

PYTHON – ΠΑΡΑΔΕΙΓΜΑ

Το παράδειγμα θα το “τρέξουμε” στο on line εργαλείο : <https://trinket.io> – Python with Turtles

Ας γνωρίσουμε τον Ρίκο, το πράσινο ψηφιακό ρομπότ-χελώνα που θα εκτελεί εντολές με σκοπό την εξοικείωσή μας με την εκπαιδευτική ρομποτική.

Χρειάζεται να γνωρίζεις τα εξής:

- Καθετί που γράφεται σε κελί κώδικα μετά το σύμβολο '#' αγνοείται από τον υπολογιστή και υπάρχει μόνο για δική μας διευκόλυνση.
- Η εντολή "import" επιτρέπει τη χρήση βιβλιοθηκών που έχουν δημιουργηθεί από τρίτους.
- Κάθε εντολή της μορφής 'α.β' μεταφράζεται ως 'χρησιμοποίησε το β από τη βιβλιοθήκη α'.
- Η Python είναι case-sensitive δηλαδή απαιτεί προσοχή στη διάκριση μεταξύ πεζών και κεφαλαίων.
- Κάθε σύνολο χαρακτήρων πριν το σύμβολο '=' αποτελεί μία μεταβλητή στην οποία εκχωρούμε ότι δηλώνεται μετά το σύμβολο '='.

```
In [1]: import turtle          # επιτρέπει τη χρήση της βιβλιοθήκης 'turtle'
wn = turtle.Screen()          # δημιουργεί τον χώρο κίνησης του Ρίκο
Rico = turtle.Turtle()        # δημιουργεί και ονομάζει ως Ρίκο τη χελώνα μας.
```

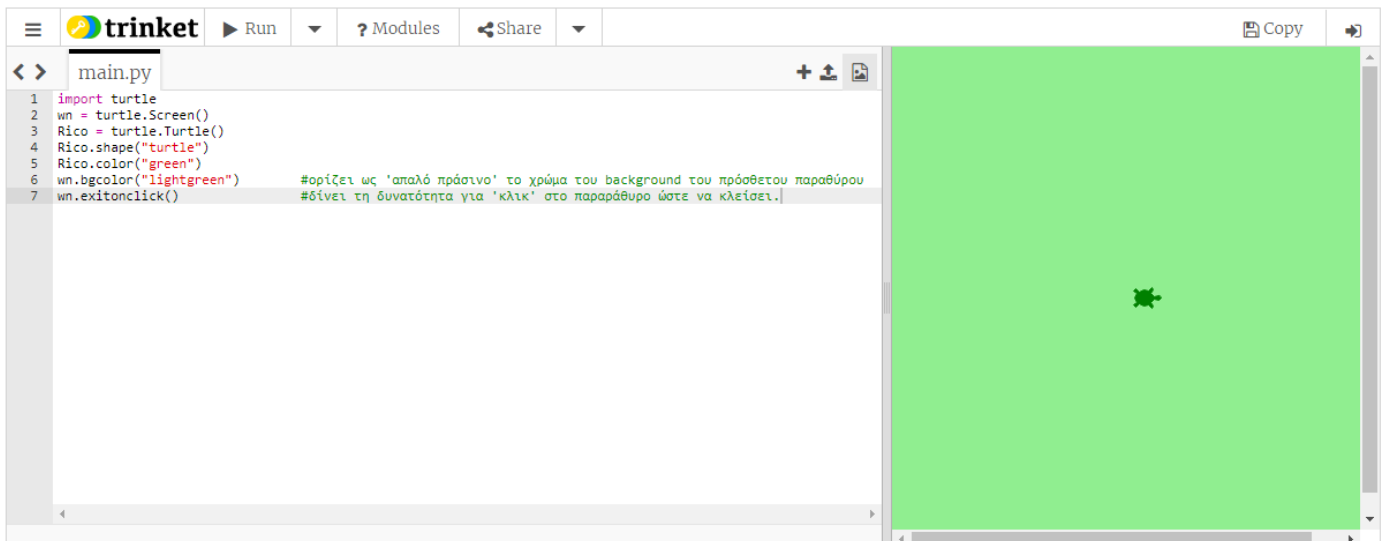
Μετά την εκτέλεση του παραπάνω κώδικα παρατηρούμε ότι παρότι δημιουργείται ένα εξωτερικό γραφικό περιβάλλον, ο Ρίκο όχι απλά δεν είναι πράσινος, αλλά δεν είναι καν χελώνα!

