Python

Il buono, il brutto, il cattivo

→ 23 février 2024

onepoint.
beyond the obvious

Bérengère MATHIEU

- Formation en analyse d'image
- Thèse analyse d'image / IA

- Makina Corpus
- R&D
- Formation

- Onepoint
- Data scientist
- Data engineering

Homem

- Feuilleter le « Tour du monde en 80 jours »
- S’arrêter sur une page
- Extraire le texte (OCR)
- Identifier le chapitre (Search)
- Illustrer ce chapitre (IA générative)

Arielle LIVIS

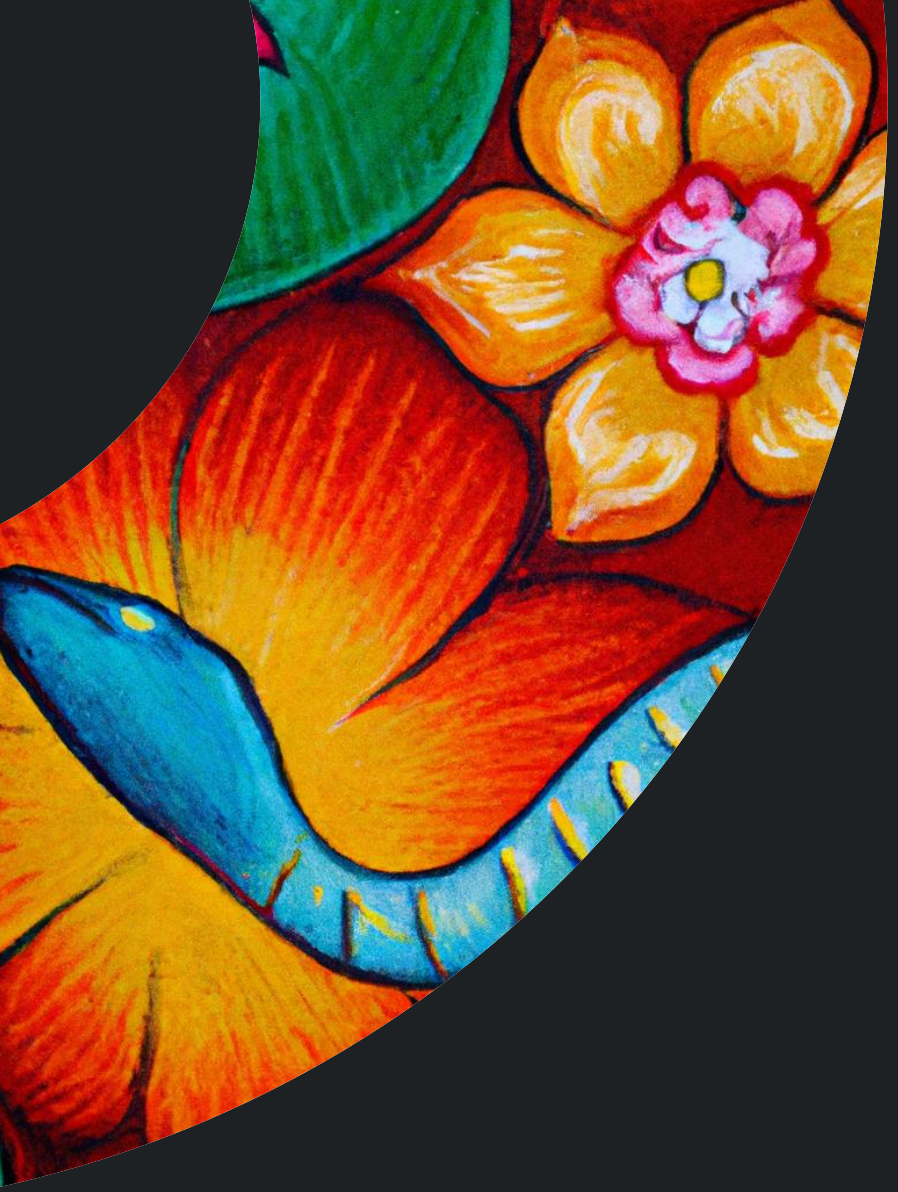
- Assistant de direction
- Agent de la fonction publique

- Devops
- Cloud
- Homem -> Prompt engineering

Bérengère : « Le Python ? C'est un langage il fait un peu tout mais pas très bien… »

