application changeantes et complexe
- montrer ses besoins en terme d'interaction et les limites du MVC
- démontrer une implémentation qui répond à ses besoins et les concepts sous-jacents
- appliquer la solution à une application de démo.
- ne pas oublier qu'il n'existe pas de balle en argent (no silver bullet)
- discuter (cad vos questions)

### Conférencier

Lionel Barret de Nazaris est un pythonnier enthousiaste depuis une dizaine d'année. Il a écrit en python des (éléments de) jeux videos AAA, des modeleurs 3D, des logiciels de téléphonie. Il est magicien logiciel (cad consultant) à LBdN Consulting, et intervenant à l'Université de Clermont Ferrand et à Gamagora.

## Wagtail - why we built another CMS

**Conférencier** : Tom Dyson

**Niveau de l'audience** : Intermediate

### Description

Wagtail was launched as an open source project in February 2014. In this talk, Tom Dyson will explain why his agency built a new Django CMS, describe what has happened since launch, and outline the plans for Wagtail's future.

### Résumé

Torchbox launched Wagtail as an open source project in February 2014. This talk covers the past, present and future of the project:

1. Why did we build a new Django CMS, when mature and popular tools already exist? Tom will compare Wagtail with some of the alternatives, both from within the Django world and elsewhere.
2. What have we learned since the initial release? We had a successful launch - a carefully-placed story on Hacker News meant Wagtail was one of Github's top trending projects for a week - but we've discovered that it's hard to maintain steady momentum on a popular open source project. In this section Tom will outline how we've prioritised features and managed development.
3. What's next? Wagtail has quickly become a viable option among the range of Django CMSs, but our ambition is for it to become a class-leading open source CMS. Tom will describe the general direction for Wagtail and some of the specific new features which form part of this ambition.

### Conférencier

Tom Dyson is the co-founder and technical director of Torchbox, a UK web agency focused on helping organisations who make the world a better place. An enthusiastic Django supporter since version 0.9.7, Tom published the framework's first screencast, and hired Django luminaries including Simon Willison and Andrew Godwin. Under Tom's technical direction, Torchbox has delivered Django projects for government, UN agencies, NGOs and world-famous universities. In February 2014 Torchbox released Wagtail, an open source Django CMS.

Outside Torchbox, Tom plays piano, rides his bike and dreams of moving to France.

# Performance des frameworks web : Python vs the world

**Conférencier** : Ronan Amicel

**Niveau de l'audience** : Intermediate

## Description

TechEmpower réalise régulièrement des tests de performance sur de nombreux frameworks web, tous langages confondus. Que nous apprennent-ils sur la performance de Python par rapport aux autres langages ? Et entre les différents frameworks Python ? Python 3 est-il plus rapide que Python 2 ? Et PyPy ? Et ces benchmarks, est-ce qu'ils servent vraiment à quelque chose ?

## Résumé

Depuis mars 2013, les gens de TechEmpower, aidés par la communauté, ont lancé le plus grand effort de mesure et de comparaison des performances de frameworks web : <http://www.techempower.com/benchmarks/>

Dans la dernière fournée de résultats (round 9), ce sont 93 combinaisons de langages, frameworks et bases de données qui ont été comparées, sur 6 tâches et sur 3 environnements matériels différents. Que nous apprennent les résultats ?

- Comment Python se place-t-il par rapport aux autres langages (PHP, Ruby, Go, Java...) ?
- Y a-t-il des écarts significatifs entre nos différents frameworks (Django, Flask, Bottle, Pyramid, Tornado...) ?
- Python 3 est-il plus rapide ou plus lent que Python 2 ? Est ce qu'on gagne beaucoup à passer à PyPy ?
- Ces benchmarks sont-ils vraiment représentatifs d'une utilisation réelle ?

Telles sont les questions qui seront abordées dans cette session, garantie sans trolls !

**Conférencier**

/

## DEPOT, story of a file.write() gone wrong

**Conférencier** : Alessandro Molina

**Niveau de l'audience** : Intermediate

### Description

DEPOT (<http://depot.readthedocs.org/en/latest/>) is a file storage framework born from a project that stored a lot of files on disk, until the day the customer system engineering team decided to switch to Heroku, which didn't support storing files on disk.

### Résumé

The DEPOT talk will cover the facets of a feature ("saving files") which has always been considered straightforward but that can become complex in the era of cloud deployment and when infrastructure migration is involved.

After exposing the major drawbacks and issues that big projects might face on short and long terms with file storage the talk will introduce DEPOT and how it tried to solve most of the issues while providing a super-easy-to-use interface for developers. We will see how to use DEPOT to provide attachments on SQLAlchemy or MongoDB and how to handle problems like migration to a different storage backend and long term evolution.

Like SQLAlchemy makes possible to switch your storage on the fly without touching code, DEPOT aims at making so possible for files and even use multiple different storages together.

### Notes

Talk will be in English

### Conférencier

Python developer since 2001 has always been interested in Python as a Web Development Platform. Currently a core developer of the TurboGears2 framework in charge of the Python3 support branch and the whole 2.3 series. Has contributed to various projects related to web development like FormEncode, ToscaWidgets, Sprox Widget Generation Library and the Ming MongoDB ORM

## Modoboa, le mail avec du python dedans

**Conférencier** : Antoine Nguyen

**Niveau de l'audience** : Intermediate

### Description

Une présentation de Modoboa, frontend web écrit en python/django permettant de gérer et d'utiliser une plate-forme de messagerie.

### Résumé

Installer et gérer une plate-forme de messagerie (un serveur de mail) demande du temps et des connaissances techniques précises. Pourtant, certains événements récents nous rappellent que confier cette tâche à un prestataire ne garantit pas nécessairement la confidentialité de nos données. Modoboa tente de lever cette barrière en proposant une interface simple et intuitive.

Cette conférence présente : L'historique de Modoboa et pourquoi Python ; Les fonctionnalités existantes (côté administrateur et utilisateur) ; Les fonctionnalités à venir ainsi que l'intégration avec l'écosystème mail/python (ex: Radicale) ; Les possibilités d'évolution.

### Conférencier

I'm a french IT profesional (1) living near Paris and working as a freelancer.

I'm using Python for about 10 years now and I'm the creator of Modoboa (2), a web frontend to manage and use an email hosting platform. It is written using Python and Django.

(1) <http://tonio.ngyn.org/about/>

(2) <http://modoboa.org/>

## Anitya, tenez vous au courant des nouvelles sorties !

**Conférencier** : Pierre-Yves Chibon

**Niveau de l'audience** : Intermediare

### Description

Que vous soyez développeur ou utilisateur se tenir au courant des sorties de nouvelles versions peut être un enjeu ou simplement amusant. Anitya offre à n'importe qui avec un compte OpenID la possibilité d'ajouter/ éditer un projet. Chaque jour, anitya surveillera et annoncera toute nouvelle version trouvée. Pour ce faire, python et ZeroMQ sont utilisés.

### Résumé

Se tenir au courant de la publication d'une nouvelle version peut avoir des conséquences importantes et parfois inattendues. En tant que développeur il est donc important de se tenir au courant des mises à jours de bibliothèques que nous utilisons.

En tant que développeur d'une bibliothèque il est aussi important de faire savoir quand une nouvelle version est sortie. Elle corrige peut être des bogues importants (voire de sécurité) ou ajoute peut être des fonctionnalités intéressantes. Et puis, il est quand même toujours pénible de voir de nouveaux rapports de bogues alors que la version réglant ce problème vient de sortir.

Anitya a pour vocation de résoudre ces problèmes. Son but est de fournir un site, central, qui ne soit pas spécifique à un système d'exploitation particulier, et sur lequel n'importe qui puisse ajouter, éditer ou simplement chercher quelle est la dernière version d'un projet. Basé sur une structure de plugins, anitya est capable de trouver la dernière version d'un logiciel sur les plateformes majeures d'hébergement (github, sourceforge, pypi...). Chaque jour, anitya va chercher pour chaque projet sa dernière version et s'il y a une nouvelle version anitya l'annonce sur un bus de messages de type fedmsg.

Vous ne me croyez pas ? Regardez donc vous même <http://release-monitoring.org>

(1) <http://fedmsg.com/>

**Conférencier**

/

## How The Oslo Program Reduces The Technical Debt

**Conférencier** : Julien Danjou

**Niveau de l'audience** : Intermediate

### **Description**

Oslo is a program within OpenStack aiming at merging common code and maintaining it in the long term

### **Résumé**

Oslo is a program within OpenStack aiming at merging common code and maintaining it in the long term. In this talk, we'll see how we managed to reduce the technical debt within OpenStack using this program, how we provided consistent libraries and API for all project and what strategies we used to do that.

### **Conférencier**

I'm a Free Software hacker, working as a senior software engineer at eNovance (now Red Hat).

I've been working on OpenStack development for the last 3 years, and I'm the author of The Hacker's Guide to Python published in 2014.

# Les docstrings pour bien documenter son projet

**Conférencier** : Adel Daouzli

**Niveau de l'audience** : Novice

## Description

En Python, les fonctions, méthodes et classes peuvent être documentées en utilisant des docstrings. Python peut ainsi générer par introspection une documentation interne. Également des outils tel que Sphinx peuvent générer une documentation dans divers formats (HTML, PDF,..). Cependant il existe différents formats (reST, Epydoc, Google,..), alors que choisir, et comment changer de format (Pymment)?

## Résumé

En Python, les fonctions, méthodes et classes peuvent être documentées en utilisant des docstrings.

Python peut ainsi générer par introspection une documentation interne dynamique. Les docstrings peuvent alors être accédées via l'attribut magique `__doc__` ou via la fonction `help()` de l'interpréteur.

Par ailleurs, des outils tel que Sphinx peuvent générer une documentation externe dans divers formats de sortie (LaTeX, HTML, PDF,..).

Cependant il existe différents formats de docstrings (reStructuredText, Epytext, Google, Numpydoc,...).

Dans cette présentation nous allons passer en revue les principaux formats de docstrings avec leurs syntaxes. Nous verrons également les outils permettant de générer une documentation externe. Enfin nous nous arrêterons sur Pymment un outils permettant de générer automatiquement des squelettes de docstrings dans un format donné sur un fichier ou un module Python. Il permet aussi de convertir d'un format vers l'autre.

## Conférencier

Docteur en sciences neuromorphiques, je suis actuellement ingénieur en systèmes embarqués, je suis co-fondateur du hackerspace Lyonnais le LOL (Laboratoire Ouvert Lyonnais) (1), j'utilise Python depuis près de 10 ans pour différents usages.

(1) <http://www.lyonopenlab.org/>

Mon travail avec Python

Professionnellement:

- J'ai développé un outil de validation de compilateurs.
- J'ai développé un ensemble de tests utilisant `py.test` pour tester un système embarqué BSP Linux et une suite d'applications fonctionnant dessus.

Contributions au libre:

- Je contribue quand j'ai du temps aux outils de tests en Python de Mozilla.
- Je développe Pymment (1) pour créer et convertir des docstrings.
- Je développe le LOLOpener (2) un système informant sur l'ouverture du LOL.

- J'ai développé un petit serveur codelauncher (3) permettant d'exécuter rapidement du code C ou Python dans un navigateur web.

(1) <https://github.com/dadadel/pyment>

(2) <https://github.com/dadadel/LOLOpener>

(3) <https://github.com/dadadel/codelauncher>

Projets persos:

- Je me suis développé une sorte de babyphone avec un côté serveur Python qui gère l'audio et l'envoi sur un serveur Internet, et de l'autre côté une application Android pour être informé que bébé pleure.
- J'ai développé différents outils et serveur en Python sur Raspberry Pi servers (Django, Flask, WSGI,...) pour gérer ma collection d'ebooks (ebooks (1)), pour jouer de la musique après récupération sur Internet, gérer des vidéos en streaming en local,...

(1) <https://github.com/dadadel/ebooks>

## **Ansible, au delà du dôme de YAML**

**Conférencier** : Michael Scherer

**Niveau de l'audience** : Intermediare

### **Description**

Ansible est un outil de gestion de configuration et d'orchestration, qui fonctionne avec des descriptions en YAML. Néanmoins, il est possible d'étendre l'outil grâce à python, en s'intégrant via des modules personnalisés, des plugins ou via l'API Python permettant d'intégrer Ansible dans un script.

### **Résumé**

La présentation va d'abord poser les bases d'Ansible et de son modèle de fonctionnement basique. Puis à travers divers exemples tirés des problématiques de mon ancien poste d'administrateur d'une infrastructure de taille conséquente, nous verrons comment étendre Ansible via l'écriture d'un module et via les différents types de plugins qu'on peut tirer. Enfin, nous terminerons par l'intégration d'Ansible dans un script python plus classique pour aller au delà de YAML.

