

# Quel est le meilleur langage?

Olivier Levillain

Agence nationale de la sécurité de systèmes d'information

15 décembre 2017

## Généralités sur les langages

- ▶ Il n'y a pas de *meilleur* langage dans l'absolu

# Généralités sur les langages

- ▶ Il n'y a pas de *meilleur* langage dans l'absolu
- ▶ Certains langages sont plus adaptés à certaines tâches
  - ▶ le C pour des tâches de bas niveau
  - ▶ certains langages objet pour les interfaces graphiques
  - ▶ certains langages fonctionnels pour manipuler des données complexes

# Généralités sur les langages

- ▶ Il n'y a pas de *meilleur* langage dans l'absolu
- ▶ Certains langages sont plus adaptés à certaines tâches
  - ▶ le C pour des tâches de bas niveau
  - ▶ certains langages objet pour les interfaces graphiques
  - ▶ certains langages fonctionnels pour manipuler des données complexes
  - ▶ le JavaScript pour participer à l'enfer du web

# Généralités sur les langages

- ▶ Il n'y a pas de *meilleur* langage dans l'absolu
- ▶ Certains langages sont plus adaptés à certaines tâches
  - ▶ le C pour des tâches de bas niveau
  - ▶ certains langages objet pour les interfaces graphiques
  - ▶ certains langages fonctionnels pour manipuler des données complexes
  - ▶ le JavaScript pour participer à l'enfer du web
- ▶ Dans tous les cas, il faut connaître les langages utilisés
  - ▶ cela passe par une connaissance des fondamentaux
    - ▶ ce n'est pas parce qu'on utilise un langage sans pointeur qu'il est inutile de comprendre son modèle mémoire
  - ▶ il faut connaître les pièges
  - ▶ et éviter de céder aux sirènes des modes

## Un peu d'expérience personnelle

Un apprentissage en plusieurs temps

## Un peu d'expérience personnelle

Un apprentissage en plusieurs temps

- ▶ de la bidouille entre 10 et 18 ans
  - ▶ du Pascal
  - ▶ de l'assembleur
  - ▶ du Visual Basic et du Delphi

# Un peu d'expérience personnelle

## Un apprentissage en plusieurs temps

- ▶ de la bidouille entre 10 et 18 ans
  - ▶ du Pascal
  - ▶ de l'assembleur
  - ▶ du Visual Basic et du Delphi
- ▶ une rencontre avec la théorie en prépa
  - ▶ un peu de théorie en option informatique
  - ▶ OCaml



# Un peu d'expérience personnelle

## Un apprentissage en plusieurs temps

- ▶ de la bidouille entre 10 et 18 ans
  - ▶ du Pascal
  - ▶ de l'assembleur
  - ▶ du Visual Basic et du Delphi
- ▶ une rencontre avec la théorie en prépa
  - ▶ un peu de théorie en option informatique
  - ▶ OCaml
- ▶ des projets collaboratifs en école d'ingénieur
  - ▶ du C, du C++, du Java
  - ▶ la découverte du multi-plateforme
  - ▶ la découverte de la gestion de version

# Un peu d'expérience personnelle

## Un apprentissage en plusieurs temps

- ▶ de la bidouille entre 10 et 18 ans
  - ▶ du Pascal
  - ▶ de l'assembleur
  - ▶ du Visual Basic et du Delphi
- ▶ une rencontre avec la théorie en prépa
  - ▶ un peu de théorie en option informatique
  - ▶ OCaml
- ▶ des projets collaboratifs en école d'ingénieur
  - ▶ du C, du C++, du Java
  - ▶ la découverte du multi-plateforme
  - ▶ la découverte de la gestion de version
- ▶ ma vie professionnelle
  - ▶ des développements en shell, C et C++
  - ▶ quelques projets opensource en OCaml et en Python