Au menu

1. Il buono (le bon Python)
2. Il brutto (le Python moche)
3. Il cattivo (le Python méchant)



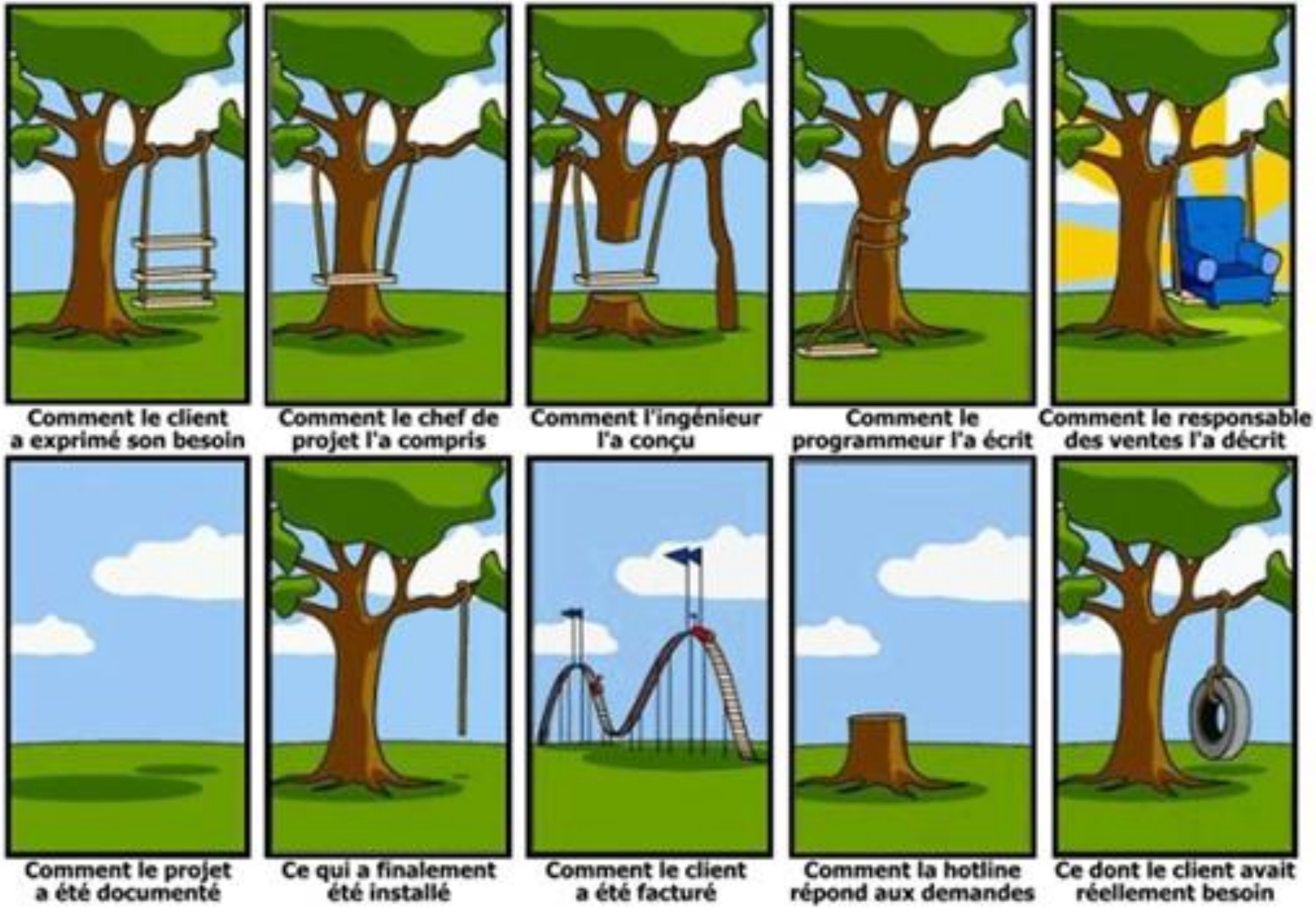
Il buono

Le bon Python

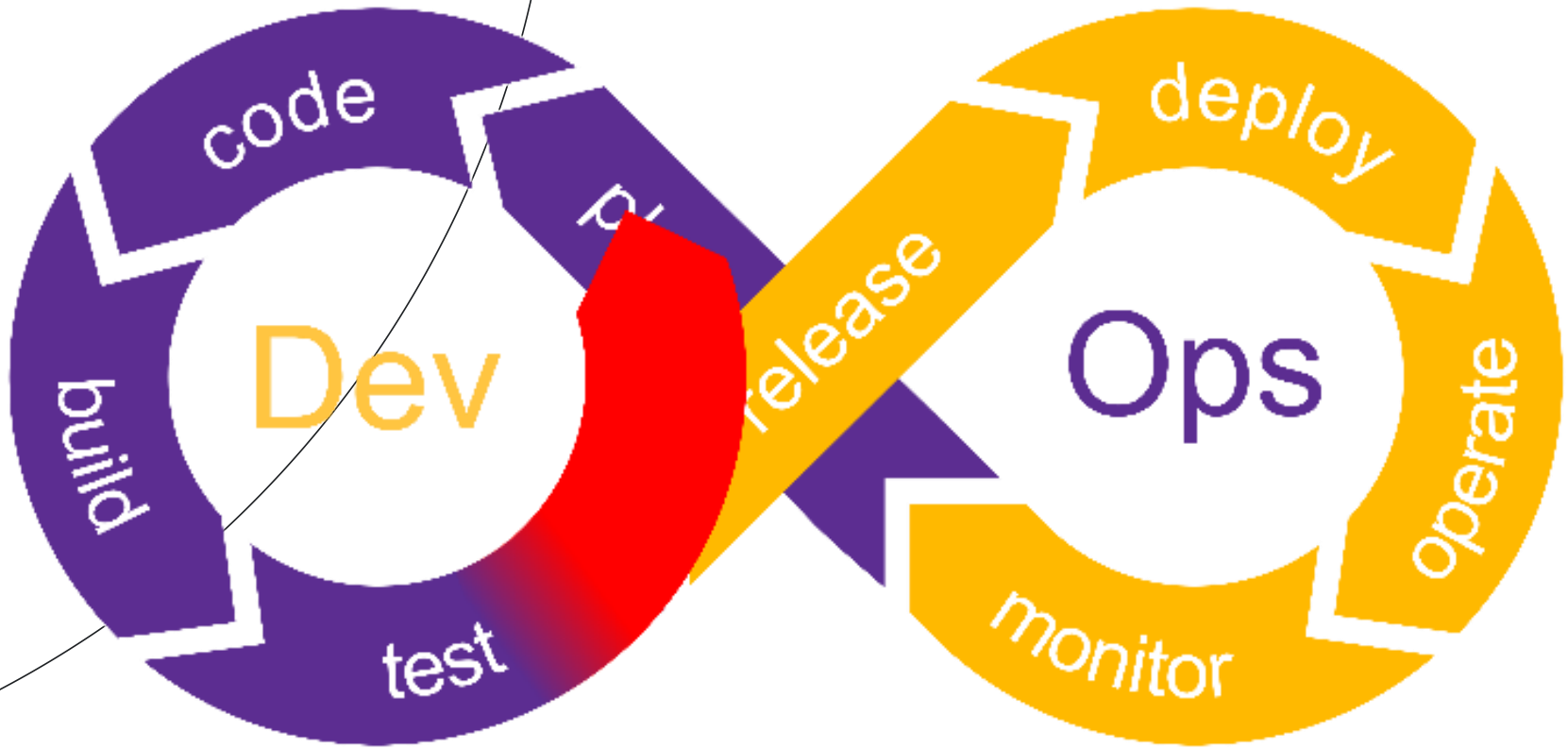
« Il avait été un ange, jadis. Il n'avait pas cherché à déchoir. Il avait eu de mauvaises fréquentations, voilà tout. »

