

ТЕМА 4: Компютърно моделиране

УРОК 14: Основни типове данни в скриптов текстов език за програмиране – числови типове данни

Връзка към решените задачи:

<https://it-schoolbooks-gmail-com.trinket.io/kompyutrno-modelirane-za-7-klas#/osnovni-tipove-danni-v-skriptov-tekstov-ezik-za-programirane/chislovi-tipove-danni>

Задача: Създайте програма за изчисляване на среден успех по география на база 3 получени оценки.

Решение:

```
1 grade_1 = 6
2 grade_2 = 5
3 grade_3 = 6
4
5 average_score = (grade_1 + grade_2+ grade_3)/3
6 average_score = round(average_score,2)
7
8 print('Среден успех: ')
9 print(average_score)
```

Резултат:

```
Среден успех:
5.67
>>>
```

УРОК 15: Основни типове данни в скриптов текстов език за програмиране – числови типове данни

Задача: Създайте програма за изчисляване на съпротивление по дадени големина на ток и напрежение.

Решение:

```
1 #Програма за изчисляване на съпротивление
2 print('Програма за изчисляване на съпротивление')
3 print('по дадени големина на ток и напрежение')
4 I = float(3)
5 V = int(2)
6
7 print('Големина на тока:')
8 print(I)
9
10 print('Напрежение:')
11 print(V)
12
13 R = V / I
14 print('Съпротивление:')
15 print(R)
```

Резултат:

```
Програма за изчисляване на съпротивление  
по дадени големина на ток и напрежение  
Големина на тока:  
3.0  
Напрежение:  
2  
Съпротивление:  
0.6666666666666666  
>>>
```

За любознателните: Въведете код, който да извежда изчисленото съпротивление, закръглено до втори десетичен знак.

Решение:

```
1 #Програма за изчисляване на съпротивление  
2 print('Програма за изчисляване на съпротивление')  
3 print('по дадени големина на ток и напрежение')  
4 I = float(3)  
5 V = int(2)  
6  
7 print('Големина на тока:')  
8 print(I)  
9  
10 print('Напрежение:')  
11 print(V)  
12  
13 R = V / I  
14 R = round(R,2)  
15 print('Съпротивление:')  
16 print(R)
```

Резултат:

```
Програма за изчисляване на съпротивление  
по дадени големина на ток и напрежение  
Големина на тока:  
3.0  
Напрежение:  
2  
Съпротивление:  
0.67  
>>> |
```

Задача: Създайте програма, която да изчислява обем на прав кръгов цилиндър по зададени височина и радиус на основата.

Решение:

```
1 #Обем на цилиндър
2 import math
3
4 R = float(10)
5 h = float(15)
6
7 B = math.pi * R * R
8 V = B * h
9
10 print('Обем:')
11 print(V)
```

Резултат:

```
Обем:
4712.38898038469
>>>
```

УРОК 16: Основни типове данни в скриптов текстов език за програмиране – низ

Връзка към решените задачи:

<https://it-schoolbooks-gmail-com.trinket.io/kompyutrno-modelirane-za-7-klas#/osnovni-tipove-danni-v-skriptov-tekstov-ezik-za-programirane/tekstov-tip-danni-niz>

Задача: Напишете програма, която извежда емотиконите:

:~)	усмивка
:-(тъга
:"D	смях до сълзи
<3	любов
@>-->--	поднасям роза
*< :-)	Честита Коледа!
("/)/	недоумение
d[-_-]b	слушам музика
~(^-^~	танцувам си

Решение:

1. Създайте нов проект на Python
2. Въведете командите

```
1 print(':-)')
2 print(':-(')
3 print(': "D')
4 print('<3')
5 print('@>-->--')
6 print('*<|:-)')
7 print('-\_(\"/)_/-'')
8 print('d[-_-]b')
9 print('~(^-^)~')
```

3. Тествайте програмата
4. Запишете файла

Резултат:

```
:-)
:-(
: "D
<3
@>-->--
*<|:-)
-\_(\"/)_/-'
d[-_-]b
~(^-^)~
>>>
```

Задача: Напишете програма, която извежда следните животни:

```
><(((('>    риба
----{,_,">  мишле
@(* O * )@  коала
@('_')@     майmunка
_/_/_/_0>  червей
```