## Enseignements pour l'enseignement ? (1/4)

Apprendre plusieurs langages est important pour un cursus de développeur

## Enseignements pour l'enseignement ? (1/4)

Apprendre plusieurs langages est important pour un cursus de développeur

Sans ordre particulier, les langages suivants me semblent pertinents :

- ▶ le C
- ▶ un langage objet (C++, Java)
- ▶ un langage de script (Shell, Python, Ruby)
- ▶ un langage fonctionnel (OCaml, Haskell, Lisp)

## Enseignements pour l'enseignement ? (1/4)

Apprendre plusieurs langages est important pour un cursus de développeur

Sans ordre particulier, les langages suivants me semblent pertinents :

- ▶ le C
- ▶ un langage objet (C++, Java)
- ▶ un langage de script (Shell, Python, Ruby)
- ▶ un langage fonctionnel (OCaml, Haskell, Lisp)

Il est plus facile d'apprendre un nouveau langage quand on a déjà touché à plusieurs langages

## Enseignements pour l'enseignement ? (2/4)

La théorie est essentielle pour ne pas réinventer la roue

## Enseignements pour l'enseignement ? (2/4)

La théorie est essentielle pour ne pas réinventer la roue

Parmi les fondamentaux, on peut citer

- ▶ les notions de complexité
- ▶ les structures de données
- ▶ certains résultats d'impossibilité théoriques
- ▶ les fondamentaux de l'informatique en général

## Enseignements pour l'enseignement ? (3/4)

Pour former un développeur, il faut qu'il apprenne à se servir d'outils et qu'il connaisse des méthodologies



## Enseignements pour l'enseignement ? (3/4)

Pour former un développeur, il faut qu'il apprenne à se servir d'outils et qu'il connaisse des méthodologies

Encore une fois, sans ordre précis, et pas de manière exhaustive

- ▶ la gestion de version
- ▶ le test
- ▶ le travail collaboratif et la relecture de code
- ▶ le déploiement
- ▶ la dure réalité du monde qui ne s'apprend que sur un projet long terme (migrations de schémas de base de données par exemple)

## Enseignements pour l'enseignement ? (4/4)

Et pour des scientifiques non informaticiens ?

## Enseignements pour l'enseignement ? (4/4)

Et pour des scientifiques non informaticiens ?

- ▶ le temps imparti rend difficile l'apprentissage de plusieurs langages

## Enseignements pour l'enseignement ? (4/4)

Et pour des scientifiques non informaticiens ?

- ▶ le temps imparti rend difficile l'apprentissage de plusieurs langages
- ▶ cela ne signifie pas qu'il faut utiliser des langages pour les nuls

## Enseignements pour l'enseignement ? (4/4)

Et pour des scientifiques non informaticiens ?

- ▶ le temps imparti rend difficile l'apprentissage de plusieurs langages
- ▶ cela ne signifie pas qu'il faut utiliser des langages pour les nuls
- ▶ deux voies semblent pertinentes
  - ▶ Python, avec ses bibliothèques nombreuses et sa courbe d'apprentissage accueillante
  - ▶ un langage fonctionnel fortement typé, pour sa rigueur et sa proximité au monde mathématique

## Enseignements pour l'enseignement ? (4/4)

Et pour des scientifiques non informaticiens ?

- ▶ le temps imparti rend difficile l'apprentissage de plusieurs langages
- ▶ cela ne signifie pas qu'il faut utiliser des langages pour les nuls
- ▶ deux voies semblent pertinentes
  - ▶ Python, avec ses bibliothèques nombreuses et sa courbe d'apprentissage accueillante
  - ▶ un langage fonctionnel fortement typé, pour sa rigueur et sa proximité au monde mathématique
- ▶ d'autres chemins me semblent moins pertinents
  - ▶ le C ou le C++ (gestion mémoire atroce, langage verbeux)
  - ▶ les langages web (PHP, JS)

Questions ?

Merci de votre attention

`olivier.levillain@ssi.gouv.fr`