Terry Pratchett & Neil Gaiman, De bons présages

Agile and how Agile works



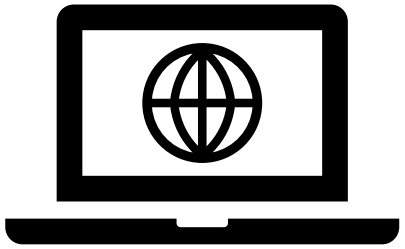
Le DevOps



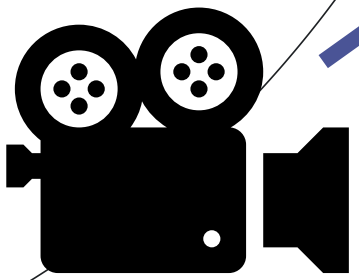
Le Devops par P.MORISSEAU

Homem

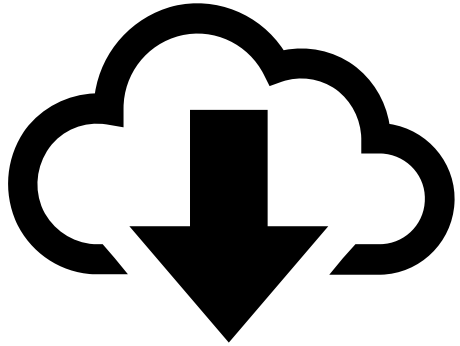
Un projet Python



Application Web



Analyse temps réel
du flux vidéo



Données générées en amont
et hébergée dans le Cloud

DEVOPS

IA

R&D

EVITER UN PROMPT TREPAS

- IA domaine très médiatisé
- Médiation avec les clients essentielles
- Vocabulaire lié à l'IA très spécifique
- Risque d'incompréhension ou de quiproquo avec le reste de l'équipe ou les autres équipes

- Point sur le vocabulaire employé
- Documentation
- Convention de nommage

- DocString
- Sphinx => documentation
- Pylint => vérifier la qualité du code

EVITER UN PROMPT TREPAS

- R&D => nombreuses pistes explorées en même temps
- Tendance à produire beaucoup de code mort
- Crée de l'instabilité dans le code

- Simplifier l'utilisation du socle technique CI/CD
- Extraire la partie prompt-engeenering du code

- Pylint => vérifier la qualité du code
- Flake8 / Black => formatage automatique
- Pytest => mise en place de tests unitaires
- Tox => automatisation des tests

Questions

Il brutto

Le Python moche

« Il y a quelque chose de profondément démoralisant, troublant, dans ces brusques transformations d'une bête paisible et que vous croyez connaitre en une créature féroce et comme entièrement autre »

Romain Gary, Chien blanc

Fondations

Communauté

- open-source (Python Software Fondation)
- Inclusif (python-docs-fr)

Responsabilité

- Peu de contrôle
- Programmation multiparadigmes

Haut niveau

- Gestion mémoire automatisée
- Structures de données très performantes

Allocation mémoire

Gestion de la mémoire particulièrement inefficace pour les nombres et les chaînes de caractères

```
i = 0
i = i +1
```

1. Allouer une zone mémoire

2. Stocker 0 dedans

3. **Mettre à jour** la valeur dans la zone mémoire déjà allouée

Le reste du monde

1. Allouer une zone mémoire

2. Stocker 0 dedans

3. Allouer **encore** une zone mémoire

4. Stocker i + 1 dedans

Python

Et vous faites de l'IA avec ça ?

NumPy

pandas

TensorFlow

Typage dynamique

```
ma_liste = [1, 2, 3]
chaine_caracteres = "Il était une fois"
entier = 2
reel = 3.14
reel = "Il était une fois" # à éviter ☺

def ma_function(devine_qui_cest):

    devine_qui_cest.append(0)
    define_qui_cest.sort()
```

Mutable VS non mutable

- De manière générale aucun objet n'est modifiable en Python *i.e* il faut allouer une zone mémoire pour chaque modification apportée => très lent

- Ajout de deux structures modifiables, les **listes** et les **dictionnaires**

MAIS

```python
def f(a):
    a += 1
    print(f"a: {a}")

v = 10
f(v)
print(f"v: {v}")
```

```
a: 11
v: 10
```

```python
def f(l):
    l.append(0)
    print(f"l: {l}")

my_list = [10, 10]
f(my_list)
print(f"my_list: {my_list}")
```

```
l: [10, 10, 0]
my_list: [10, 10, 0]
```

