JE FAIS ENCORE DU PYTHON2 EN 2024

Pourquoi s'infliger cette douleur ?

Julien Lenormand

KAIZEN SOLUTIONS

INTRODUCTION

PRÉSENTATION DU SPEAKER

Julien Lenormand

Ingénieur informatique chez Kaizen Solutions

Pythonista depuis ~2009

J’ai répondu en 2022 à une offre d’emploi pour faire du Python …

2 !

MINUTE DE SILENCE POUR PYTHON2

Il nous a quitté il y a presque 5 ans …

PETIT SONDAGE

- Qui a déjà fait du Python 2 ?
- Qui en a fait il y a moins de 4 ans ?
- Moins d’1 an ?

UNE HISTOIRE DE PYTHON

UNE LONGUE HISTOIRE (VERSIONS 0.X À 2.X)

Python démarre en Décembre 1989

rendu public en 1991

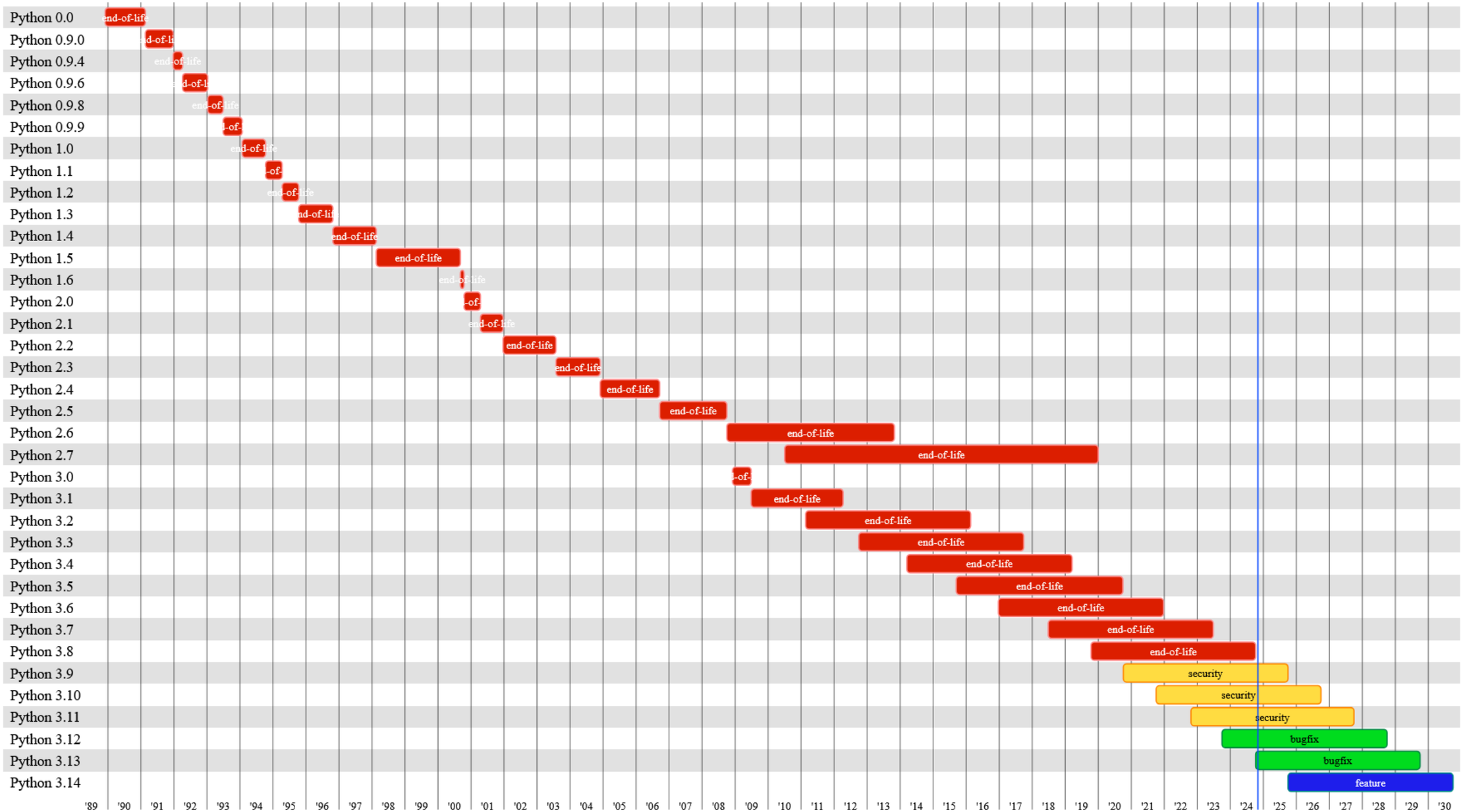
2.0 en 2000 (What's new ?) :

