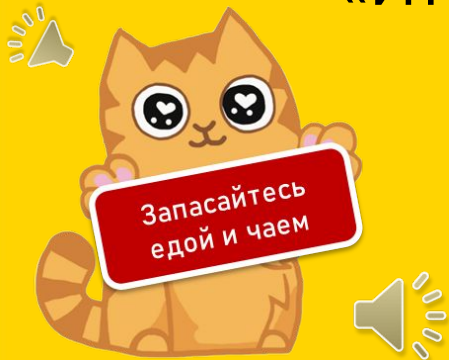


## Занятие №3

«Итеративные конструкции. Цикл for»

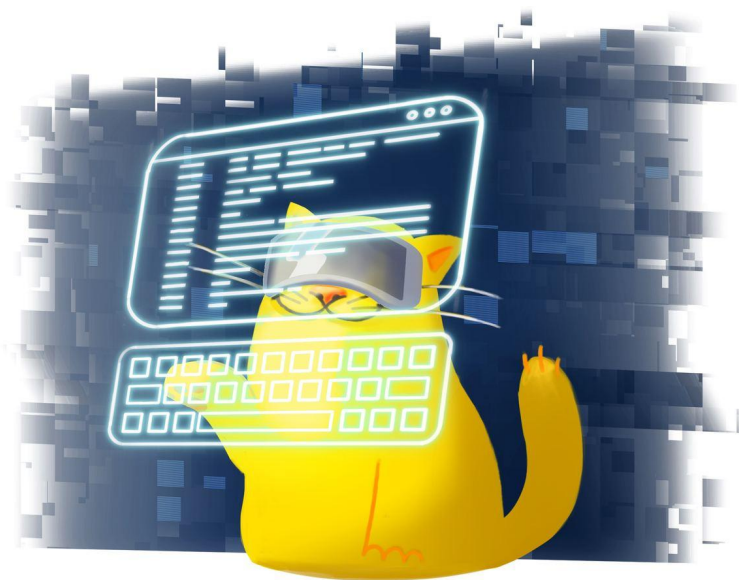




# О чем будем говорить сегодня

**Тема:** итеративные конструкции, цикл for

**Цель занятия:** повторить итеративные конструкции и изучить цикл for





## Результаты работы

- Повторим итеративные конструкции
- Познакомимся с циклом **for**
- Познакомимся с условием **if – else**
- Рассмотрим применение арифметических операций



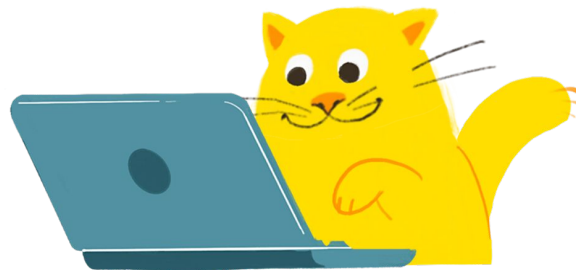
## Это нужно знать

**Цикл** – управляющая конструкция, при помощи которой можно выполнять часть кода несколько раз.

**Оператор** – команда, выполняющая свою внутреннюю инструкцию.

### Самые популярные циклы:

- while
- for
- циклы с 'if – else – elif'



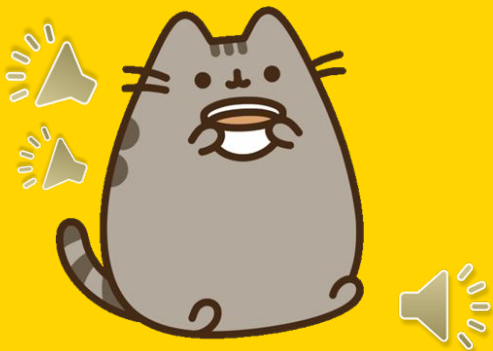


## Что помните с прошлого раза?

1. Какую роль выполняет команда <b>in</b> в языке <i>Python</i> ?
2. Что такое <b>оператор выхода из цикла</b> ?
3. Как можно передавать аргументы циклу?
4. Какую роль выполняет цикл <b>while</b> в языке <i>Python</i> ?
5. Как можно использовать оператор <b>not</b> в языке <i>Python</i> ?

