

Python – Циклы

```
>>> for i in range(14, 18):  
    print(i)
```

ИЗУЧАЕМ:

- Список List;
- Цикл For;
- Функцию range()

ПОВТОРЯЕМ (ЗАКРЕПЛЯЕМ):

- input и print;
- работу с векторной графикой.



TURTLE



СПИСКИ (List)

Списки в Python — упорядоченные изменяемые коллекции объектов произвольных типов.

Списки можно создавать и генерировать с помощью специальных функций.

```
s = [] # Пустой список
```

```
l = ['s', 'p', ['isok'], 2]
```

```
l = list('лист') # ['л', 'и', 'с', 'т']
```

Как видно, список может быть и пустым.

Python

Списки

```
>>> data = [5, 'a', ['python'], 20]  
>>> data  
[5, 'a', ['python'], 20]
```

Использование списка

```
# Выводим элементы  
списка list = ['1', '2', '3', '4']  
for item in list:  
    print(item)
```

РЕЗУЛЬТАТ: 1 2 3 4

```
word = input('Введите слово: ')  
for letter in word:  
    print(letter)
```

РЕЗУЛЬТАТ: вывод слова по буквам



Второй тип цикла в Python.

Цикл **for** в языке программирования Python это не цикл со счетчиком. Как было видно из примера ранее, цикл **for** нужен для перебора элементов структур данных. В нашем случае он осуществлял перебор элементов в СПИСКЕ.

СИНТАКСИС FOR:

for переменная **in** объект

Еще пример:

```
for number in range(5):  
    print(number)
```



Python range() Function

```
range(start, stop[, step])
```

Функция: range

Возвращает список чисел с заданным диапазоном и шагом

range (начало, окончание, шаг)

Если начало и окончание явно не заданы, то (по умолчанию):

- начало = 0;
- шаг = 1;
- окончания нет? Тогда перебираем все!

Вывод чисел от 0 до 10:

```
for num in range(10):  
    print(num)
```

РЕЗУЛЬТАТ: 0 1 2 3 4 5 6 7 8 9 10

Вывод чисел от 1 до 10 с шагом 2:

```
for num in range(1, 10, 2):  
    print(num)
```

РЕЗУЛЬТАТ: 1 3 5 7 9



ВЕРНЕМСЯ К ЧЕРЕПАШКЕ

Мы уже знаем:

1. Как «вызывать» черепашку

```
import turtle as t
```

2. Как «командовать» черепашкой

```
t.forward(100) #вперед на 100 шагов (пикселей)  
t.left(90) #поворот налево на 90 градусов
```

3. Задавать цвет и ширину пера

```
t.pencolor('red') #рисуем красным цветом  
t.width(3) #ширина линий в пикселях
```

Добавим сюда скорость и фон «холста»:

```
t.bgcolor('black') #фон холста [черный  
t.speed(10) #скорость рисования
```



ДОБАВИМ КРАСОК

Поскольку мы теперь знаем:

1. **СПИСКИ**
2. цикл **FOR**
3. функцию **range()**

то сможем менять цвет и толщину линий в процессе рисования узора:

```
import turtle as t

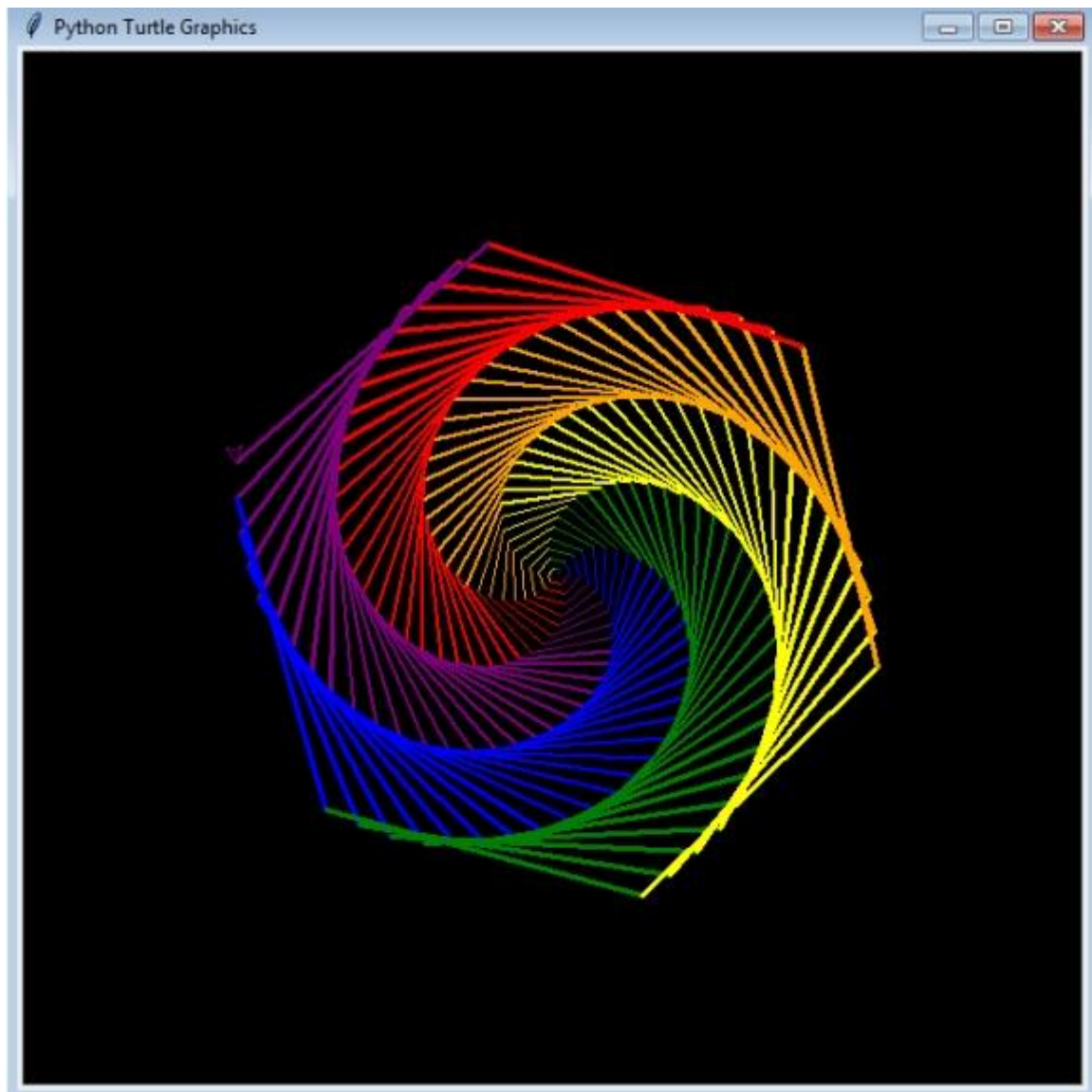
colors = ['red', 'purple', 'blue', 'green', 'yellow', 'orange']
t.bgcolor('black')
t.speed(25)
angle = 59 # Нестандартный поворот :)

for x in range(200):
    t.pencolor(colors[x%6])
    t.width(x/100+1)
    t.forward(x)
    t.left(angle)

t.mainloop()
```




ОБСУЖДАЕМ РЕЗУЛЬТАТ



Смотрим, что еще можно
поменять, чтобы получить
другие узоры