- passage sur SourceForge pour faciliter la collaboration
- introduction des PEPs (PEP-1 - PEP Purposes and Guidelines)
- type Unicode pour les strings
- et bien d'autres choses …

Et les versions s'enchaînent jusqu'à …

- 2.6 en 2008
- 2.7 en 2010, PEP-373 : dernière de la branche 2.x
- pas de 2.8 (PEP-404)

UNE LONGUE HISTOIRE (EN PERSPECTIVE)



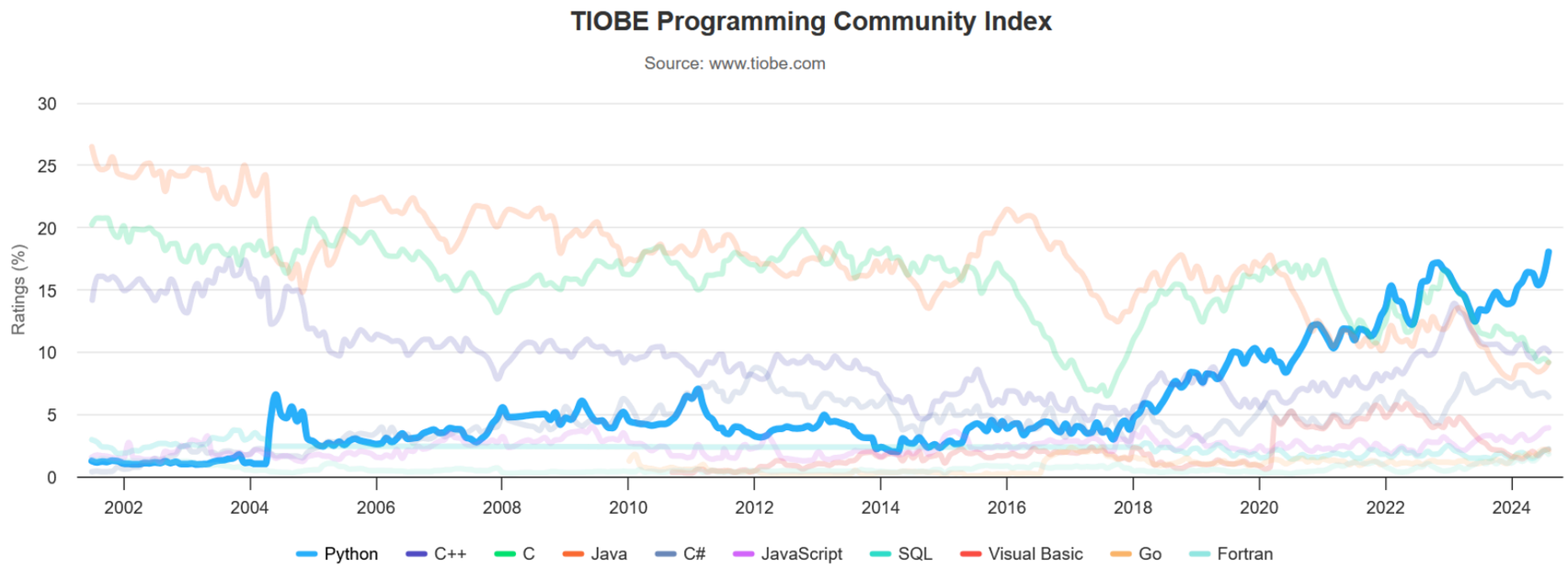
UNE BELLE HISTOIRE (V3.X)

