Цикл for

Python

Цикл for

- Копирует каждое значение итерируемого объекта в переменную
- Заканчивается, когда в итерируемом объекте больше нет элементов

```
for i in iter_object:
    ...
```

Генератор range

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

```
range(5)         # 0 1 2 3 4
range(5, 10)     # 5 6 7 8 9
range(2, 9, 2)   # 2 4 6 8
range(7, 2, -1)  # 7 6 5 4 3
```

Применение range

```
for i in range(2, 20, 2):
    print(i, end=' ')
```

```
for i in range(10, -1, -1):
    print(i, end=' ')
```

for с другими итераторами

Со строками:

```
for c in 'string':
    if c in 'crying':
        print(c, end='')

# ring
```

Со списками:

```
for e in [1, 3, 10, 15, 19]:
    if e % 3 == 0:
        print(e, end=' ')

# 3 15
```

И с другими…

break и continue

Оператор break завершает выполнение цикла и переводит выполнение на следующую после него инструкцию

```
for i in range(10):
    if i == 4:
        break
    print(i, end=' ')

# 0 1 2 3
```

Оператор continue завершает выполнение текущей итерации цикла и переходит к выполнению следующей

```
for i in range(10):
    if i % 2:
        continue
    ...
    print(i, end=' ')

# 1 3 5 7 9
```

Конструкция for-else

Конструкция else выполняется после завершения выполнения цикла

```
for i in range(5):
    print(i, end=' ')
else:
    print('end.')

# 0 1 2 3 4 end.
```

Но если цикл был завершен с помощью оператора break, то выполнение конструкции else будет пропущено

```
for i in range(5):
    if i == 3:
        break
    print(i, end=' ')
else:
    print('end.')

# 0 1 2
```