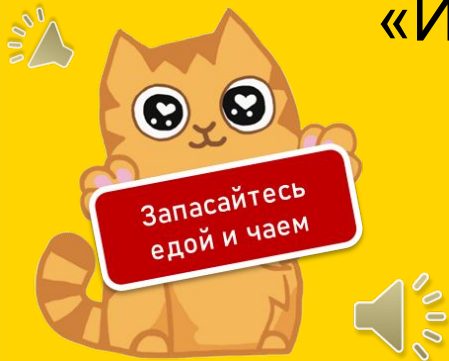


Занятие №3

«Итеративные конструкции. Циклы»

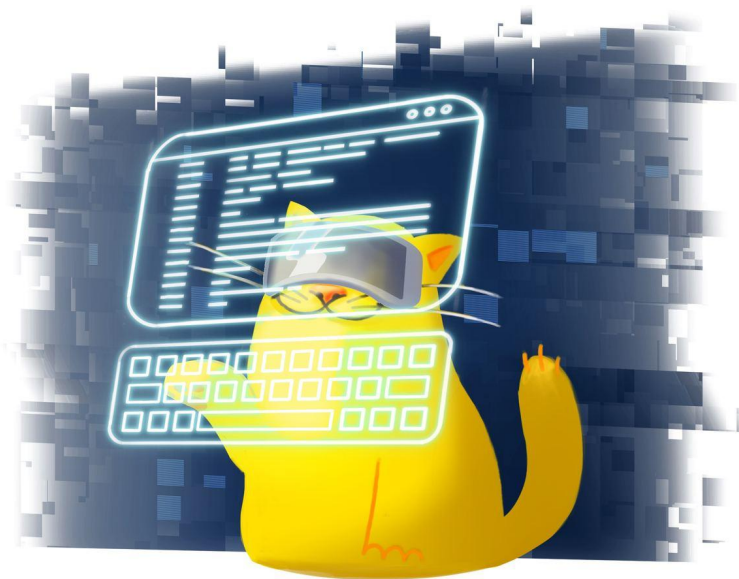




О чем будем говорить сегодня

Тема: итеративные конструкции и циклы

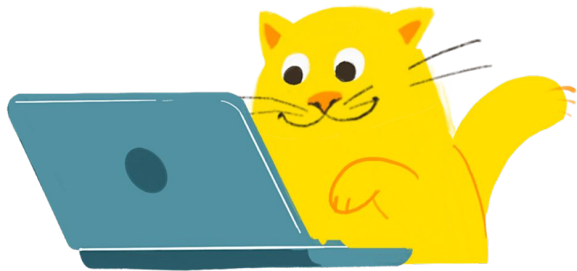
Цель занятия: изучить как обращаться с новыми конструкциями
и циклами





Результаты работы

- Научимся работать с циклами и выходить из них
- Познакомимся с вложенными циклами (циклы внутри цикла)
- Посмотрим на новые программы с циклом **'while'**
- Научимся работать с циклом **'for'**

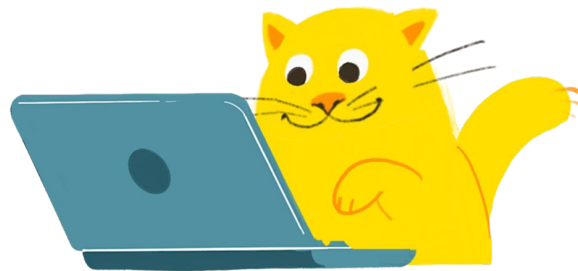




Это нужно знать

Цикл – часть кода, которая выполняется несколько раз.

Оператор – команда, которая может выполняться в цикле.



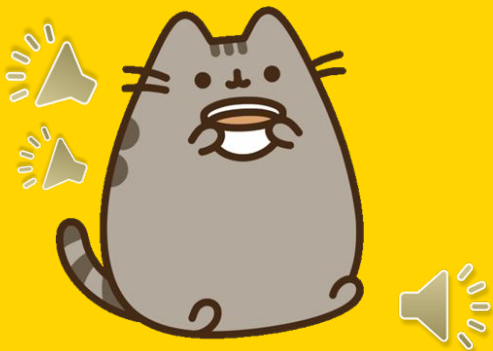


Что помните с прошлого раза?

1. Какую роль выполняет команда float в языке <i>Python</i> ?
2. Что такое логическая операция ?
3. Как можно передавать аргументы программе?
4. Какую роль выполняет цикл while в языке <i>Python</i> ?
5. Какое минимальное количество аргументов можно передать программе на языке <i>Python</i> за один раз?

