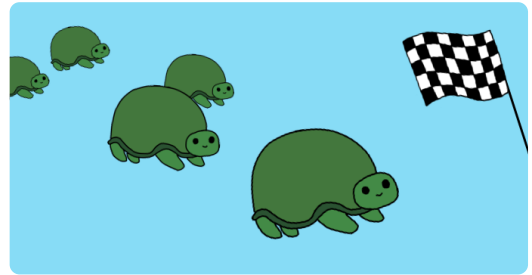


 Projets

Course de tortues!

Fais courir les tortues les unes contre les autres!

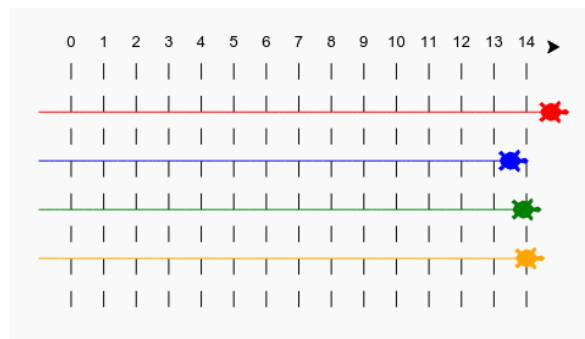


Étape 1 Introduction

Utilise des boucles pour dessiner une piste de course et crée un jeu de course de tortues.

Ce que tu feras

Ce projet introduit les boucles dans un jeu amusant de course de tortues. Les boucles sont utilisées pour dessiner la piste de course et pour faire en sorte que les tortues fassent un nombre aléatoire de pas à chaque tour. Si tu as un groupe de personnes pour jouer au jeu, chaque personne choisit une tortue et celle qui obtient le plus loin est le gagnant.



Ce que tu vas apprendre

En réalisant ton jeu de course de tortues, tu apprendras à:

- Écrire des boucles `for` en Python
- Utiliser des numéros aléatoires en Python
- Dessiner des lignes en différentes couleurs avec Python Turtle

Ce projet traite des éléments des sections suivantes du **Programme de Création Numérique Raspberry Pi** (<https://rpf.io/curriculum>):

- **Utilise des concepts de base de programmation pour créer des programmes simples** (<https://www.raspberrypi.org/curriculum/programming/creator/>)

Informations supplémentaires pour les éducateurs

Si vous besoin d'imprimer ce projet, merci d'utiliser la **version imprimable**. (<https://projects.raspberrypi.org/fr-FR/projects/turtle-race/print>).

Utilise le lien dans au bas de la page pour accéder au dépôt GitHub pour ce projet, qui contient toutes les ressources (y compris un exemple d'un projet terminé) dans le dossier 'fr-FR/resources'.

Étape 2 Ce dont tu auras besoin

Matériel informatique

- Un ordinateur connecté à internet

Logiciel

Ce projet utilise Python 3. Nous recommandons d'utiliser **Trinket** (<https://trinket.io/>), qui te permet d'écrire du code Python en ligne.

Étape 3 Piste de course

Tu vas créer un jeu avec des tortues de course. D'abord, elles auront besoin d'une piste de course.

- Ouvre le modèle de Trinket Python vierge : jumpto.cc/python-new (<http://jumpto.cc/python-new>).
- Ajoute le code suivant pour dessiner une ligne à l'aide de "turtle":

```
from turtle import *
forward(100)
```



- Utilisons maintenant la tortue pour dessiner des marques de piste pour la course.

La fonction turtle `write` écrit le texte à l'écran.

Essaie-le:

```
from turtle import *
write(0)
forward(100)
write(5)
```



- Maintenant, tu dois remplir les numéros entre les deux pour créer des marquages:

```
write(0)
forward(20)
write(1)
forward(20)
write(2)
forward(20)
write(3)
forward(20)
write(4)
forward(20)
write(5)
forward(20)
```




- As-tu remarqué que ton code est très répétitif? La seule chose qui change est le nombre à écrire.

Il y a une meilleure façon de faire cela en Python. Tu peux utiliser une boucle `for`.

Mets à jour ton code pour utiliser une boucle `for`:

```
from turtle import *
for etape in range(5):
    write(etape)
    forward(20)
```



- Hmm, cela n'imprime que des nombres allant jusqu'à 4. En Python `range(5)` renvoie cinq nombres, allant de 0 à 4. Pour que le code retourne également 5, tu dois utiliser `range(6)`:

```
for etape in range(6):
    write(etape)
    forward(20)
```



- Nous pouvons maintenant dessiner des marques de piste. La tortue commence aux coordonnées (0,0) au milieu de l'écran.