# Цикл **for**

Учимся работать с новым циклом





## Немного теории: **цикл for**

Этот цикл производит перебор некого набора значений и помещает каждое значение в свою переменную. Посмотрим на синтаксис цикла **for**:

```
1  # синтаксис цикла for
2  # for переменная in значения:
3  # инструкции
```



## Попробуем на практике **цикл for**

```
1.py × third.py × 07.08.py ×
1
2 # итерации по числам от 0 до 5
3 for i in range(0, 6):
4     print(i)
5
6
7 |
07.08 ×
0
1
2
3
4
5
```

При выполнении цикла на языке Python происходит последовательное получение всех значений из набора и передача их переменной.





## Пример использования: цикл **for**

```
1 # задаем слово
2 m = 'Python'
3 # конструкция цикла for
4 for i in m:
5     # вывод символа
6     print(i)
7
```

07.08

P  
y  
t  
h  
o  
n

Данному циклу можно  
передавать не только  
цифры, но и **СИМВОЛЫ**



## Начинаем практику: **цикл for**

### Задание.

Посмотрите на код программы и предположите - что выведет данная программа?

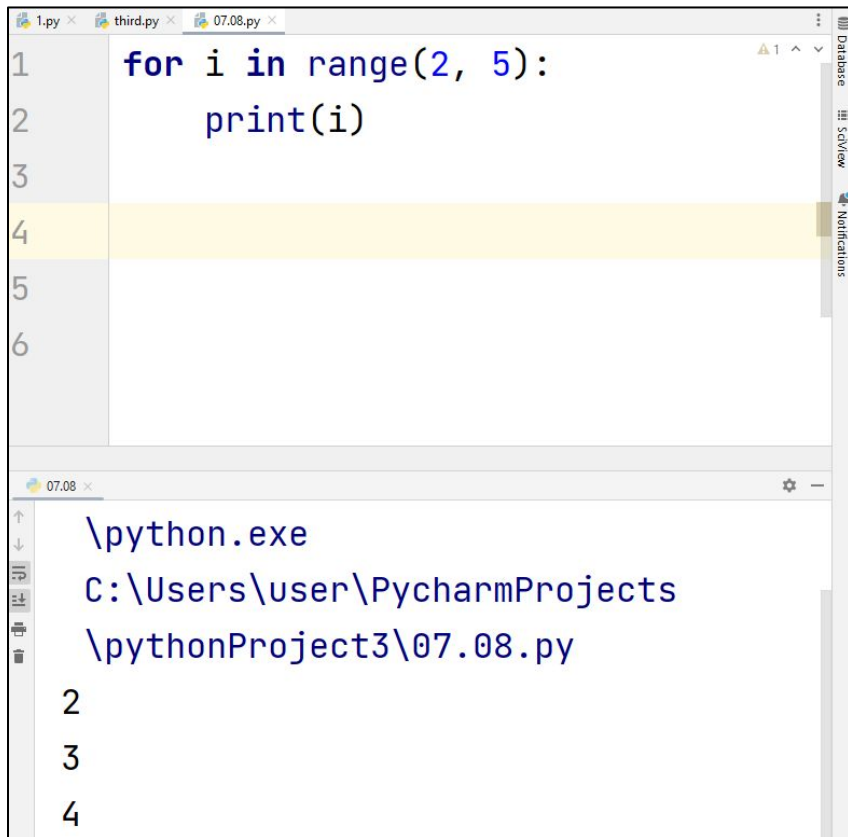
```
1  for i in range(2, 5):  
2      print(i)  
3
```