Mutable VS non mutable

- De manière générale aucun objet n'est modifiable en Python *i.e* il faut allouer une zone mémoire pour chaque modification apportée => très lent

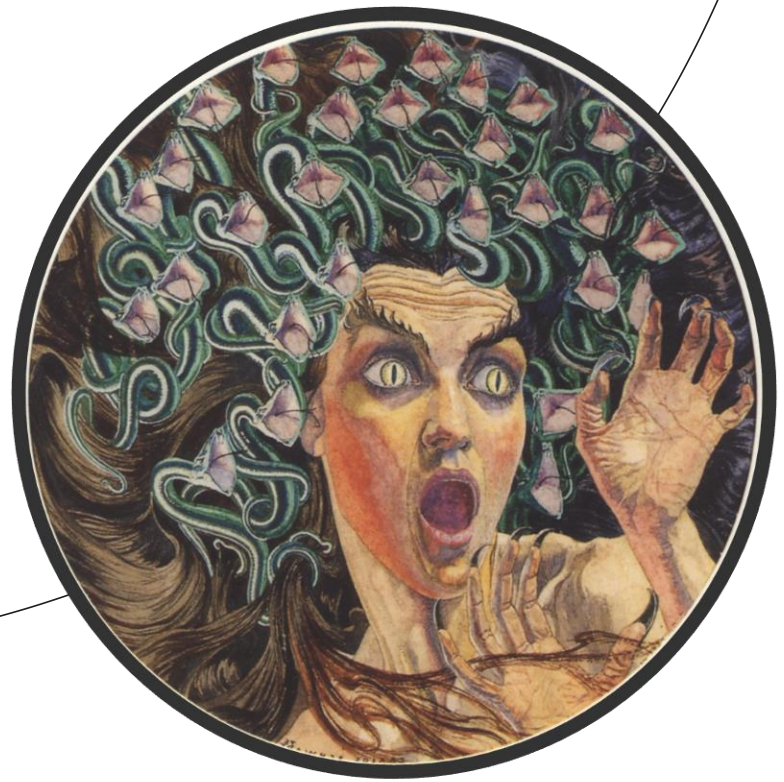
- Ajout de deux structures modifiables, les **listes** et les **dictionnaires**

MAIS

```
from copy import deepcopy
list_1 = [[1, 2, 3], [4, 5, 6]]
list_2 = deepcopy(list_1)
list_2[0].append("Soleil !")
print(f"liste_1 : {list_1}")
print(f"liste_1 : {list_2}")
```

```
liste_1 : [[1, 2, 3], [4, 5, 6]]
liste_1 : [[1, 2, 3, 'Soleil !'], [4, 5, 6]]
```

Mutable VS non mutable

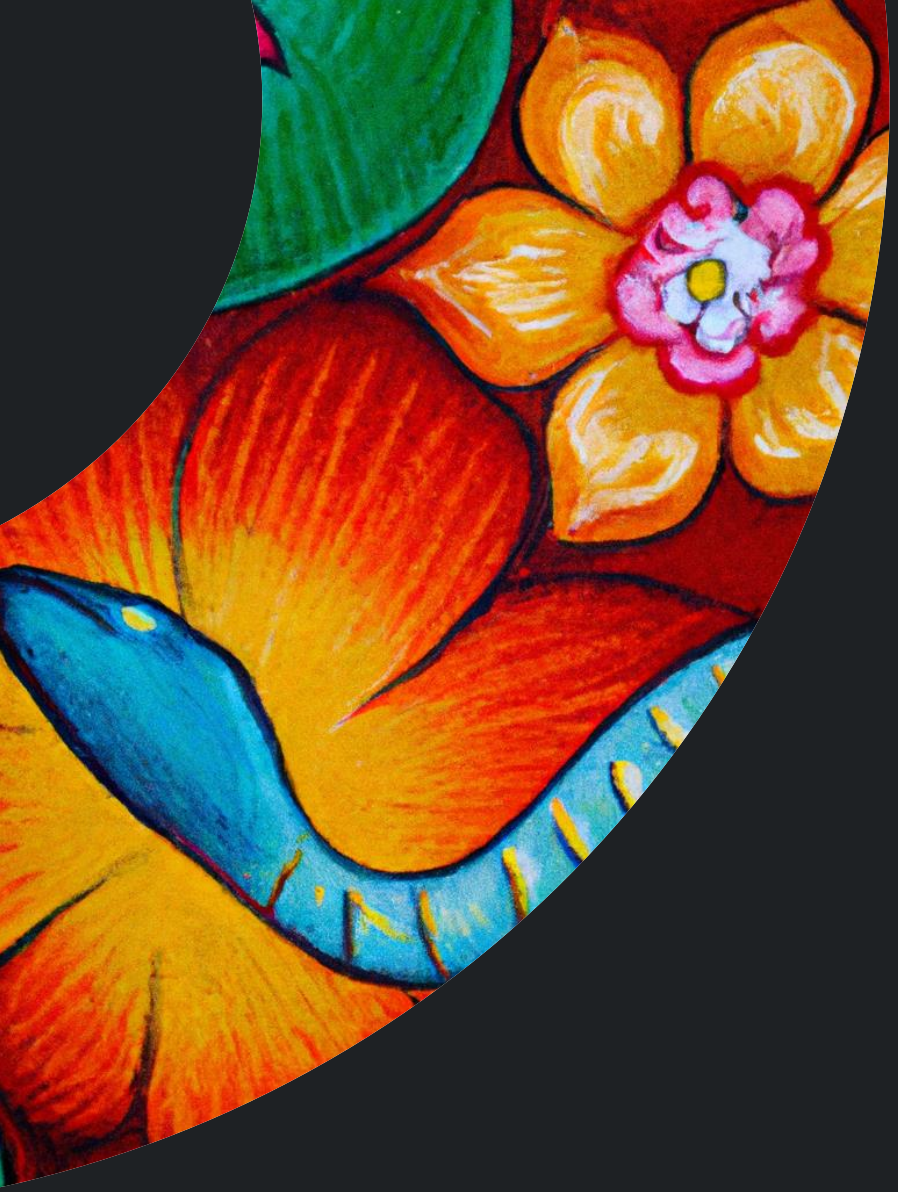


```python
def f1(list_1=None):
    if list_1 is None:
        list_1 = []
    list_1.append(0)
    print(list_1)

f1([1])
f1()
f1()
```

```
[1, 0]
[0]
[0]
```

