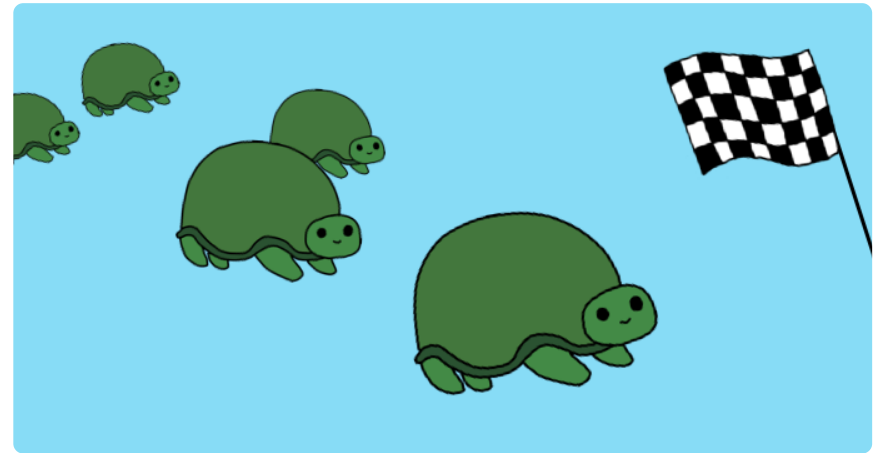


Αγώνας χελώνων!

Χελώνες που αγωνίζονται η μία εναντίον της άλλης!

Python



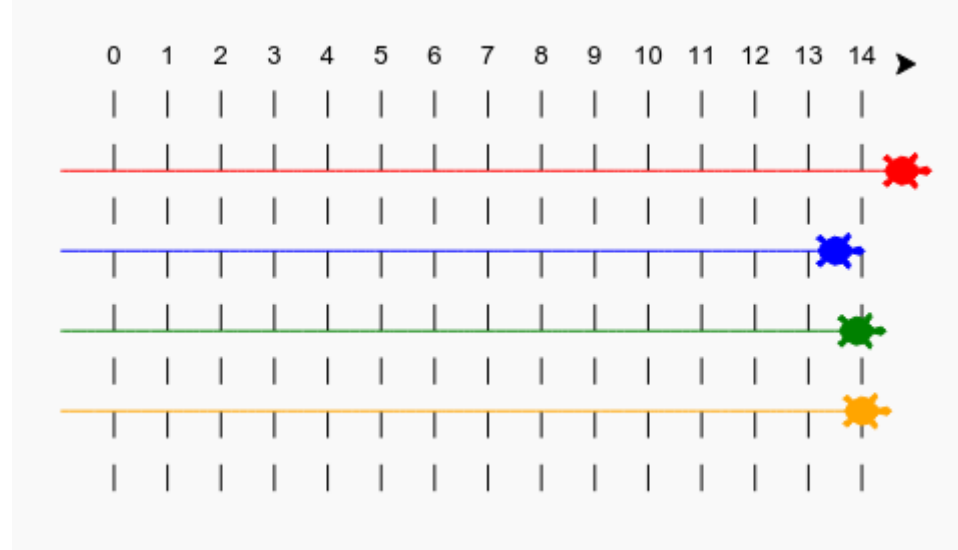
Βήμα 1

Εισαγωγή

Χρησιμοποίησε βρόχους επανάληψης για να σχεδιάσεις μια αγωνιστική πίστα και να δημιουργήσεις έναν αγώνα χελώνων.

Τι θα φτιάξεις

Αυτό το έργο κάνει μια εισαγωγή στους βρόχους επανάληψης μέσα από ένα διασκεδαστικό αγώνα χελώνων. Οι βρόχοι χρησιμοποιούνται για το σχεδιασμό της αγωνιστικής διαδρομής και για την κίνηση των χελώνων για τυχαίο αριθμό βημάτων σε κάθε στροφή. Εάν έχεις μια ομάδα ανθρώπων για να παίξεις το παιχνίδι, κάθε άτομο επιλέγει μια χελώνα και αυτή που φτάνει πιο μακριά είναι η νικήτρια.