Выход из цикла

Учимся вовремя выходить из
цикла





Немного теории: **выход из цикла**

Для управления циклом мы можем использовать специальные операторы **break** и **continue**. Оператор **break** осуществляет выход из цикла. Оператор **continue** выполняет переход к следующей итерации цикла.

```
1.py × third.py × 07.08.py ×
1  # цикл 'while'
2  num = 0
3  while num < 6:
4      num += 1
5      if num == 3:
6          break          # если значение равно 3, выходим из цикла
7      print(f"Значение num = {num}")
8
9
```



Попробуем на практике оператор **'continue'**

```
1  # цикл 'while'
2  num = 0
3  while num < 6:
4      num += 1
5      if num == 3:
6          # если значение равно 3, не выводим только данное значение
7          continue
8      print(f"Значение num = {num}")
9
```

07.08

Значение num = 1
Значение num = 2
Значение num = 4
Значение num = 5
Значение num = 6

Оператор **continue** выполняет переход к следующей итерации цикла без его завершения. Например, в предыдущем примере заменим **break** на **continue**:



Начинаем практику: **выход из цикла**

Задание.

Напишите программу, в которой используется переменная с именем 'a'.
Значение данной переменной равно числу **3**. При помощи цикла '**while**'
и оператора **continue** выводятся значения данной переменной
от **3** до **5** и от **7** до **10** включительно.



Время на выполнение - 5 минут





Посмотрим на решение

```
1  # решение задания
2  a = 2
3  while a < 10:
4      a += 1
5      if a == 6:
6          continue
7      print('Значение переменной "a":', a)
8  print('Работа цикла завершена!')
```

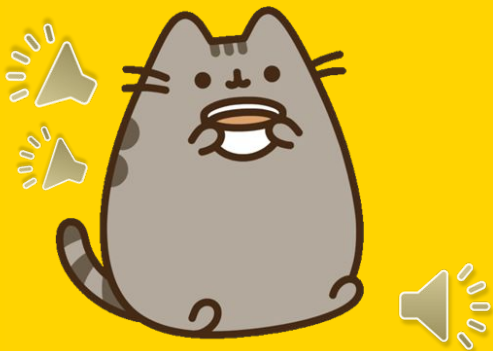
07.08

Значение переменной "a": 3
Значение переменной "a": 4
Значение переменной "a": 5
Значение переменной "a": 7
Значение переменной "a": 8
Значение переменной "a": 9
Значение переменной "a": 10
Работа цикла завершена!



Вложенные циклы

Учимся писать циклы внутри
циклов





Немного теории: **вложенные циклы**

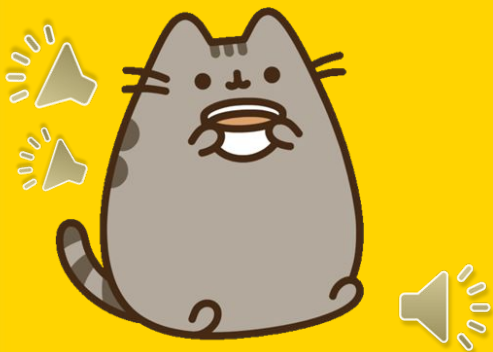
```
1  c = 1
2  d = 1
3  while c < 5:
4      while d < 5:
5          print(c * d, end = '\t')
6          d += 1
7      print('\n')
8      d = 1
9      c += 1
```

```
07.08 x
1  2  3  4
2  4  6  8
3  6  9  12
4  8  12 16
```

Циклы внутри себя
могут содержать другие
циклы. Пример вывода
простой **таблицы**
умножения:

Цикл **while**

Смотрим на новые программы с
циклом **while**





Пример использования: цикл **while**

Рассмотрим пример использования цикла **while** и ключевого слова **else**:

```
1  # цикл while и ключевое слово else
2  z = 1
3
4  while z < 4:
5      print('условие выполнено')
6      z = z + 1
7
8  else:
9      print('условие не выполнено')
```

```
условие выполнено
условие выполнено
условие выполнено
условие не выполнено
```

Здесь выводится сообщение
'условие выполнено', либо
'условие не выполнено'



Начинаем практику: **цикл while**

Задание.

Написать программу, в которой есть переменная 'а', которая равна 2.

Если значение переменной меньше 7, то выводится удвоенное значение этой переменной. В противном случае, выводится утроенное значение этой переменной.

При каждой итерации значение 'а' увеличивается на 2.



