

Python & Java {4} Teachers



Näin luodaan hirviö

Level 2 - Python



Funded by the
Erasmus+ Programme
of the European Union

{4}

Python & Java 4 Teachers

Tehtävä

- Projektissa hyödynnät muotojen ja taiteen tuntemustasi ja luot hirviön naaman!

Laajennos:

- Kokeile käyttää eri muotoja ja värejä ja luo täysin omannäköisesi otus.
- Esimerkki: yksi timantin muotoinen silmä, kolmen kolmion muodostama nenä, kolmion mallinen pää. Saat vapaat kädet.



Prosessi

Ohjelman pitäisi

- ✓ Käyttää Python-kielen Turtle-grafiikkakirjastoa,
- ✓ Hyödyntää silmukoita ja aliohjelmia, joilla teet säännöllisiä kuvioita,
- ✓ Kääntelet Turtlea, jotta voit piirtää eri paikkoihin ruutua.



Python-kirjastot

Kirjastoissa on hyödyllisiä funktioita, joita saatat tarvita koodatessa. Sinun ei tarvitse kirjoittaa kaikkea alusta saakka, koska voit hyödyntää muiden tekemiä kirjastoja.

```
1 from random import *  
2  
3 import Tkinter  
4  
5 import statistics  
6
```

Python-kielessä kirjastot tuodaan käytettäväksi 'import'-käskyllä.

Eräs hyvin paljon käytetty kirjasto on 'random'-moduuli eli satunnaislukuja tuottava kirjasto.

1. vaihe

Tuo Turtle-kirjasto

Tuodaan Turtle-grafiikkakirjasto ohjelmaamme, joten voimme piirtää kuvioita Python-kielellä.

```
1 #monster Code
2 #Imports the Turtle library
3 from turtle import *
4 #Change sthe cursor shape to a turtle
5 shape("turtle")
6 #Increases the speed of the turtle
7 delay(0)
8 #Changes the thickness of the pen
9 pensize(4)
```

Muotoa (shape) muuttamalla voidaan vaihtaa kursorin ulkoasua. Kynän paksuutta (pen size) suurennettiin, jotta hirviö näkyy paremmin.



Funded by the
Erasmus+ Programme
of the European Union

{4}

Python & Java 4 Teachers





Muutama Turtle-käsky

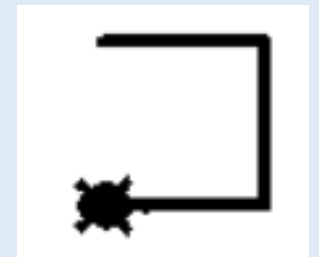
Turtle ymmärtää joitain yksinkertaisia käskyjä, kuten käänny oikealle tai vasemmalle, kulje eteenpäin ja taaksepäin. Englannin kielellä, tietenkin.

Yksinkertaisia Turtle-komentoja:

- `fd(10)` – kulje eteenpäin 10 pikseliä,
- `backward(5)` – kulje taaksepäin 5 pikseliä,
- `right(35)` – käänny kellon suuntaan (myötäpäivään) 35 astetta,
- `left(55)` – käänny vastapäivään 55 astetta;
- `goto(x, y)` – mene pisteeseen (x, y).

Tärkeä komento on myös `pu()` ja `pd()`, eli `pen up` (kynä ylös) ja `pen down` (kynä alas). Turtle saadaan siis kulkemaan ilman, että se piirtää viivaa.

```
54 fd(50)
55 rt(90)
56 fd(50)
57 lt(90)
58 backward(50)
```



2. vaihe

Määritellään aliohjelma

Riviltä 13 alkaa aliohjelman määrittely. Se tehdään `def()`-käskyllä, ja aliohjelman nimi on neliö, `square()`.

```
11 #These square and miniSquare subroutines are going to be used for the eyes of the monster
12 #Subroutines can be called within the program so that we don't have to keep rewriting code.
13 def square():
14     color('maroon', 'red')
15     begin_fill()
16     for loop in range(4):
17         fd(50)
18         rt(90)
19     end_fill()
```

Väritetään neliö punaiseksi ja ulkoreuna rusehtavaksi (maroon). Koska neliö on yksinkertainen, säännöllinen kuvio, hyödynnetään silmukkaa sen piirtämisessä.