Déplace la tortue en haut à gauche à la place:

```
from turtle import *
goto(-140, 140)
for etape in range(6):
    write(etape)
    forward(20)
```



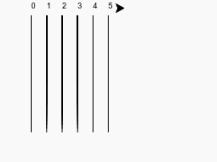
- Ah, tu voudras d'abord lever le stylo!

```
penup()
goto(-140, 140)
for etape in range(6):
    write(etape)
    forward(20)
```



- Au lieu de dessiner une ligne horizontalement, dessinons des lignes verticales pour créer une piste:

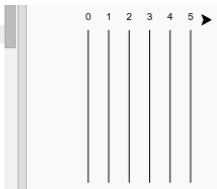
```
for etape in range(6):
    write(etape)
    right(90)
    forward(10)
    pendown()
    forward(150)
    penup()
    backward(160)
    left(90)
    forward(20)
```



right(90) fait tourner la tortue à droite de 90 degrés (un angle droit.) Avance **forward(10)** avant de poser le stylo, laisse un petit espace entre le nombre et le début de la ligne. Après avoir tracé la ligne, relève le stylo et va **backward(160)** la longueur de la ligne plus l'écart.

- Cela semble plus net si tu centres les chiffres:

```
for etape in range(2):
    write(etape, align='center')
    right(90)
    forward(10)
    pendown()
    forward(150)
    penup()
    backward(160)
    left(90)
    forward(20)
```



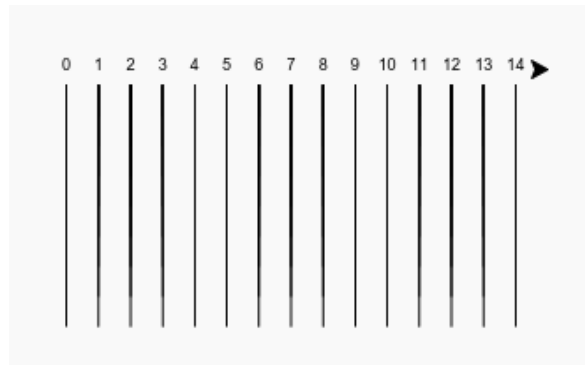
- Et tu peux accélérer la tortue pour qu'elle s'approche plus vite:

```
from turtle import *

speed(10)
penup()
goto(-140, 140)
```

Étape 4 Défi: plus de lignes

Peux-tu changer ton code pour que les lignes de piste traversent l'écran?



Si tu veux que la tortue aille encore plus vite, tu peux utiliser `speed(0)`.

Étape 5 Tortues de course

Maintenant pour le plaisir. Ajoutons des tortues de course. Ce serait vraiment ennuyeux que les tortues fassent la même chose à chaque fois pour pouvoir faire un nombre aléatoire de pas à chaque tour. Le gagnant est la tortue la plus éloignée en 100 tours.

- Lorsque tu utilises des commandes telles que `forward(20)` tu utilises une seule tortue. Mais tu peux créer plus de tortues. Ajoute le code suivant à la fin de ton script (mais assure-toi qu'il ne soit pas en retrait):

```
forward(20)

ada = Turtle()
ada.color('red')
ada.shape('turtle')
```




La première ligne crée une tortue appelée 'ada'. Les lignes suivantes définissent la couleur et la forme de la tortue. Maintenant, ça ressemble vraiment à une tortue!

- Envoyons la tortue à la ligne de départ:

```
ada = Turtle()
ada.color('red')
ada.shape('turtle')

ada.penup()
ada.goto(-160, 100)
ada.pendown()
```



- Maintenant, tu dois faire la course de tortue en déplaçant un nombre aléatoire de pas à la fois. Tu auras besoin de la fonction `randint` de la bibliothèque Python `random`. Ajoute cette ligne `import` en haut de ton script:

```
from turtle import *
from random import randint
```

- La fonction `randint` renvoie un entier aléatoire (nombre entier) entre les valeurs choisies. La tortue avancera de 1, 2, 3, 4 ou 5 pas à chaque tour.

```
ada.penup()
ada.goto(-160, 100)
ada.pendown()

for tourner in range(100):
    ada.forward(randint(1,5))
```

