

ОПЕРАТОРЫ BREAK И CONTINUE

 Модуль 1. Занятие 9.



Тема **и цель**

Тема:

Знакомство с операторами на языке Python.

Цель занятия:

Изучение основ по работе с операторами на языке Python



Глоссарий

- 1. Цикл** — разновидность управляющей конструкции в высокоуровневых языках программирования, предназначенная для организации многократного исполнения набора инструкций.
- 2. Программа** — данные, которые используются процессором как инструкции по управлению компьютерной системой.

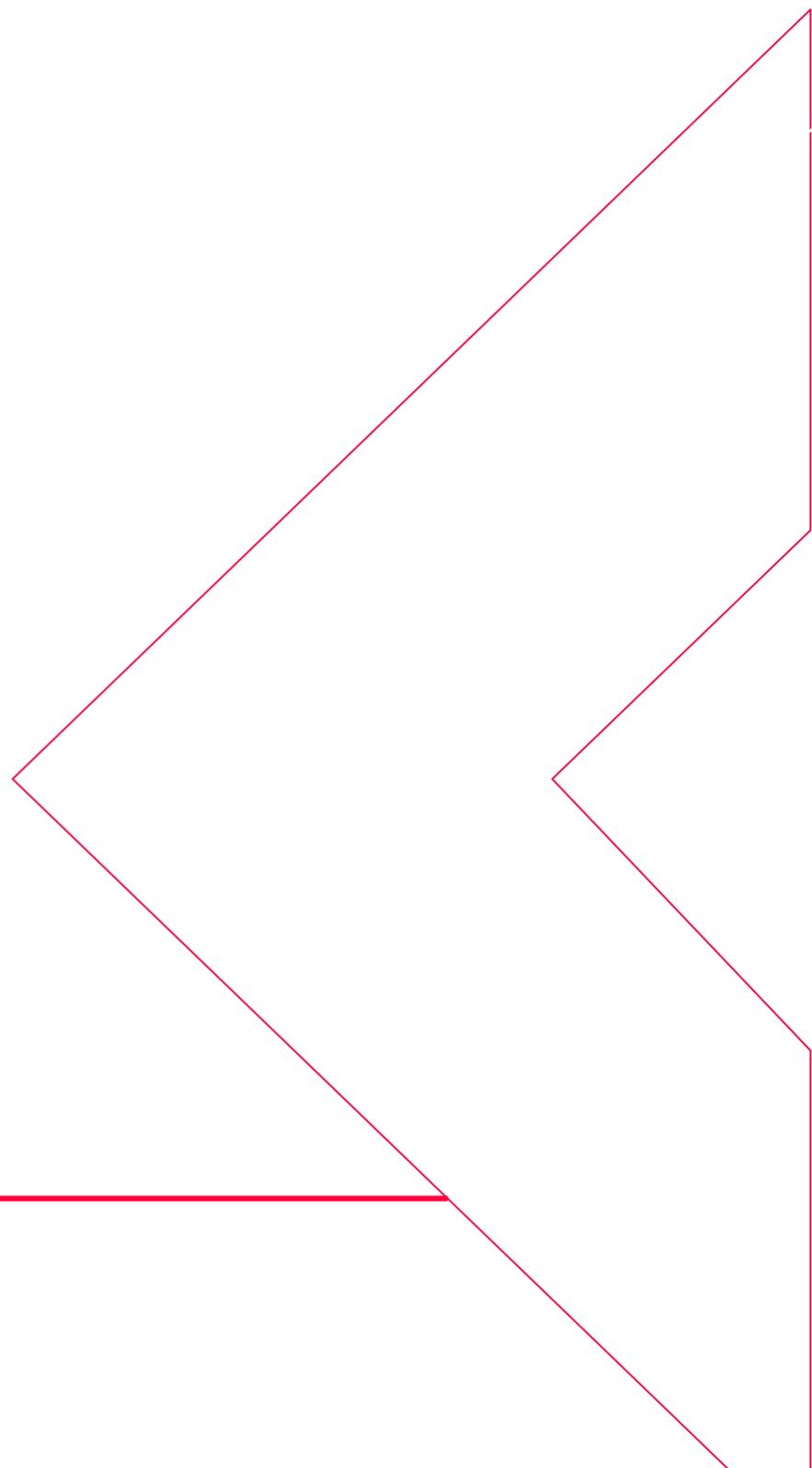


Вопросы

1. Какую роль выполняет цикл **for** в языке Python?
2. Что такое **цикл**?
3. Как можно передавать аргументы циклам?
4. Какую роль выполняет функция **len(x)** в языке Python?
5. Какое **максимальное количество аргументов** можно передать циклу на языке Python за один раз?



