



Les nouveautés de Python 3.11



www.communautesmetiersbyextia.fr



- 1 Optimization and Performance
- 2 Exceptions
- 3 Typing improvements
- 4 Major Language and Modules Changes



Faster CPython, un projet ambitieux

- Initiative: Microsoft et Guido Van Rossum
- Objectif: **x5** en 2025 \Rightarrow 1.5/an
- basé sur **HotPy** et **HotPy2**
- Incrémental sur les différentes version de python a venir.

Statut actuel

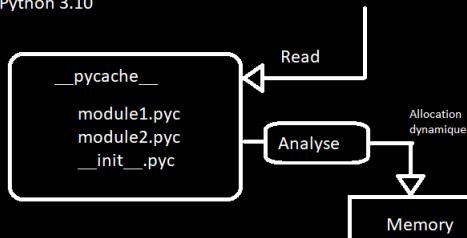
- Speedup **x1.25** en moyenne
- Entre **x1.10** et **x1.60** selon le workload
- Python 3.11 Faster Cpython



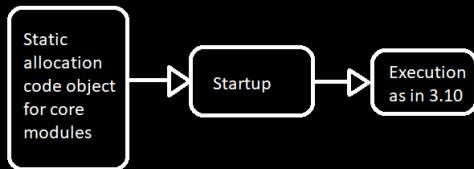
Speedup du démarrage

- code object et bytecode statiquement alloué par l'interpréteur pour les modules essentiels au démarrage

Python 3.10



Python 3.11





Speedup du runtime

- Reduction de l'information d'exécution en mémoire (debug ou on-demand). **3-7%**
- mécanisme de "jump" lors d'un appel de fonction (skip fonction d'interprétation C). **1-3%**
- Specializing Adaptive Interpreter

Misc

- les objets nécessitent moins d'espace mémoire
- Représentation des exceptions plus concise, améliore le catchin de **10%**



Optimisation

- C-style formatting **%s**, **%r**, **%a** est maintenant aussi efficace que les f-string
- "Zero-cost" exception => cout du **try** casi null lorsqu'il n'y a pas d'exception
- Les **dict** ne stock plus les **hash** lorsqu'il ne manipule que des objet unicode => reduction espace mémoire
- module **re** refactorisé en partie, **10%** plus rapide.



Exception Groups and except*

- PEP 654

Enriching Exceptions with Notes

- ajout d'une methode **add_note** pour les exceptions

Improved Error Locations in Tracebacks

- Error Locations in Tracebacks Exemple



Variadic generics

- Introduit **TypeVarTuple** pour les Variadic Generic
- Variadic Generic: Nombre variable d'entrée de type variable

Self type

- Ajout de l'annotation **Self** pour les retour du type de l'objet



Individual TypedDict items

- **TypedDict** peut marqué ses items comme **Required** ou **NotRequired**
- Les items sont **Required** par défaut, sauf si **total=False**

Arbitrary literal string type

- l'annotation **LiteralString** permet de marquer une variable comme une string *statique*



Major Language Changes

- * utilisable a présent dans les **for statement**
- **PYTHONSAFEPATH** / **-P** option pour les **path** potentiellement non secure pour le **sys.path**
- **sys.path** ne supporte plus les **Bytes**

Major Modules Updates

- **AsyncIO** => **TaskGroup** context manager
- **contextlib** => **chdir** context manager
- **re** => atomic groups et possessive quantifier
- **unittest** => **enterContext()** et **enterClassContext()**