Решение:

```
1 print("><(((('>")    #fish
2 print('----{,_,">') #mouse
3 print('@(* O * )@') #koala
4 print("@('_')@")    #monkey
5 print('_/_/_/_0>') #worm
```

Резултат:

```
><(((('>
----{,_,">
@(* O * )@
@('_')@
_/_/_/_0>
```


УРОК 18: Основни типове данни в скриптов текстове език за програмиране – низ и числови – конвертиране

Връзка към решените задачи:

<https://it-schoolbooks-gmail-com.trinket.io/kompyutrno-modelirane-za-7-klas#/osnovni-tipove-danni-v-skriptov-tekstov-ezik-za-programirane/tipove-danni-konvertirane>

Задача: Роботко обича да сглобява различни механизми. Първо подготвя две части, а след това ги свързва. Но когато изчислява колко болтчета ще му трябват, вечно се обърква. Затова иска да създаде програма, в която да въвежда броя на болтчетата за всяка част и извежда общия им брой.

Решение:

```
1 part_1 = input('Брой болтчета в първа част: ')
2 part_1 = int(part_1)
3 part_2 = input('Брой болтчета във втора част: ')
4 part_2 = int(part_2)
5
6 totalBolts = part_1 + part_2
7 print('Общо: ' + str(totalBolts))
```

Резултат:

```
Брой болтчета в първа част: 5
Брой болтчета във втора част: 4
Общо: 9
>>> |
```

Задача: Модифицирайте програмата за учене на думи, така че и броят повторения на дума да се въвежда от клавиатурата.

Решение:

```
1 def learn(count, word):
2     print(count * word)
3
4 word = input('Дума: ')
5 count = input('Брой: ')
6 count = int(count)
7 learn(count, word)
```

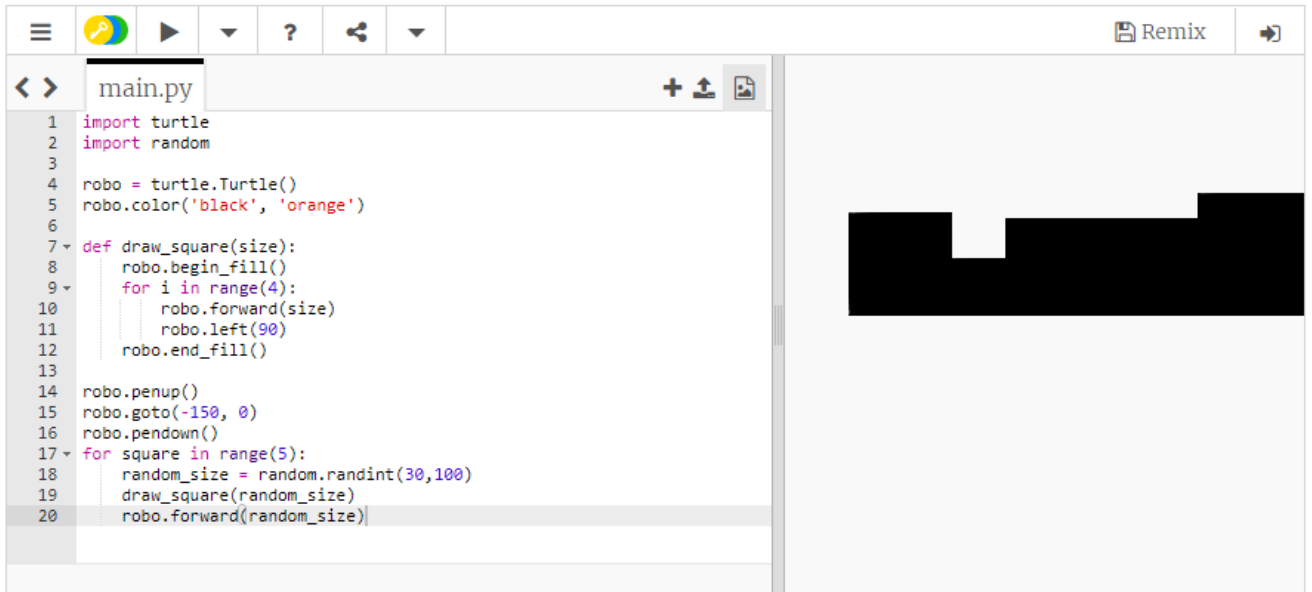
УРОК 19: Приложение на циклични конструкции. Цикъл for

Връзка към решените задачи:

<https://it-schoolbooks-gmail-com.trinket.io/kompyutrno-modelirane-za-7-klas#/prilozhenie-na-ciklichni-konstrukcii/cikl-for>

Задача: Квадратчо моли Роботко да построи ограда около къщата му. Преди да започне със строежа, Роботко иска да направи проект на оградата на Python и да го покаже на Квадратчо. Оградата трябва да се състои от 5 долепени една до друга квадратни плоскости с дължина на страната 50.

