

Micro cours Python

Thierry Dumont

Aldil /06/02/25/

Tous les langages de programmation (ou presque) ont les mêmes capacités

Tous les langages de programmation (ou presque) ont les mêmes capacités

Vieille question: que peut-on *calculer* ?

- Leibnitz (fin XVII^e siècle).

Tous les langages de programmation (ou presque) ont les mêmes capacités

Vieille question: que peut-on *calculer* ?

- Leibnitz (fin XVII^e siècle).
- Turing (1936) : *on ne peut pas tout calculer.*

Machines à registres

- Des registres R_i en nombre fini qui peuvent contenir des nombres entiers *quelconques*.

Machines à registres

- Des registres R_i en nombre fini qui peuvent contenir des nombres entiers *quelconques*.
- On peut incrémenter chaque registre: $R_i \leftarrow R_i + 1$ et $R_i \leftarrow R_i - 1$.
- Faire $R_i = x$ et $R_i = R_j$.

Machines à registres

- Des registres R_i en nombre fini qui peuvent contenir des nombres entiers *quelconques*.
- On peut incrémenter chaque registre: $R_i \leftarrow R_i + 1$ et $R_i \leftarrow R_i - 1$.
- Faire $R_i = x$ et $R_i = R_j$.

Test :

- *if $R_i = 0$ then S else S' .*

Machines à registres

- Des registres R_i en nombre fini qui peuvent contenir des nombres entiers *quelconques*.
- On peut incrémenter chaque registre: $R_i \leftarrow R_i + 1$ et $R_i \leftarrow R_i - 1$.
- Faire $R_i = x$ et $R_i = R_j$.

Test :

- *if $R_i = 0$ then S else S' .*

Boucles :

- *for $i = 1$ to R_i do S .*

Machines à registres

- Des registres R_i en nombre fini qui peuvent contenir des nombres entiers *quelconques*.
- On peut incrémenter chaque registre: $R_i \leftarrow R_i + 1$ et $R_i \leftarrow R_i - 1$.
- Faire $R_i = x$ et $R_i = R_j$.

Test :

- *if $R_i = 0$ then S else S' .*

Boucles :

- *for $i = 1$ to R_i do S .*
- *while $R_i \neq 0$ do S .*

Les machines à registres sont équivalentes à des machines de Turing.

Donc, ce qui est calculable est ce qui est calculable avec une machine à registres.

Les machines à registres sont équivalentes à des machines de Turing.

Donc, ce qui est calculable est ce qui est calculable avec une machine à registres.

Donc: un ordinateur + langage de programmation “*équivalent à une machine à registres*” peut **tout** calculer.

Apprendre par l'exemple : quelques exercices

- les types en python.
- les fonctions.

Apprendre par l'exemple : quelques exercices

- les types en python.
- les fonctions.
- factorielle : $n! = 1 \times 2 \times 3 \times \dots \times n$.

4 manières de calculer.

Apprendre par l'exemple : quelques exercices

- les types en python.
- les fonctions.
- factorielle : $n! = 1 \times 2 \times 3 \times \dots \times n$.

4 manières de calculer.

- Tri par la méthode des bulles (peu efficace).

2 programmes légèrement différents.

Apprendre par l'exemple : quelques exercices

- les types en python.
- les fonctions.
- factorielle : $n! = 1 \times 2 \times 3 \times \dots \times n$.

4 manières de calculer.

- Tri par la méthode des bulles (peu efficace).

2 programmes légèrement différents.

- Manipulation de chaînes de caractères.

- Le *théorème de l'arrêt* (Turing, 1936)

“Il n'existe pas de programme (d'algorithme) permettant de prouver qu'un programme s'arrête ou non”

Une preuve en moins de 10 lignes de Python !

Python est vendu avec accessoires...

- Doctests