Aliohjelmat

Aliohjelmat ovat koodilohkoja eli ohjelman käskyjä, mitkä toteuttavat tiettyjä asioita.

- Ne ovat pienoishjelmia ison ohjelman sisällä.
- Niissä on koodirivejä, jotka tekevät aiemmin määrättyjä asioita.
- Jos näitä samoja asioita käytetään useasti, ne kannattaa kirjoittaa aliohjelmaksi.
- Jokaiselle aliohjelmalle annetaan oma kuvaava nimi, jotta sitä on helppo kutsua koodista.
- Aliohjelmat lyhentävät koodin pituutta, mutta myös tekevät ne loogisemmaksi ja helpommaksi lukea ja ylläpitää.

```
#Määritellään aliohjelma:  
def nameOfSubroutine():  
    code goes here  
#Kusutaan aliohjelmaa:  
nameOfSubroutine()
```

```
1 def nameOfSubroutine(): #declaring  
2     print("hello")  
3 nameOfSubroutine()      #calling  
4
```

Silmukat



Funded by the
Erasmus+ Programme
of the European Union

Silmukka on koodilohko, jota toistetaan, kunnes silmukassa esitetty ehto toteutuu. Python-kielessä on kaksi perussilmukkatyyppiä. WHILE-silmukat (niitä on tässä ohjelmassa) ja FOR-silmukat.

WHILE-silmukan päättymistä tutkii ehtolause (condition). Silmukkaa toistetaan, kunnes ehto on epätosi (FALSE).

```
1
2 condition = True
3 while condition:
4     print("Repeating...")
5
6     print("Finish loop?")
7     finished = input()
8
9
10    if finished == "Y":
11        condition = False
12
```

```
Repeating...
Finish loop?
N
Repeating...
Finish loop?
N
Repeating...
Finish loop?
N
Repeating...
Finish loop?
Y
```

FOR-silmukoita toistetaan jonkun lukumäärän verran.

```
1
2 for i in range(5):
3     print(i)
4
5
```

```
0
1
2
3
4
```

3. vaihe

Määritellään toinen aliohjelma

Seuraava aliohjelma tekee pienemmän neliön, joka onkin hirviön silmä. Aliohjelma määritellään samanlaisesti kuin edellinen.

```
20 def miniSquare():
21     color('gold', 'yellow')
22     begin_fill()
23     for loop in range(4):
24         fd(25)
25         rt(90)
26     end_fill()
```

Neliön muoto on sama, mutta koko ja väritys eri.



4. vaihe

Kolmas aliohjelma

Kolmas ja viimeisin aliohjelma tekee nenän.

```
27 def triangle():  
28     fillcolor("Orange")  
29     begin_fill()  
30     for loop in range(3):  
31         fd(75)  
32         lt(120)  
33     end_fill()
```

Väritys on oranssi. Koska kolmiossa on kolme kulmaa, käännytään vain kolme kertaa ja silmukka käy kolme askelta.



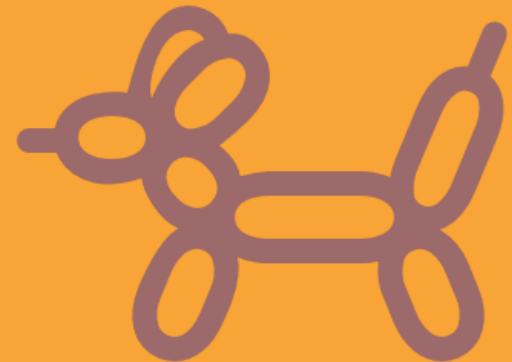
5. vaihe

Turtle oikeaan paikkaan.

Kun kaikki aliohjelmat on tehty, voidaan rauhassa keskittyä nenän, silmien ja sun laittamiseen oikeaan paikkaan.

```
34 #Main Code
35 #We have to move the turtle down the page so the face fits properly
36 pu()
37 rt(90)
38 fd(100)
39 lt(90)
40 pd()
```

Muista hyödyntää pen up() -komentoa, jotta naamaan ei tule ylimääräisiä viivoja. Toki muutama arpi voisi hirviölle sopia. Huomaa vielä, että koko naama nenineen mahtuu ikkunaan.