Αυτό πρέπει να το διορθώσουμε άμεσα.

```
In [ ]: import turtle
wn = turtle.Screen()
Rico = turtle.Turtle()
Rico.shape("turtle")
Rico.color("green")
```

Τίσως θα επιθυμούσαμε να μπορούμε να επεξεργαστούμε και τον χώρο κίνησης του Ρίκο ώστε να αναβαθμίσουμε την ποιότητα της σύντομης ζωής του.

```
In [ ]: import turtle
wn = turtle.Screen()
Rico = turtle.Turtle()
Rico.shape("turtle")
Rico.color("green")
wn.bgcolor("lightgreen")      #ορίζει ως 'απαλό πράσινο' το χρώμα του background του πρόσθετου παραθύρου
wn.exitonclick()             #δίνει τη δυνατότητα για 'κλικ' στο παράθυρο ώστε να κλείσει.
```

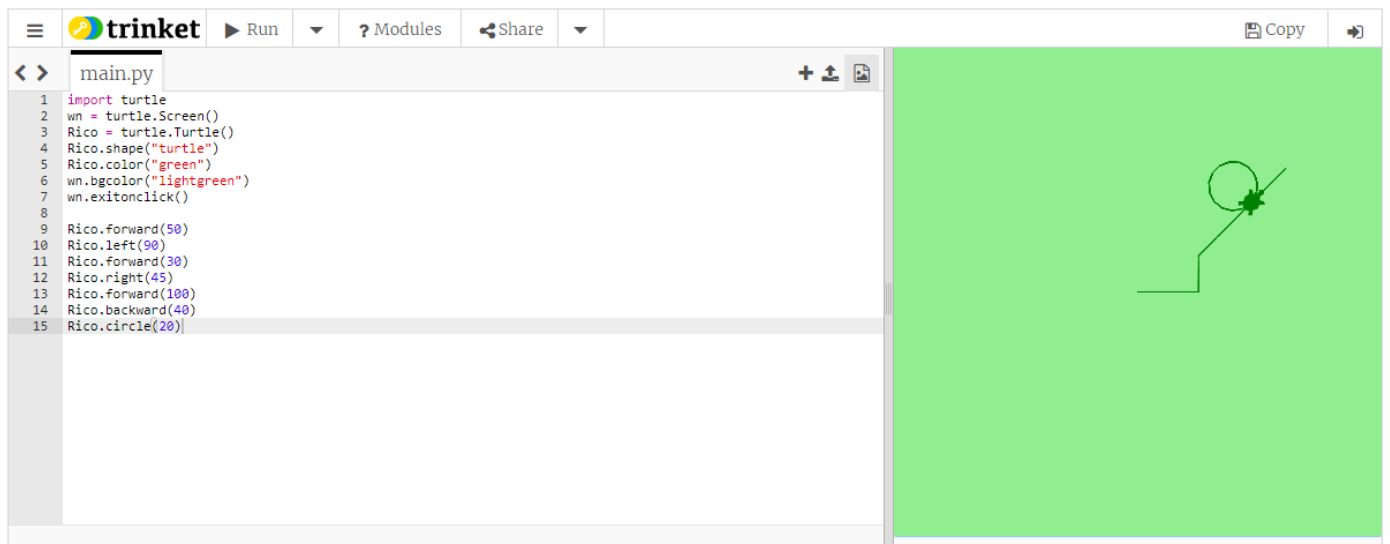


Οι τρόποι με τους οποίους μπορεί να κινηθεί ο Ρίκο είναι λίγοι αλλά αποτελεσματικοί.

Δες τον παρακάτω κώδικα (εκτέλεσέ τον εάν χρειαστεί) και προσπάθησε να καταλάβεις τι κάνει κάθε μία από τις διαθέσιμες εντολές κίνησης.

```
In [ ]: import turtle
wn = turtle.Screen()
Rico = turtle.Turtle()
Rico.shape("turtle")
Rico.color("green")
wn.bgcolor("lightgreen")
wn.exitonclick()

Rico.forward(50)
Rico.left(90)
Rico.forward(30)
Rico.right(45)
Rico.forward(100)
Rico.backward(40)
Rico.circle(20)
```



Έχουμε τη δυνατότητα να επιλέξουμε πότε θα αφήνει σημάδι περπατώντας και πότε όχι με τις εντολές **penup** και **pendown**.