Решение:



```
1 import turtle
2 import random
3
4 robo = turtle.Turtle()
5 robo.color('black', 'orange')
6
7 def draw_square(size):
8     robo.begin_fill()
9     for i in range(4):
10         robo.forward(size)
11         robo.left(90)
12     robo.end_fill()
13
14 robo.penup()
15 robo.goto(-150, 0)
16 robo.pendown()
17 for square in range(5):
18     random_size = random.randint(30,100)
19     draw_square(random_size)
20     robo.forward(random_size)
```

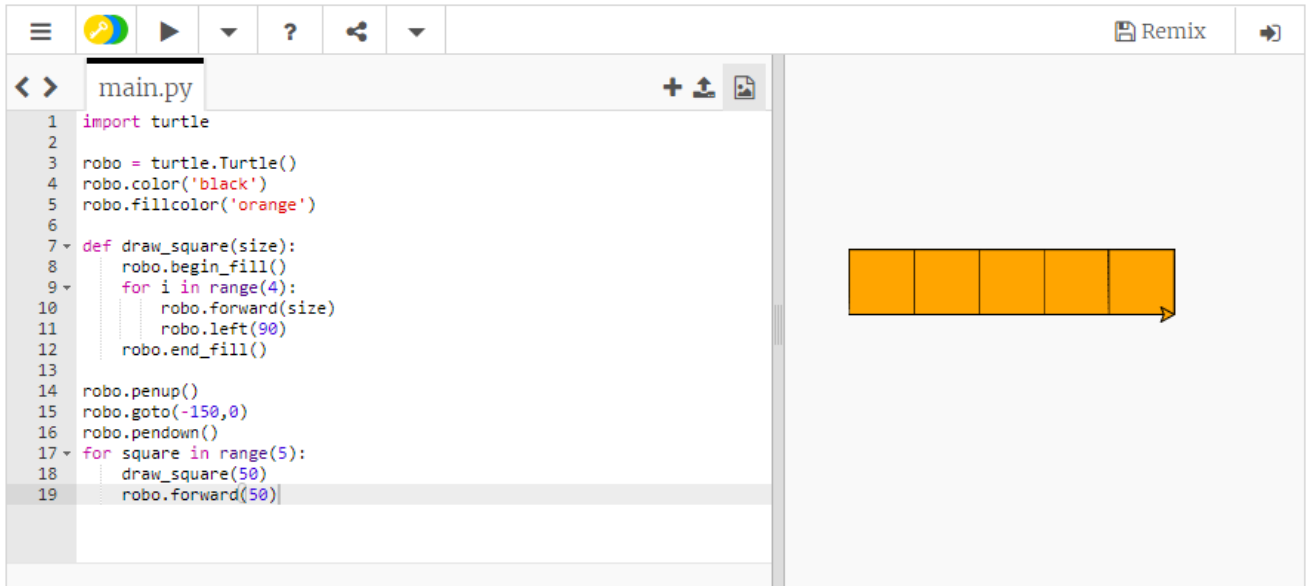
The output shows a black fence with five irregularly shaped panels of varying widths and heights.

Цветна ограда

Подобрете кода, като:

- оцветите панелите на оградата;
- укажете типа на параметъра на функция `draw_square`