6. vaihe

Naaman piirtäminen

Turtle-kirjastossa on sisäänrakennettu ympyrän piirto -funktio, joten hyödynnetään sitä naaman piirroksessa.

```
41 #Starting with the shape of the head
42 #Turtle Library already has a circle method within it
43 #Pen changes to blue and fills in aqua
44 color('blue', 'aqua')
45 begin_fill()
46 circle(200)
47 end_fill()
48 #Completed the face shape of the monster
```

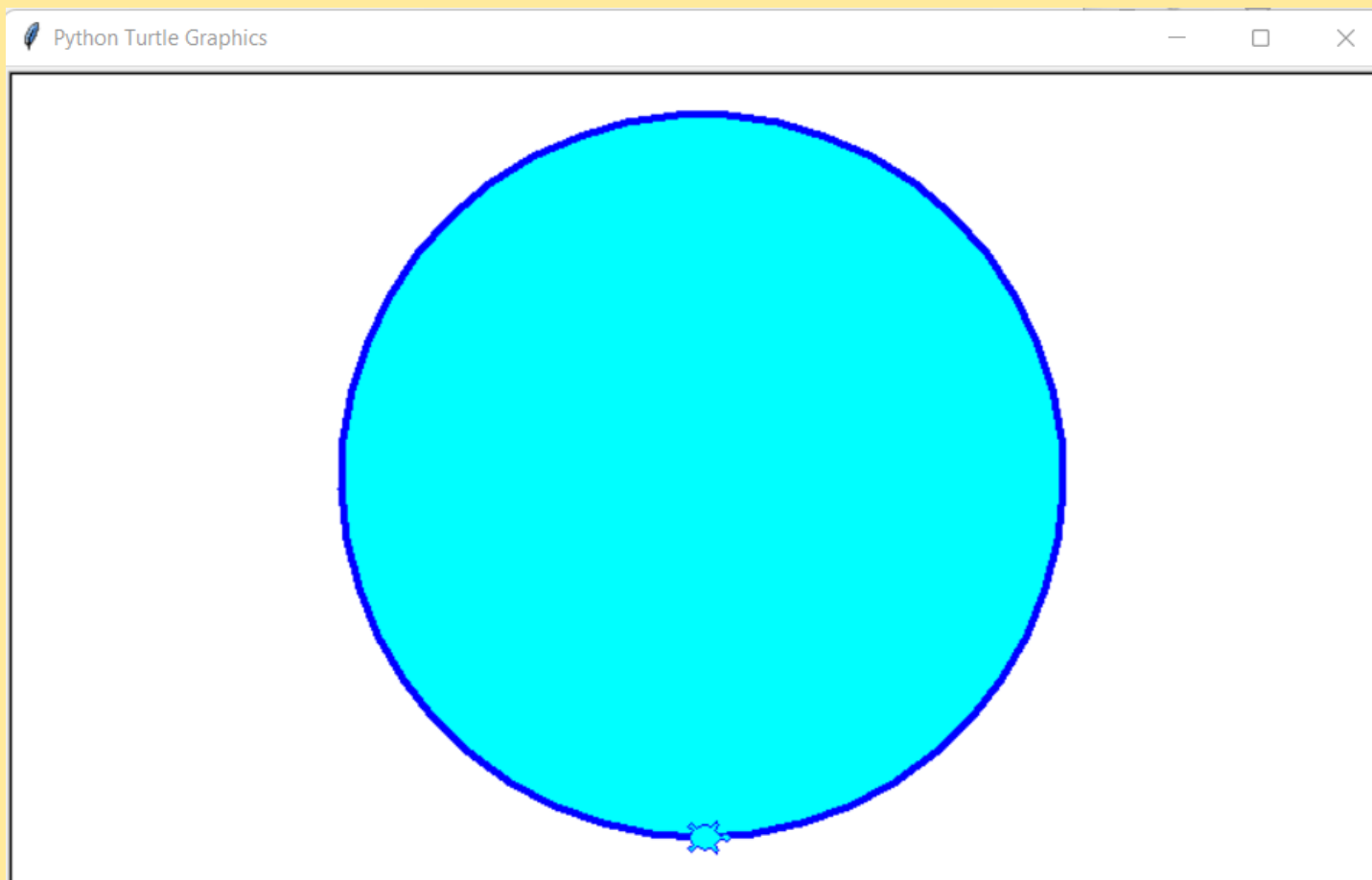
Ympyrän väri on 'aqua' ja viivan väri sininen.