Επίσης έχουμε τη δυνατότητα να τον κατευθύνουμε σε συγκεκριμένη θέση (x,y) του επιπέδου με την εντολή **goto(x,y)**. Είναι σημαντικό να γνωρίζετε πως ο χώρος κίνησης της χελώνας αποτελείται από ένα τετράγωνο με διαγώνια σημεία (-200,-200),(200,200).

Τα παραπάνω, σε συνδυασμό με την εντολή δημιουργίας κύκλων, μπορεί να δημιουργήσει αστεία πρόσωπα!

```
In [ ]: import turtle
Rico = turtle.Turtle()
Rico.shape('turtle')

Rico.penup()
Rico.goto(30,-150)
Rico.pendown()
Rico.circle(130)
Rico.penup()
Rico.goto(0,0)
Rico.pendown()
Rico.circle(20)
Rico.circle(10)
Rico.penup()
Rico.forward(60)
Rico.right(45)
Rico.pendown()
Rico.circle(30)
Rico.circle(10)
Rico.penup()
Rico.right(90)
Rico.forward(90)
Rico.pendown()
Rico.circle(40)
Rico.penup()
Rico.goto(25,-25)
```

trinket

Run

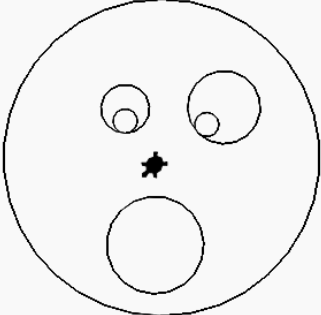
Modules

Share

Copy

main.py

```
1 import turtle
2 Rico = turtle.Turtle()
3 Rico.shape('turtle')
4
5 Rico.penup()
6 Rico.goto(30,-150)
7 Rico.pendown()
8 Rico.circle(130)
9 Rico.penup()
10 Rico.goto(0,0)
11 Rico.pendown()
12 Rico.circle(20)
13 Rico.circle(10)
14 Rico.penup()
15 Rico.forward(60)
16 Rico.right(45)
17 Rico.pendown()
18 Rico.circle(30)
19 Rico.circle(10)
20 Rico.penup()
21 Rico.right(90)
22 Rico.forward(90)
23 Rico.pendown()
24 Rico.circle(40)
25 Rico.penup()
26 Rico.goto(25,-25)
```



Τώρα ας προσπαθήσουμε να κάνουμε τον Ρίκο να δημιουργήσει μερικά βασικά σχήματα. Αφότου καταλάβεις τον τρόπο δημιουργίας των σχημάτων, θα κληθείς να τροποποιήσεις λίγο τον κώδικα ώστε να λύσεις τις πρώτες σου ασκήσεις.

In []: *#Δημιουργία τετραγώνου*

```
import turtle
wn = turtle.Screen()
Rico = turtle.Turtle()
Rico.shape("turtle")
Rico.color("green")
wn.bgcolor("lightgreen")
wn.title("Rico's World")
wn.exitonclick()

turtle.forward(50)
turtle.left(90)
turtle.forward(50)
turtle.left(90)
turtle.forward(50)
turtle.left(90)
turtle.forward(50)
turtle.left(90)
```

≡

trinket

▶ Run

▼

? Modules

↔ Share

▼

Copy

🔗

< >

main.py

+ ↕ 🖼

```
1 #Δημιουργία τετραγώνου
2
3 import turtle
4 wn = turtle.Screen()
5 Rico = turtle.Turtle()
6 Rico.shape("turtle")
7 Rico.color("green")
8 wn.bgcolor("lightgreen")
9
10 wn.exitonclick()
11
12 turtle.forward(50)
13 turtle.left(90)
14 turtle.forward(50)
15 turtle.left(90)
16 turtle.forward(50)
17 turtle.left(90)
18 turtle.forward(50)
19 turtle.left(90)
```

In []: #Δημιουργία τριγώνου

```
import turtle
wn = turtle.Screen()
Rico = turtle.Turtle()
Rico.shape("turtle")
Rico.color("green")
wn.bgcolor("lightgreen")
wn.title("Rico's World")
wn.exitonclick()

Rico.forward(25)
Rico.left(120)
Rico.forward(25)
Rico.left(120)
Rico.forward(25)
```