Questions



Il cattivo

Le Python méchant

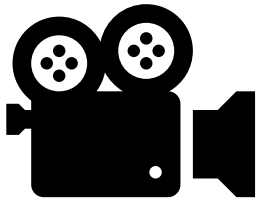
« Pour qui sont ces serpents qui sifflent sur vos têtes »

Jean Racine, Oreste

Environnement virtuel

- Besoin :
  - Cohabitation de plusieurs projets
  - Partager un projet
- Principe :
  - Répertoire contenant interpréteur Python & bibliothèques associées à un projet
  - Construction de ce répertoire à partir d'un fichier de dépendances (requirements.txt)
- Outils :
  - les basiques : venv, virtualenv
  - les sophistiqués : conda, pipenv , poetry

Homem

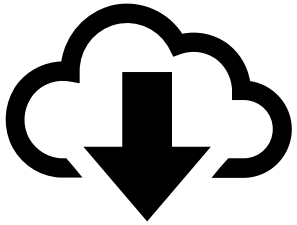


Analyse temps réel du flux vidéo

- Accès au driver de la caméra nécessaire
- Environnement virtuel faute de mieux

Application web

- Passage de environnement virtuel à docker
- En cours : automatisation « mise en production »



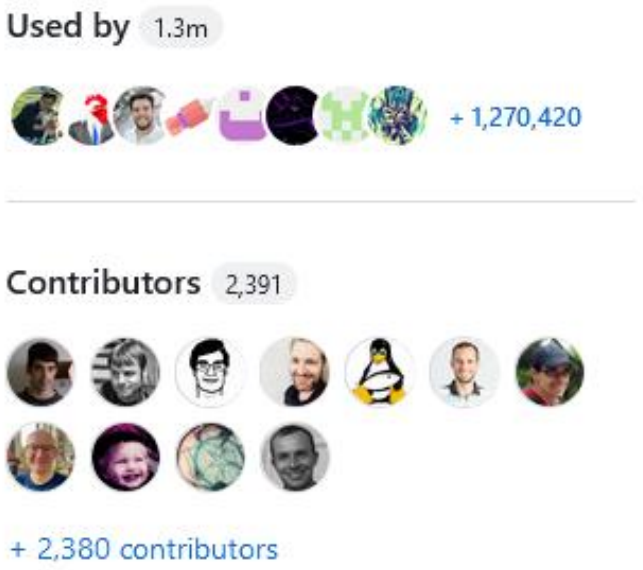
Données illustration chapitres

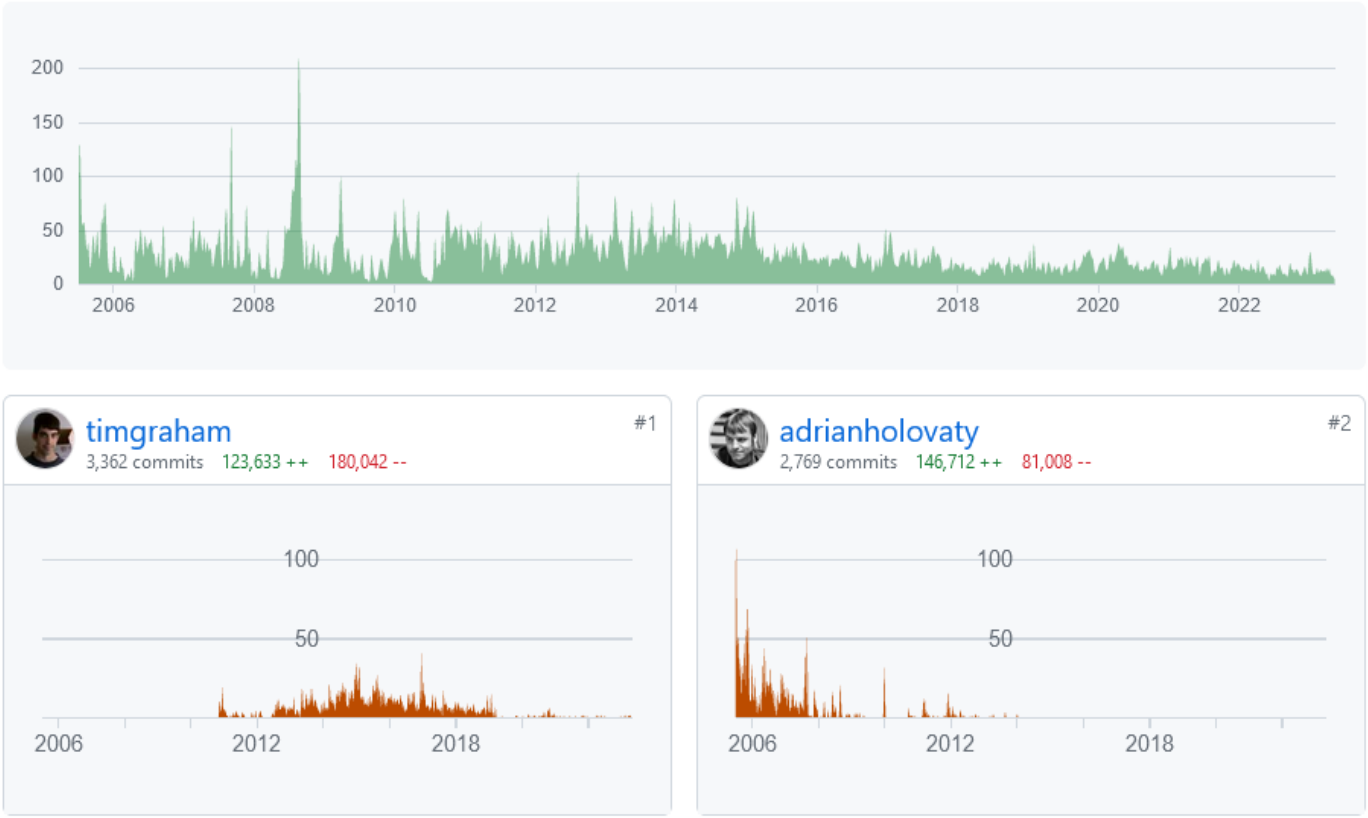
- Environnement virtuel
- Passage à docker en discussion
- MLOps