Время на выполнение - 5 минут





Посмотрим на код для решения задания

Программа написана верно?

```
1  # цикл while и ключевое слово else
2  a = 2
3
4  while a < 700:
5      print('Условие выполнено:', a*a)
6      a = a + 3
7  else:
8      print('Условие не выполнено:', a*a*a)
9
```

Если **нет**, то **что нужно исправить**,
чтобы программа выполнила свою задачу?



В программе было **2 ошибки**:

```
1.py x third.py x 07.08.py x
1  # цикл while и ключевое слово else
2  a = 2
3
4  while a < 700:
5      print('Условие выполнено:', a*a)
6      a = a + 3
7  else:
8      print('Условие не выполнено:', a*a*a)
9
```

1-ая ошибка

2-ая ошибка

1. В строке 4 число **700** нужно заменить на **7**;
2. В строке 6 число **3** нужно заменить на **2**.



Исправим ошибки и посмотрим на вывод

```
1  # цикл while и ключевое слово else
2  a = 2
3
4  while a < 7:
5      print('Условие выполнено:', a*a)
6      a = a + 2
7  else:
8      print('Условие не выполнено:', a*a*a)
9
```

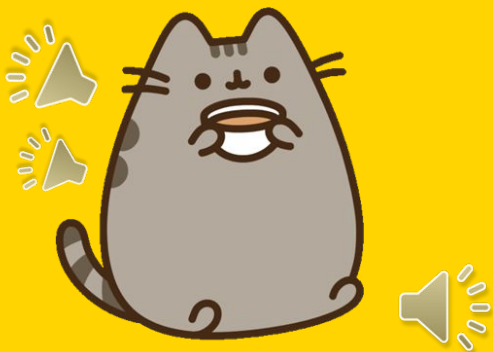
07.08

```
Условие выполнено: 4
Условие выполнено: 16
Условие выполнено: 36
Условие не выполнено: 512
```

Прекрасно!
Теперь программа выполняет
свои функции!

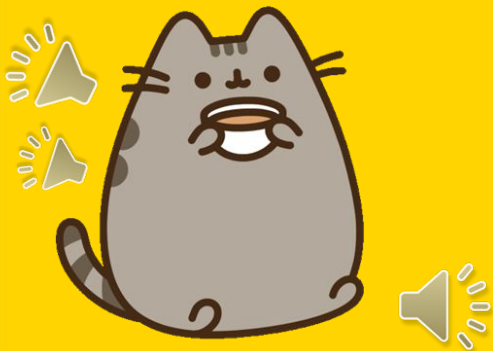


Перерыв **10** минут



Цикл **for**

Учимся работать с новым
циклом





Немного теории: цикл **for**

Цикл **for** способен перебирать элементы последовательности

```
1  # реализация цикла for
2  for i in 1, 2, 3, 4, 'three':
3      print(i)
```

```
1
2
3
4
three
```