Выход из цикла. Break и continue



Программа с использованием оператора `break`



```
1 number = 0
2 while number < 5:
3     number += 1
4     if number == 3 :      # если number = 3, выходим из цикла
5         break
6     print(f"number = {number}")
```

Рисунок 1

Консольный вывод при работе с оператором `break`



```
number = 1  
number = 2
```

Рисунок 2

Программа с использованием оператора `continue`



```
1 number = 0
2 while number < 5:
3     number += 1
4     if number == 3 :    # если number = 3, переходим к новой итерации цикла
5         continue
6     print(f"number = {number}")
```

Рисунок 3

Консольный вывод с использованием оператора `continue`



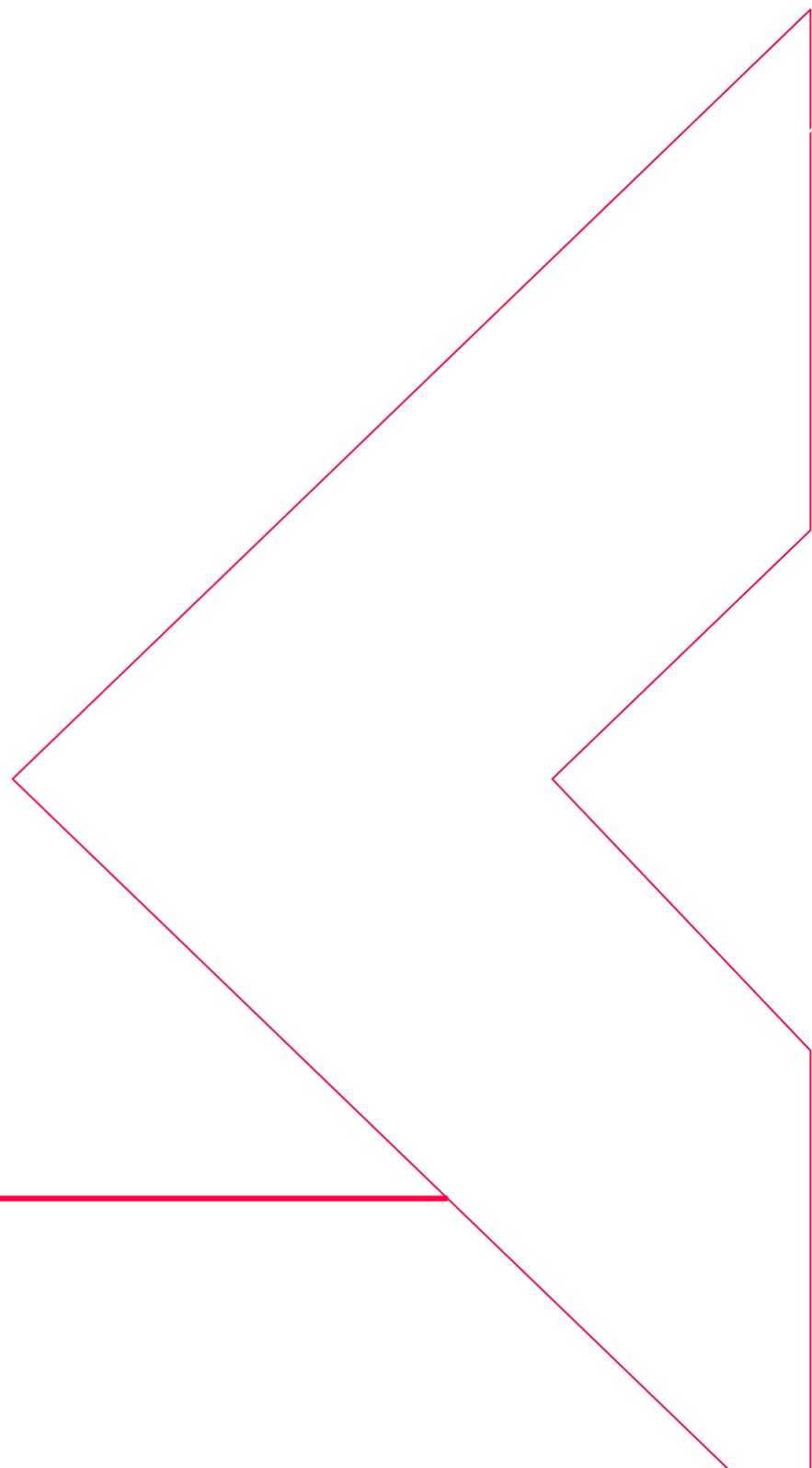
```
number = 1  
number = 2
```

Рисунок 4



Массивы.