### **Notes**

J'ai aussi fait une proposition de ce talk pour pycon.us. J'ai contribué à Ansible, j'ai déjà fait plusieurs présentations sur ansible et je m'en sert régulièrement dans le cadre de mon travail pour gérer des infrastructures de projets libres.

### **Conférencier**

Sysadmin en charge des projets communautaires chez Red hat, codeur python/perl/ruby occasionnel et packager chez Fedora.

## Comment le packaging m'a simplifié la vie

**Conférencier** : Xavier Ordoquy

**Niveau de l'audience** : Intermediare

### **Description**

Cette présentation discute de l'intérêt pratique de packager du code Python ainsi que les bonnes pratiques associées.

### **Résumé**

Après une présentation rapide de ce qu'est le packaging Python, je présenterai un retour d'expérience sur les efforts de packaging portant sur différents projets (opensource ou privés).

Par la suite, nous regarderons un cas pratique de packaging d'un projet complexe et comment le packaging à rendu le projet simple à utiliser.

**Conférencier**

# Amphi T9 - Samedi 25 octobre - Session enseignement – recherche

## Sage, un logiciel libre de mathématiques

**Conférencier** : Thierry Dumont

**Niveau de l'audience** : Intermediare

### Description

Sage se veut une alternative libre à Matlab, Maple et Mathematica. Il combine la puissance de nombreux programmes libres dans une interface commune basée sur Python.

### Résumé

Sage utilise Python pour interfacier de nombreuses bibliothèques existantes dans tous les domaines du Calcul ("computer algebra", méthodes numériques, calculs dans divers ensembles etc.).

En utilisant de manière intelligente l'orientation objet et la généricité de Python, on a pu rendre transparente l'utilisation des différents outils de calcul fédérés dans Sage. Ceci démontre l'extrême souplesse et la puissance de Python.

Pour l'utilisateur, qui peut-être un étudiant débutant, un lycéen ou un chercheur, les avantages sont nombreux: outre la liberté et l'accessibilité des sources, le caractère typé du langage rend les calculs beaucoup plus sûrs. Enfin, les performances s'avèrent meilleurs que celles des logiciels commerciaux, la plupart du temps. Les avantages pédagogiques sont aussi nombreux: en apprenant Sage, on apprend Python.

Python permettant de tout relier, Sage est aussi un serveur web: le notebook, qui a été le précurseur du notebook de ipython, a la manière standard d'interagir avec le logiciel.

Sage: <http://www.sagemath.org/fr/>

### Conférencier

Thierry Dumont, Ingénieur de Recherches à l'Institut Camille Jordan (mathématiques), CNRS et Universités de Lyon et St. Etienne.

# Pattes de mouche: une modélisation de biophysique avec python scientifique

**Conférencier** : Guillaume Gay

**Niveau de l'audience** : Novice

## Description

Je présenterai un projet en collaboration avec un laboratoire de recherche en biologie dans lequel nous avons utilisé les outils python scientifique pour développer la simulation d'un phénomène biologique

## Résumé

Dans cette présentation, je montrerai comment l'écosystème python scientifique a permis de développer un modèle biophysique de la morphogénèse de la patte de la mouche. Une présentation du code peut être parcourue ici : <http://nbviewer.ipython.org/DamCB/leg-joint/notebooks>

Je présenterai d'abord le problème biologique: comment la mort cellulaire programmée (a.k.a l'apoptose) est impliquée dans la formation des organes de la mouche, et pourquoi ça intéresse les biologistes, puis je détaillerai un peu la stratégie de développement des simulations, en insistant sur quelques outils centraux (notamment la bibliothèque graph-tool (1)), suivie de quelques perspectives, principalement concernant l'intégration de la visualisation au logiciel Blender.

(1) <http://graph-tool.skewed.de/>

## Conférencier

I am currently working as an independent researcher and software developer in physical biology and cell biophysics.

I hold a PhD in physics, and about 7 years of work with biologist, building microscopes and developing software to help them analyse data and create models. I started using the scientific python stack around 2007, and nether regret it.

You can see a bit more on my website, <http://damcb.com/>, and of course on github, <https://github.com/glyg>.

# Python dans la recherche, l'enseignement, et l'innovation en biotechnologie

**Conférencier** : Marc-André Delsuc

**Niveau de l'audience** : Novice

## Description

Dans le cadre de l'équipe de recherche en biophysique et biologie structurale, de nombreux outils Python sont développés. Le même cadre, et souvent les mêmes codes, sont utilisés en initiation informatique aux élèves-ingénieurs en biotechnologie. De même une spin-off du laboratoire, travaille dans le domaine des "big-Data" scientifique, en utilisant des outils et des méthodes communes

## Résumé

Dans notre équipe de recherche, nous utilisons quasi exclusivement Python/numpy/scipy pour l'analyse de données et la modélisation en biologie structurale et en biophysique. La possibilité de passer le même code du laptop au cluster régional est inappréciable, la souplesse de l'environnement est vitale pour des non-informaticiens de métiers, et l'efficacité est cruciale pour les flux de données que nous gérons.

Dans le cadre de l'Ecole de Biotechnologie de Strasbourg ESBS (1), la même équipe propose à tous les élèves entrant en 1ere année, une initiation à la programmation. Souvent totalement débutants, nous leur présentons les rudiments de l'algorithmique et de la structuration de données. L'environnement iPython notebook, proche de leur habitudes facilite beaucoup les choses.

Le même environnement est finalement utilisé dans les développements réalisés dans le cadre d'une société spin-off adossée au laboratoire, et qui fournit des méthodes algorithmiques et mathématiques pour l'analyse de données en chimie analytique et en biophysique.

Je présenterai les exemples de codes développés, et les avantages et difficultés rencontrés dans ces différentes approches.

## Notes

la spin-off société CASC4DE (2) est créée mais encore très jeune, et la partie communication n'est pas du tout en place (comme vous pouvez le voir).

Cette société remplace la société NMRTEC (3), créée il y a 15 ans par la même équipe, qui continue encore cependant son activité. Cette dernière utilise aussi intensivement le python dans ses développements interne, externe. Le logiciel NMRNoteBook, largement diffusé, est articulé autour de jython (conception du produit il y a ~10 ans).

Les choses vont très vite, je présenterai plus ou moins d'exemples provenant de la première ou de la deuxième société, suivant l'état d'avancement du projet CASC4DE.

## Conférencier

As a CNRS research director I work at the IGBMC Institute (3) in the Biomolecular NMR group. There we use Nuclear Magnetic Resonance to decipher the way the large biological molecules interact intimately to produce all the wonderful patterns observed in life.

We have created many pieces of code, mostly for the processing and analysis of spectroscopic data in NMR and in Mass Spectrometry. These codes are used worldwide by other research groups.

As a funder of the NMRTEC company, I supervise the software development there, mostly the same ideas, but implemented in a manner compatible with commercial development.

As a teacher in ESBS (5) biotechnology school, I present the fundamental biophysical laws, and show how computer simulation is a way of understanding complex systems.

For all these tasks the python language, thanks to its versatility is central to my activity.

(1) <http://www-esbs.u-strasbg.fr/>

(2) <http://www.casc4de.eu/>

(3) <http://www.igbmc.fr/>

(4) <http://www.nmrtec.com/>

(5) <http://www-esbs.u-strasbg.fr/>

## Retour d'expérience sur TOMUSS

**Conférencier** : Thierry Excoffier

**Niveau de l'audience** : Novice

### Description

TOMUSS est un tableur web permettant le suivi des étudiants. L'exposé présente les avantages d'avoir utilisé Python pour programmer cette application.

### Résumé

TOMUSS : The Online Multi User Simple Spreadsheet.

Présentation rapide de l'application (1) et de ses contraintes.

C'est une application en production depuis 2008 à la faculté des sciences et technologies de l'université Lyon 1 et utilisée par 700 enseignants et 7000 étudiants

Les choix techniques qui ont été permis par l'utilisation de Python :

- Faire le maximum de plugin.
- Faire du Monkey Patching.
- Ne pas utiliser de base de données mais stocker les données comme des programmes Python
- Des conseils sur l'utilisation des threads
- Les outils de debugage développés.

Ensuite, on présente ce qui aurait été fait différemment si c'était à refaire :

- Utiliser un traducteur Python vers JavaScript.
- Créer ou utiliser un framework de synchronisation automatique entre les objets coté server en Python et coté client en Javascript
- Utiliser Memcached pour gérer les caches

(1) <https://www.projet-plume.org/relier/tomuss>

### Conférencier

Enseignant en informatique à l'université de Lyon 1.

# De la gestion de clusters à l'examen in silico des processeurs : la polyvalence de Python

**Conférencier** : Emmanuel Quemener

**Niveau de l'audience** : Intermediare

## Description

"Python est à l'informatique scientifique ce que les mathématiques sont à la physique" : provocateur comme comparaison, mais nous le montrerons dans deux exemples issus du calcul scientifique : Python comme langage transformant des noeuds en architecture de stockage distribué, Python comme outil d'investigation pour explorer les limites des matériels d'aujourd'hui, notamment les GPU faces au CPU.

## Résumé

L'informatique scientifique doit faire face à deux enjeux majeurs dans un contexte paradoxal : l'explosion des volumes de données à traiter d'une part, et la multiplication des unités de traitement permettant de les traiter d'autre part. Le paradoxe, c'est que les programmes associés ont souvent plusieurs dizaines d'années, sont éprouvés, s'enchaînent les uns les autres mais que les composants matériels pour les exécuter n'ont, eux, une durée de vie n'excédant pas quelques années ! Comment rapidement, évaluer la performance de ces nouvelles infrastructures pour ensuite mettre à disposition ces vieux outils de manière efficace ? Notre approche s'inspire de ce que l'aéronautique a développé depuis un siècle : l'exploration de "domaines de vol", ou la délimitation du "là où ça marche". C'est là, dans le rapide prototypage de tests d'évaluation que Python s'avère très précieux et nous illustrerons son exploitation sur deux cas particuliers. Le premier concerne le stockage : pour que le stockage soit efficace, il faut désormais le distribuer. Nous verrons comment, en quelques lignes, Python, utilisé comme "liant", permet de déployer une infrastructure de stockage distribué en quelques lignes puis d'en évaluer la performance, et ce en ne s'appuyant que sur des composants externes : Python révèle sa puissance de langage de script. Le second concerne l'exploitation des circuits graphiques, ces fameux GPU, comme accélérateurs de calcul : nous montrerons comment Python peut permettre la programmation expresse de ces composants, leur comparaison avec les processeurs traditionnels mais surtout dans quelles conditions leur usage est optimal. Ces deux cas d'usage révéleront chacun leur lot de surprises. Notre conclusion offrira un constat nuancé mais finalement rassurant : sans une connaissance approfondie de son infrastructure matérielle, sans une maîtrise parfaite de l'environnement logiciel et son optimisation application par application, il est bien difficile de les exploiter efficacement. En résumé, "sans la maîtrise, la puissance n'est rien".

## Conférencier

Emmanuel Quemener définit la partie la plus "sexy" de son travail comme celle d'un "pilote d'essais informatique". Ingénieur de Recherches au Centre Blaise Pascal de l'Ecole Normale Supérieure de Lyon, la maison de la modélisation lyonnaise, il est au service de celles et ceux pour qui l'outil informatique, dans la recherche, est devenu incontournable. Son activité gravite autour de la création et l'administration de plateaux techniques dédiés à tous les aspects de l'informatique scientifique (du réseau à l'intégration de processus complets de calcul scientifique), le stockage & le calcul distribué, le calcul sur accélérateurs, la librification de codes et le "transfert technologique". C'est un fervent développeur, défenseur et promoteur d'une approche SIDUS (1) (pour "Single Instance Distributing Universal System") pour simplifier l'administration de groupes de machines ou réaliser des expérimentations sur des recherches en reproductibilité. Enfin, il exploite quotidiennement Python exactement comme un physicien utilise les

mathématiques : un outil aussi puissant que polyvalent.

(1) <http://www.cbp.ens-lyon.fr/sidus/>

## IRPF90 : a Fortran code generator for HPC

**Conférencier** : Anthony Scemama

**Niveau de l'audience** : Intermediate

### Description

IRPF90 is a Fortran code generator written in Python. It aims at simplifying the collaborative development of large Fortran programs.

### Résumé

IRPF90 est un générateur de code Fortran écrit en Python qui aide le développement d'applications de grande taille. En Fortran, le programmeur doit se concentrer sur l'ordre dans lequel les instructions sont exécutées : avant d'utiliser une variable, le programmeur doit être sûr que cette variable a déjà été construite pour tous les chemins d'exécution possibles. Pour les gros codes, c'est une énorme source d'erreur qui croît avec la taille du code.