В последовательностях
могут быть числа,
символы, строки и их
комбинации



Попробуем на практике цикл **for**

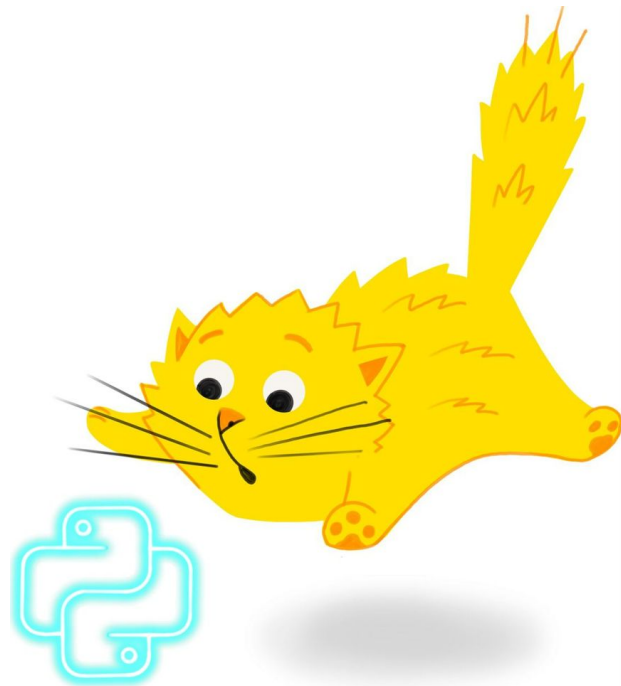
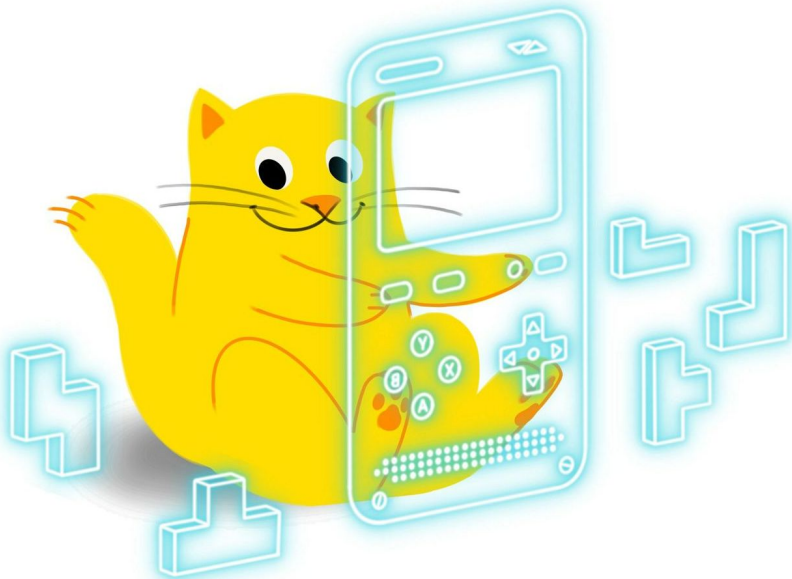
```
1  # реализация цикла for
2  for a in range(1, 4):
3      print('a =', a)
4  for b in range(3, 5):
5      print('b =', b)
6
7
8
```

```
a = 1
a = 2
a = 3
b = 3
b = 4
```

В этом примере цикл **for** выводит значения переменных **'a'** и **'b'** в указанных диапазонах



ПРАКТИЧЕСКИЕ ЗАДАЧИ





Практика. Задача №1

Создать программу, в которой реализован оператор выхода из цикла **break**. Изначально вводится переменная '**z**' со значением **2**. При каждой итерации цикла значение переменной '**z**' увеличивается на 2. Когда значение переменной '**z**' становится равным **10**, оператор **break** завершает работу цикла (на консоли должны быть выведены значения от **4** до **8** включительно).





Решение задачи №1

Напишем код для решения и посмотрим на вывод:

```
1  # ввод данных
2  z = 2
3  # реализация оператора 'break'
4  while z < 15:
5      z += 2
6      if z == 10:
7          break
8      print(f'z = {z}')
9
```

```
07.08
z = 4
z = 6
z = 8
```



Практика. Задача №2

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





Решение задачи №2

Напишем код для
решения и посмотрим на
вывод:

```
1  # решение задачи
2  words = ['one', 'two', 'three', 'four']
3  i = 0
4  while i < len(words):
5      print(words[i])
6      i += 1
7
8
9
```

```
07.08
↑
↓
one
two
three
four
```



Практика. Задача №3

Создать программу, в которой реализован оператор выхода из цикла **break**. Изначально вводится переменная **'num'** со значением **0**. При каждой итерации цикла значение переменной **'num'** увеличивается на единицу. Когда значение переменной **'num'** становится равным **3**, оператор **break** завершает работу цикла (на консоли должны быть выведены значения **1** и **2**).





Решение задачи №3

Напишем код для решения и посмотрим на вывод:

```
1.py third.py 07.08.py
1  # решение задачи
2  num = 0
3  while num < 5:
4      num += 1
5      if num == 3:
6          break
7      print(f'num = {num}')
8
9

07.08
C:\Users\user\PycharmProjects\pythonProject3\07
.08.py
num = 1
num = 2
```



Практика. Задача №4

Создать программу с использованием цикла **while** и оператора **continue**. Цикл **while** работает с переменной 'z', значение которой изначально равно 3. Данный цикл выводит значение переменной 'z' во второй степени. При каждой итерации значение переменной 'z' увеличивается на единицу. Когда значение переменной становится равным 6, оператор **continue** приостанавливает работу цикла **while**.