Время на выполнение - 5 минут



## Посмотрим на код



The screenshot shows the PyCharm IDE interface. The top editor window displays a Python script in a file named `07.08.py`. The code is as follows:

```
1 for i in range(2, 5):  
2     print(i)  
3  
4  
5  
6
```

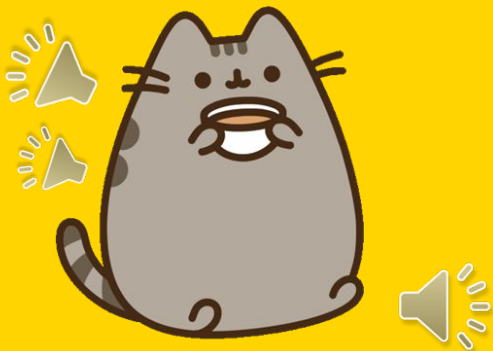
The line numbers 1 through 6 are visible on the left margin. The code on line 4 is highlighted in yellow. The bottom editor window shows the output of the script, indicating the command `python.exe` was run on the file `C:\Users\user\PycharmProjects\pythonProject3\07.08.py`, resulting in the following output:

```
2  
3  
4
```

Цикл **for** выводит числа  
от 2 до 4.

# Операторы **if – else**

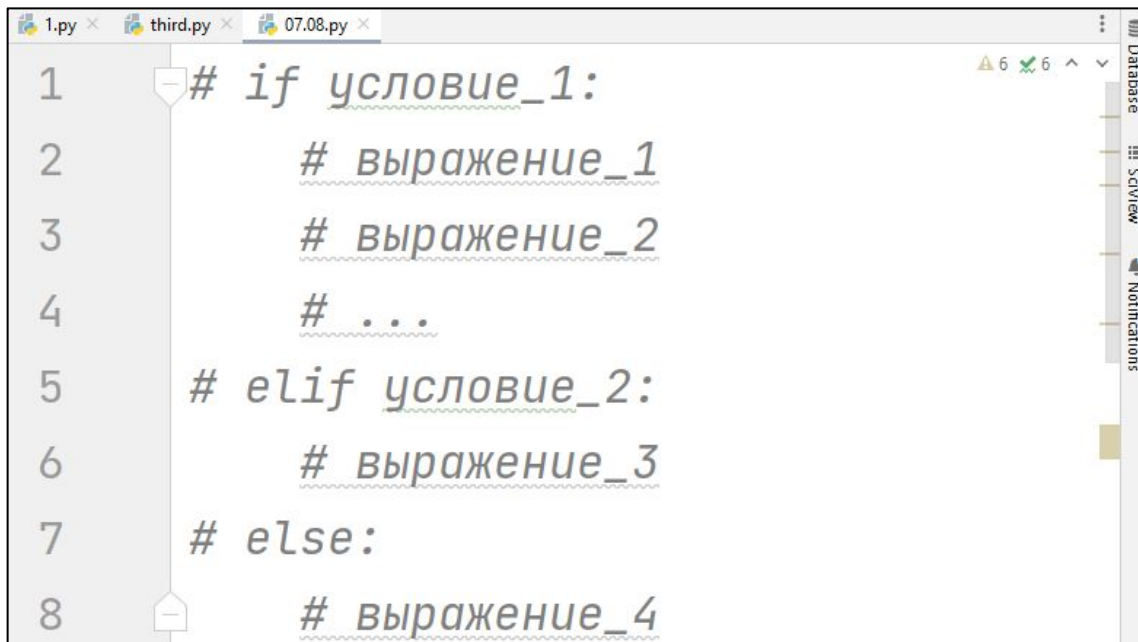
Учимся работать с условием





## Немного теории: операторы **if – else**

Оператор **'if'** записывается с условным выражением, а далее следуют одна или несколько частей **'elif'** и необязательная часть **'else'**.



```
1  # if условие_1:
2      # выражение_1
3      # выражение_2
4      # ...
5  # elif условие_2:
6      # выражение_3
7  # else:
8      # выражение_4
```



## Попробуем на практике операторы **if – else**

```
1  # Ввод данных
2  a = 25
3  b = 15
4  # реализация оператора 'if'
5  if a > b:
6      print('Переменная a:', a)
7
8
```

```
\pythonProject3\venv\Scripts
python.exe
C:\Users\user\PycharmProjects
\pythonProject3\07.08.py
Переменная a: 25
```

Если значение **'a'** больше, чем значение переменной **'b'** (что является истинной), то выводится значение переменной **'a'**.