Avec IRPF90, l'ordre dans lequel s'exécute les instructions n'est qu'en partie contrôlé par l'utilisateur, et un mécanisme automatique garantit que chaque entité est construite avant d'être utilisée. Ce mécanisme repose sur les relations {a besoin de / est utilisé par} entre les entités, qui sont construites automatiquement. Par conséquent, le programmeur n'a plus besoin de connaître l'arbre de production de chaque entité.

De nombreuses "features" ont été ajoutées au langage, notamment pour la métaprogrammation. Par exemple, des templates ou des scripts Python peuvent être insérés dans le code source. Ces scripts sont exécutés à la compilation et le résultat de la sortie standard est inséré dans le code généré.

Les codes écrits avec IRPF90 s'exécutent généralement plus vite que les programmes Fortran standard, sont plus rapides à écrire et sont beaucoup plus faciles à maintenir.

### References

- IRPF90 : Un générateur de code Fortran pour le calcul scientifique (1)
- IRPF90: a programming environment for high performance computing (2)
- Software home page : <http://irpf90.ups-tlse.fr>

(1) <http://www.hpcmagazine.fr/hpc-labs/irpf90-un-generateur-de-code-fortran-pour-le-calcul-scientifique/>

(2) <http://arxiv.org/abs/0909.5012>

### Conférencier

Ingénieur de recherche / Calcul scientifique

Laboratoire de Chimie et Physique quantiques

IRSAMC / CNRS / Université de Toulouse

Home page : <http://scemama.mooo.com/>



# Ipython Notebook pour l'enseignement

**Conférencier** : Marc Buffat

**Niveau de l'audience** : Novice

## **Description:**

Utilisation de Python et Ipython-Notebook dans les enseignements au département de mécanique et dans le MOOC InPros (Introduction à la Programmation scientifique)

## **Résumé**

Nous présenterons le retour d'expérience sur l'utilisation de Python et Ipython-Notebook dans les enseignements au département de mécanique de l'université Claude Bernard Lyon 1 ainsi que dans le MOOC InPros (Introduction à la Programmation scientifique): <http://INPROS.univ-lyon1.fr>

## **Conférencier**

### Enseignement

- Professeur au département de mécanique de l'UCB Lyon 1
- Responsable master M1 MEGA, enseignant en mécanique des fluides, méthodes numériques

### Recherche

- Membre du laboratoire de mécanique des fluides LMFA UMR 5509
- Directeur de la FLMSN, meso-centre de calcul HPC partenaire de l'équipex [EQUIP@MESO](mailto:EQUIP@MESO)
- Expertise en recherche: CFD, calcul HPC, calcul parallèle

# Python et les supernovae

**Conférencier** : Yannick Copin

**Niveau de l'audience** : Novice

## Description

Ces dernières années ont largement consacré Python comme un des langages de référence pour la recherche en astronomie. Je présenterai un retour d'expérience dans le cadre d'un projet de cosmologie de taille intermédiaire, dans lequel Python et son écosystème ont joué un rôle structurant.

## Résumé

Durant la dernière décennie, Python a réussi le tour de force de structurer la communauté de la recherche astronomique mondiale autour d'un écosystème couvrant une grande partie de ses besoins d'acquisition, de gestion, de traitement, d'analyse et de publications des données. J'illustrerai mon propos à partir de mon expérience personnelle de data scientist dans le cadre d'un projet de recherche en cosmologie observationnelle -- The Nearby Supernova Factory (1)-- où Python & Co. (numpy/scipy, matplotlib, django, astropy, etc.) a joué un rôle majeur dans toutes les étapes. À terme, Python sera au cœur des plus grandes expériences actuellement en développement (p.ex. LSST et Euclid).

## Notes

Je ne compte pas dans cette intervention faire une présentation exhaustive des outils utiles à l'astronome, mais juste rapporter une success story dans le domaine exigeant et pragmatique de la recherche professionnelle. Depuis 10 ans, je pense que cette expérience positive a contribué à asseoir le rôle de Python comme langage de référence maintenant utilisé dans les plus grandes expériences

## Conférencier

Maitre de conférences, équipe de Cosmologie observationnelle (2) de l'Institut de Physique Nucléaire (3) de Lyon, Université Claude Bernard Lyon 1. Utilisateur et promoteur de Python dans le domaine scientifique depuis 2005, enseignement de Python à l'École Normale Supérieure de Lyon et au Master de physique fondamentale (M2) de l'Université Lyon 1 (cours en ligne (4)).

(1) <http://snfactory.lbl.gov/>

(2) <http://www.ipnl.in2p3.fr/spip.php?rubrique51>

(3) <http://www.ipnl.in2p3.fr/>

(4) <http://informatique-python.readthedocs.org/>

## Du python qui ne manque pas d'air

**Conférenciers** : David Romaric , Bernard BARROIS

**Niveau de l'audience** : Intermediate

### Description

Cet exposé présente un petit pré-processeur réalisé en python permettant de produire un simulateur de flux d'air dans une salle machine. Le simulateur lui-même tourne en FreeFem++. La description de la salle machine se fait dans un format de fichier facilement analysable par python. L'exposé vise à recueillir des bonnes idées pour améliorer le développement.

### Résumé

Les Datacenters sont des installations regroupant un grand nombre de serveurs informatiques, qu'il faut ensuite refroidir. La simulation des flux d'air en salle machine est un problème complexe, et il est important de pouvoir tester plusieurs scénarios. Le développement réalisé ici ne vise pas à écrire un simulateur en python, mais à produire automatiquement un code FreeFem++ (1). Ce code de simulation, conçu par Bernard Barrois (PSA), sera exécuté sous FreeFem++. En entrée, le pré-processeur écrit en Python lit une description de salle machine dans un format crée pour l'occasion, dans le double but : - d'être facilement productible par un être humain normal ; - d'être facilement analysé par un interpréteur python normal.

Le code du pré-processeur lui-même représente un peu plus de 1200 lignes de python, est très probablement améliorable, les éventuels retours de la salle (pas trop de tomates SVP) seront profitables pour cela.

### Conférencier

I'm the head of the HPC Computing Center of the University of Strasbourg. My main interrests are: - scientific computing - datacenters - scientific training - public procurements

I organize a lot of technical schools on several topics.

(1) <http://www.freefem.org/ff++/>

## Mathématiques expérimentales : Sage pour la combinatoire

**Conférencier** : Viviane Pons

**Niveau de l'audience** : Novice

### Description

On pourrait penser que les mathématiques fondamentales ne sont faites que d'immenses formules griffonnées sur d'interminables cahiers. Et pourtant, derrière les théorèmes, se cache parfois du Python ! Cet exposé se propose d'explorer l'expérimentation mathématique en combinatoire à travers le logiciel Sage.

### Résumé

Qu'est-ce que vient faire l'expérimentation en mathématique ? Dans ce monde où tout n'est que déduction, où intervient la machine ?

A travers l'exemple de la combinatoire et des objets combinatoires en Sage, nous verrons que la modélisation informatique de structures mathématiques est essentielle pour la résolution de certains problèmes. Nous nous baserons sur des exemples concrets : énumération d'objets, génération aléatoire... Nous utiliserons Sage pour explorer ces questions et émettre des conjectures. Nous montrerons aussi des exemples liés à de véritables questions de recherche actuelle et expliquerons comment la machine est notre alliée à toutes les étapes de la preuve.

Par ailleurs, cet exposé sera une occasion de découvrir le logiciel mathématique Sage. Tous les exemples seront donnés en Sage et nous explorerons un peu la structure de certains objets au sein du logiciel.

Note : aucune connaissance de Sage au préalable n'est nécessaire et le niveau mathématique sera maintenu à un seuil accessible à tous.

### Conférencier

Je suis chercheur en combinatoire algébrique, maître de conférence en informatique à l'université Paris-Sud Orsay.

L'exploration par ordinateur est essentielle à ma recherche et en particulier Python que j'utilise à travers le logiciel Sage. A la frontière des mathématiques et de l'informatique, je suis venue partager mon expérience et suis impatiente de rencontrer les autres pythonistas français(es) !

## Visualisation avec python

**Conférencier** : Nicolas Rougier

**Niveau de l'audience** : Intermediare

### Description

Figures, dessins et animations (2D/3D) avec python.

### Résumé

Matplotlib est une librairie graphique 2D (avec un support partiel de la 3D) qui est la librairie standard pour la visualisation scientifique. Elle permet notamment d'exporter vers tous les formats standards de publications (vectoriel ou bitmap) et peut être aussi utilisée pour simplement faire du graphisme. C'est aussi un outil très précieux pour l'enseignement puisqu'il permet aux élèves de très rapidement visualiser un résultat lorsque la situation s'y prête.

Cependant, lorsque l'on souhaite faire de la 3D, les choses se compliquent un petit peu et il faut alors se tourner vers OpenGL qui est l'un des standards de l'industrie (OpenGL, GL ES, WebGL). OpenGL est très facilement accessible sous python, mais sa programmation l'est elle beaucoup moins. Il faut alors se tourner vers des librairies de plus haut niveau telles que pyglet, nodebox, pyprocessing ou le tout récent projet vispy.

Cette proposition d'intervention a pour but de présenter un panorama de la visualisation avec python.

Quelques liens:

- Tutoriel matplotlib: <http://www.loria.fr/~rougier/teaching/matplotlib/matplotlib.html>
- Tutorial OpenGL: <http://www.loria.fr/~rougier/teaching/opengl/index.html>

### Conférencier

Chargé de recherche en neurosciences computationnelles à l'INRIA ([www.inria.fr](http://www.inria.fr)) et je travaille au sein de l'Institut des Maladies Neurodégénératives à Bordeaux (<http://imn-bordeaux.org>).

# SymPy une bibliothèque pure Python pour le calcul symbolique

**Conférencier** : Kamel Ibn Aziz Derouiche

**Niveau de l'audience** : Intermediate

## Description

SymPy est une bibliothèque de mathématiques symboliques et un outil de calcul formel (Computer Algebra System) écrit en Python. Cette présentation donnera un aperçu sur les fonctionnalités de SymPy et les multiples opportunités pour l'éducation et la recherche

## Résumé

SymPy est une bibliothèque de mathématiques symboliques et un outil de calcul formel (Computer Algebra System) écrit en Python. Le projet contient quatre composants un interpréteur en ligne de commande (isympy) deux applications Web, SymPy Gamma et SymPy Live, permettent aux étudiants, enseignants et chercheurs d'effectuer des expériences mathématiques. SymPy Gamma (<http://gamma.sympy.org/>) est une interface d'apprentissage pour le calcul et la visualisation baser sur des exemples: algébrique, géométrique, trigonométrie et théorie des nombres. SymPy Live (<http://live.sympy.org/>) offre un web-shell avec des fonctionnalités tel-que: l'écriture des équations en LaTeX, manipulation symbolique avec le langage programmation Python. Et une API très riche qui facilite la modélisation et la programmation de nombreux problèmes mathématiques (géométrie différentielle, intégration numérique, théorie des catégories, combinatoire) et physiques (optique, mécanique classique, mécanique et informatique quantique ).

Cette présentation donnera un aperçu sur les fonctionnalités de SymPy et les multiples opportunités pour l'éducation et la recherche. Elle couvrira les points suivants:

- Pourquoi choisir SymPy pour calculer formellement ?
- Comparaison entre la philosophie SymPy et Sagemath
- Manipulation des expressions symboliques
- Brefs aperçu de l'architecture de SymPy
- L'outil web SymPy Gamma
- Résolution de problème mathématique avec SymPy

## Conférencier

Architecte R&D chez Algerian IT Security Group, développeur-contributeur dans le projet NetBS et le Framework pkgsrc intéressé par le portage et la maintenance des paquets scientifiques python-math, python-physics, python-biology. Membre de l'Association Algérienne pour le développement de l'enseignement, des mathématiques et technologies de l'information

## Retour d'expérience : l'école d'été python scientifique

**Conférencier** : Nelle Varoquaux

**Niveau de l'audience** : Novice

### Description

Pour la 7ème année consécutive, une vingtaine d'étudiants ont pu participer à l'école d'été sur python scientifique: 5 jours de python scientifique intensif et de "bonnes pratiques en informatique"

### Résumé

Je discuterai les choix techniques (pair programming, utilisation de l'ipython notebook, selection des étudiants) et difficultés rencontrés lors de l'organisation de cette école d'été.

### Conférencier

Présidente de l'AFPy

# Quelques idées d'optimisation d'un code numérique écrit en Python

**Conférencier** : Loic Gouarin

**Niveau de l'audience** : Experienced

## Description

NumPy est le module incontournable lorsque l'on veut faire un code numérique en utilisant Python. Néanmoins, pour avoir des codes performants, il est souvent nécessaire de repasser à des langages bas niveau. Nous présenterons ici différents outils Python qui le font à notre place et nous permettent d'avoir des gains significatifs sur les temps de calcul.