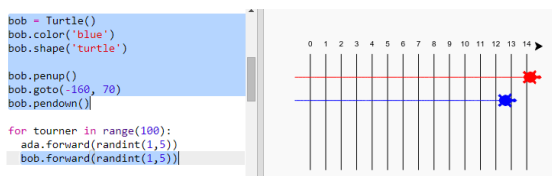


- Une tortue, ce n'est pas assez! Ajoutons-en une autre:

```
bob = Turtle()
bob.color('blue')
bob.shape('turtle')

bob.penup()
bob.goto(-160, 70)
bob.pendown()

for tourner in range(100):
    ada.forward(randint(1,5))
    bob.forward(randint(1,5))
```

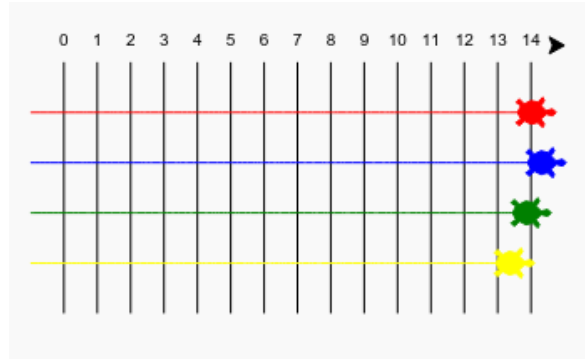


Note que le code pour déplacer la tortue bleue doit être dans **la même** boucle `for` comme le code pour déplacer la tortue rouge afin qu'elles bougent à chaque tour.

Étape 6 Défi: c'est l'heure de la course!

Maintenant, tu es prêt à courir. Choisis une tortue et un adversaire et vois qui gagne.

Peux-tu ajouter plus de tortues pour pouvoir faire la course avec plus d'amis?



Les couleurs comprennent: orange, violet, tomate, turquoise, magenta et brun - ou tu peux aller à jumpto.cc/colours (<http://jumpto.cc/colours>) et choisis n'importe quelle couleur que tu aimes!

Étape 7 Défi: faire un tourbillon

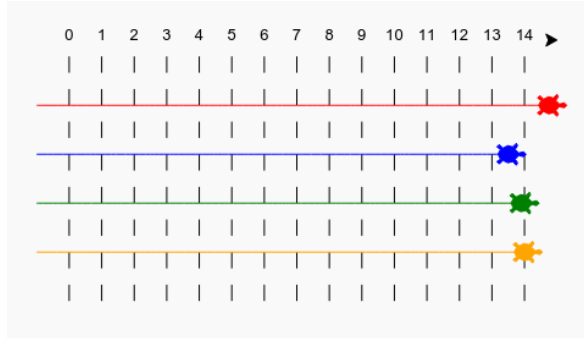
Peux-tu utiliser une boucle `for tourner in range()` : pour que chaque tortue fasse un tourbillon de 360 degrés après avoir atteint la ligne de départ? Tu devras t'assurer qu'elles font face dans la bonne direction au début de la course!

`ada.right(36)` tournera la tortue rouge à droite de 36 degrés.

Indice: Un tour complet est à 360 degrés. Une tortue peut tourner à droite 36 fois de 10 degrés, ou à gauche de 5 degrés 72 fois, ou n'importe quel autre chiffre faisant 360!

Étape 8 Défi: lignes pointillées

Peux-tu utiliser une boucle pour que les lignes de tracé soient pointillées au lieu d'être pleines?



Indice: Trouve le code qui trace une ligne droite. Essaie d'utiliser: `for`, `forward()`, `penup()` et `pendown()`

Ce projet a été traduit par des bénévoles:

Jonathan Vannieuwerkerke

Michel Arnols

Antoine Even

Grâce aux bénévoles, nous pouvons donner aux gens du monde entier la chance d'apprendre dans leur propre langue. Vous pouvez nous aider à atteindre plus de personnes en vous portant volontaire pour la traduction - plus d'informations sur rpf.io/translate (<https://rpf.io/translate>).

Publié par [Raspberry Pi Foundation](https://www.raspberrypi.org) (<https://www.raspberrypi.org>) sous un [Creative Commons license](https://creativecommons.org/licenses/by-sa/4.0/) (<https://creativecommons.org/licenses/by-sa/4.0/>).

Voir le projet et la licence sur GitHub (<https://github.com/RaspberryPiLearning/turtle-race>).