## Пример использования: операторы **if – else**

```
1  # ВВОД ДАННЫХ
2  a = 8
3  b = 15
4  # реализация оператора 'if'
5  if a >= b:
6      print('Переменная a:', a)
7  else:
8      print('Переменная b:', b)
9
```

```
python.exe
C:\Users\user\PycharmProjects
\pythonProject3\07.08.py
Переменная b: 15
```

Здесь работа идет с переменными 'a' и 'b'



## Пример использования: другой вариант исполнения операторов **if – else**

Здесь добавляется  
логическая ветка **elif**

```
1  # ввод данных
2  a = 5
3  b = 8
4  # реализация оператора 'if'
5  if a > b:
6      print('Переменная a:', a)
7  elif a == b:
8      print('Переменные равны')
9  elif a**2 == 25:
10     print('Значение переменной:', a)
11 else:
12     print('Переменная b:', b)
```

```
07.08 <
Значение переменной: 5

Process finished with exit code 0
```





## Начинаем практику: операторы **if – else**

### Задание.

Напишите программу, в которой реализован цикл **for**, а также логические ветки **else** и **elif**. Количество переменных произвольно.



Время на выполнение - 5 минут





## Вариант решения задания

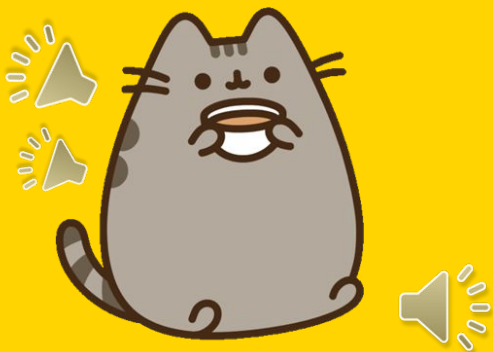
```
1  # ввод данных
2  c = 12
3  d = 9
4  # реализация оператора 'if'
5  if c > d:
6      print('Переменная c:', c)
7  elif c == d:
8      print('Переменные равны')
9  elif c**2 == 25:
10     print('Значение переменной:', c)
11 else:
12     print('Переменная d:', d)
```

↑  
↓  
[Icons]  
Process finished with exit code 0



# Арифметические операции

## Повторяем простые математические команды





## Немного теории: арифметические операции

Посмотрим на основные арифметические операции

Операция	Обозначение	Пример
Сложение	+	$3 + 4 = 7$
Вычитание	-	$7 - 2 = 5$
Умножение	*	$2 * 2 = 4$
Деление	/	$8 / 2 = 4$
Целочисленное деление	//	$9 // 2 = 4$
Остаток от деления	%	$9 \% 2 = 1$
Возведение в степень	**	$2 ** 3 = 8$



## Пример использования: арифметические операции

```
1 q = 'abc'
2 w = 'def'
3 print('Результат:', q + w)
4
5
```

07.08

```
C:\Users\user\PycharmProjects
\pythonProject3\venv\Scripts\python.exe
C:\Users\user\PycharmProjects
\pythonProject3\07.08.py
Результат: abcdef
```

В этой программе  
используется оператор **+**,  
он способен выводить  
строки слитно



## Немного теории: арифметические операции

Посмотрим на синтаксис оператора `*`, он способен выводить символы или строки несколько раз

```
1  
2 # строка * кол-во повторений  
3 # print(строка и кол-во повторений)  
4  
5
```

The screenshot shows a code editor with three tabs: '1.py', 'third.py', and '07.08.py'. The '07.08.py' tab is active. The code is written in a monospaced font. A red rectangular box highlights lines 2 and 3. Line 2 is a comment: `# строка * кол-во повторений`. Line 3 is a comment: `# print(строка и кол-во повторений)`. The right sidebar of the IDE shows 'Database', 'ScView', and 'Notifications' panels.