Τι θα μάθεις

Με το παιχνίδι αγώνα χελώνων, θα μάθεις πώς να:

- Γράψε `for` βρόχους επανάληψης στην Python
- Χρησιμοποιείς τυχαίους αριθμούς στην Python
- Σχεδιάζεις γραμμές με διαφορετικά χρώματα με την χελώνα στην Python

Αυτό το πρόγραμμα καλύπτει στοιχεία από τις ακόλουθες πτυχές του **Raspberry Pi Digital Making Curriculum** (<http://rpf.io/curriculum>):

- Χρησιμοποίησε βασικές δομές προγραμματισμού για να δημιουργήσεις απλά προγράμματα (<https://www.raspberrypi.org/curriculum/programming/creator/>)



Η ολοκλήρωση αυτού του έργου θα σου δώσει πόντους για το χάλκινο σήμα “Digital Maker” iDEA. Για περισσότερες πληροφορίες, μπες στο idea.org.uk (<https://idea.org.uk>).

Πρόσθετες πληροφορίες για εκπαιδευτικούς

Αν χρειαστεί να εκτυπώσεις αυτό το έργο, χρησιμοποίησε την εκτυπώσιμη έκδοση (<https://projects.raspberrypi.org/el-GR/projects/turtle-race/print>).

Χρησιμοποίησε το σύνδεσμο στο υποσέλιδο για να αποκτήσεις πρόσβαση στο αποθετήριο GitHub για αυτό το έργο, το οποίο περιέχει όλους τους πόρους (συμπεριλαμβανομένου ενός έργου που έχει ολοκληρωθεί στο παράδειγμα) στο φάκελο ‘en/resources’.

Πρόσθετες πληροφορίες για τους συντονιστές των ομάδων

Αν χρειαστεί να εκτυπώσεις αυτό το έργο, χρησιμοποίησε την εκτυπώσιμη έκδοση (<https://projects.raspberrypi.org/el-GR/projects/turtle-race/print>).

Υλικό

- Ένας υπολογιστής συνδεδεμένο στο διαδίκτυο

Λογισμικό

Αυτό το έργο χρησιμοποιεί Python 3. Συνιστούμε να χρησιμοποιήσεις την εφαρμογή **Trinket** (<https://trinket.io/>), που σου επιτρέπει να γράψεις κώδικα Python online.

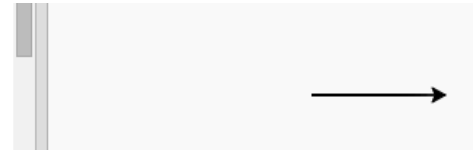
Βήμα 3

Πίστα αγώνα

Πρόκειται να δημιουργήσεις ένα παιχνίδι με αγωνιστικές χελώνες. Πρώτα θα χρειαστείς μια αγωνιστική πίστα.

- Άνοιξε το κενό πρότυπο Python Trinket: jumpto.cc/python-new (<http://jumpto.cc/python-new>).
- Πρόσθεσε τον ακόλουθο κώδικα για να σχεδιάσεις μια γραμμή χρησιμοποιώντας τη 'χελώνα':

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

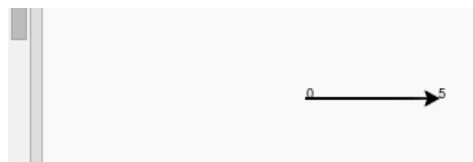


- Τώρα ας χρησιμοποιήσουμε τη χελώνα για να σχεδιάσουμε μερικές σημάνσεις στην πίστα για τον αγώνα.

Η λειτουργία `write` της χελώνας γράφει κείμενο στην οθόνη.

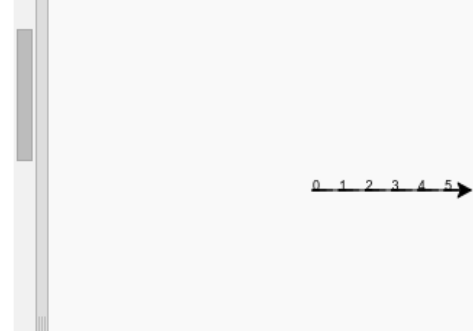
Δοκίμασέ το:

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



- Τώρα πρέπει να συμπληρώσεις τους αριθμούς για να δημιουργήσεις σημάνσεις:

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



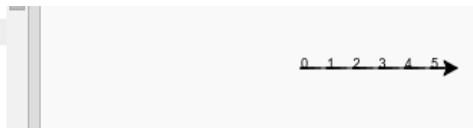
- Παρατήρησες ότι ο κώδικας είναι επαναλαμβανόμενος; Το μόνο που αλλάζει είναι ο αριθμός που γράφεις. Υπάρχει καλύτερος τρόπος να το κάνεις αυτό στην Python. Μπορείς να χρησιμοποιήσεις ένα βρόχο `for`. Ενημέρωσε τον κώδικα για να χρησιμοποιεί ένα επαναληπτικό βρόχο `for`:

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



- Χμμ, αυτό εκτυπώνει αριθμούς μέχρι 4. Στην Python η `range(5)` επιστρέφει πέντε αριθμούς, από 0 έως 4. Για να το κάνεις να επιστρέφει και το 5, θα χρειαστεί να χρησιμοποιήσεις την `range(6)`:

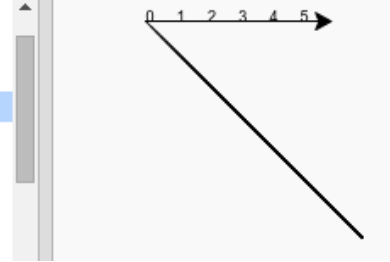
```
for step in range(6):
    write(step)
    forward(20)
```



- Τώρα μπορούμε να σχεδιάσουμε κάποια σημάδια στην πίστα. Η χελώνα ξεκινά με συντεταγμένες (0,0) στη μέση της οθόνης.

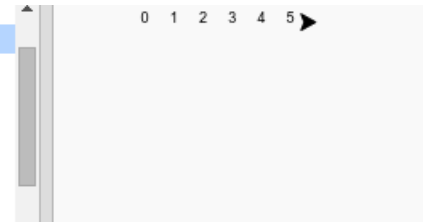
Μετακίνησε τη χελώνα επάνω αριστερά:

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



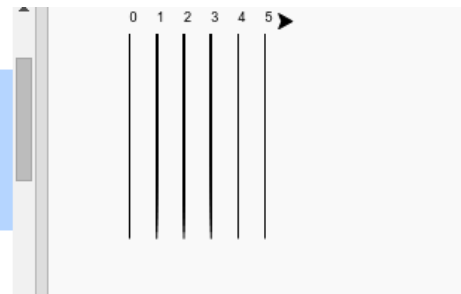
- Πρώτα, θα χρειαστεί να ανασηκώσεις την πένα!

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



- Αντί να σχεδιάζουμε μια γραμμή οριζόντια, ας σχεδιάσουμε κάθετες γραμμές για να δημιουργήσουμε μια πίστα:

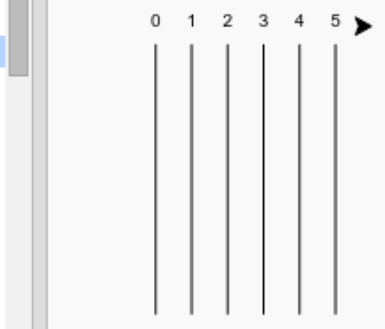
```
for step in range(6):  
    write(step)  
    right(90)  
    forward(10)  
    pendown()  
    forward(150)  
    penup()  
    backward(160)  
    left(90)  
    forward(20)
```



Η `right(90)` κάνει τη χελώνα να στρίψει δεξιά 90 μοίρες (δεξιά γωνία). Η κίνηση `forward(10)` πριν κατεβάσεις την πένα κάτω αφήνει ένα μικρό κενό μεταξύ του αριθμού και της αρχής της γραμμής. Αφού τραβήξεις τη γραμμή, σήκωσε την πένα και πήγαινε πίσω `backward(160)` όσο το μήκος της γραμμής συν το κενό.

- Φαίνεται πιο καθαρό αν κεντράρεις τους αριθμούς:

```
for step in range(6):
    write(step, align='center')
    right(90)
    forward(10)
    pendown()
    forward(150)
    penup()
    backward(160)
    left(90)
    forward(20)
```



- Και μπορείς να επιταχύνεις τη χελώνα έτσι ώστε να σχεδιάζει πιο γρήγορα:

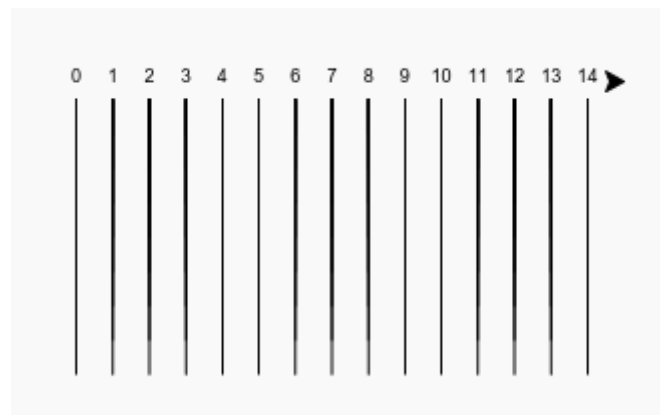
```
from turtle import *

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

Βήμα 4

Πρόκληση: περισσότερες γραμμές

Μπορείς να αλλάξεις τον κώδικά σου έτσι ώστε οι γραμμές του στίβου να πηγαίνουν δεξιά στην οθόνη;



Εάν θέλεις να κάνεις την χελώνα να πηγαίνει ακόμα πιο γρήγορα μπορείς να χρησιμοποιήσεις την `speed(0)`.

Ας το κάνουμε πιο διασκεδαστικό. Ας προσθέσουμε μερικές αγωνιστικές χελώνες. Θα ήταν πραγματικά βαρετό αν οι χελώνες έκαναν το ίδιο πράγμα κάθε φορά, για αυτό θα κινούνται έναν τυχαίο αριθμό βημάτων. Νικήτρια είναι η χελώνα που φτάνει πιο μακριά σε 100 στροφές.

- Όταν χρησιμοποιείς εντολές όπως `forward(20)` χρησιμοποιείς μία χελώνα. Μπορείς ωστόσο να δημιουργήσεις περισσότερες χελώνες. Πρόσθεσε τον ακόλουθο κώδικα στο τέλος του προγράμματός σου (βεβαιώσου ότι δεν έχει εσοχή):

```
forward(20)
ada = Turtle()
ada.color('red')
ada.shape('turtle')
```

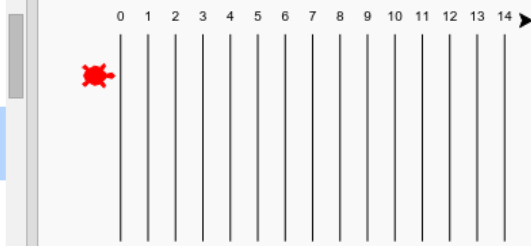


Η πρώτη γραμμή δημιουργεί μια χελώνα που ονομάζεται 'ada'. Οι επόμενες γραμμές ορίζουν το χρώμα και το σχήμα της χελώνας. Τώρα αληθινά μοιάζει με μια χελώνα!

- Ας στείλουμε τη χελώνα στην γραμμή εκκίνησης:

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

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



- Τώρα πρέπει να κάνεις την χελώνα να κινείται έναν τυχαίο αριθμό βημάτων κάθε φορά. Θα χρειαστείς τη συνάρτηση `randint` από τη βιβλιοθήκη `random` της Python. Πρόσθεσε την γραμμή `import` στην κορυφή του προγράμματός σου:

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

- Η συνάντηση `randint` επιστρέφει έναν τυχαίο (ακέραιο) αριθμό μεταξύ των επιλεγμένων τιμών. Η χελώνα θα προχωρά 1, 2, 3, 4 ή 5 βήματα κάθε φορά.

```
ada.penup()  
ada.goto(-160, 100)  
ada.pendown()  
  
for turn in range(100):  
    ada.forward(randint(1,5))
```

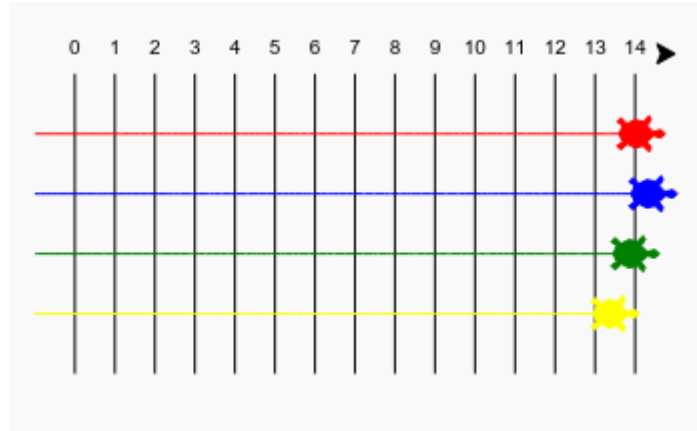
- Με μία χελώνα δεν γίνεται αγώνας! Ας προσθέσουμε άλλη μία:

```
bob = Turtle()  
bob.color('blue')  
bob.shape('turtle')  
  
bob.penup()  
bob.goto(-160, 70)  
bob.pendown()  
  