Par rapport à la `2.x` (PEP 3100) :

- division flottante vs entière
- que des *new-style* `classes`
- fonction `print`
- plus de *string exceptions*
- `!=` plutôt que `<>`
- plus de `long`
- **Unicode** par défaut !
- tout est lazy (“x”)
- plus de `raw_input`
- *context manager*: `with`
- *PEP process*
- *implicit namespace packages*
- `set` et `dict` *literals*
- `super()` sans paramètre
- `nonlocal`
- …

Jusqu’à la `3.13` :

UNE HISTOIRE TRAGIQUE



UNICODE

L'absolu minimum que chaque dev doit vraiment, absolument savoir à propos d'Unicode et des jeux de caractères (pas d'excuses !)

Joel Spolsky - 2003

Arrêter avec l'ASCII (cf)

C’EST LA GUERRE !

TODO musique de sirène d’alarme, de tirs

PREMIÈRES ÉCHAUFFOURÉES

- les versions 3.0 (2008), 3.1 (2009) et 3.2 (2011) :
  - API peu stables, mauvaises perf, et bugs

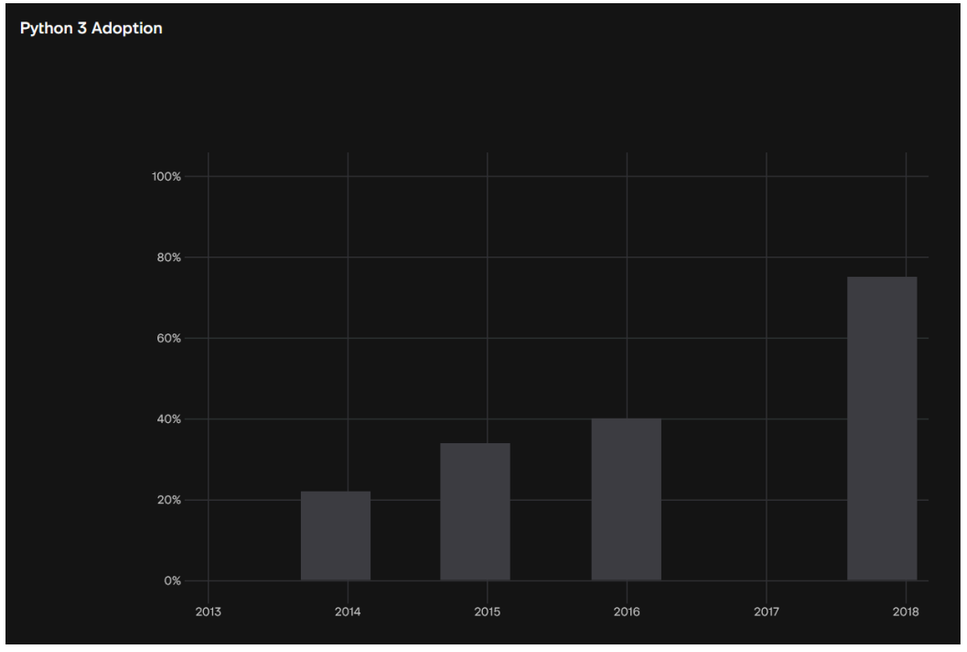
faible adoption

PEP 404 - Python2.8 Un-release Schedule

LE CONFLIT S'ENLISE

ça marche, pourquoi payer pour changer ?

- Second-system effect
- Things You Should Never Do, Part I : do not rewrite
- 2.6 EOL en 2013, 2.7 EOL en 2016 (2010) puis EOL en 2020 (2014)



LA COURSE À L'ARMEMENT

Des efforts sur la compatibilité entre 2.6/7 et 3.x :

- la 3.3 a une syntaxe assez proche des 2.6/2.7
  - front-porting : `u""`
- des backports sont réalisés
  - `from __future__ import` : `division`, `absolute_import`, `print_function`, `unicode_literals`
  - `backports`: `abc`, `csv`, `inspect`, `os`, `range`, `statistics`, …

Des outils pour assister à la migration :