## Пример использования: арифметические операции

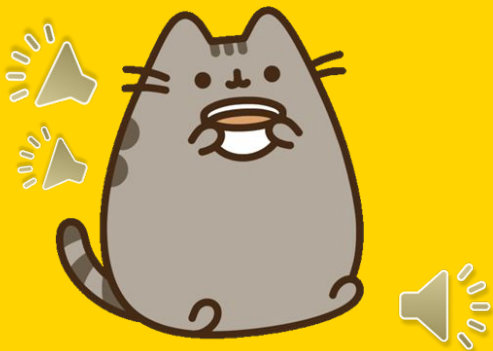
Здесь оператор \*  
выводит значение  
переменной 'z' 4  
раза

```
1  z = 'abc'
2  print('Результат: ', z * 4)
3
4
5
```

```
C:\Users\user\PycharmProjects
\pythonProject3\venv\Scripts\python.exe
C:\Users\user\PycharmProjects
\pythonProject3\07.08.py

Результат:  abcabcabcab
```

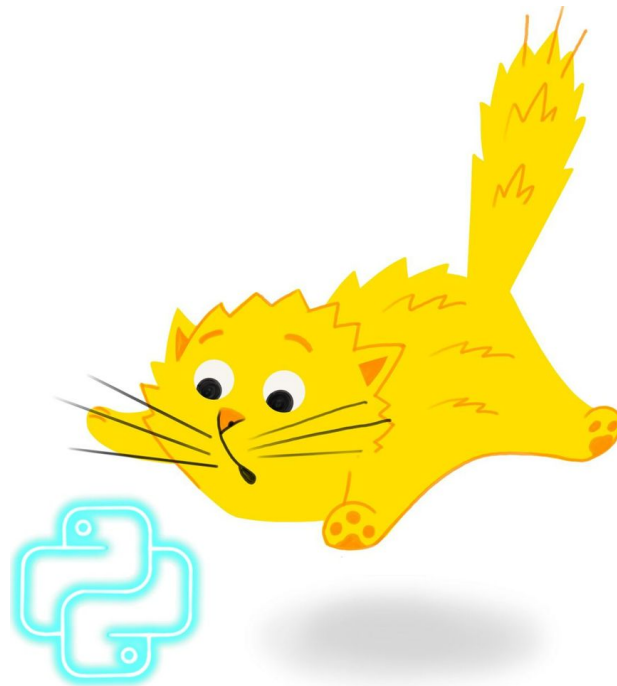
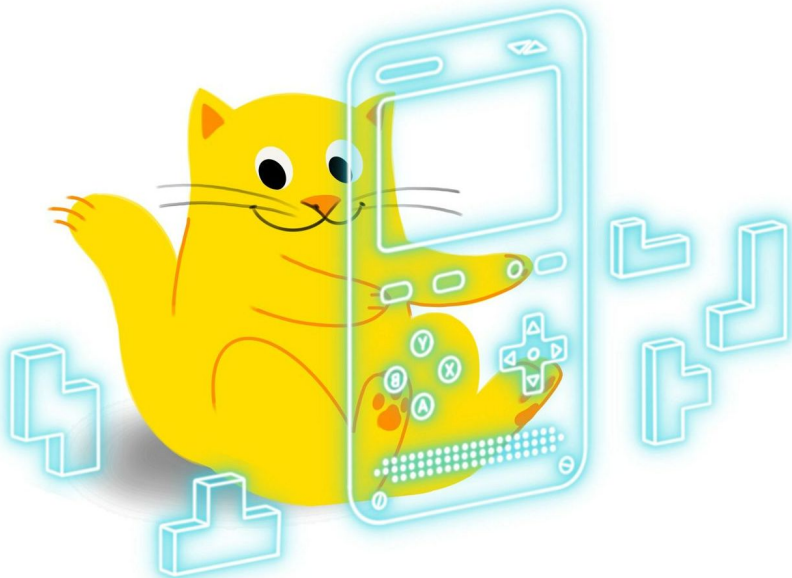
Перерыв **10** минут







