
Machine de Turing en Python

Version 0.1

Tristan LAROCHE, Loris ADAM

févr. 08, 2023

1	Pré-requis	3
2	Utilisation	5
2.1	Lancement du script	5
2.2	Fonctionnement	5
2.3	Écriture d'un script (facultatif)	6
2.4	Écriture du ruban	7
2.5	Exécution :	7

La **Machine de Turing en Python** est un programme implémentant en python la Machine de Turing, inspirée par l'expérience de pensée de son auteur.

CHAPITRE 1

Pré-requis

Pour télécharger le programme exécutez la commande :

```
git clone https://github.com/Yoween/PythonTuringMachine
```

ou rendez-vous dans l'onglet « Releases », téléchargez le fichier « PythonTuringMachine.zip » et extrayez le fichier dans un dossier nommé à votre guise.

Pour l'utiliser, vous devez disposer de **Python >= 3.9** et installer les deux modules requis via la commande :

```
pip install python-i18n[YAML] logging2
```


2.1 Lancement du script

- Windows : Double-cliquez sur « start.bat » ou exécutez la commande :

```
$ python UI_Window.py
```

- Linux : Double-cliquez sur « start.sh » ou exécutez la commande :

```
$ python3 UI_Window.py
```

- MacOS : Le script utilisé pour Linux devrait fonctionner mais aucune certitude.

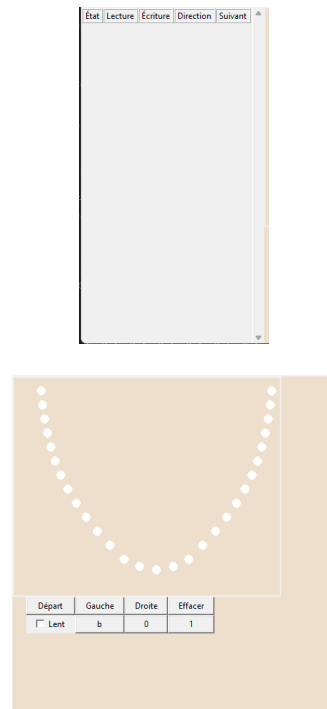
2.2 Fonctionnement

Nous partirons du principe que vous savez utiliser une machine de Turing et que vous connaissez son fonctionnement. Sinon, cliquez [ici](#) pour en savoir plus.

Comme vous pouvez le voir, l’affichage est divisé en deux parties :

La partie de gauche est réservée à la prévisualisation des instructions du code que vous avez sélectionné. Vous ne pouvez pas interagir avec cette partie.

La partie de droite, elle, est destinée à afficher le « ruban » virtuel (« tape » en anglais) et dispose de boutons ; vous pouvez donc interagir avec.



2.3 Écriture d'un script (facultatif)

L'interpréteur dispose d'un système de programmation personnalisé où les instructions sont écrites sous une forme définie :

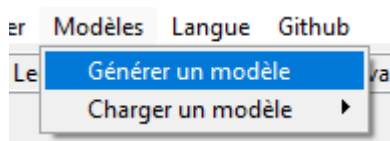
<état actuel (nombre entier)> : <valeur lue (« b », « 0 » ou « 1 »)> : <valeur à écrire (« - » pour ne rien faire, « b », « 0 » ou « 1 » sinon)> : <sens de déplacement (chevrons : « < » ou « > »)> : <état suivant (nombre entier ou « f » pour terminer le programme)>

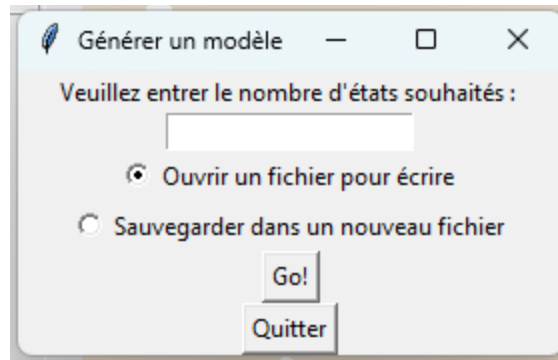
et placées par « paquets » de 3 pour chaque valeur lue possible, ce qui nous donne par exemple :

```
1 : b : - : < : 1
1 : 0 : - : < : 1
1 : 1 : 0 : < : f
```

Ici, ce programme va aller vers la gauche jusqu'à trouver un « 1 », le remplacer par un « 0 » et s'arrêter.

Vous pouvez aussi générer un modèle de la taille souhaitée en cliquant sur « Générer un modèle » et en précisant la taille souhaitée, ou bien en charger un existant en cliquant cette fois-ci sur « Charger un modèle ».





2.4 Écriture du ruban

Comme vous pouvez le voir, l'interface est composée de 8 boutons mais seuls 6 vont nous intéresser : Leurs actions sont explicites par leur nom mais nous allons quand même décrire leur fonctionnement. Les boutons « Gauche » et « Droite » déplacent le curseur d'une case sur le ruban virtuel, et les boutons « b », « 0 » et « 1 » écrivent la valeur indiquée à la position actuelle du ruban. Le bouton « Effacer », lui, efface le ruban virtuel. C'est aussi simple que ça !

2.5 Exécution :

Notre interpréteur dispose de deux types d'exécution : Une exécution « instantanée », à la vitesse réelle à laquelle votre ordinateur exécute le programme, et une exécution « ralentie » afin de voir les différentes étapes de l'exécution. Cette fonction peut être activée en cliquant sur le bouton « Lent ». Une fois votre vitesse d'exécution sélectionnée, il vous suffit de cliquer sur « Départ » pour exécuter le programme !