- `2to3` : transpile du 2.x en 3.x
- `python-modernize` : comme 2to3 mais en 3.x compatible 2.x
- `3to2` : transpile du 3.x en 2.7
- `lib3to6` : pareil que lib3to2
- `future` : `futurize/pasteurize` : transpile du x.y en 3.x compatible 2.x
- `six` : couche de compatibilité
- `nine` : comme six, mais avec le nommage 3.x
- `eight` : polyfills automatiques
- `py3c` : couche de compatibilité pour l'API C
- `pylint --py3k` : warnings sur le 2.x-only
- `python2.6 -3` : warnings sur le 2.x-only

Des guides :

- PEP 3000, Compatibility and Transition : “The recommended development model …”
- doc officielle
- “conservative”
- Python3 Statement - Practicalities
- Porting to Python 3 Redux
- Twisted’s Reviewer check list

Des listes :

- Python3 statement : librairies qui s’engagent à dropper le support de Python2
- Python3 Wall of Shame / Python3 Wall of Superpowers : Top100 PyPI
- Python3 Readiness : Top 360
- Can I Use Python 3?

LA VICTOIRE

pas de spoiler : Python3 a gagné, Python2 a été sunset

dernière release de la 2.7 (.18) le 04 Avril 2020, juste avant la PyCon US

- les libs et les applications ont été portées
- la bi-compatibilité a été dropée au fur-et-à-mesure
  - 2016 : Django, 2017 : Pandas, 2018 : Numpy, 2020 : Pip, Celery et Twisted, …
- les environnements (hébergeurs Web, OS, compute, …) ne proposent plus Python2
  - 2018 : RHEL, 2022 : Debian, 2024 : Google App Engine, …
- les conférences n'en parlent plus (🙃)

DES POCHES DE RÉSISTANCE

- Ubuntu 18.04 (2018) a son support commercial (incluant Python2) jusqu'en 2028
- RHEL 7 (2014) a son support commercial (incluant Python2) jusqu'en 2028
- PyPy étant basé sur Python2, annonce son support "pour toujours"
- ActiveState continue de proposer une version 2.7 maintenue (payante)
- Tauthon est "une sorte de 2.8"
    - > *Ok, bring in the lawyers. - gvanrossum*
- OilShell utilise une version forkée et réduite de la 2.7
- d'après le JetBrains Python Survey 2023, ~6% des Pythonistas continuent de faire du Python2

LE PRIX DE LA GUERRE

- la communauté a été fracturée, et blessée
- prise d'otage ?
  - non, merci l'Open Source !
- servi d'exemple de ce qu'il ne faut PAS faire (cf Perl/Raku)
  - très bon exemple de network effect
- a coûté TRÈS cher ("billion-dollar mistake" ?)
  - Guido reconnaît avoir fait le mauvais choix, avoir sous-estimé les efforts requis
- indicateur de qualité des projets : env (version et dépendances), tests, hacks
  - LES TESTS !!
  - mais aussi les environnements et le packaging
  - et de la non-contribution à l'open-source en général par les entreprises
- la GIL est restée (jusqu'en 3.12~13+)
- il n'y aura jamais de Python 4 ! (en fait si, en cas de changement majeur de la C-API)

CONCLUSION

Should I use Python 2 or Python 3 for my development activity?

Python 3 is strongly recommended for any new development.

- Python2 n'est pas mort, et ne mourra peut-être pas avant 2040.
- Mais sauf si on me l'impose, Python2 n'est plus un "choix".
- Je pratique le Python3.6+, avec l'outillage et le typing moderne, et j'en suis heureux.

Et ça s'inscrit dans la raison d'être de Python :

- la glue entre différents langages ayant une C-FFI
- un langage plaisant à utiliser (comparé aux autres)

SOURCES

Principales, en plus des liens dans les slides :

- *.python.org
- Guido van Rossum: BDFL Python 3 retrospective
- Python FAQ: Why should I use Python 3?
- Python 3 Q & A

CRÉDITS

Images :

- Image de couverture (“Python Meetup Grenoble”) par Pierre-Loïc Bayart
- Image d’introduction (“Python is hell”) par Pierre-Loïc Bayart
- Photo de Simon Hurry sur Unsplash
- Photo de Walkator sur Unsplash
- Photo de Hasan Almasi sur Unsplash
- Photo de Vladislav Babienko sur Unsplash
- Photo de Christina @ wocintechchat.com sur Unsplash
- Capture d’écran du site de la PyCon Fr 2024 par moi-même