# ПРАКТИЧЕСКИЕ ЗАДАЧИ





## Практика. Задача №1

Создать программу, в которой вводятся списки **'m' = [1, 3, 5, 7, 9]** и **'x' = [11, 23, 35]**. Необходимо рассчитать количество элементов в данных списках и вывести данные значения на консоль.





## Решение задачи №1

Напишем код для решения и посмотрим на вывод консоли:

```
1  m = [1, 3, 5, 7, 9]
2  x = [11, 23, 35]
3  print('Список "m":', m)
4  print('Список "x":', x)
5  print('Количество элементов в "m":', len(m))
6  print('Количество элементов в "x":', len(x))
7
8
9
10
```

```
07.08
Список "m": [1, 3, 5, 7, 9]
Список "x": [11, 23, 35]
Количество элементов в "m": 5
Количество элементов в "x": 3
```



## Практика. Задача №2

Создать программу, в которой происходит перебор элементов списка с именем **'w'** при помощи цикла **while**.





## Решение задачи №2

Напишем код для решения и посмотрим на вывод консоли:

```
1  W = ['one', 'two', 'three']
2  i = 0
3  while i < len(w):
4      print(w[i])
5      i += 1
```

C:\Users\user\PycharmProjects  
pythonProject3\venv\Scripts  
python.exe  
C:\Users\user\PycharmProjects  
pythonProject3\07.08.py

one  
two  
three



## Практика. Задача №3

Создать программу, в которой вводится список  $x = [1, 2, 3, 4, 5, 6]$ .  
Необходимо найти минимальный элемент в данном списке и вывести его на консоль.





## Решение задачи №3

Напишем код для решения и посмотрим на вывод консоли:

```
1 x = [1, 2, 3, 4, 5, 6]
2 print('Список "x":', x)
3 print('Минимальный элемент:', min(x))
4
5
```

07.08

```
C:\Users\user\PycharmProjects\pythonProject3\venv
\Scripts\python.exe
C:\Users\user\PycharmProjects\pythonProject3\07
.08.py
Список "x": [1, 2, 3, 4, 5, 6]
Минимальный элемент: 1
```



## Практика. Задача №4

Создать программу, в которой используется строка **'z' = 'abc'**.  
Необходимо применить оператор **"\*"** и вывести данную строку 4 раза подряд в строчку.







## Решение задачи №4

Напишем код для решения и посмотрим на вывод консоли:

The screenshot shows the PyCharm IDE interface. The top pane displays a Python script in a file named 07.08.py. The script consists of two lines of code: `z = 'abc'` and `print('Результат:', z * 4)`. The bottom pane shows the console output for the same file, displaying the file path and the execution result: `Результат: abcabcabcab`.

```
1  z = 'abc'
2  print('Результат:', z * 4)
3
4  |

C:\Users\user\PycharmProjects
pythonProject3\venv\Scripts
python.exe
C:\Users\user\PycharmProjects
pythonProject3\07.08.py
Результат: abcabcabcab
```



## Практика. Задача №5

Создать программу, в которой реализован оператор выхода из цикла **break**. Изначально вводится переменная 's' со значением 0. При каждой итерации цикла значение переменной 's' увеличивается на единицу. Когда значение переменной 's' становится равным 5, оператор **break** завершает работу цикла (на консоли должны быть выведены значения с 1 по 4 включительно).