Bibliothèques & packages

Django

- Il y a (pratiquement) une bibliothèque Python pour tout
- Toutes sont cools, mais certaines le sont plus que d'autres

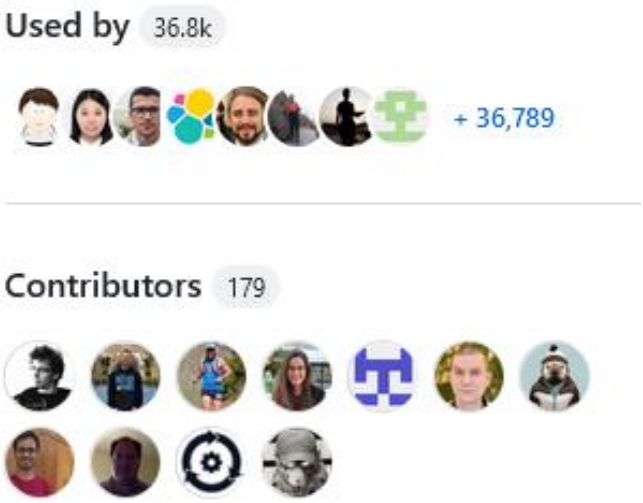


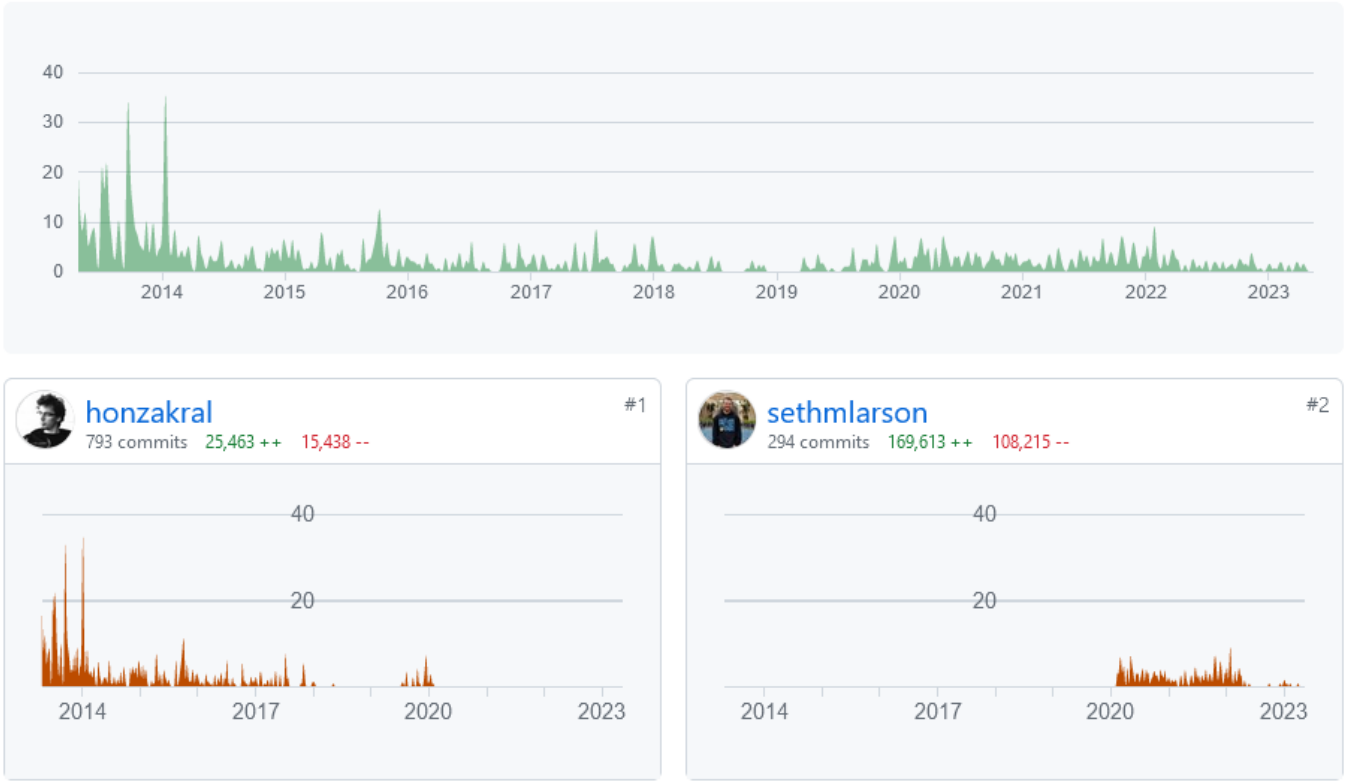


Bibliothèques & packages

Elasticsearch-py

- Il y a (pratiquement) une bibliothèque Python pour tout