Miltä se näyttää...

{4}

Python & Java 4 Teachers



7. vaihe

Tee silmät

Naaman piirrettyään kilpikonna on ruudun alareunassa, joten sitä täytyy vähän siirtää ylöspäin, että saadaan silmät oikeaan paikkaan.

```
49 #Starting on the eyes
50 #Moving the turtle so its in the right place
51 pu()
52 lt(90)
53 fd(300)
54 lt(90)
55 fd(175)
56 rt(180)
57 pd()
```

Siirretään Turtlea ylös ja vasemmalle, jotta silmät tulisivat oikeaan paikkaan.

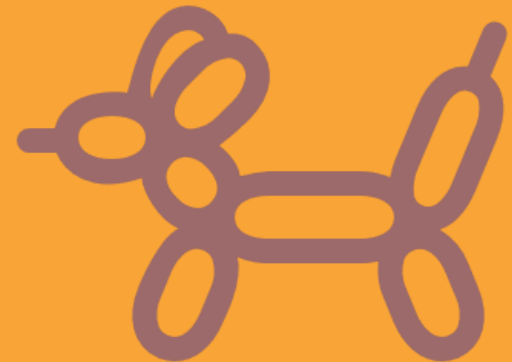


8. vaihe

Tee silmät

```
58 #eyes
59 #The 1st FOR loop draws the eye
60 for loop in range(3):
61     pu()
62     fd(75)
63     pd()
64     square()
65 miniSquare()
66 #2nd FOR loop draws pupils
67 for loop in range(2):
68     pu()
69     lt(180)
70     fd(75)
71     rt(180)
72     pd()
73     miniSquare()
```