## Résumé

Le langage Python est maintenant largement utilisé au sein de la communauté du calcul numérique en s'appuyant fortement sur le module NumPy qui permet de gérer les tableaux. Bien que ce module soit optimisé pour les opérations élémentaires (addition de deux vecteurs, multiplication de matrices, ...), il est souvent nécessaire d'optimiser certaines parties du code avec d'autres outils pour obtenir des performances proches d'un langage bas niveau (C ou Fortran). Une large palette est offerte aux développeurs pour optimiser leur code: Cython, Numba, PyPy, Pythran, Parakeet, ... La plupart de ces outils utilise de la compilation "just in time" permettant de générer du code optimisé à l'exécution du programme. Leur syntaxe est très simple et ils peuvent être utilisés sans connaissances particulières. Lors de cette présentation, nous aborderons quelques uns de ces outils ainsi que leurs utilisations sur des cas simples. Nous ferons des comparaisons de performances entre ces outils et un code écrit en langage bas niveau.

## Conférencier

Ingénieur de Recherche CNRS en calcul scientifique au laboratoire de Mathématiques d'Orsay, j'utilise le langage Python dans la plupart de mes développements. Le langage me permet d'avoir des logiciels simples d'utilisation (notamment pour des personnes ayant l'habitude d'utiliser Matlab ou Scilab) tout en ayant des codes performants et parallèles. J'organise également des formations à destination du monde académique dans le domaine du calcul scientifique.

## Poppy - la plateforme robotique humanoïde qui s'adresse à tous

**Conférencier** : Stéphane Ribas , Clément Moulin-Frier

**Niveau de l'audience** : Novice

### Description

Dans cette présentation, nous parlerons brièvement de l'histoire de poppy, robot humanoïde (84 cm, 3 kg) open-source et imprimé en 3D . Nous présenterons ensuite la plateforme réunissant informatique (python), électronique (arduino), "mécanique" (moteur) et fabrication (imprimante 3D). Nous insisterons sur la partie programmation du robot en python.

### Résumé

Poppy – coquelicot en français – est un petit robot humanoïde (84 cm, 3 kg) open-source et imprimé en 3D créé par l'équipe de recherche Flowers. C'est une plateforme robotique commune d'échange. Elle a pour but de tester différentes morphologies mécaniques et algorithmes de contrôle du robot et de partager entre utilisateurs les résultats obtenus.

Depuis l'ouverture de la plateforme, plusieurs projets ont vu le jour dans le monde entier et dans des univers très variés (recherche, art, éducation,...). Poppy arrive également dans les écoles et les FabLabs pour l'éducation au monde numérique.

Dans cette présentation, nous parlerons brièvement de l'histoire de poppy et ses objectifs. Nous présenterons ensuite la plateforme réunissant informatique (python), électronique (arduino), "mécanique" (moteur) et fabrication (imprimante 3D). Nous insisterons sur la partie programmation du robot en python.

### Conférencier

Stéphane Ribas

Formation : DEA sons, images et paroles.

Situation actuelle : "gestion" de projet technologique à l'Inria, conseils sur OSS et création de communauté pour le développement technologique/logiciel au sein de l'institut.

Rôle dans Poppy-project.org : agitateur d'idées, community manager :-)

Participe aux axes : heu... chais pas, j'ai vu de la lumière alors je suis rentré. nan, community manager :-)

Compétences : oss, gestion projet, animation, promotion, un peu de programmation.

Langages de programmation : C, python, php, java (android), MAX/MSP (bcq!), mais auss français, anglais et un peu d'espagnol avec les machines humaines.

# Salle 51 - Samedi 25 octobre - Session enseignement – recherche

## BoF retour expérience enseignement Python

**Conférencier** : Olivier Berger

**Niveau de l'audience** : Novice

### Description

L'objectif sera de se regrouper entre personnes intéressées par l'enseignement de Python Le format est celui d'une BoF (birds of a feather) pour échanger : retours d'expérience, présentation outils, prise de contacts pour mutualisation, etc.

### Résumé

L'objectif serait de recenser / mutualiser des efforts autour de l'enseignement de Python, par exemple autour des outils pour les travaux pratiques.

Le format est celui d'une BoF (birds of a feather) pour échanger : retours d'expérience, présentation outils, prise de contacts pour mutualisation, etc.

Venez avec vos idées (et vos sandwiches)

Mon intérêt perso (Olivier) est principalement centré sur l'utilisation d'exerciceurs pour l'enseignement de Python en non-présentiel, typiquement pour un MOOC ou un SPOC

### Conférencier

Vieux pythoniste, mais éloigné longtemps de Python... j'essaye de me rattraper, et suis pas mal intéressé par : - l'enseignement à l'informatique en/avec Python (Notebooks IPython, etc.) - l'utilisation de Python pour des applications Linked Data

## **Amphi T7 – Dimanche 26 octobre**

### **Présentation de Nao**

**Conférencier** : Nao Sur-Seine

**Niveau de l'audience** : Novice

### **Description**

Présentation du robot Nao.

### **Résumé**

Présentation du robot Nao, d'Aldebran Robotics. Ses capacités (motrices, interactives, detections, etc), mais aussi le SDK permettant de le programmer, sans oublier l'interface native avec Python.

### **Conférencier**

Groupe d'utilisateurs de Nao dans la région parisienne

# Obscurcissement de bytecode Python

**Conférencier** : Nicolas Szlifierski

**Niveau de l'audience** : Intermediate

## Description

Il est simple de récupérer le code source d'un programme Python depuis son bytecode, mais ce n'est pas toujours souhaitable. Nous verrons qu'en utilisant un interpréteur généré spécifiquement pour une application, nous pouvons rendre le bytecode d'une application obscurci et non exécutable par l'interpréteur standard sans modifier significativement le temps d'exécution.

## Résumé

Lorsqu'on livre une application Python, il est simple d'en récupérer le code source depuis le bytecode, ce qui n'est pas toujours souhaitable. L'obscurcissement (en anglais, obfuscation) permet de pallier cette limitation. Nous pouvons modifier l'interpréteur Python afin qu'il puisse exécuter le bytecode de l'application obscurcie via différentes techniques :

- Permutation des opcodes : On peut permuter les représentations entières des opcodes, afin que les actions effectuées par ceux ci ne correspondent plus à la représentation de l'interpréteur standard.
- Ajout de nouveaux opcodes : Les opcodes étant codé sur 8 bits, il reste plus d'une centaine de codes libres, ce qui nous laisse la possibilité d'en ajouter de nouveaux. On peut ainsi remplacer les suites d'opcodes récurrentes dans l'application par un seul nouvel opcode qui effectuera les mêmes opérations.
- Bytecode auto-modifiant : Pendant l'exécution de l'application, on peut modifier le bytecode chargé en mémoire ; ainsi le bytecode exécuté ne sera pas celui contenu dans le fichier.

On obtient un interpréteur spécifique à l'application, capable d'exécuter le bytecode obscurci sans pertes de performances, et la rétro-ingénierie est impossible sans un travail sur la compréhension de l'interpréteur modifié.

**Conférencier**

/

## Asynco: offrez des tulipes à vos entrées sorties asynchrones

**Conférencier** : Thierry Chappuis

**Niveau de l'audience** : Intermediare

### Description

Bien que parfois limitée par la capacité de calcul, la vitesse d'exécution d'un morceau de code est bien plus souvent limitée par des opérations d'entrées sorties qui n'en peuvent plus d'attendre. Co-routines, Futures et une nouvelle pièce au puzzle, asynco, proposent une réponse élégante à ce problème. Cette présentation vous propose de l'explorer et de la rendre accessible au plus grand nombre.

### Résumé

En tant que scientifique, mon univers est constamment confronté au besoin de paralléliser pour faire plus dans un temps fini et généralement bien occupé. Bien que souvent confronté à des tâches de programmation limitées par la capacité de calcul de la machine, la vitesse d'exécution d'un morceau de code est limitée, dans de très nombreuses applications, par des opérations d'entrées sorties qui attendent une réponse, par exemple d'un serveur distant. Ici, l'utilisation de threads ou de multiprocessing n'est pas toujours la solution la plus indiquée ni la plus intuitive à implanter. Afin de répondre à cette problématique, de nombreux frameworks asynchrones ont vu le jour, comme Twisted, Tornado, ZeroMQ, gevent, pour n'en citer que quelques uns. Si chaque opus se propose d'apporter son lot de solutions à certaines problématiques spécifiques, leur interopérabilité peut être rendue difficile par l'utilisation de modèles événementiels bien différents. C'est pour répondre à ce besoin de standardisation que le PEP-3156 a été rédigé et que la bibliothèque asynco a trouvé sa place dans la bibliothèque standard de python dès sa mouture 3.4.

Que reproche-on à asynco? Quel sont les apports de asynco? Que deviennent Twisted, gevent, Tornado dans cette équation? Pouvez-vous m'expliquer les générateurs, les co-routines, les Futures ou encore "yield from"? Quels problèmes ces abstractions essaient-elles de résoudre? Ce sont les questions que je me propose d'explorer avec vous, dans un style accessible au plus grand nombre.

### Conférencier

Après des études d'ingénieur et un doctorat à l'Ecole Polytechnique Fédérale de Lausanne, je suis actuellement Professeur de Génie Chimique et de modélisation à l'Ecole d'Ingénieurs et d'Architectes de Fribourg. Dans le contexte de mes travaux de recherche et d'enseignement, j'utilise essentiellement Python pour ses qualités en matière de programmation scientifique et de calcul numérique, notamment pour la simulation et l'optimisation de réacteurs chimiques industriels (Numpy, Scipy, FEniCS ou encore scikit-learn). En marge des équations différentielles partielles, j'utilise également python pour notre système interne de gestion de connaissances et de publications scientifiques, Solv.re. Ce service home made permet l'indexage, la recherche, le classement et la citations des SOPs, des publications scientifiques et des brevets référencés dans nos bases de données (ZeroMQ, ZeroRPC, PyLucene, Django). Solv.re m'a amené à beaucoup m'intéresser à l'architecture des applications distribuées et à la manière de gérer la communication entre les briques de tels systèmes. Dans ce domaine, la programmation asynchrone propose des outils excitants, particulièrement depuis l'arrivée dans le paysage d'un nouveau chef d'orchestre, asynco.

De manière générale, à ma sortie du bureau, je m'intéresse aux différents aspects du langage: interface

avec des langages bas niveau, pypy, programmation réseau ou programmation d'applications web. Passionné de musique et de photos, j'utilise également python pour scripter ma vie de tous les jours: retoucher des fichiers, renommer, organiser, envoyer, archiver. Bref! Python est mon couteau suisse digital.

# Exploration de la boucle d'événement asyncio

**Conférencier** : Victor Stinner

**Niveau de l'audience** : Experienced

## Description

Le nouveau module asyncio de Python 3.4 est haut-niveau et complexe. Le coeur d'asyncio est composé de plusieurs briques simples, la complexité vient de la composition élégante de ces briques. Des versions allégées de ces briques vont être réécrites pour introduire différents concepts.

## Résumé

Programme :

- Ordonnanceur basé sur des callbacks
- Ordonnanceur simple : `call_soon()`
- Minuteur : `call_at()`
- Minteur 2 : `call_later()`
- Multiplexage : `select()`
- Boucle infinie : `run_forever()`
- Résultat asynchrone
- Future
- Future : intégration avec la boucle d'événement
- Générateur Python et `yield-from`
- Générateur
- `yield-from`
- Tâche asyncio
- Coroutine
- Tâche
- Coroutine attendant un objet future
- Tâche : intégration avec la boucle d'événement

## Notes

Je travaille sur le projet asyncio depuis fin 2013, directement dans le projet asyncio. J'ai aidé à corriger des bugs, implémenté le module `asyncio.subprocess` (gestion des sous-processus), amélioré et complété l'implémentation Windows, etc.

J'ai également backporté le projet asyncio sur Python 2 : projet Trollius dont la version 1.0 est sortie fin juillet dernier.

## Conférencier

Core developer Python depuis 2010, je suis l'auteur de plusieurs applications et bibliothèques Python. Voyez mes profils Bitbucket (1) et Github (2). Je travaille actuellement sur OpenStack pour Enovance (Paris).

(1) <https://bitbucket.org/haypo/>

(2) <http://github.com/haypo/>

# Gunicorn, le problème du thundering herd et autres challenges liés à la concurrence

**Conférencier** : Benoit Chesneau

**Niveau de l'audience** : Intermediare

## Description

Le design de Gunicorn n'a pas beaucoup évolué depuis que le projet a démarré. Avec le nombre croissant de CPUS et l'apparition de nouvelles bibliothèques de programmation concurrente sous Python 2 et 3 , il est temps de répondre à ces nouveaux défis.

## Résumé

Le design de Gunicorn n'a pas beaucoup évolué depuis que le projet a démarré. Avec le nombre croissant de CPUS et l'apparition de nouvelles bibliothèques de programmation concurrente sous Python 2 et 3, il est temps de répondre à ces nouveaux défis.