Создание списка





Создание списка

The screenshot shows an IDE window for a project named 'pythonProject3'. The file 'main.py' is open. The code in the editor is as follows:

```
1 # создание списка
2 n = [1, 3, 5, 7, 9]
3
```

The words 'создание' and 'списка' in the comment are underlined with a green wavy line. The numbers 1, 3, 5, 7, and 9 in the list are highlighted in blue. The IDE interface includes a menu bar with options like File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. A project explorer on the left shows the file structure.

Рисунок 1

Создание списка с текстовой информацией



The screenshot shows an IDE window titled 'pythonProject3 - main.py'. The menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The breadcrumb path is 'pythonProject3 > main.py'. The Project Explorer on the left shows a folder 'Project' containing a file 'main.py'. The code editor displays the following Python code:

```
1 # создание списка
2 z = ['First', 'Second', 'Third']
3
```

Рисунок 2

Создание списка при помощи функции-конструктора list()



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
pythonProject3 > main.py
Project
main.py x
1 # создание списка
2 x_1 = []
3 x_2 = list()
4
```

Рисунок 3

Создание списка с разными ТИПАМИ ДАННЫХ



The screenshot shows an IDE window titled 'pythonProject3 - main.py'. The menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The breadcrumb path is 'pythonProject3 > main.py'. The editor shows a file named 'main.py' with the following code:

```
1 # создание списка
2 x = [1, 'Tom', 2.6, 'Python']
3
```

Рисунок 4

Программа с использованием СПИСКОВ



```
pythonProject3 - main.py
pythonProject3 > main.py
Project
  > main.py x
  > |
1  # создание списков
2  x = [1, 3, 5, 7]
3  y = ['First', 'Second', 'Third']
4
5  # вывод списков
6  print('Список x: ', x)
7  print('Список y: ', y)
Run: main x
C:\Users\user\PycharmProjects\pythonProject3\venv\
Список x: [1, 3, 5, 7]
Список y: ['First', 'Second', 'Third']
Process finished with exit code 0
```

Рисунок 5

Программа с введенными элементами списков



```
pythonProject3 - main.py
pythonProject3 main.py
1 # создание списков
2 x = list(input('Введите элементы списка x: '))
3 y = list(input('Введите элементы списка y: '))
4
5 # вывод списков
6 print('Список x: ', x)
7 print('Список y: ', y)

Run: main x
C:\Users\user\PycharmProjects\pythonProject3\venv\
Введите элементы списка x: 1357
Введите элементы списка y: 2468
Список x: ['1', '3', '5', '7']
Список y: ['2', '4', '6', '8']
```

Рисунок 6

Программа с использованием конструктора list()



```
pythonProject3 - main.py
pythonProject3 > main.py
Project > main.py x
1 # создание списка
2 z_1 = [1, 3, 5, 7, 9]
3 z_2 = list(z_1)
4
5 # вывод списка
6 print('Список z_1:', z_2)

Run: main x
C:\Users\user\PycharmProjects\pythonProject3\
Список z_1: [1, 3, 5, 7, 9]
Process finished with exit code 0
```

Рисунок 7

Программа с созданием списков с одинаковыми элементами



```
pythonProject3 - main.py
pythonProject3 > main.py
Project: pythonProject3
main.py x
1 # создание списков
2 n = [3] * 5
3 z = ['Zack'] * 3
4
5 # вывод списков
6 print('Список n:', n)
7 print('Список z:', z)

Run: main x
C:\Users\user\PycharmProjects\pythonProject3\
Список n: [3, 3, 3, 3, 3]
Список z: ['Zack', 'Zack', 'Zack']

Process finished with exit code 0
```

Рисунок 8



Обращение к элементам списка



Программа с обращением к элементам списка



```
pythonProject3 - main.py
pythonProject3 > main.py
Project: pythonProject3
main.py x
1 # создание списка
2 x = ['First', 'Second', 'Third']
3
4 # вывод списка и обращения к элементам
5 print('Список x: ', x)
6 print('Первый элемент: ', x[0])
7 print('Последний элемент: ', x[2])

Run: main x
C:\Users\user\PycharmProjects\pythonProject3\venv\Scripts\python.exe
Список x: ['First', 'Second', 'Third']
Первый элемент: First
Последний элемент: Third
```

Рисунок 9

Программа с изменением элемента списка



```
pythonProject3 - main.py
pythonProject3 main.py
Project
main.py
1 # создание списка и вывод до изменения
2 x = ['First', 'Second', 'Third']
3 print('Список x до изменения: ', x)
4 # изменение элемента списка и вывод списка после изменения
5 x[1] = 2
6 print('Список x после изменения: ', x)