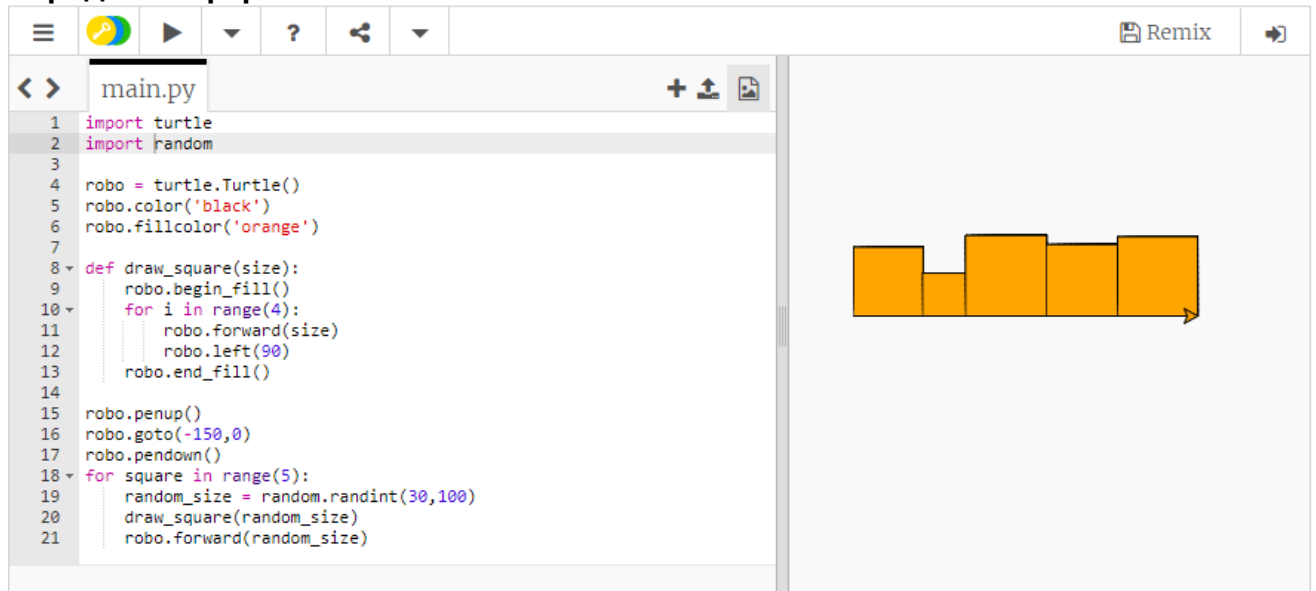
Решение:



```
1 import turtle
2
3 robo = turtle.Turtle()
4 robo.color('black')
5 robo.fillcolor('orange')
6
7 def draw_square(size):
8     robo.begin_fill()
9     for i in range(4):
10         robo.forward(size)
11         robo.left(90)
12     robo.end_fill()
13
14 robo.penup()
15 robo.goto(-150,0)
16 robo.pendown()
17 for square in range(5):
18     draw_square(50)
19     robo.forward(50)
```

The output shows a fence with five uniform orange square panels of size 50, each with a black outline.

Ограда за Сферичка



```
1 import turtle
2 import random
3
4 robo = turtle.Turtle()
5 robo.color('black')
6 robo.fillcolor('orange')
7
8 def draw_square(size):
9     robo.begin_fill()
10    for i in range(4):
11        robo.forward(size)
12        robo.left(90)
13    robo.end_fill()
14
15 robo.penup()
16 robo.goto(-150,0)
17 robo.pendown()
18 for square in range(5):
19     random_size = random.randint(30,100)
20     draw_square(random_size)
21     robo.forward(random_size)
```

The screenshot shows a Python IDE with a code editor on the left and a canvas on the right. The code defines a function `draw_square` that takes a size parameter and draws a square with a black outline and orange fill. The main program sets up a turtle named `robo` at `(-150, 0)` and uses a `for` loop to draw five squares of random sizes (between 30 and 100 units) along the x-axis.