Cette présentation décrira le nouveau design de Gunicorn les différentes solutions fournies par celui-ci pour gérer toujours plus de connexions concurrentes et comment le problème du "thundering herd" (1) a été résolu. Par ailleurs elle décrira l'amélioration du support de Python 3 avec l'introduction des "workers" gaihhttp et gaihttp et comment vous pouvez les utiliser. Dans la foulée, elle récapitulera les différentes stratégies offertes par Gunicorn pour permettre a votre application de monter a l'échelle facilement en fonction de la plateforme ou des bibliothèques utilisées.

Enfin cette presentation fera un retour sur les recent changements apportés sur le support des différents "frameworks" Python et la nouvelle architecture de plugins.

(1) [http://en.wikipedia.org/wiki/Thundering\\_herd\\_problem](http://en.wikipedia.org/wiki/Thundering_herd_problem)

## Notes

1. Le design de gunicorn: presente l'architecture interne, son fonctionnement, les workers et comment ce design peut permettre a votre application de monter a l'échelle.
2. "1 commande pour les gouverner tous" met en avant les recents changements apportés sur le support des différents "frameworks" Python.
3. Des plugins partout: la nouvelle architecture de plugins

## Conférencier

Benoît Chesneau est un artisan web vivant près de Paris. Il a des années d'expérience dans la construction et la montée en charge de services web et systèmes de données. Il est le fondateur d'une société spécialisée dans le développement de service web et d'applications open-sources innovantes. Benoît Chesneau est aussi le fondateur du projet Refuge et l'auteur de Gunicorn et de multiples projets opensources.

Benoit est un membre "fellow" de la PSF.

# Bootstrapping Machine Learning

**Conférencier** : Louis Dorard

**Niveau de l'audience** : Intermediate

## Description

Prediction APIs are democratizing Machine Learning. They make it easier for developers to build smart features in their apps by abstracting away some of the complexities of building and deploying predictive models. In this talk we'll look at the possibilities and limitations of ML, how to use Prediction APIs, how to prepare data to send to them, and how to assess performance.

## Résumé

Last year, Forrester introduced the term “predictive apps” and described them as “the next big thing in app development” (1). They are apps that can provide the right functionality and content at the right time, for the right person, by continuously learning about them. Now, in the same way that we talked of the importance of mobile-first development, we are talking of the importance of predictive-first development (2).

Fortunately for developers, machine learning is being democratized through Prediction APIs that abstract away some of the complexities of building predictive models from data (and deploying them). They make it easier for developers to build smart features in their apps (for instance: detecting spam, predicting missing values, predicting a user's intent, his interests, etc.). As demand for predictive apps is growing, more and more of these APIs are coming out. Microsoft and Wolfram have just entered the space of Machine Learning as a Service and Prediction APIs, with Azure ML (3) and Wolfram Programming Cloud (4) respectively. They are thus joining Google Prediction API (5) and BigML (6), a promising startup which is rapidly expanding.

We'll start this talk by looking at the possibilities and limitations of Machine Learning. Then, we'll study how Prediction APIs work. We'll go through an IPython notebook to illustrate this with the BigML API: we'll see how to create a (white-box) predictive model from data and how to use that model. The crucial point here is to have “good” data to send to the API. We'll discuss what this means and how to prepare data, from “sanity checks” you can make in Excel to scripting data processing tasks with the Pandas library (7). Finally, we'll go over the criteria and methodology to assess the performance of a Machine Learning system prior to its deployment in production.

(1) [http://blogs.forrester.com/mike\\_gualtieri/13-08-28-predictive\\_apps\\_are\\_the\\_next\\_big\\_thing\\_in\\_app\\_development](http://blogs.forrester.com/mike_gualtieri/13-08-28-predictive_apps_are_the_next_big_thing_in_app_development)

(2) <http://techcrunch.com/2014/08/29/predictive-first-how-a-new-era-of-apps-will-change-the-game/>

(3) <http://azure.microsoft.com/en-us/services/machine-learning/>

(4) <http://www.wolframcloud.com/>

(5) <https://developers.google.com/prediction/>

(6) <http://www.bigml.com/>

(7) <http://pandas.pydata.org/>

## Notes

This talk is not a traditional machine learning talk. I won't cover ML libraries such as scikit-learn, but I will take a higher level view of ML — made possible by prediction APIs.

Applying ML to data used to be very demanding and required a lot of technical knowledge. Prediction APIs are disrupting things as they are removing barriers to entry (such as knowledge of algorithms). For these reasons, ML talks have always had restricted audiences in the past, but considering Prediction APIs now makes it possible to talk about ML in a simpler way and to a wider audience.

This is what I have started to do at a few meetups and conferences (see below) and in a new book I wrote, *Bootstrapping Machine Learning* (1). I'm also a co-founder of PAPIs.io (2), the 1st international conference on predictive APIs and predictive apps, and I'm general chair of the 2014 edition.

Speaker experience:

- Invited speaker at scientific conferences (ICML and NIPS), universities (University College London, Universitat Autònoma de Barcelona, Universitat Politècnica de València), companies (Telefonica)
- Speaker at meetups: Paris Hackers, Paris Machine Learning, BordeauxJS, HumanTalks
- Upcoming talks at APIcon UK and Bdx.io

(1) <http://www.louisdorard.com/machine-learning-book>

(2) <http://www.papis.io/>

## Conférencier

I am a data guy who has fully switched from Matlab to Python. I am the author of *Bootstrapping Machine Learning* (1) and a co-founder of PAPIs.io (2), the International Conference on Predictive APIs and Apps. Previously, I served as Chief Science Officer in a tech startup, I bootstrapped a web business, and I studied Machine Learning at University College London where I obtained my PhD.

I am passionate about technological innovation in web and mobile apps. I am now on a mission to help developers create smarter apps with the use of Predictive APIs.

(1) <http://www.louisdorard.com/machine-learning-book>

(2) <http://www.papis.io/>

## Kivy - a very short introduction

**Conférencier** : Gabriel Pettier

**Niveau de l'audience** : Intermediare

### Description

Kivy est un framework d'interactions multitouch, multiplateforme, sous licence MIT. Cette présentation introduira les concepts pour utiliser Kivy, et les différents outils autour du projet.

### Résumé

Concepts GUI

Les différents concepts de programmation d'interfaces graphiques et leur implémentation dans kivy.

- Évènements
- Graphiques
- Widgets
- Properties
- Language Kv

Outils

modules

- inspector
- WebDebugger
- Screen

Garden

- utiliser les fleurs du jardin
- partager ses fleurs

Plyer - APIs multiplateformes

- Pourquoi ?
- Fonctionnalités

### Conférencier

Core developpeur Kivy depuis 2011, développeur python depuis 2005, libriste touche-à-tout.

Travaillant actuellement pour Tangible Display (<http://tangibledisplay.com>), je travaille sur les interfaces tangibles, les objets connectés, et d'autres sujets a haute valeur ludique et interactive.

# Amphi T8 – Dimanche 26 octobre

## Une interface expérimentale pour lier OpenGL à Python

**Conférencier** : Fabrice Salvaire

**Niveau de l'audience** : Intermediare

### Description

PyOpenGLng est un module visant à expérimenter de nouveaux concepts autour de l'intégration de l'API OpenGL au sein de Python. Les objectifs de ce module sont d'apporter une simplification par rapport à l'implémentation de référence PyOpenGL, d'implémenter une interface basée sur CFFI et d'offrir une API de haut niveau pour les versions modernes d'OpenGL

### Résumé

Jusqu'à présent les interfaces Python de l'API OpenGL étaient générées à partir d'un script qui lit le fichier d'en-tête C afin d'en extraire l'API. Cette approche n'a rien de choquant en soi et c'est d'ailleurs l'approche choisie par l'interface CFFI. Mais pour l'API OpenGL nous pouvons utiliser une autre approche puisque l'organisme en charge de la spécification d'OpenGL, le groupe Khronos, fournit un fichier XML appelé Registry qui décrit l'API et ses extensions officielles en fonction de la version, du profile (compatibilité ou core) et de sa variante desktop ou embarquée. Ce point est d'une grande importance car une des particularités de l'API OpenGL est qu'il n'y a pas une seule API mais plusieurs. L'autre particularité de l'API OpenGL est qu'elle est relativement simple puisque elle est uniquement composée d'énumérants et de fonctions mettant en œuvre des types fondamentaux. L'API ne définit par exemple aucun type de données composées.

L'interface de PyOpenGLng repose sur deux composants essentiels, le premier est une interface orientée-objet de l'OpenGL Registry et le deuxième est la partie qui gère l'interface C-Python à proprement parlé.

L'interface orientée-objet de l'OpenGL Registry permet de générer au vol l'API OpenGL souhaitée par l'utilisateur sous la forme d'un module Python, c.à.d. la liste des énumérants et des fonctions correspondant à l'API demandée. En comparaison PyOpenGL compte pas moins de mille fichiers pour couvrir toute l'API OpenGL et ses extensions. De plus il est relativement aisé d'obtenir la dernière version de l'API en téléchargeant le fichier XML depuis le serveur CVS de Khronos.

Au fil du temps l'API OpenGL a subi une révolution avec l'arrivée des GPUs qui a introduit à partir de la version 3 une approche basée sur un pipeline programmable. Cette nouvelle approche a occasionné des changements radicaux au niveau de l'API et sur la manière de travailler avec. Aujourd'hui OpenGL est essentiellement une interface orientée graphisme pour les processeurs de flux (GPU), de la même manière qu'OpenCL est une interface orientée calcul. Par conséquent l'usage de l'API en Python va essentiellement consister à modifier des variables d'états et échanger des tableaux Numpy via leurs adresses.

Cette relative simplicité est mise à contribution dans l'interface C-Python via une translation automatique basée sur les prototypes des fonctions. Ce mécanisme tient dans un seul fichier et requiert actuellement moins de 500 lignes de codes. Par conséquent il est relativement aisé de comprendre ce qui se passe entre le code utilisateur et l'API OpenGL. De plus il est implémenté avec ctypes et CFFI, ce qui permet d'être compatible avec l'interpréteur pypy.

Cette présentation sera l'occasion de faire un tour d'horizon des différentes approches possibles pour lier une bibliothèque C à Python et avec dans ce cas précis une approche originale puisque elle est basé sur une API décrite par un fichier XML.

Afin d'illustrer l'utilisation de ce module, je présenterais brièvement un module Python qui permet d'interpréter les fichiers DVI générés par LaTeX et un viewer mettant en œuvre un rendu du texte accéléré par GPU.

Lien vers les dépôts Git et leurs documentations:

- PyOpenGLng dépôt et doc
- PyDVI dépôt et doc

## Conférencier

Je suis chef de projets dans une biotech pour laquelle j'ai conçu une plate-forme logicielle de traitement et visualisation d'images issues d'un microscope automatisé. Ce logiciel est essentiellement implémenté en langage Python et repose sur des composants open source tel que Numpy, Scipy, Matplotlib, H5py, Pyqt, PyOpenGL et OpenCV.

J'ai participé l'année dernière à la conférence Euroscipy 2013 et publié un article dans le proceedings de la conférence: High-Content Digital Microscopy with Python, Fabrice Salvaire, Part of the Proceedings of the 6th European Conference on Python in Science (EuroSciPy 2013 (1)), Pierre de Buyl and Nelle Varoquaux editors, (2014) arXiv:1404.6385v2 (2).

Mes contributions à l'open source en relation avec le langage Python sont:

- PyOpenGLng (3) un wrapper expérimental pour l'API OpenGL,
- PyDVI (4) un module qui permet de lire les fichiers DVI générés par LaTeX et de les afficher avec un rendu accéléré par GPU en utilisant l'API OpenGL,
- PySpice (5) un module qui permet de simuler des circuits électroniques via Ngspice,
- PyResistorColorCode (6) un utilitaire pour décoder les codes de couleurs des résistances,
- LaptopControlPanel (7) un control panel pour laptop Lenovo,
- quelques plugins Sphinx tel que sphinx-css3image (8),
- et pour finir un gestionnaire de PDF (9) en gestation.

Mon attrait pour l'open source ne se limite pas au software mais aussi à l'open hardware. C'est pourquoi je suis un membre actif (10) du projet Electrolab V2 (11) géré par une association loi 1901 qui consiste à construire le plus grand hackerspace de France et d'Europe située à Nanterre en proche banlieue parisienne.