trinket

Run

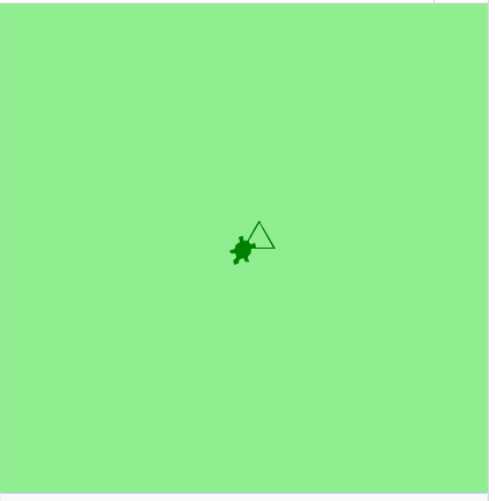
Modules

Share

Copy

main.py

```
1 #Δημιουργία τριγώνου
2
3 import turtle
4 wn = turtle.Screen()
5 Rico = turtle.Turtle()
6 Rico.shape("turtle")
7 Rico.color("green")
8 wn.bgcolor("lightgreen")
9
10 wn.exitonclick()
11
12 Rico.forward(25)
13 Rico.left(120)
14 Rico.forward(25)
15 Rico.left(120)
16 Rico.forward(25)
```



Και τώρα η σειρά σου. Προσπάθησε να τροποποιήσεις τους παραπάνω κώδικες ώστε να φτιάξεις ένα παραλληλόγραμμο και έπειτα τρία διαδοχικά -ελαφρώς απομακρυσμένα- τρίγωνα.

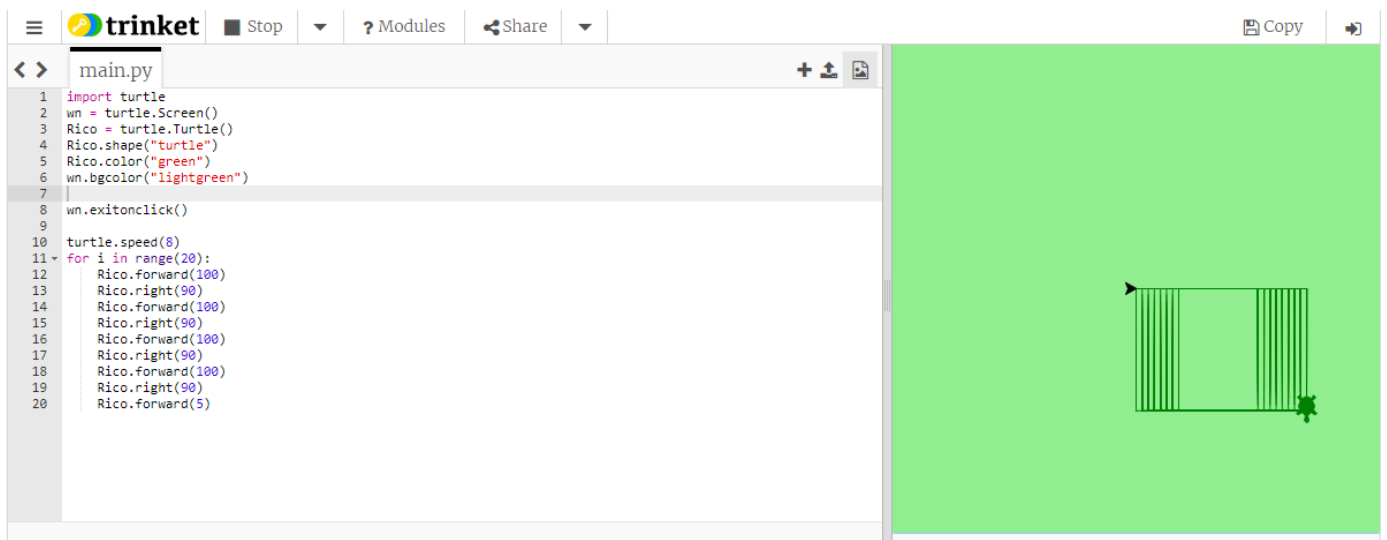
Έπειτα δοκίμασε να δημιουργήσεις ένα πρόσωπο (αντίστοιχο με αυτό που δημιουργήθηκε) που θα έχει τριγωνική μύτη!

Τέλος προσπάθησε να δημιουργήσεις ένα τετράγωνο του οποίου οι γωνίες δεν θα είναι εμφανείς, σηκώνοντας και κατεβάζοντας το μολύβι κατάλληλα.