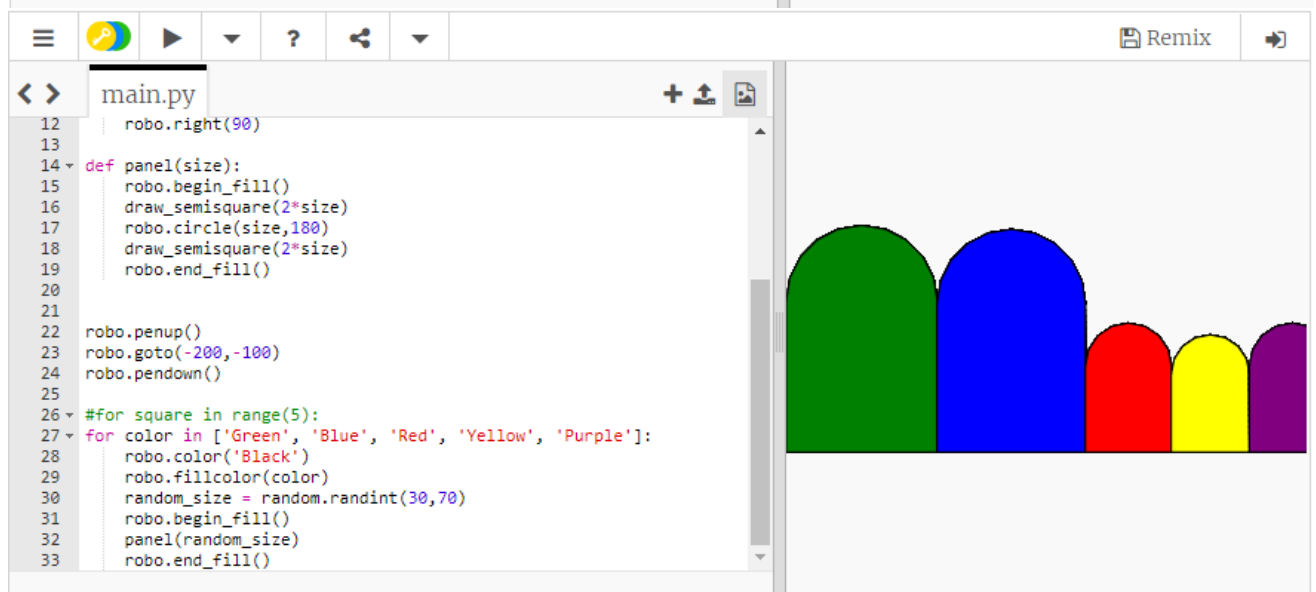
УРОК 20: Приложение на циклични конструкции. Цикъл for

Мисия: Заоблена ограда



```
1 import turtle
2 import random
3
4 robo = turtle.Turtle()
5 robo.color('black')
6 robo.fillcolor('orange')
7
8 def draw_semisquare(size):
9     for i in range(2):
10        robo.forward(size)
11        robo.left(90)
12        robo.right(90)
13
14 def panel(size):
15     robo.begin_fill()
16     draw_semisquare(2*size)
17     robo.circle(size,180)
18     draw_semisquare(2*size)
19     robo.end_fill()
20
21 robo.penup()
22
```

The screenshot shows a Python IDE with a code editor on the left and a canvas on the right. The code defines a function `draw_semisquare` that draws a semi-square (a square with one side open) and a function `panel` that draws a rounded panel by combining two semi-squares and a semi-circle. The main program sets up a turtle named `robo` and uses a `for` loop to draw five panels of random sizes (between 30 and 100 units) along the x-axis. The panels are colored green, blue, red, yellow, and purple.



```
12     robo.right(90)
13
14 def panel(size):
15     robo.begin_fill()
16     draw_semisquare(2*size)
17     robo.circle(size,180)
18     draw_semisquare(2*size)
19     robo.end_fill()
20
21 robo.penup()
22 robo.goto(-200,-100)
23 robo.pendown()
24
25
26 #for square in range(5):
27 for color in ['Green', 'Blue', 'Red', 'Yellow', 'Purple']:
28     robo.color('Black')
29     robo.fillcolor(color)
30     random_size = random.randint(30,70)
31     robo.begin_fill()
32     panel(random_size)
33     robo.end_fill()
```

The screenshot shows a Python IDE with a code editor on the left and a canvas on the right. The code defines a function `panel` that draws a rounded panel by combining two semi-squares and a semi-circle. The main program sets up a turtle named `robo` at `(-200, -100)` and uses a `for` loop to draw five panels of random sizes (between 30 and 70 units) along the x-axis. The panels are colored green, blue, red, yellow, and purple.

УРОК 21: Приложение на циклични конструкции. Цикъл *WHILE*

Връзка към решените задачи:

<https://it-schoolbooks-gmail-com.trinket.io/kompyutрно-modelirane-za-7-klas#/prilozhenie-na-ciklichni-konstrukcii/cikl-while>

Мисия: Касов апарат

```
main.py
1 # Касов апарат
2 total = 0
3
4 price = input('Цена: ')
5 while price != '#':
6     total = total + float(price)
7     print('Текуща сума: ' + str(total))
8     price = input('Цена: ')
9 print('Дължите общо: ' + str(total) + '')
```

Powered by trinket
Цена: 25
Текуща сума: 25.0
Цена: 12
Текуща сума: 37.0
Цена: 12
Текуща сума: 49.0
Цена: 1
Текуща сума: 50.0
Цена: █