J'en profite pour faire un peu de publicité à cette aventure humaine: L'Electrolab V2 sera un lieu unique doté d'une surface de 1500 m<sup>2</sup> qui mettra à disposition de ces membres des moyens de production en mécanique et en électronique ainsi qu'un laboratoire de chimie et une salle blanche. Une salle serveur de bonne dimension supportera l'infrastructure informatique et l'appétit des hackers. Le lieu disposera aussi de salles de formation et de boxes entreprises. Pour nous soutenir (12)

Lien vers mon site perso : <http://fabrice-salvaire.pagesperso-orange.fr/>

- (1) <https://www.euroscipy.org/2013/>
- (2) <http://arxiv.org/abs/1404.6385v2>
- (3) <https://github.com/FabriceSalvaire/PyOpenGLng>
- (4) <https://github.com/FabriceSalvaire/PyDVI>
- (5) <https://github.com/FabriceSalvaire/PySpice>
- (6) <https://github.com/FabriceSalvaire/PyResistorColorCode>
- (7) <https://github.com/FabriceSalvaire/LaptopControlPanel>
- (8) <https://github.com/FabriceSalvaire/sphinx-css3image>
- (9) <https://github.com/FabriceSalvaire/Biblio>
- (10) <http://www.pycon.fr/2014/reviews/review/86/>
- (11) <http://www.electrolab.fr/>
- (12) <http://donate.electrolab.fr/>

## APIs hypermédia

**Conférencier** : Olivier Hervieu

**Niveau de l'audience** : Novice

### Description

Le design et l'implémentation d'APIs est au coeur du travail d'un développeur. Les travaux de Roy Fielding ont permis ainsi de redéfinir la manière d'écrire des webservices autour du principe d'architecture REST et posent aussi le principe des APIs Hypermedia. Cette présentation offrira une explication de ce qu'est une API Hypermedia.

### Résumé

Après un rappel des concepts du principe d'architecture RESTs, les concepts des APIs Hypermedias seront développés. Notamment en montrant comment ces formats sont actuellement utilisés par chacun d'entre nous; soit dans notre vie quotidienne (en surfant simplement sur le WEB), soit en utilisant certaines APIs, déjà hypermedias (c'est le cas des APIs Google) et comment ce concept vie avec REST. Plusieurs formats seront alors vus; dont Collection+JSON, AtomPub, JSON-LD. Enfin, un tour rapide des solutions permettant d'implémenter ce type d'API en python sera effectué.

### Conférencier

Olivier Hervieu est actuellement à la tête de l'équipe de développement de tinyclues. Depuis 10 ans (dont 6 années de python), il développe des systèmes pour des applications critiques (drones, applications biométrique, solution de sécurité). Grâce à une approche pragmatique du développement logiciel et des pratiques Agile/TDD il s'assure aujourd'hui que tinyclues soit à même de fournir un environnement technique permettant de packager une solution de machine learning simple à l'usage pour ces clients.

# Grimper en haut de la Pyramid

**Conférencier** : Gaston TJEJBES

**Niveau de l'audience** : Novice

## Description

Tour d'horizon des fonctionnalités de Pyramid.

## Résumé

Pyramid est un framework web fruit de la jonction du projet repoze.bfg et du projet Pylons. C'est un framework sans opinion qui fournit une structure adaptable aux besoins, et aux habitudes. Il permet aussi bien le développement rapide de micro-application que le développement de projets complexes. Il donne aux développeurs une grande liberté dans le choix de l'architecture de leur application. Cette présentation aura pour objectif de décrire les différents outils que Pyramid met à disposition, de partager des retours d'expérience sur son usage et de faire un rapide tour de son écosystème.

Le plan de la présentation :

Présentation des principales fonctionnalités de pyramid

Cette partie a pour objectif de montrer la pertinence des choix faits dans le framework et de donner une idée des choix que Pyramid fait ou ne fait pas pour nous.

- Introduction : présentation de la communauté et de la philosophie du projet : 100% tested, interfaces, qualité du code, sans-opinion
- Hello world
- L'url dispatch : déclaration de route auxquelles on associe ensuite des vues
- L'url traversal : les vues sont associées à un contexte qui est récupéré par le biais d'un arbre d'objet
- Les vues : déclaration des vues sous forme de fonctions ou de méthodes, les prédicats et leur finesse
- Les acls et l'authentification
- Les outils architecturaux (event, tween, ...)
- Le templating (chameleon ou autres)

Introduction de fonctionnalités complémentaires

Présentation de quelques bibliothèques utiles par le biais de cas pratique

- Accéder à une base de données (SQL, NOSQL)
- Construire une CRUD pour Pyramid
- Les outils pour construire un backend pour une One Page Application

## **Conférencier**

Après des études mathématiques et d'arts plastiques, j'ai travaillé durant 5 ans comme prestataire auprès du projet Eole (distribution linux de l'éducation nationale), c'est par ce biais que j'ai découvert le langage Python, qui est utilisé à tous les niveaux du projet, notamment dans la partie serveur des différentes applications web développée en interne (outil d'administration, serveur SSO, service xmlrpc ...).

Depuis 3 ans, avec d'anciens collègues d'Edenwall, nous avons fondé Majerti, notre activité se concentre essentiellement sur les systèmes d'informations, que ce soit en terme d'administration de serveur, ou en terme de développement de solution de gestion personnalisée.

# Pyramid, OpenERP, des métaclasses et encore plus !

**Conférencier** : Houzéfa Abbasbhay

**Niveau de l'audience** : Intermediate

## Description

Implémentation de pyramid\_openerp, une librairie "pont" qui permet de lier un site Pyramid à OpenERP / Odo.

## Résumé

pyramid\_openerp facilite la création d'un site Pyramid qui affiche les différents éléments d'un back-end OpenERP / Odo. Cette présentation vous en montrera les usages, une implémentation et vous emmènera au fin fond de Pyramid et de Deform. Authentic sera aussi intégré pour montrer la gestion de l'authentification SAML 2.

## Conférencier

Houzéfa a appris à coder avec JavaScript et en PHP. Il a suivi l'évolution des sites web, des plus basiques aux plus évolués de nos jours. Il est d'autre part passionné par l'informatique et tout ce qui gravite autour. Il se sert maintenant énormément de Python, même si le C++ reste gravé dans son coeur ! Il contribue (voire gère) des projets open-source sur du P2P et sera ravi d'en raconter davantage. Il travaille chez XCG depuis 1 an.

## Documents

Slides : <http://www.pycon.fr/2014/proposals/document/1/b06197a4-8904-43d1-940e-79dfa3b21282.gz>

# Brood, une application web par des gens qui n'y connaissent pas grand chose

**Conférencier** : David Cournapeau

**Niveau de l'audience** : Intermediare

## Description

Enthought gère une distribution python appelée Canopy/ex-EPD. Nous raconterons l'histoire de nos efforts pour remplacer les scripts/hacks pour la distribution de ces paquets par une application web, ce qui a marché, et ce qui n'a pas marché. Nous parlerons de nos efforts (avortés) pour utiliser asyncio, de nos problèmes avec eventlet, et aussi de qui a marché (nginx, SQLAlchemy, locust).

## Résumé

La présentation sera organisée comme suivant

Introduction

Brève description du projet, simple démonstration.

Description de l'architecture asynchrone

L'architecture initiale de l'application a jusqu'à récemment été basée sur un processus "backend" utilisant asyncio + SQLAlchemy, avec des "workers" "front end" basés sur flask. Les processus "backend" et "frontend" communiquent au travers de Omq + un protocole spécifique RPC.

Cette architecture était justifiée sur la base des problèmes spécifiques à l'application (génération de gros payload json, contraintes de vérification de gros fichiers en background, nombre/taille de fichier atteignant plusieurs dizaines de Go, etc...), et pour forcer une stricte séparation entre les parties sans état ("stateless") et avec état ("stateful").

Nous détaillerons aussi comment nous avons pu utiliser nginx pour non seulement servir de gros fichiers, mais aussi de manière moins classique accepter, et calculer les checksums à la volée de manière à ce qu'aucun processus python n'ait besoin de lire le fichier en mémoire.

Architecture courante

Des tests ont montré que l'architecture initiale n'était pas aussi robuste qu'espéré. En particulier, le processus "backend" causait des erreurs de type "segfault" régulières qui se sont révélées difficiles à réparer.

L'architecture a été simplifiée pour utiliser celery pour gérer les "jobs" en tâche de fond, et n'utiliser que des processus flask derrière nginx+unicorn. L'architecture générale séparant les parties stateful/stateless a été conservée, mais sans une séparation stricte entre processus.

Conclusion

Nous concluons la présentation avec quelques chiffres de performance.

## Conférencier

Utilisateur de python depuis 2004, contributeur initial du paquet scikit-learn (GSoc), et un des contributeurs principaux à NumPy et SciPy. Egalement auteur des paquets python 'bento' (packaging),

'audiolab' (IO pour fichiers audio).

Depuis 2011, je travaille à Enthought, Cambridge (Royaume-Uni), entreprise spécialisée dans le développement d'applications scientifiques.

github <https://github.com/cournape>

twitter <https://twitter.com/cournape>

## Brython (Python dans le navigateur)

**Conférencier** : Pierre Quentel

**Niveau de l'audience** : Intermediare

### Description

Brython est une implémentation de Python 3 pour les navigateurs web, avec une interface vers les objets et événements DOM. Il permet de développer des applications web côté client en Python au lieu de Javascript.

### Résumé

Le développement web côté client prend une place de plus en plus importante avec le développement des smartphones et tablettes, l'arrivée de HTML5 et des initiatives comme Firefox OS. Aujourd'hui Python est peu présent sur cette plateforme, où Javascript a un quasi monopole

Brython permet de réaliser des programmes en Python 3.x dans des pages web aussi simplement qu'en Javascript, en intégrant du code Python dans des balises `<script type="text/python">`

Rendu public fin 2012, le projet couvre maintenant la quasi totalité de la syntaxe de Python 3 et exécute une bonne partie des modules de la librairie standard sans modification

La présentation couvrira les sujets suivants :

- pourquoi Brython ?
- comment l'utiliser dans une page web
- les outils fournis en ligne : éditeur, console
- quelques exemples (inspirés de ceux fournis sur le site [brython.info](http://brython.info))
- la couverture de la syntaxe et de la librairie standard Python 3
- l'interface avec les objets et événements du navigateur
- interaction avec les librairies Javascript
- performance (comparée à Javascript, et à d'autres solutions de conversion Python vers Javascript)
- comment contribuer

### Notes

Le site Brython (1) permet de tester l'implémentation de Python sur un éditeur ou une console, fournit une documentation complète et une galerie d'exemples

Le développement a migré début septembre 2014 sur Github : <https://github.com/brython-dev/brython>

Brython a déjà été présenté à la conférence Python Espagne fin 2013 et à PyCon à Montréal en avril 2014

Parmi mes projets précédents j'ai développé le framework web Karrigell (2) (1ère version en 2002) qui a eu son petit succès avant l'arrivée de Django, et sur lequel je ne fais plus que de la maintenance. J'ai aussi contribué à la correction du module cgi de la librairie standard Python dans la version 3.2

Je ne suis pas développeur dans la société où je travaille, mais j'y ai monté une formation d'initiation à Python depuis quelques mois

(1) <http://brython.info/>

(2) <http://www.karrigell.fr/>

### **Conférencier**

Programmeur amateur et autodidacte depuis le début des années 80, j'ai développé en Basic sur Alice, Amstrad et Atari ; repris le virus avec mes premières pages web fin des années 90

J'ai découvert Python en 2000 et suis tombé tout de suite sous le charme. Avant Brython, je l'utilisais essentiellement pour la programmation de sites web côté serveur .

# Lean data science, or how to do machine learning with what you have to hand

**Conférencier** : Christophe Bourguignat

**Niveau de l'audience** : Intermediate

## Description

That's life : it's not uncommon to have to do complex data science or machine learning on a capacity-limited platform : a few Gb of RAM, 2 or 3 cores only, no Hadoop Map / Reduce cluster, or an old laptop. In these situations, you must do more with less, and do kind of "lean machine learning".

## Résumé

In this talk, we will show typical illustrations of such "worst-cases", encountered in real life, or in data science challenges like Kaggle (1). We will give some tips & tricks, and best-practices to make the most of Python, and its flagship data analysis libraries : scikit-learn, pandas, numpy and scipy.

On the menu for this session : sparse matrix, incremental learning, chunk reads, manual caching, ... and much more !