- Toutes sont cools, mais certaines le sont plus que d'autres

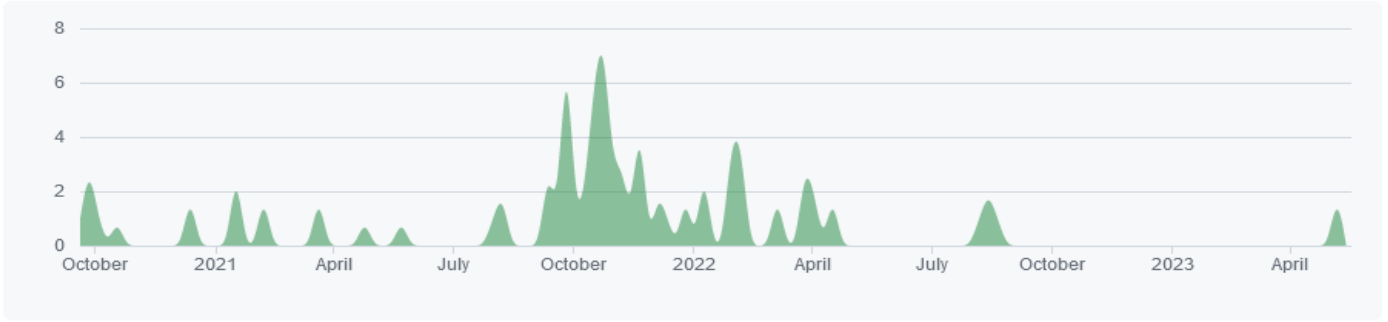


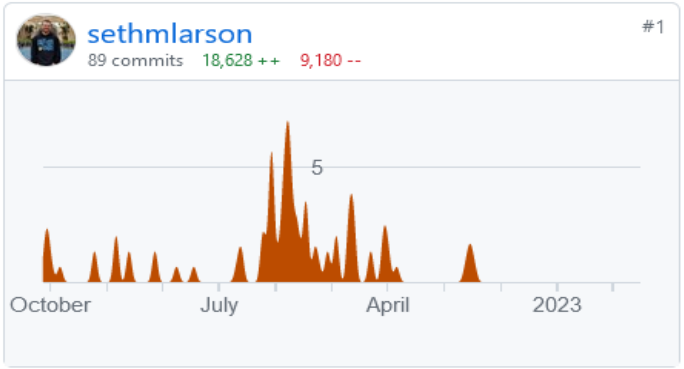


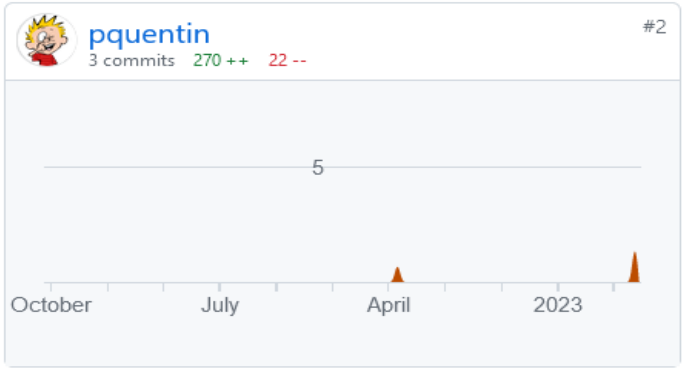
Bibliothèques & packages

elastic-transport-python

- Il y a (pratiquement) une bibliothèque Python pour tout
- Toutes sont cools, mais certaines le sont plus que d'autres