Щастието на Сферичка

```
main.py
1 # Число на деня
2 day_num = input('Дата (ddmm): ')
3 day_num = int(day_num)
4 sum = 0
5 while (day_num > 0):
6     sum = sum + day_num % 10
7     day_num = day_num // 10
8 print('Щастливо число: ' + str(sum))
```

Powered by trinket
Дата (ddmm): 2307
Щастливо число: 12

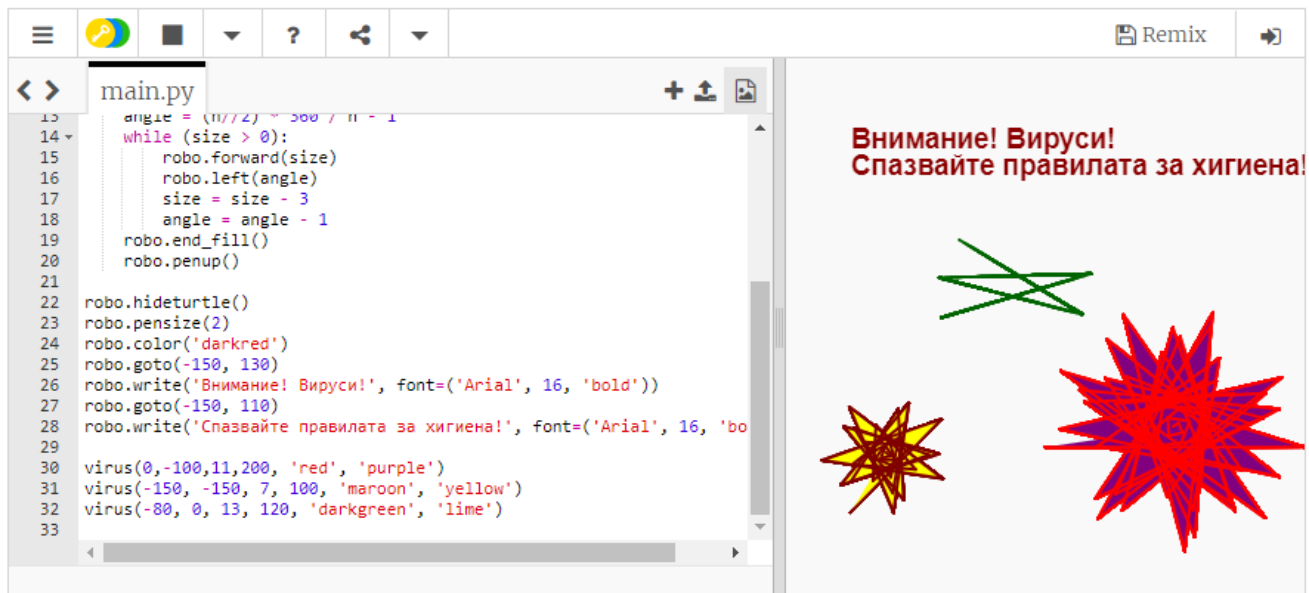
УРОК 22: Приложение на циклични конструкции. Цикъл *WHILE*

Мисия: Вирус!

```
main.py
1 import turtle
2 import random
3
4 robo = turtle.Turtle()
5 robo.penup()
6
7 def virus(x, y, n, size, contour, filling):
8     robo.goto(x,y)
9     robo.pendown()
10    robo.color(contour)
11    robo.fillcolor(filling)
12    robo.begin_fill()
13    angle = (n//2) * 360 / n - 1
14    while (size > 0):
15        robo.forward(size)
16        robo.left(angle)
17        size = size - 3
18        angle = angle - 1
19    robo.end_fill()
20    robo.penup()
21
22
```

Внимание! Вируси!
Спазвайте правилата за хигиена!





УРОК 23: Приложение на циклични конструкции.