(1) <https://www.kaggle.com/>

## Notes

Lors de ce talk, je souhaite présenter quelques techniques avancées de manipulation et de traitement de données, utiles pour l'analyse de gros fichiers avec Python. Voici une liste provisoire de thématiques qui pourront être abordées : - Utilisation des formats de matrices numpy « sparse » (creuses), pour optimiser la gestion de la mémoire - « incremental learning », permettant des traitements en flux (méthodes scikit-learn `partial_fit`) - Techniques de cache manuel, ou comment éviter de calculer plusieurs fois la même chose - Lectures par « chunks », pour ne pas tout charger en mémoire - « warm restart » de modèles prédictifs - Instrumentation du code, ou comment prédire la durée du traitement - Map / reduce « à la main » - Parallélisation avec `joblib`, - Etc. .... Je souhaite que la présentation soit didactique, fouillée techniquement mais sans entrer dans des détails trop complexes. Il s'agit de faire découvrir aux participants une panoplie de techniques, pour qu'ils puissent creuser ensuite. Des extraits de codes seront proposés, et un notebook IPython sera mis en ligne. La présentation sera en français, mais les supports seront en anglais. Quelques informations complémentaires me concernant : j'ai atteint le niveau « master » sur Kaggle (1), une plateforme de référence en Data Science. J'aime faire partager ma passion sur mon compte twitter `@chris_bour`, lors de meetups, ou bien au cours de conférences spécialisées telles que PyCon (par exemple, j'ai donné un talk lors de Devovx 2014, également sur une thématique « machine learning ». La vidéo est en ligne ici (2)). Par ailleurs j'exerce également la Data Science à titre professionnel, au Data Innovation Lab d'AXA.

(1) <https://www.kaggle.com/users/45978/christophe-bourguignat>

(2) <http://parleys.com/play/536763d7e4b04bb59f50270e>

## Conférencier

Christophe est un "data fan". Il pratique le machine learning au Data Innovation Lab du groupe AXA, ainsi qu'au travers de compétitions Kaggle ou Datascience.net.

Il fait partager sa passion à l'occasion de conférences spécialisées, ou sur son compte twitter @chris\_bour.

Ses outils préférés sont Python, scikit-learn, et pandas.

## Ce que je n'aime pas dans Python

**Conférencier** : Sébastien Douche

**Niveau de l'audience** : Intermediare

### Description

Pendant 7 ans, j'ai participé au développement d'un produit développé en Python / C (~180k SLOC) chez un éditeur logiciel. Je vous propose un retour d'expérience et quelques réflexions personnelles sur l'utilisation de Python sur un projet de cette taille. Je détaillerai ce qui est pour moi des points de friction avec le langage et les moyen de les contourner.

### Résumé

Python est un langage simple et flexible qui permet de prototyper rapidement une application. Sa puissance accompagne la montée en complexité de l'application. Mais que se passe t'il quand vous maintenez une application sur des années et qu'elles dépassent les 100 000 lignes de code ? Certains atouts du langage deviennent moins flagrants, certaines absences se font ressentir avec plus de force.

### Notes

L'objectif de cette session est de donner un retour d'expérience sur les points à mon sens négatif dans l'utilisation de Python sur un projet relativement gros et je l'espère faire réfléchir l'assistance sur ses pratiques de développement logiciel.

### Conférencier

Développeur amateur depuis 84, utilisateur de logiciel libre depuis 95, je suis un vieux Geek. Fainéant par nature, je suis à la recherche constante pour faire mieux avec moins d'effort. Actuellement en année sabbatique, j'ai occupé les postes de directeur technique et de responsable R&D chez un éditeur Français, me permettant d'assouvir mes passions pour le développement logiciel, l'administration système, le coaching technique et la gestion d'organisation.

# Posters

## Project of software for research in fluid dynamics

**Auteur** : Pierre Augier

**Niveau de l'audience** : Intermediate

### Description

Research in fluid dynamics would benefit from a Python (Cython and C) software developed collaboratively, with good documentation and testing framework. I am starting to develop this project. The program will in particular be useful to control motors, power supplies and acquisition boards, to carry out numerical simulations, to analyse data and to produce latex tables, figures and movies.

### Résumé

Research in fluid dynamics would benefit from a Python (Cython and C) software developed collaboratively, with good documentation and testing framework. I am starting to develop this project and will need some advice.

The program will in particular be useful to

- open and write files in many formats (for example using hdf5),
- control motors, power supplies and acquisition boards,
- carry out numerical simulations (efficient computation with an object-oriented solver using mpi4py, cProfile, Cython and C),
- analyse data (Particle Image Velocimetry and Light Induced Fluorescence, spectral analysis, energy budget, probability density functions, structure functions, flow decompositions as POD and DMD, wavelets) and
- produce latex tables, figures and movies.

The aim is to gather in one unified software efficient and tested tools written by the Python programmers working in the field.

### Auteur

Post-doc at DAMTP, University of Cambridge; Chercheur CNRS au LEGI, Grenoble

## **OSRM, le routing par excellence**

**Auteurs** : Anael Lorimier , Alexandre Allouche

**Niveau de l'audience** : Intermediare

### **Description**

Présentation d'une manière simple et efficace d'intégrer sur une appli web une interface de routage dynamique

### **Résumé**

Vous souhaitez intégrer sur votre site web un système de calcul d'itinéraire ?

Voire simplement un moyen de vérifier une adresse postale ?

Alors cette présentation est faite pour vous.

Après une rapide revue des serveurs de routage Open Source, je vous montrerai les fonctionnalités de l'un d'entre eux, OSRM, et une façon d'utiliser ses résultats pour faire du routing intelligent.

### **Auteurs**

Après avoir étudié le fonctionnement des ordinateurs avec un oeil externe, Anael s'est enfermé dans une cave pendant 2 ans afin de soulager sa soif de connaissances.

Un apprentissage entièrement dédié au C/C++ et Asm, ce qui lui a permis de comprendre les méandres de l'informatique.

Maintenant sur Python tout le long de la journée, et bluffé par les capacités de ce langage, il sera ravi d'échanger ses expériences avec quiconque.

## Science reproductible

**Auteur** : Nicolas Rougier

**Niveau de l'audience** : Intermediare

### Description

Un compte rendu sur une aventure e en "science reproductible" dans le domaine des neurosciences computationnelles.

### Résumé

Si la reproductibilité des expériences est une des propriétés fondamentales de la Science, la non-reproductibilité semble être un des marqueurs des modèles issus des neurosciences computationnelles. En voulant étudier un modèle de la littérature concernant les ganglions de la base (une structure du cerveau responsable de la sélection de l'action), nous nous sommes rendus compte qu'il était presque impossible de reproduire le modèle, même après avoir lu les 6000 lignes du code original (Delphi/Pascal). Après 2 mois de réécriture intense, nous avons finalement réussi à reproduire les résultats en 200 lignes de python, simplement en se servant des outils standards.

Lien vers le poster: <http://www.loria.fr/~rougier/tmp/Reproductible-Science.pdf>

### Auteur

Chargé de recherche en neurosciences computationnelles à l'INRIA ([www.inria.fr](http://www.inria.fr)) et je travaille au sein de l'Institut des Maladies Neurodégénératives à Bordeaux (<http://imn-bordeaux.org>).

# Une interface expérimentale pour lier OpenGL à Python

**Auteur** : Fabrice Salvaire

**Niveau de l'audience** : Intermediare

## Description

PyOpenGLng est un module visant à expérimenter de nouveaux concepts autour de l'intégration de l'API OpenGL au sein de Python. Les objectifs de ce module sont d'apporter une simplification par rapport à l'implémentation de référence PyOpenGL, d'implémenter une interface basée sur CFFI et d'offrir une API de haut niveau pour les versions modernes d'OpenGL

## Résumé

Jusqu'à présent les interfaces Python de l'API OpenGL étaient générées à partir d'un script qui lit le fichier d'en-tête C afin d'en extraire l'API. Cette approche n'a rien de choquant en soi et c'est d'ailleurs l'approche choisie par l'interface CFFI. Mais pour l'API OpenGL nous pouvons utiliser une autre approche puisque l'organisme en charge de la spécification d'OpenGL, le groupe Khronos, fournit un fichier XML appelé Registry qui décrit l'API et ses extensions officielles en fonction de la version, du profile (compatibilité ou core) et de sa variante desktop ou embarquée. Ce point est d'une grande importance car une des particularités de l'API OpenGL est qu'il n'y a pas une seule API mais plusieurs. L'autre particularité de l'API OpenGL est qu'elle est relativement simple puisque elle est uniquement composée d'énumérants et de fonctions mettant en œuvre des types fondamentaux. L'API ne définit par exemple aucun type de données composées.

L'interface de PyOpenGLng repose sur deux composants essentiels, le premier est une interface orientée-objet de l'OpenGL Registry et le deuxième est la partie qui gère l'interface C-Python à proprement parlé.

L'interface orientée-objet de l'OpenGL Registry permet de générer au vol l'API OpenGL souhaitée par l'utilisateur sous la forme d'un module Python, c.à.d. la liste des énumérants et des fonctions correspondant à l'API demandée. En comparaison PyOpenGL compte pas moins de mille fichiers pour couvrir toute l'API OpenGL et ses extensions. De plus il est relativement aisé d'obtenir la dernière version de l'API en téléchargeant le fichier XML depuis le serveur CVS de Khronos.

Au fil du temps l'API OpenGL a subit une révolution avec l'arrivée des GPUs qui a introduit à partir de la version 3 une approche basée sur un pipeline programmable. Cette nouvelle approche a occasionné des changements radicaux au niveau de l'API et sur la manière de travailler avec. Aujourd'hui OpenGL est essentiellement une interface orientée graphisme pour les processeurs de flux (GPU), de la même manière qu'OpenCL est une interface orientée calcul. Par conséquent l'usage de l'API en Python va essentiellement consister à modifier des variables d'états et échanger des tableaux Numpy via leurs adresses.

Cette relative simplicité est mise à contribution dans l'interface C-Python via une translation automatique basée sur les prototypes des fonctions. Ce mécanisme tient dans un seul fichier et requiert actuellement moins de 500 lignes de codes. Par conséquent il est relativement aisé de comprendre ce qui se passe entre le code utilisateur et l'API OpenGL. De plus il est implémenté avec ctypes et CFFI, ce qui permet d'être compatible avec l'interpréteur pypy.

Cette présentation sera l'occasion de faire un tour d'horizon des différentes approches possibles pour lier une bibliothèque C à Python et avec dans ce cas précis une approche originale puisque elle est basée sur

une API décrite par un fichier XML.

Afin d'illustrer l'utilisation de ce module, je présenterais brièvement un module Python qui permet d'interpréter les fichiers DVI générés par LaTeX et un viewer mettant en œuvre un rendu du texte accéléré par GPU.

Lien vers les dépôts Git et leurs documentations:

- PyOpenGLng dépôt et doc
- PyDVI dépôt et doc

### **Auteur**

Je suis chef de projets dans une biotech pour laquelle j'ai conçu une plate-forme logicielle de traitement et visualisation d'images issues d'un microscope automatisé. Ce logiciel est essentiellement implémenté en langage Python et repose sur des composants open source tel que Numpy, Scipy, Matplotlib, H5py, Pyqt, PyOpenGL et OpenCV.

J'ai participé l'année dernière à la conférence Euroscipy 2013 et publié un article dans le proceedings de la conférence: High-Content Digital Microscopy with Python, Fabrice Salvaire, Part of the Proceedings of the 6th European Conference on Python in Science (EuroSciPy 2013 (1)), Pierre de Buyl and Nelle Varoquaux editors, (2014) arXiv:1404.6385v2 (2).

Mes contributions à l'open source en relation avec le langage Python sont:

- PyOpenGLng (3) un wrapper expérimental pour l'API OpenGL,
- PyDVI (4) un module qui permet de lire les fichiers DVI générés par LaTeX et de les afficher avec un rendu accéléré par GPU en utilisant l'API OpenGL,
- PySpice (5) un module qui permet de simuler des circuits électroniques via Ngspice,
- PyResistorColorCode (6) un utilitaire pour décoder les codes de couleurs des résistances,
- LaptopControlPanel (7) un control panel pour laptop Lenovo,
- quelques plugins Sphinx tel que sphinx-css3image (8),
- et pour finir un gestionnaire de PDF (9) en gestation.

Mon attrait pour l'open source ne se limite pas au software mais aussi à l'open hardware. C'est pourquoi je suis un membre actif (10) du projet Electrolab V2 (11) géré par une association loi 1901 qui consiste à construire le plus grand hackerspace de France et d'Europe située à Nanterre en proche banlieue parisienne.

J'en profite pour faire un peu de publicité à cette aventure humaine: L'Electrolab V2 sera un lieu unique doté d'une surface de 1500 m<sup>2</sup> qui mettra à disposition de ces membres des moyens de production en mécanique et en électronique ainsi qu'un laboratoire de chimie et une salle blanche. Une salle serveur de bonne dimension supportera l'infrastructure informatique et l'appétit des hackers. Le lieu disposera aussi de salles de formation et de boxes entreprises. Pour nous soutenir (12)

Lien vers mon site perso : <http://fabrice-salvaire.pagesperso-orange.fr/>

(1) <https://www.euroscipy.org/2013/>

- (2) <http://arxiv.org/abs/1404.6385v2>
- (3) <https://github.com/FabriceSalvaire/PyOpenGLng>
- (4) <https://github.com/FabriceSalvaire/PyDVI>
- (5) <https://github.com/FabriceSalvaire/PySpice>
- (6) <https://github.com/FabriceSalvaire/PyResistorColorCode>
- (7) <https://github.com/FabriceSalvaire/LaptopControlPanel>
- (8) <https://github.com/FabriceSalvaire/sphinx-css3image>
- (9) <https://github.com/FabriceSalvaire/Biblio>
- (10) <http://www.pycon.fr/2014/reviews/review/86/>
- (11) <http://www.electrolab.fr/>
- (12) <http://donate.electrolab.fr/>

# Code de conduite

En tant qu'événement organisé par l'Association Francophone Python (AFPY), la charte de l'AFPY (1) s'applique. Pour mémoire, cette charte est reprise dans la suite de cette page.