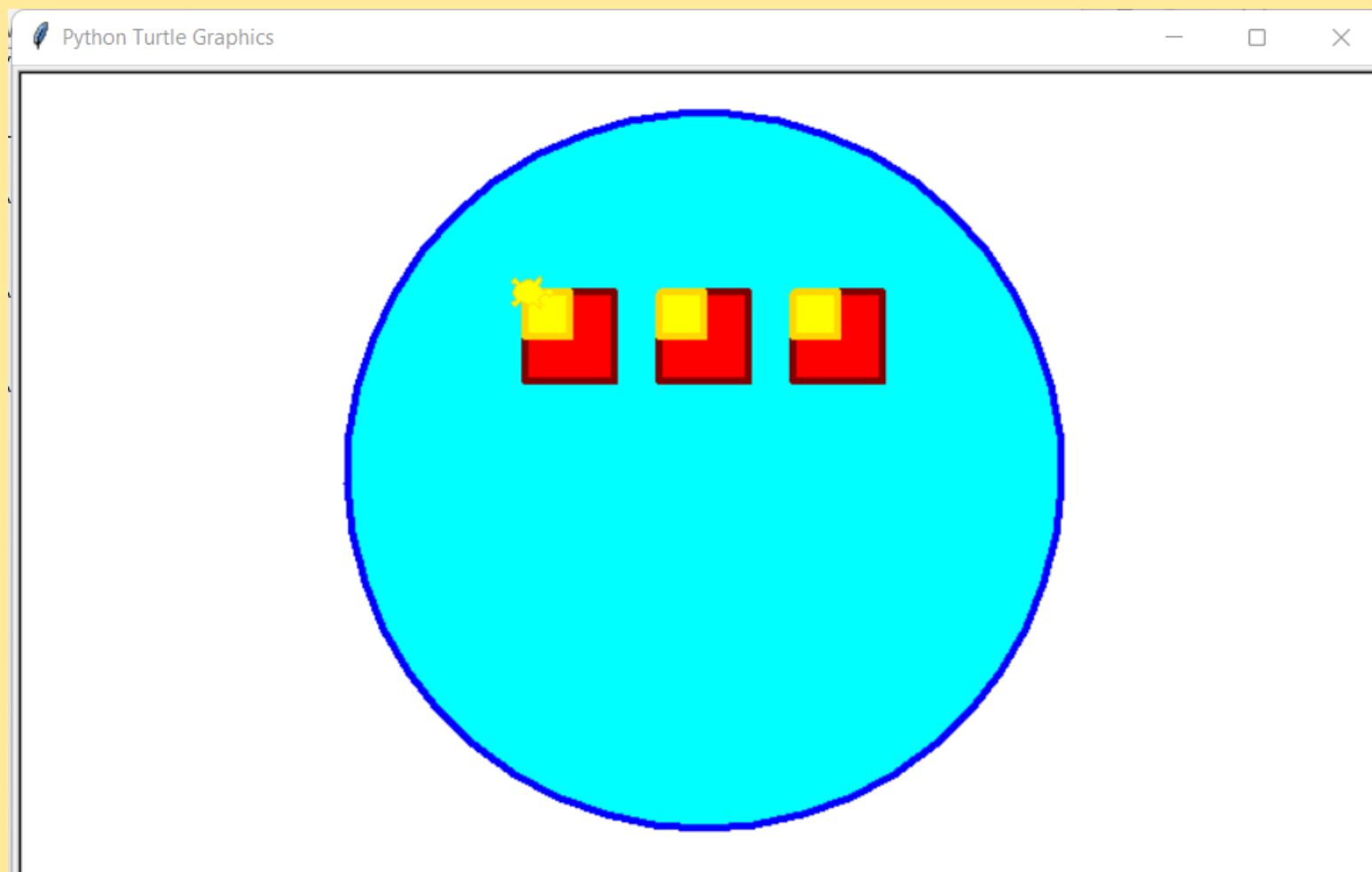
Ensimmäinen silmukka piirtää kolmiosaisen silmän kolmesta neliöstä. Piirretään keltainen pupilli ensin viimeiseksi piirrettyyn silmään ja silmukalla siirrytään muihin silmiin piirtämään pupilleja.



Miltä se näyttää...

{4}

Python & Java 4 Teachers



9. vaihe

Siirretään kilpikonnan

Silmät ovat valmiit. Nostetaan kynää pen up() -funktiolla, jotta kuvaan ei tule ylimääräisiä viivoja. Siirrytään keskelle naamaa ja piirretään seuraavaksi nenä.

```
75 #Completed the eyes
76 #Turns turtle in the right place for the mouth and nose to sit properly
77 pu()
78 rt(90)
79 fd(150)
80 lt(90)
81 fd(60)
82 pd()
```



Funded by the
Erasmus+ Programme
of the European Union

{4}

Python & Java 4 Teachers



10. askel

Piirrä nenä

Kolmio- eli triangle()-funktio määriteltiin rivillä 27.
Piirretään nenä kohdilleen.

```
83 triangle()  
84 pu()  
85 rt(90)  
86 fd(25)  
87 lt(90)  
88 fd(180)  
89 rt(105)  
90 pd()
```