Решение задачи №4

Напишем код для решения и посмотрим на вывод:

```
1  # решение задачи
2  z = 3
3  while z < 8:
4      print('Результат: ', z**2)
5      z += 1
6      if z == 6:
7          continue
8
9
```

```
Результат:  9
Результат:  16
Результат:  25
Результат:  36
Результат:  49
```



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:

```
1.py x third.py x 07.08.py x
1  # решение задачи
2  s = 0
3  while s < 9:
4      s += 1
5      if s == 5:
6          break
7      print(f's = {s}')
8
9

07.08 x
.08.py
s = 1
s = 2
s = 3
s = 4
```



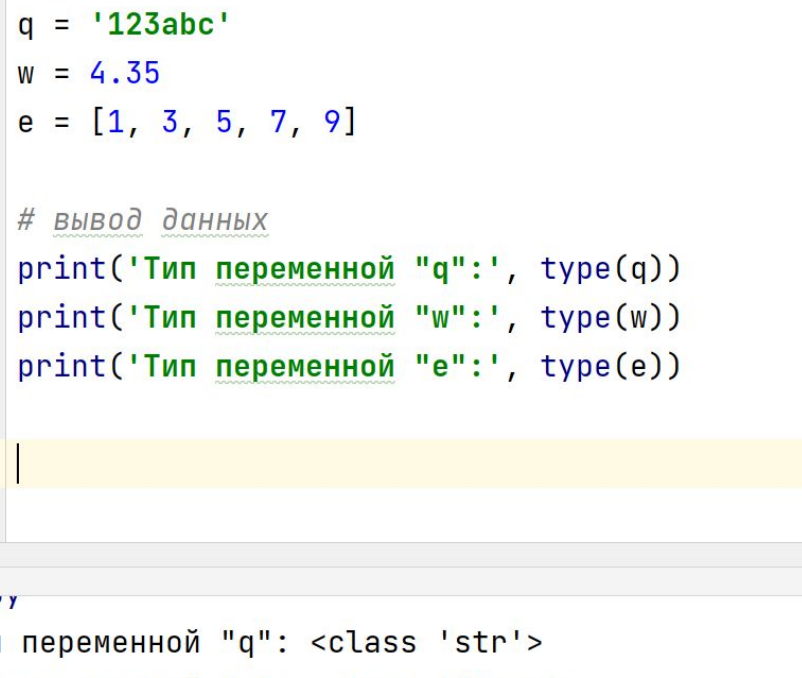
Практика. Задача №6

Создать программу, в которой используются: переменная **'q'** с типом **str**, переменная **'w'** с типом **float** и переменная **'e'** с типом **list**. Необходимо при помощи функции **type()** вывести на консоль типы данных переменных.





Напишем код для решения и посмотрим на вывод:



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

```
1 # ввод переменных
2 q = '123abc'
3 w = 4.35
4 e = [1, 3, 5, 7, 9]
5
6 # вывод данных
7 print('Тип переменной "q":', type(q))
8 print('Тип переменной "w":', type(w))
9 print('Тип переменной "e":', type(e))
10
11
```

The bottom window, also titled '07.08.py', shows the output of the code:

```
Тип переменной "q": <class 'str'>
Тип переменной "w": <class 'float'>
Тип переменной "e": <class 'list'>
```



Практика. Задача №7

Создать программу с использованием цикла **for**. Данный цикл перебирает значения переменной 'x' от 1 до 5 и переменной 'y' от 2 до 5 и выводит соответствующие значения на консоль.





Решение задачи №7

Напишем код для
решения и посмотрим на
вывод:

```
1  # решение задачи
2  for x in range(1, 6):
3      print('x =', x)
4
5  for y in range(2, 6):
6      print('y = ', y)
7
```

```
x = 1
x = 2
x = 3
x = 4
x = 5
y = 2
y = 3
y = 4
y = 5
```



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:

```
1  # решение задачи
2  y = 2
3  while y < 10:
4      print('Значение "y":', y**3)
5      y += 2
6
7
```

C:\Users\user\PycharmProjects\pythonProject3\07
.08.py

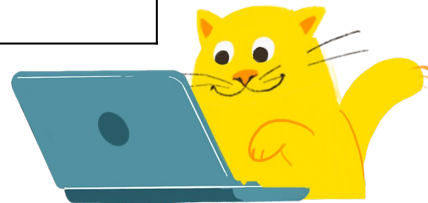
Значение "y": 8
Значение "y": 64
Значение "y": 216
Значение "y": 512

Process finished with exit code 0



Закрепление знаний

- | |
|---|
| 1. Какую роль выполняет цикл for в языке <i>Python</i> ? |
| 2. Что такое оператор выхода из цикла ? |
| 3. Как можно передавать аргументы циклам? |
| 4. Какую роль выполняет цикл while в языке <i>Python</i> ? |
| 5. Как можно использовать оператор and в языке <i>Python</i> ? |



Спасибо за внимание!

**Следующее занятие во
вторник
в 16:00 (по Москве)**