Web :

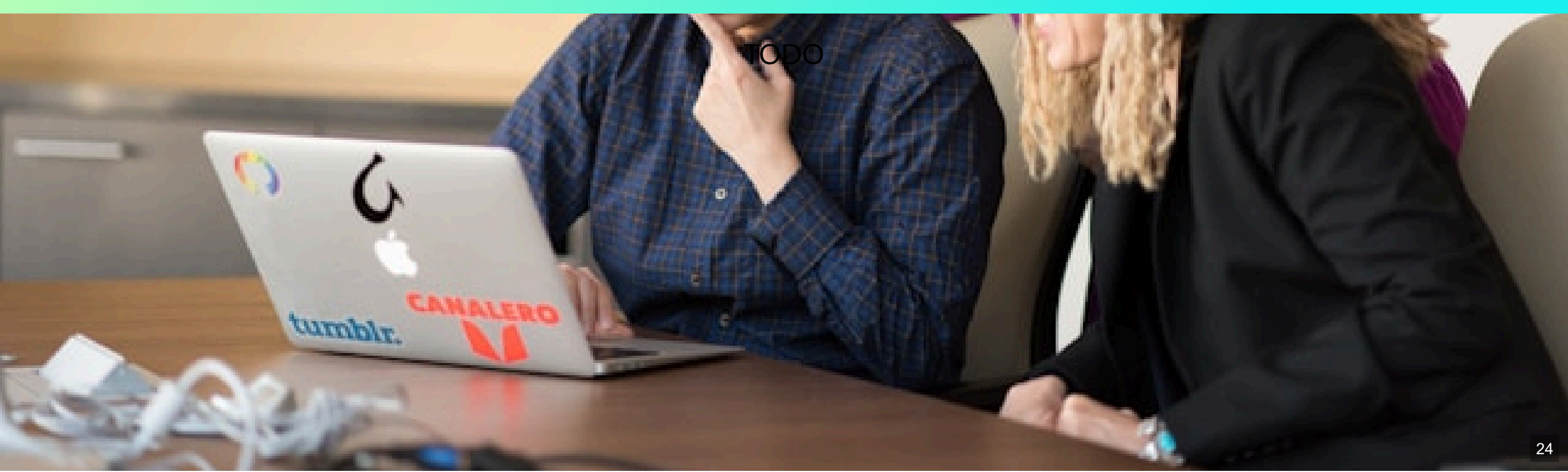
- The Wayback Machine: Web.Archive.org
- GitHub.com
- PyPI.org

POUR ALLER PLUS LOIN

- Top 10 articles de Joel “on Software” Spolsky
- ma conférence “Rambo Python”
- faire l’exercice de coder quelque chose en Python 2.6 !
- porter du code legacy

LIVE CODING

TODO



Platine

UFR de mathématique et d'informatique
Université de Strasbourg

EuroPython Society

python SOFTWARE FOUNDATION

Or

KRAKEN

SWEPY Software Engineering Python

Argent

Alma

entr'ouvert

MAKINA CORPUS

Bronze

gandi.net

camp to camp
INNOVATIVE SOLUTIONS BY OPEN SOURCE EXPERTS

TeMPO CONSULTING
Conseils en Services en Logiciels Libres

YAAL

pass Culture

API SOCIETY

django

Julien Lenormand
#human #craft

Cœur

Association PauLLA Pour Les Logiciels Libres

Women Techmakers
Paris and Global Friends

Partenaires

AD ALSACEDIGITALE
CONSTRUIRE L'ÉCOSYSTÈME NUMÉRIQUE EN ALSACE

ATUS

QUESTIONS

?

MERCI

JULIEN LENORMAND

Dev / DevOps / Craft / Talk

LinkedIn/julien-lenormand

KAIZEN SOLUTIONS

www.kaizen-solutions.net

--

contact@kaizen-solutions.net

KAIZEN SOLUTIONS

26, avenue Jean Kuntzmann

38330 Montbonnot-Saint-Martin