Contributors 2

sethmlarson Seth Michael Larson

pquentin Quentin Pradet

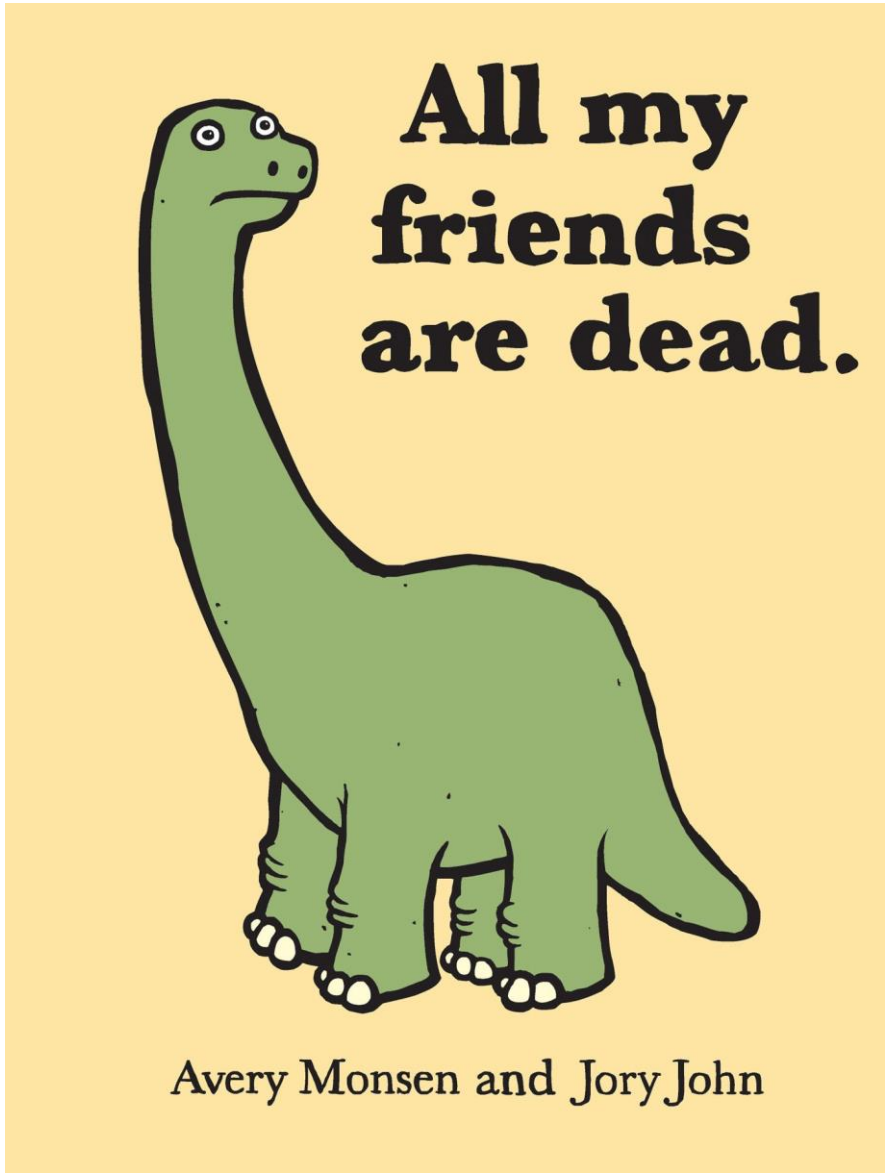
Bibliothèques & packages

elastic-transport-python

Contributors 2


sethmlarson Seth Michael Larson


pquentin Quentin Pradet



Arbitrage du juge de paix

- Arbitrage concernant la prise de risques, les coûts, les bénéfices, l'apport innovation
- Arbitrage vis-à-vis des attentes du client / équipes Dev & Ops
- Arbitrage entre l'équipe Dev et l'équipe Ops
- Assurer la cohérence du projet de bout en bout

Programmation orientée objet

- ✓ Créer une classe avec attributs et méthodes
- ✓ Attributs & méthodes privées

```python
class Livre:
    nb_livres = 0

    def __init__(self, titre, auteur):
        self.titre = titre
        self.auteur = auteur
        Livre.nb_livres += 1
        self.__private_attribut = 'something'

    def __str__(self):
        return f"{self.titre} - auteur: {self.auteur}"

    def afficher_nb_livres():
        print(f"Nombre de livres {Livre.nb_livres}")

    def __private_joke(self):
        print("here be dragons")

livre_1 = Livre("Chien Blanc", "Romain Gary")
print(livre_1)
Livre.afficher_nb_livres()
```

```
Chien Blanc - auteur: Romain Gary
Nombre de livres 1
```

Programmation orientée objet

- ✓ Créer une classe avec attributs et méthodes
- ✓ Attributs & méthodes privées
- ✓ Héritage

```python
class BD(Livre):

    def __init__(self, dessinateur, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.dessinateur = dessinateur

    def __str__(self):
        livre_str = super().__str__()
        return f"{livre_str}, dessinateur: {self.dessinateur}"


bd_1 = BD("Le grand pouvoir du Chninkel", "Jean  Van Hamme", "Grzegorz Rosinski")
print(bd_1)
Livre.afficher_nb_livres()
```

```
Jean  Van Hamme - auteur: Grzegorz Rosinski, dessinateur: Le grand pouvoir du Chninkel
Nombre de livres 2
```

Programmation orientée objet

- ✓ Créer une classe avec attributs et méthodes
- ✓ Attributs & méthodes privées
- ✓ Héritage

- ☹ Possibilité de modifier une instance
- ☹ Pas d'attributs ou méthodes protégés
- ☹ Pas d'interface
- ☹ Difficultés liées au typage dynamique

PEP 3124, 3119

```
livre_1.EAN = '9782070360505'
print(f"EAN {livre_1.titre} : {livre_1.EAN}")
```

```
EAN Chien Blanc : 9782070360505
```

Médiagraphie

Agile and how Agile works
Agile_PPT1.pptx (slideshare.net)

Le DevOps
https://pmorisseau.ineaweb.net/articles/devops/vrille/02.pourquoi.devops/

Arbitrage du juge de paix
File:Honoré Daumier, Les Voisines devant le Juge de paix, 1845, NGA 57245.jpg - Wikimedia Commons

Questions ?

« De temps en temps, j'allais rendre visite à mon python.
... je m'installais, les jambes croisées, en face de lui et nous nous regardions longuement avec un étonnement, une stupéfaction sans bornes

[...]

Se trouver dans la peau d'un python ou dans celle d'un homme était un avatar tellement ahurissant que cet effarement partagé devenait une véritable fraternité. »

Romain Gary, Chien blanc