## Решение задачи №5

Напишем код для решения и посмотрим на вывод

The screenshot shows an IDE with two windows. The top window, titled '07.08.py', contains the following Python code:

```
1  # ввод данных
2  s = 0
3  # реализация оператора 'break'
4  while s < 9:
5      s += 1
6      if s == 5:
7          break
8      print(f's = {s}')
9
```

The bottom window, titled '07.08', shows the output of the program:

```
\pythonProject3\07.08.py
s = 1
s = 2
s = 3
s = 4
```



## Практика. Задача №6

Создать программу, в которой используются переменные **'a'** и **'b'**. Реализовать в данной программе оператор **'if'** со следующим условием: если значение переменной **'a'** больше, чем значение переменной **'b'**, то выводится значение переменной **'b'**; если значения переменных равны, то выводится сообщение **'Значения переменных равны'**; в противном случае, выводится значение переменной **'b'**. Реализовать все три итерации (в каждой итерации выполняется свое условие).

Результаты продемонстрировать на консоли.





## Решение задачи №6

Напишем код для  
решения и  
посмотрим на  
вывод консоли:

```
1  # Ввод данных
2  a = 15
3  b = 8
4  # реализация оператора 'if'
5  if a > b:
6      print('Переменная a:', a)
7  elif a == b:
8      print('Переменные равны')
9  else:
10     print('Переменная b:', b)
```

```
pythonProjects\venv\scripts
python.exe
C:\Users\user\PycharmProjects
pythonProject3\07.08.py
Переменная a: 15
```



## Решение задачи №6

Напишем код для  
решения и  
посмотрим на  
вывод консоли:

```
1  # ввод данных
2  a = 15
3  b = 15
4  # реализация оператора 'if'
5  if a > b:
6      print('Переменная a:', a)
7  elif a == b:
8      print('Переменные равны')
9  else:
10     print('Переменная b:', b)
```

07.08 x

C:\Users\user\PycharmProjects  
\pythonProject3\07.08.py

Переменные равны



## Практика. Задача №7

Создать программу, в которой происходит перебор элементов списка с именем **'words' = ['one', 'two', 'three', 'four']** при помощи цикла **while**.





## Решение задачи №7

Напишем код для  
решения и  
посмотрим на  
вывод консоли:

```
1 words = ['one', 'two', 'three', 'four']
2 i = 0
3 while i < len(words):
4     print(words[i])
5     i += 1
6
```

```
one
two
three
four
```





## Практика. Задача №8

Создать программу с использованием цикла **while** и переменной 'y', значение которой, изначально, равно 2. Данный цикл выводит значения переменной 'y' в третьей степени до тех пор, пока значение переменной 'y' не станет равным 10. При каждой итерации значение переменной 'y' увеличивается на 2.





## Решение задачи №8

Напишем код для  
решения и  
посмотрим на  
вывод консоли:

```
1  # реализация цикла 'while'
2  y = 2
3  while y < 10:
4      print('Значение:', y**3)
5      y += 2
6
```

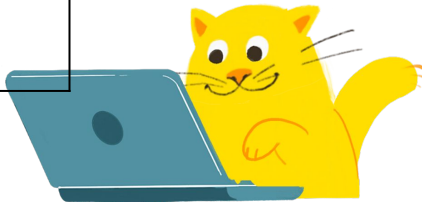
07.08 x

Значение: 8  
Значение: 64  
Значение: 216  
Значение: 512



## Закрепление знаний

1. Какую роль выполняет цикл <b>while</b> в языке <i>Python</i> ?
2. Что такое <b>оператор прерывания цикла</b> ?
3. Как можно передавать аргументы циклу?
4. Какую роль выполняет функция <b>max(x)</b> в языке <i>Python</i> ?
5. Как можно использовать оператор <b>and</b> в языке <i>Python</i> ?



**Спасибо за  
внимание!**  
**До новых встреч!**