Η εντολή **for i in range(n)** επαναλαμβάνει οτιδήποτε βρίσκεται στην εμβέλειά της ως εξής:

```
In [ ]: import turtle
wn = turtle.Screen()
Rico = turtle.Turtle()
Rico.shape("turtle")
Rico.color("green")
wn.bgcolor("lightgreen")
wn.title("Rico's World")
wn.exitonclick()

turtle.speed(8)
for i in range(20):
    Rico.forward(100)
    Rico.right(90)
    Rico.forward(100)
    Rico.right(90)
    Rico.forward(100)
    Rico.right(90)
    Rico.forward(100)
    Rico.right(90)
    Rico.forward(5)
```



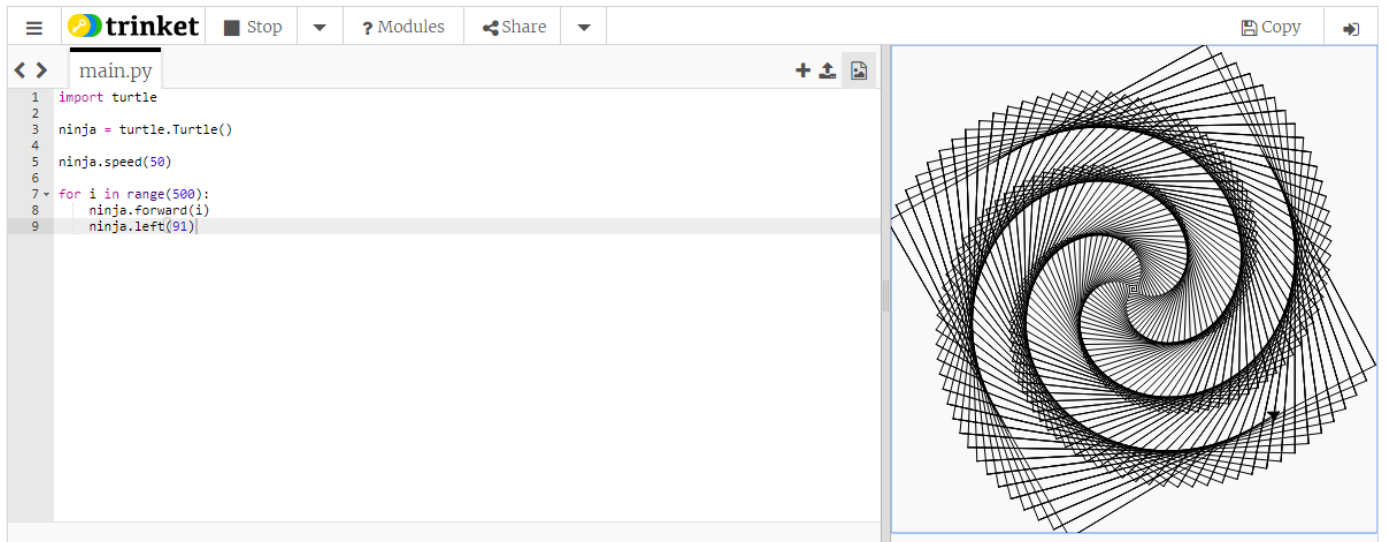
Με επανάληψη απλών εντολών μπορείς να πετύχεις εντυπωσιακά σχέδια όπως το παρακάτω.

```
In [ ]: import turtle

ninja = turtle.Turtle()

ninja.speed(50)

for i in range(500):
    ninja.forward(i)
    ninja.left(91)
```



Δοκίμασε να δημιουργήσεις τα δικά σου σχέδια και να αποθηκεύσεις τις εικόνες ως ενθύμιο από το μάθημα.

Η τελευταία εντολή που χρειάζεται να μάθεις είναι η εντολή επιλογής. Παρατήρησε τον παρακάτω κώδικα και πειράζοντας την τιμή της μεταβλητής `height`, κάνε τον Ρίκο να αλλάζει επιλογές.

```
In [ ]: import turtle
Rico=turtle.Turtle()
Rico.shape('turtle')
Rico.penup()

#πείραξε την παρακάτω γραμμή
height = -10

try:
    if height > 150 and height <= 200:
        Rico.write("This is very high!")
    elif height > 100 and height <= 150:
        Rico.write("This is high!")
    elif height > 0 and height <= 100:
        Rico.write("This is high but not too high!")
    elif height > -100 and height <= 0:
        Rico.write("This is low but not too low!")
    elif height > -150 and height <= -100:
        Rico.write("This is low!")
    elif height >= -200 and height <= -150:
        Rico.write("This is very low!")
    else:
        raise
except:
    Rico.backward(100)
    Rico.write("Hey, that's not a number between 200 and -200!")
    Rico.backward(200)
```