(1) <http://www.afpy.org/doc/afpy/charte.html>

## Charte AFPY

Le but de l'Association Francophone Python (AFPY) est d'assurer la promotion du langage de programmation Python auprès du plus large public possible. Nous diffusons nos connaissances et sommes ouverts aux apports extérieurs car la diversité, l'écoute réciproque, la solidarité et la démocratie élargissent les horizons et accroissent le potentiel de l'AFPY et de chacun de ses membres. En conséquence, nous encourageons donc la participation aux divers événements que nous (co-)organisons de toutes les personnes de la communauté Python ou souhaitant en faire partie. Dans ce cadre, l'AFPY souhaite que toute personne intéressée soit accueillie de manière agréable et enrichissante.

L'AFPY s'attend donc à ce que chaque membre de l'association fasse preuve de respect et de courtoisie envers les autres membres de l'association et les personnes rencontrées dans le cadre de tout événement organisé (formellement ou non) par l'association.

Ce texte est notamment inspiré par le code de conduite établi par l'Ada Initiative et par la loi française de lutte contre les discriminations.

### La version "courte"

L'AFPY souhaite éviter tout type de discrimination, que ce soit sur le sexe, l'orientation sexuelle, le handicap, l'apparence physique, l'origine ou la religion, et sous quelque forme que ce soit (parole, image, texte et autre). Bien évidemment, ceci n'est pas une liste exhaustive.

Bien que l'AFPY ait pour but la promotion de Python dans le monde francophone, l'association s'engage à accueillir chacun, quelle que soit son langage de programmation ou sa langue maternelle.

L'AFPY attend de ses membres un langage et un style adaptés à une audience variée. Les allusions sexuelles ou à caractère raciste ne sont en aucun cas tolérées, quel que soit leur support.

Soyez poli(e). N'insultez personne. Discrimination et harcèlement sont à exclure aussi bien dans les écrits, les visuels, les paroles et les actes, et ce quelle que soit leur nature (sexuel, raciste, religieuse, sociale...).

Le non respect de ces règles de bienséance pourra entraîner l'exclusion de l'évènement, à la seule discrétion d'un ou plusieurs organisateurs, ou de l'association, après délibération du comité directeur, ainsi que de tout service proposé par l'AFPY (planète AFPY, blog, canaux IRC).

## La version “longue”

### *Champ d'application*

Ces bonnes pratiques s'appliquent dans le cadre de tout évènement (co-)organisé par l'AFPy de manière directe ou indirecte à savoir :

- les conférences (PyconFR, évènement à la cantine et autre)
- les canaux IRC ou XMPP ou autre messagerie instantanée
- planète python
- le blog
- la liste de diffusion
- tout évènement ou vous représentez l'AFPy, y compris les rencontres informelles (notamment AFPyros)

### *Définition de la discrimination*

La discrimination inclut, selon la loi française, « \_la situation dans laquelle, sur le fondement de son appartenance ou de sa non-appartenance, vraie ou supposée, à une ethnie ou une race, sa religion, ses convictions, son âge, son handicap, son orientation ou identité sexuelle ou son sexe, une personne est traitée de manière moins favorable qu'une autre ne l'est, ne l'a été ou ne l'aura été dans une situation comparable.» Est donc proscrit «tout agissement à connotation sexuelle, [subi] par une personne et ayant pour objet ou pour effet de porter atteinte à sa dignité ou de créer un environnement intimidant, hostile, dégradant, humiliant ou offensant ; Le fait d'enjoindre à quiconque d'adopter un [tel] comportement. »

### *Comportements à éviter (proscrire ?)*

En référence à la définition qui précède, nous vous invitons à vous abstenir

- de toute expression ou attitude inappropriées,
- de présenter des images sexuelles ou choquantes (notamment violentes),
- d'intimidation délibérée, de harcèlement, des contacts physiques inconvenants et non consentis.

### *Traitement des incidents*

En cas de harcèlement ou de discrimination commis par un participant, les organisateurs de l'évènement ou membres du comité directeur pourront prendre immédiatement les mesures qu'ils jugent appropriées et qui peuvent aller jusqu'à l'exclusion temporaire ou définitive d'un évènement.

### *Signaler*

Si vous êtes harcelé(e) ou victime de discrimination, ou si vous êtes témoin de harcèlement ou de discrimination à l'égard d'autre(s) participant(e-s), ou si vous craignez la survenue d'un incident, contactez sans attendre un membre organisateur (dans le cas d'un évènement) ou un membre du comité directeur de l'AFPy.

**Contact** : [diversite@afpy.org](mailto:diversite@afpy.org)

# Sponsors

## Gold



**HexACK**

<http://www.hexack.fr/>



**Akretion**

<http://www.akretion.com/>

Akretion est une Société de Services en Logiciels Libres spécialisée dans l'intégration et le déploiement d'Odoo (anciennement appelé OpenERP) dans les PME de 5 à 200 employés. Akretion France se spécialise plus particulièrement dans deux secteurs d'activité : l'e-commerce et la gestion de contrats de maintenance.

Nos compétences en comptabilité française et brésilienne nous permettent d'accompagner nos clients de bout en bout dans leur déploiement d'OpenERP dans ces deux pays. Nous sommes également habitués à travailler conjointement avec d'autres experts locaux pour intégrer les spécificités nationales ; nous avons ainsi des références aux Etats-Unis, en Espagne et en Autriche.



**ASK'interactive**

<https://www.askindev.com/>



**Yaal**

<http://www.yaal.fr/>

Yaal s'adresse aux créateurs de startups qui cherchent à concrétiser leur projet par la conception d'une solution web ou mobile. Nous nous engageons à leurs côtés en apportant notre expérience de la création d'entreprise, en partageant le risque financier et en prenant en charge la direction technique pour faire de leur idée un projet unique et durable.



**Fédération Lyonnaise de Modélisation et Sciences Numériques**

<https://flmsn.univ-lyon1.fr/>

La "Fédération Lyonnaise de Modélisation et Sciences Numériques" est une structure fédérative reconnue par le ministère et le CNRS. Elle a pour vocation de fédérer et soutenir les activités de calcul HPC et de modélisation dans la région lyonnaise. Elle regroupe le méso-centre de calcul HPC de

Lyon, partenaire du projet EQUIPEX EQUIP@MESO, ainsi que l'IXXI et le CBP.



## Faculté des Sciences et Technologies de l'Université Lyon 1

<http://sciences.univ-lyon1.fr/>

La Faculté des Sciences et Technologies de l'Université Lyon 1 rassemble 36 unités de recherches, dont 29 unités mixtes avec le CNRS. Certaines unités sont coordonnées dans des structures fédératives de recherche, ou partagent des ressources via des plateformes techniques, des centres communs, des collections.

Les structures de recherche sont en liaison étroite avec les sept départements disciplinaires : Biologie, Chimie-Biochimie, Génie Électrique et des Procédés, Informatique, Mathématiques, Mécanique, Physique, de façon à assurer la cohérence et la fertilisation croisée entre formation et recherche.



### alwaysdata

<https://www.alwaysdata.com/>

Modern hosting platform.



### XCG Consulting

<http://xcg-consulting.fr/>

Comme toujours, XCG Consulting est sponsor de PyconFR.

Cet événement est l'occasion d'appuyer et de promouvoir le langage Python utilisé dans le développement des solutions open source déployées par nos équipes afin d'optimiser les systèmes d'information de nos clients.

De par son savoir-faire technique et fonctionnel, XCG parvient à développer de véritables logiciels métiers à la fois robustes et flexibles.

PyconFR est donc le rendez-vous incontournable pour les développeurs passionnés de chez XCG.



### UNLISH

<https://www.unlish.com/>

UNLISH est un service mobile destiné à faciliter l'organisation de la pratique sportive libre.

Trouver des partenaire, des lieux, des activités adaptées à ses envies, son temps disponible ou son niveau, avec UNLISH nous vous aidons à (re)faire du sport !



## Clever Cloud

<http://www.clever-cloud.com/>

Clever Cloud is a Europe-based PaaS company. We help developers deploy and run their apps with bulletproof infrastructure, automatic scaling, fair pricing and other cool features. We aim to make an easy-to-use service, without any vendor lock-in and able grow with your needs.



## Bearstech

<http://bearstech.com/>

Bearstech est une jeune entreprise innovante, SCOP d'ingénieurs du libre, dont l'activité principale est l'infogérance et l'optimisation d'applications web complexes. (Services à valeur ajoutée autour de l'hébergement d'applications web et leur infogérance, Organisations et outils pour les équipes de développement et de production, Développement, Hébergement, Gestion de la montée en charge, Audits).

## Bronze



## Anybox

<http://anybox.fr/>

Anybox est un centre de services Odoo (ex-OpenERP) à forte orientation technique, pour associations, entreprises et collectivités. Nous proposons une maintenance corrective et préventive, audit, optimisations, évolutions, migrations, synchronisations, formations, développements de modules ou applications spécifiques, hébergement dédié ou mutualisé, ainsi qu'une expertise Python et Javascript.



## Novapost

<http://www.people-doc.fr/>

Novapost est le leader de la digitalisation des documents et demandes RH en Europe.

Novapost édite PeopleDoc, une plateforme RH SaaS globale qui facilite l'accès, le partage et la distribution sécurisée de l'information RH. PeopleDoc aide les grandes entreprises françaises et internationales à digitaliser leur fonction RH, de la gestion en ligne des processus de recrutement (signatures de contrats, collecte des justificatifs...) à celle des dossiers et des échanges avec leurs salariés (bulletins de salaire, entretiens annuels...). Elle assure leur conformité réglementaire, simplifie leurs processus et réduit leurs coûts.

En 2014, Novapost a été nommé « Cool Vendor » en Gestion du Capital

Humain par Gartner. L'éditeur affiche un portefeuille impressionnant de plus de 150 clients français et internationaux dont Bouygues Telecom, Canal+ Groupe, Eiffage, GDF Suez, Lyonnaise des Eaux, Nexity, Schneider Electric, Starbucks Coffee, Total et Vinci Energies GSS.

Novapost est présent aux Etats-Unis, à New-York, sous le nom de PeopleDoc.



Red Hat

<http://www.redhat.com/>

Red Hat is the world's leading provider of open source software solutions, using a community-powered approach to reliable and high-performing cloud, Linux, middleware, storage and virtualization technologies. Red Hat also offers award-winning support, training, and consulting services. As the connective hub in a global network of enterprises, partners, and open source communities, Red Hat helps create relevant, innovative technologies that liberate resources for growth and prepare customers for the future of IT.

Learn more at <http://www.redhat.com/>

## Partenaires



### Département informatique de la Faculté des Sciences et Technologie

<http://fst-informatique.univ-lyon1.fr/>

Le département informatique est l'un des 8 départements de la Faculté des Sciences et Technologie. Le département accueille environ 600 étudiants, répartis dans 11 formations qui s'appuient sur 4 laboratoires de recherche et une cinquantaine d'enseignants-chercheurs, assurant ainsi le développement de la recherche et de la formation en informatique de l'Université Claude Bernard Lyon 1.



### LIRIS

<https://liris.cnrs.fr/presentation/presentation-du-laboratoire>

Le LIRIS, regroupant 320 membres, est une unité mixte de recherche (UMR 5205) dont les tutelles sont le CNRS, l'INSA de Lyon, l'Université Claude Bernard Lyon 1, l'Université Lumière Lyon 2 et l'Ecole Centrale de Lyon. Le champ scientifique de l'unité est l'Informatique et plus généralement les Sciences et Technologies de l'Information.



## Éditions Diamond

<http://boutique.ed-diamond.com/>

Le magazine de référence technique pour les Développeurs et les Administrateurs sur systèmes Open Source !



## Kozea

<http://kozea.org/>

L'expertise de l'Open Source au Service de l'Entreprise.

Dès 2007, Kozea a su constituer une équipe d'experts réunis autour d'un seul objectif : développer des solutions Web performantes adaptées aux besoins des entreprises.

Kozea intervient dans divers domaines d'activité (Sécurité industrielle, Pharmacie, Médecins, etc.) en proposant des solutions personnalisées, innovantes et compétitives.