Langages de script – TD 1

achille.falaise@cnrs.fr

https://pro.aiakide.net => Cours

Évaluation

- Interros en fin de TD (~1 TD sur 2)
- Exam final

Qu’est-ce qu’un langage de script ?

Larry Wall [modifier | modifier le code]

Larry Wall qui est le concepteur du langage de programmation Perl a dit :

- « *when I was a RSTS programmer on a PDP-11, I certainly treated BASIC as a scripting language, at least in terms of rapid prototyping and process control. I'm sure it warped my brain forever* »

  (Quand je programmais en RSTS sur un PDP-11, j'ai effectivement considéré le BASIC comme un langage de script, au moins à cause du prototypage facile et de la commande de processus pour lesquels on l'employait. Je suis certain que ça m'a déformé intellectuellement à long terme.)

- « *basically, scripting is not a technical term. When we call something a scripting language, we're primarily making a linguistic and cultural judgment, not a technical judgment* »

  (L'expression "Langage de script" ne constitue pas un terme technique ; quand on utilise cette expression, on fait part d'une appréciation linguistique et culturelle, on ne porte pas un jugement technique.)

— « *Programming is Hard, Let's Go Scripting...* » [archive]

https://fr.wikipedia.org/wiki/Langage_de_script

Objectifs du cours

- Python « niveau intermédiaire »
- D’autres langages de script
  - Bash, Awk, PHP, Kotlin
- Communication entre scripts (≠ langages)
  - *DevOps* : on va faire du *dev*, parfois évaluer son efficacité (vitesse), et le mettre en (pré-)production (*ops*)
- Traitement de langues orientales
  - Arabe (normalisation), Chinois (tokenisation)
  - Spacy

Rappels

- https://loicgrobol.github.io/python-im-2/
  - https://mybinder.org/v2/gh/loicgrobol/python-im-2/master?filepath=slides/1-man-1.ipynb

Rappels

- Écrivez un programme Python qui répète *1 2 3 1 2 3 1 2 3* … 99 fois en utilisant le modulo.
- Écrivez un programme Python qui répète *1 2 3 1 2 3 1 2 3* … 99 fois en utilisant la multiplication de chaînes.

Fizz Buzz

- Écrire un script en Python, qui :
  - Pour les nombres entre 1 et 100:
    - on écrit Fizz si le nombre est divisible par 3
    - on écrit Buzz si le nombre est divisible par 5
    - on écrit Fizz Buzz si le nombre est divisible par 3 et 5
    - Sinon, on écrit le nombre
  - Ex :
    - 1 2 Fizz 4 Buzz Fizz 7 8 Fizz Buzz 11 Fizz 13 14 Fizz Buzz 16
    - 1 2 (3) 4 (5) (6) 7 8 (9) (10) 11 (12) 13 14 (15) 16

Fizz Buzz

- Convertir en Python
  - Et vérifier que ça marche...

Fizz Buzz

- Pour n de 1 à 100
  - si n%3==0 et n%5==0 : Fizz Buzz
  - elsi n%3==0 : Fizz
  - elsi n%5==0 : Buzz
  - sinon : n

- Pour n de 1 à 100
  - si n%3==0 ou n%5==0 :
    - si n%3==0: Fizz
    - si n%5==0 : Buzz
  - sinon : n

Fizz Buzz

- Pour n de 1 à 100
  - si n = 3 et n = 5 :
    Fizz Buzz
  - elsi n = 3 : Fizz
  - elsi n = 5 : Buzz
  - sinon : n

- Pour n de 1 à 100
  - si n = 3 ou n = 5
    - si n = 3 : Fizz
    - si n = 5 : Buzz
  - sinon : n

si au lieu de *elsi* pour traiter les multiples de 15

En rouge : tests effectués dans le cas général où n ≠ 3 et n ≠ 5

Fizz Buzz

- Pour n de 1 à 100
  - si n = 3 et n = 5 :
    Fizz Buzz
  - elsi n = 3 : Fizz
  - elsi n = 5 : Buzz
  - sinon : n

- Pour n de 1 à 100
  - si n = 3 ou n = 5
    - si n = 3 : Fizz
    - si n = 5 : Buzz
  - sinon : n

si au lieu de *elsi* pour traiter les multiples de 15

En rouge : tests effectués dans le cas où n = 3

Fizz Buzz

- Pour n de 1 à 100
  - si n = 3 et n = 5 :
    Fizz Buzz
  - elsi n = 3 : Fizz
  - elsi n = 5 : Buzz
  - sinon : n

- Pour n de 1 à 100
  - si n = 3 ou n = 5
    - si n = 3 : Fizz
    - si n = 5 : Buzz
  - sinon : n

si au lieu de *elsi* pour traiter les multiples de 15

En rouge : tests effectués dans le cas où n = 5

Fizz Buzz

- Pour n de 1 à 100
  - si n = 3 et n = 5 :
    Fizz Buzz
  - elsi n = 3 : Fizz
  - elsi n = 5 : Buzz
  - sinon : n

- Pour n de 1 à 100
  - si n = 3 ou n = 5
    - si n = 3 : Fizz
    - si n = 5 : Buzz
  - sinon : n

si au lieu de *elsi* pour traiter les multiples de 15

Fizz Buzz

- Quelle est l’approche la + rapide ?
  - python3 -m timeit "$(cat *mon_script.py*)"

Fonctions

- Heures
  - Écrire une fonction heures(secondes) qui prend un nombre de secondes (entier) et le convertit en heures, minutes et secondes sous le format H:M:S où H est le nombre d'heures, M le nombre de minutes et S le nombre de secondes.
    - On suppose que secondes est positif ou nul (secondes >= 0).
  - Écrire une fonction secondes(heure) qui prend une heure au format H:M:S et renvoie le nombre de secondes correspondantes (entier).
    - On suppose que l'heure est bien formatée. On aura toujours un nombre d'heures valide, un nombre de minutes valide et un nombre de secondes valide.

- https://hub.ovh2.mybinder.org/user/loicgrobol-python-im-2-pbjsyuqi/notebooks/slides/2-man-2.ipynb#/slide-0-0

SSH