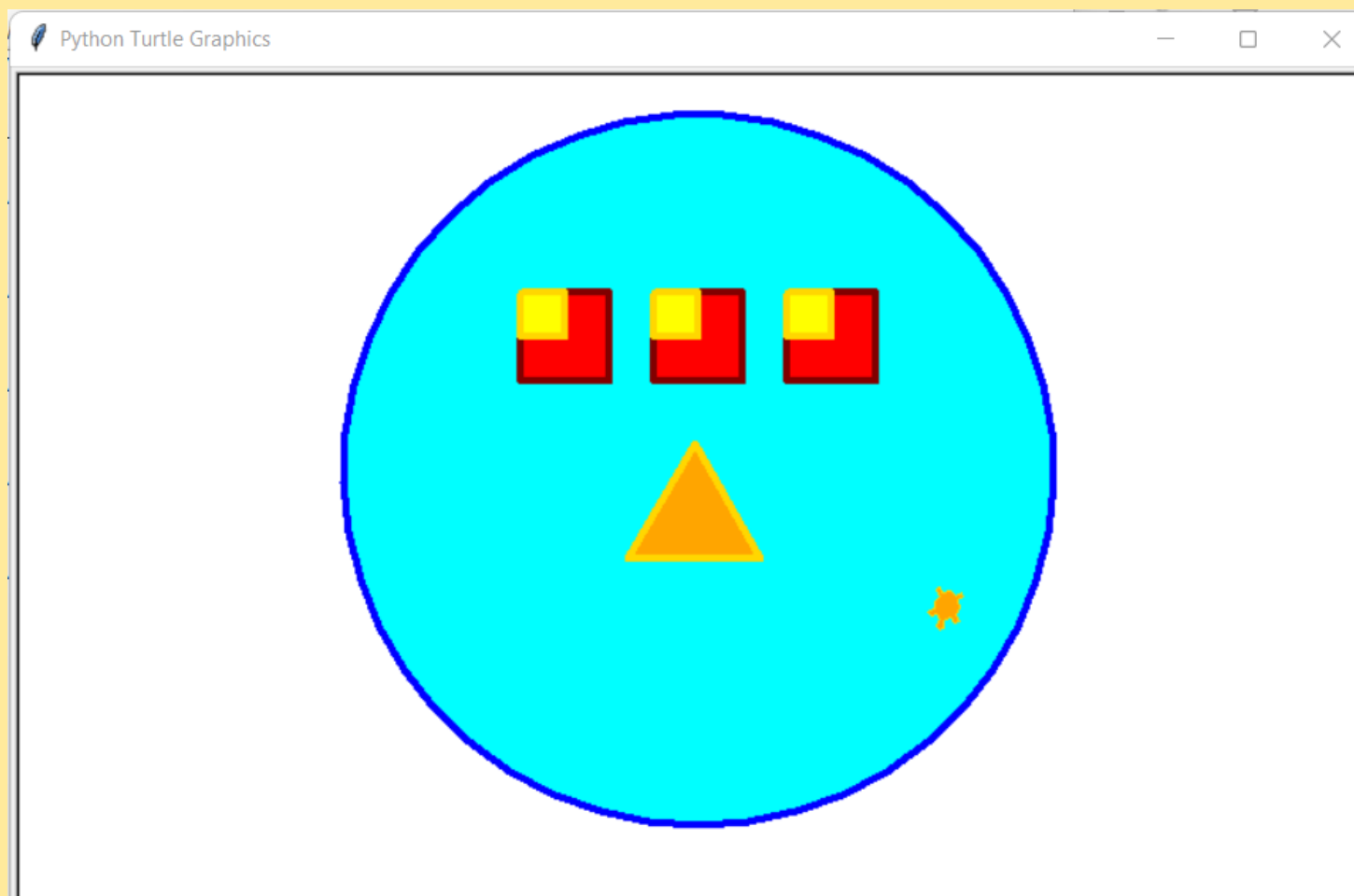
Seuraavaksi koodataan hirviön suu,
ja sitten se alkaakin olemaan jo valmis.



Miltä se näyttää...

{4}

Python & Java 4 Teachers



11. vaihe

Suu

Kynän väri vaihdetaan vihreäksi (green), ja täyttöväriksi tulee keltaisen vihreä.

```
91 #Changes the colour for the mouth
92 color('green', 'greenyellow')
93 begin_fill()
94 circle(-150, 150)
95 end_fill()
96 #Completed the mouth
```