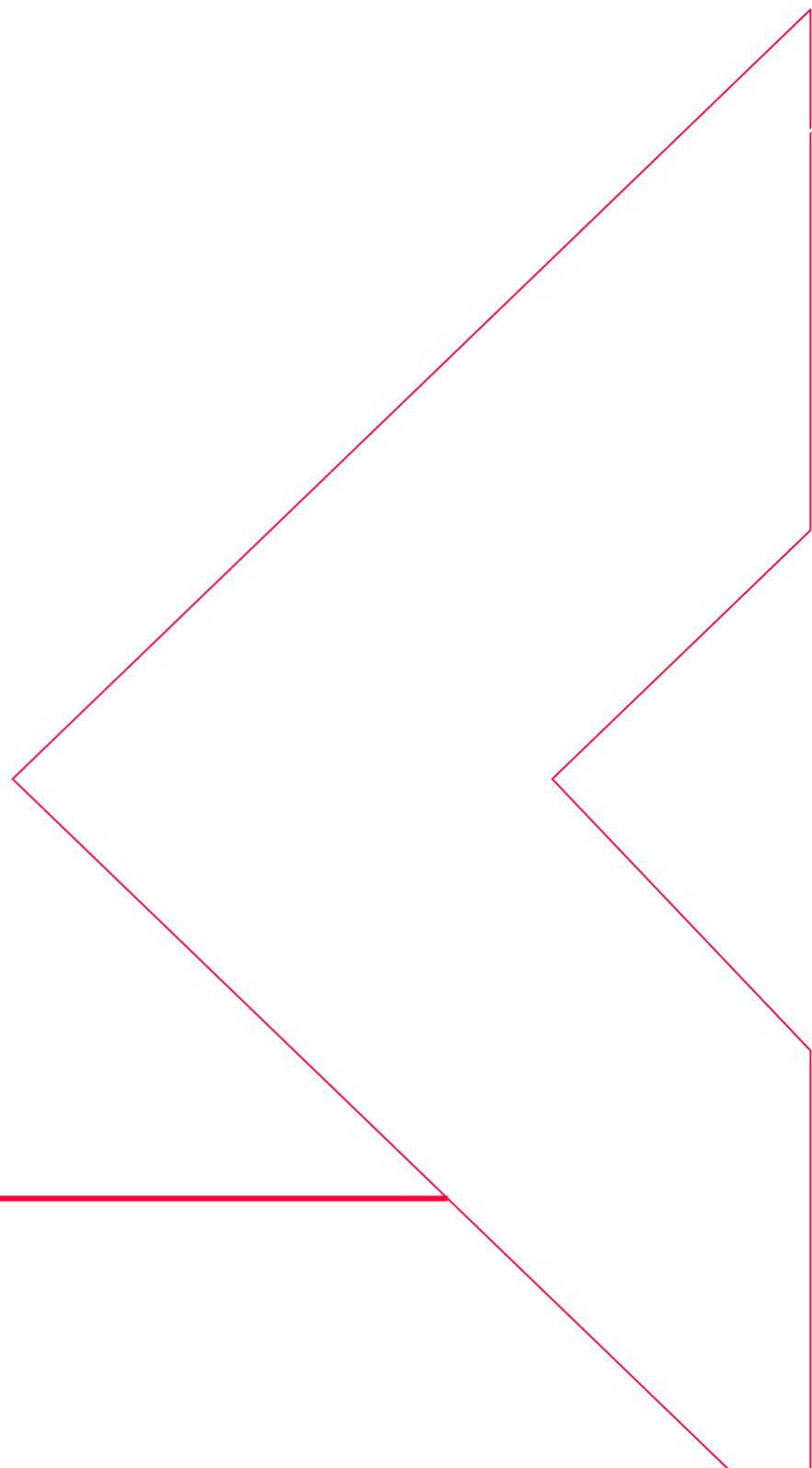
Run: main
C:\Users\user\PycharmProjects\pythonProject3\venv\Scripts\python.exe
Список x до изменения: ['First', 'Second', 'Third']
Список x после изменения: ['First', 2, 'Third']

Process finished with exit code 0
```

Рисунок 10



Разложение СПИСКА



Программа с разложением списка



