

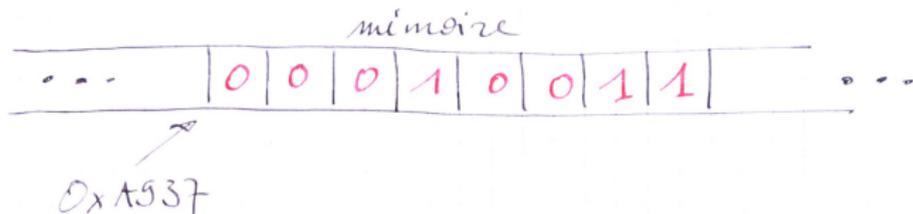
- Retour sur les variables
- Entrées/sortie: affichage
- Notion de type
- Exercices boucles et variables

Une variable est une **zone de la mémoire** de l'ordinateur dans laquelle une **valeur** est stockée. Aux yeux du programmeur, cette variable est définie par un **nom**, alors que pour l'ordinateur, il s'agit en fait d'une **adresse**, c'est-à-dire d'une zone particulière de la mémoire.

En Python, la déclaration d'une variable et son initialisation (c'est-à-dire la première valeur que l'on va stocker dedans) se font en même temps.

Affectation/initialisation

toto = 19



"table des symboles":

:	:
toto	$0xA937$
:	:

Nom de variable : **identificateur**

- Lettres, chiffres, underscore
- Ne doit pas commencer par un chiffre
- Majuscule et minuscules différenciées
- Eviter les accents (non-ascii)
- Eviter les mots-clés du langage (ou les fonctions usuelles → masquage)
- Choisir des noms informatifs:
 - `entier`
 - `toto`
 - `while`
 - `x`
- Respecter les conventions (`i` : plutôt un indice) et être cohérent

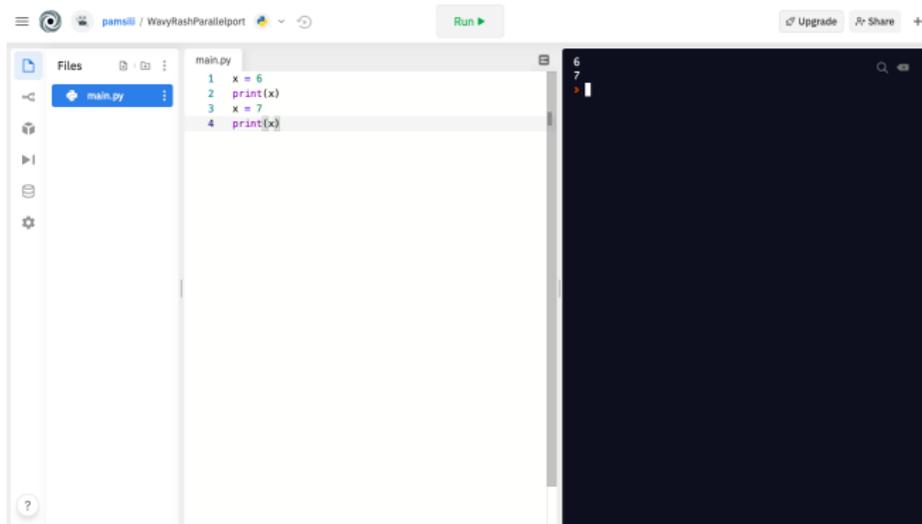
Les ordinateurs passent beaucoup de temps à modifier leur état interne (mémoire), mais il faut aussi qu'ils puissent communiquer avec l'« extérieur ».

- obtenir des informations nouvelles (ou des instructions)
→ entrées (*input*)
- communiquer des résultats
→ sorties (*output*)

- préhistoire: cartes perforées, puis bandes magnétiques - imprimante
- aujourd'hui: clavier - écran
- mémoire de masse (disques dur, clés USB)
- périphériques :
 - I: tablette graphique, souris, micro...
 - O: imprimante, haut-parleur...
 - I/O: réseau, écran tactile,

- Récupération d'une information au clavier: `input()`
- Affichage d'une information à l'écran: `print()`
- Lecture/écriture d'un fichier
open(), close(), read(), write()...

Démos sur repl.it



The screenshot shows a Repl.it Python environment. The browser address bar displays 'pamsili / WavyRashParallelport'. The interface includes a 'Files' sidebar with 'main.py' selected, a code editor with the following Python code, and a terminal window.

```
main.py
1 x = 6
2 print(x)
3 x = 7
4 print(x)
```

The terminal window shows the output of the code execution:

```
6
7
```

Type:

- Mode de codage
- Ensemble de valeurs
- Opérations

Types scalaires:

- Entiers (int)
- Réels (float)
- Booléens (bool)

Type dynamique:

- Chaîne de caractères (str)

Types structurés : listes, ensembles, « dictionnaires »