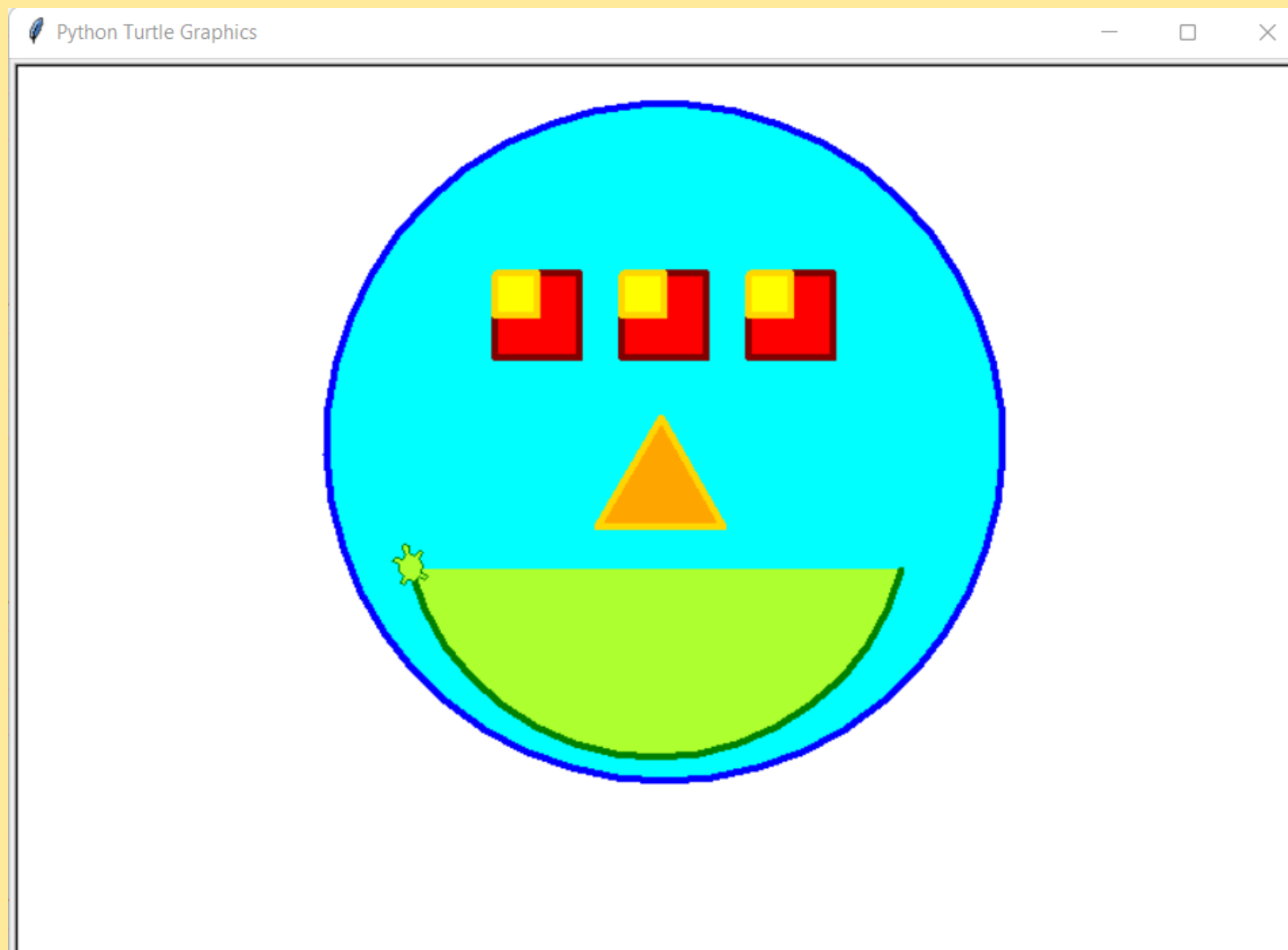
Rivillä 94 piirretään puoliympyrä, joka viimeistelee hirviön hymyn.



Miltä se näyttää...

{4}

Python & Java 4 Teachers



Lopuksi

Osaat

- ✓ Osaat tuoda Turtle-kirjaston,
- ✓ Osaat käyttää aliohjelmia kuvioiden piirtämiseen,
- ✓ Osaat käyttää kulmia ja käännellä kilpikonnan,
- ✓ Osaat säädellä Turtlen nopeutta ja kynän paksuutta.



Onnittelut!

Olet tehnyt hirviön naaman



{4}

Python & Java 4 Teachers