Връзка към решените задачи:

<https://it-schoolbooks-gmail-com.trinket.io/kompyutрно-modelirane-za-7-klas#/prilozhenie-na-ciklichni-konstrukcii/prilozhenie-na-ciklichni-konstrukcii>

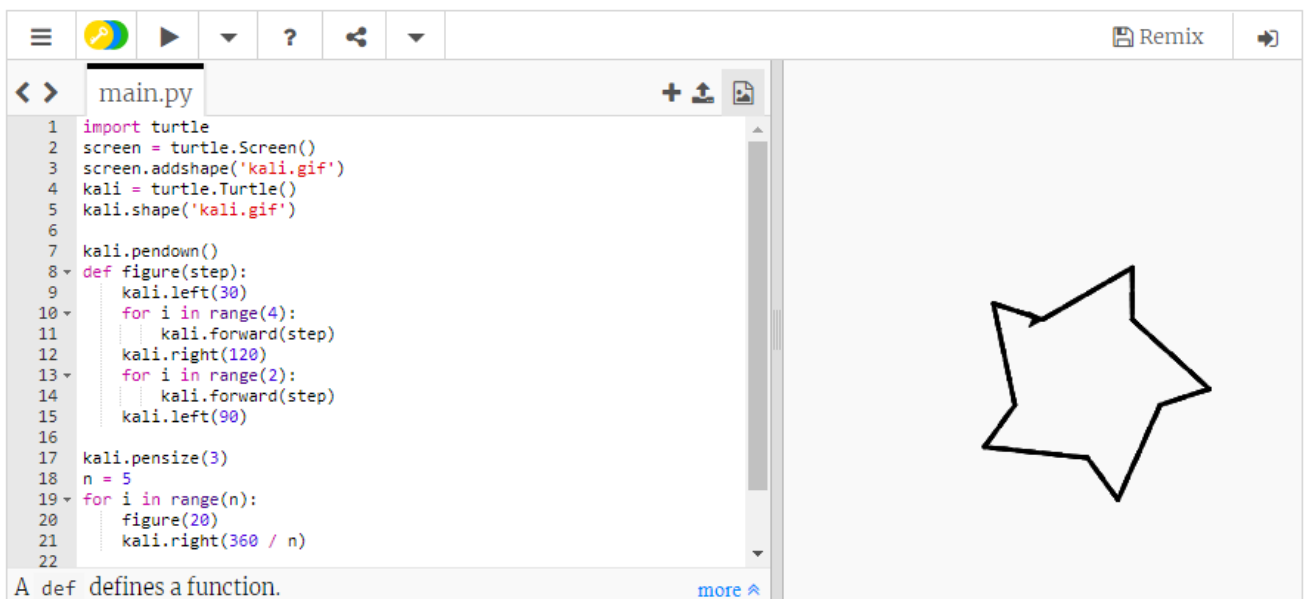
ТЕМА 5: Работа по проект

УРОК 26: Използване на скриптов текстов език за програмиране за реализиране на конкретен проект

Връзка към решените задачи:

<https://it-schoolbooks-gmail-com.trinket.io/kompyutрно-modelirane-za-7-klas#/proekt-praznik-na-khoroto/po-stpkite-na-taratanci>

По стъпките на Таратанци



main.py

```

1 import turtle
2
3 screen = turtle.Screen()
4
5 screen.addshape('stoyo.gif')
6 screen.addshape('kali.gif')
7
8 stoyo = turtle.Turtle()
9 stoyo.shape('stoyo.gif')
10
11 kali = turtle.Turtle()
12 kali.shape('kali.gif')
13
14 stoyo.penup()
15 stoyo.goto(-80, 110)
16 stoyo.pencolor('black')
17 stoyo.pensize(3)
18 stoyo.pendown()
19
20 kali.penup()
21 kali.goto(-100, 100)
22 kali.pencolor('red')

```

main.py

```

21 kali.goto(-100, 100)
22 kali.pencolor('red')
23 kali.pensize(2)
24 kali.pendown()
25
26 def people_left(degree):
27     for hero in screen.turtles():
28         hero.left(degree)
29
30
31 def people_right(degree):
32     for hero in screen.turtles():
33         hero.right(degree)
34
35
36 def people_forward(steps):
37     for hero in screen.turtles():
38         hero.forward(steps)
39
40 def figure_1(step):
41     people_left(30)
42     for i in range(4):

```

main.py

```

41     people_left(30)
42     for i in range(4):
43         people_forward(step)
44     people_right(120)
45     for i in range(2):
46         people_forward(step)
47     people_left(90)
48
49 def horo(n):
50     for i in range(n):
51         figure_1(20)
52         people_right(360/n)
53
54
55 horo(10)
56
57 kali.penup()
58 stoyo.penup()
59
60 stoyo.goto(-100, 180)
61 kali.goto(-120, 170)
62

```