for turn in range(100):  
    ada.forward(randint(1,5))  
    bob.forward(randint(1,5))
```

Σημείωσε ότι ο κώδικας για την κίνηση της μπλε χελώνας πρέπει να είναι ο ίδιος `for` βρόχος όπως στον κώδικα για την κίνηση της κόκκινης χελώνας έτσι ώστε καθεμιά να κινείται με τη σειρά.

Τώρα είσαι έτοιμος να αγωνιστείς. Επέλεξε μία χελώνα και μία αντίπαλο και δεξ ποιος θα κερδίσει.

Μπορείς να προσθέσεις περισσότερες χελώνες ώστε να αγωνίζεσαι με περισσότερους φίλους;



Τα χρώματα περιλαμβάνουν: πορτοκαλί, μοβ, βιολετί, ντοματί, τουρκουάζ, φούξια και καφέ - ή μπορείς να μεταβείς στο jump.to/cc/colours και να επιλέξεις οποιοδήποτε χρώμα θέλεις!

Βήμα 7

Πρόκληση: Κάνε ένα στριφογύρισμα

Μπορείς να χρησιμοποιήσεις ένα `for turn in range()`: βρόχο για να κάνεις κάθε χελώνα να κάνει μια περιστροφή 360 μοιρών αφού φτάσει στη γραμμή εκκίνησης; Θα πρέπει να βεβαιωθείς ότι θα κοιτούν προς τον τερματισμό!

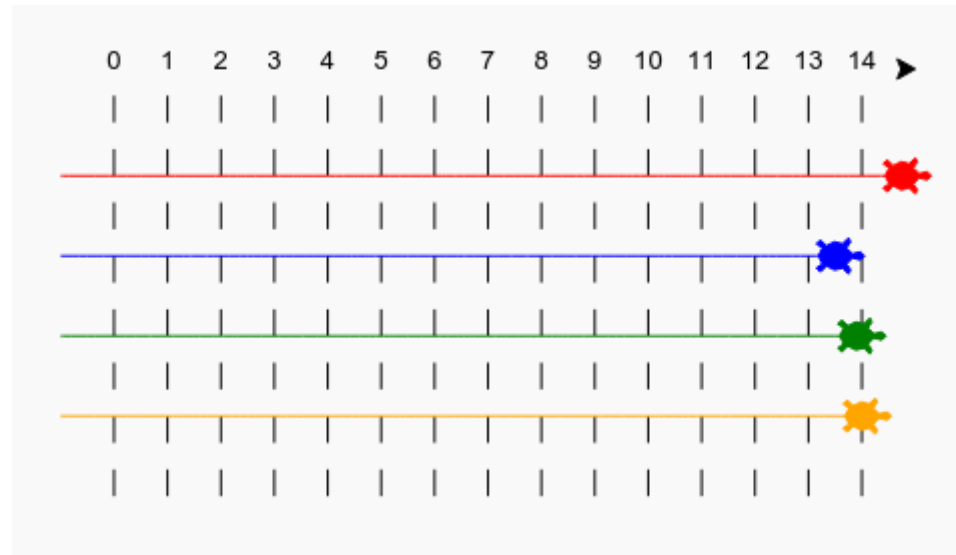
Με την `ada.right(36)` η κόκκινη χελώνα θα στρίψει δεξιά κατά 36 μοίρες.

Συμβουλή: Μια πλήρης στροφή είναι 360 μοίρες. Μια χελώνα θα μπορούσε να στρίψει δεξιά 10 μοίρες για 36 φορές, ή αριστερά 5 μοίρες για 72 φορές, ή όποιο άλλο συνδυασμό κάνει 360!

Βήμα 8

Πρόκληση: Διακεκομμένες γραμμές

Μπορείς να χρησιμοποιήσεις ένα βρόχο για να κάνεις τις γραμμές αγώνα διακεκομμένες αντί για συμπαγείς;



Συμβουλή: Βρες τον κώδικα που σχεδιάζει μια ευθεία γραμμή. Δοκίμασε να χρησιμοποιήσεις τις εντολές: `for`, `forward()`, `penup()` και `pendown()`

Κοινοτική συνεισφορά μετάφρασης

Το έργο αυτό μεταφράστηκε από Γιώργος Νικολαΐδης και αναθεωρήθηκε από Νίκος Γρηγορόπουλος.

Οι εκπληκτικοί μας εθελοντές-μεταφραστές μας βοηθούν να δώσουμε στα παιδιά όλου του κόσμου την ευκαιρία να μάθουν να προγραμματίζουν. Μπορείτε να μας βοηθήσετε να προσεγγίσουμε περισσότερα παιδιά μεταφράζοντας τα έργα μας - διαβάστε περισσότερα στο rpf.io/translators (<https://rpf.io/translators>).

Δημοσιεύθηκε από **Raspberry Pi Foundation** (<https://www.raspberrypi.org>) κάτω από ένα **Creative Commons license** (<https://creativecommons.org/licenses/by-sa/4.0/>).

Δες το έργο και την άδεια χρήσης στο **GitHub** (<https://github.com/RaspberryPiLearning/turtle-race>)