```
pythonProject3 - main.py
pythonProject3 main.py
1 # создание списка
2 elements = ['Андрей', 'Макар', 'Олег']
3
4 # разложение списка
5 first, second, third = elements
6 # отдельный вывод элементов
7 print(first)
8 print(second)
9 print(third)
```

Run: main

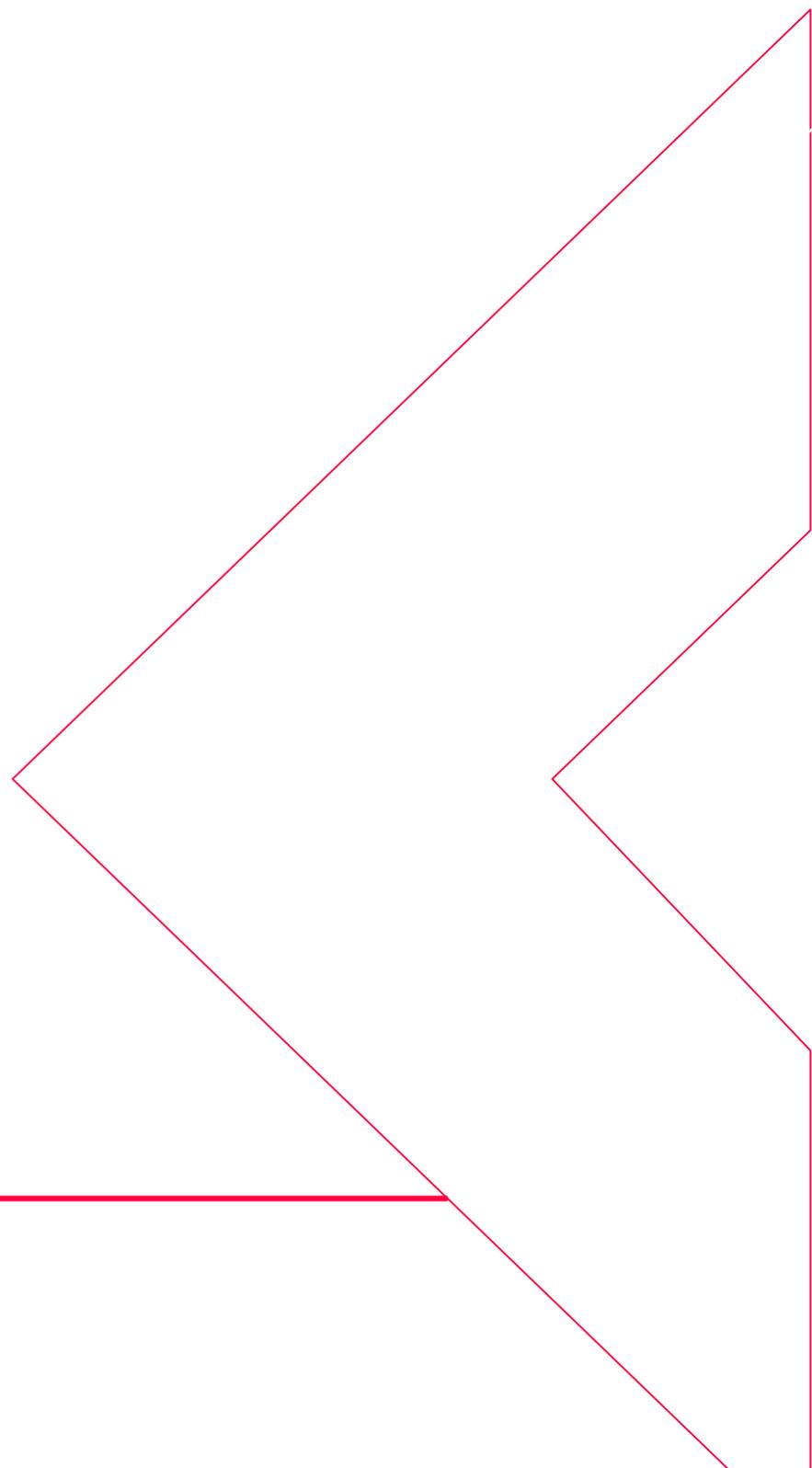
```
C:\Users\user\PycharmProjects\pythonProject3\venv\Scri
Андрей
Макар
Олег

Process finished with exit code 0
```

Рисунок 11



Перебор ЭЛЕМЕНТОВ



Программа с перебором при помощи цикла for



```
pythonProject3 - main.py
pythonProject3 > main.py
1 # создание списка
2 elements = ['Андрей', 'Макар', 'Олег']
3
4 # реализация цикла 'for'
5 for i in elements:
6     print(i)

Run: main
C:\Users\user\PycharmProjects\pythonProject3\venv\Scripts\python.exe
Андрей
Макар
Олег

Process finished with exit code 0
```

Рисунок 12

Программа с перебором при помощи цикла `while`

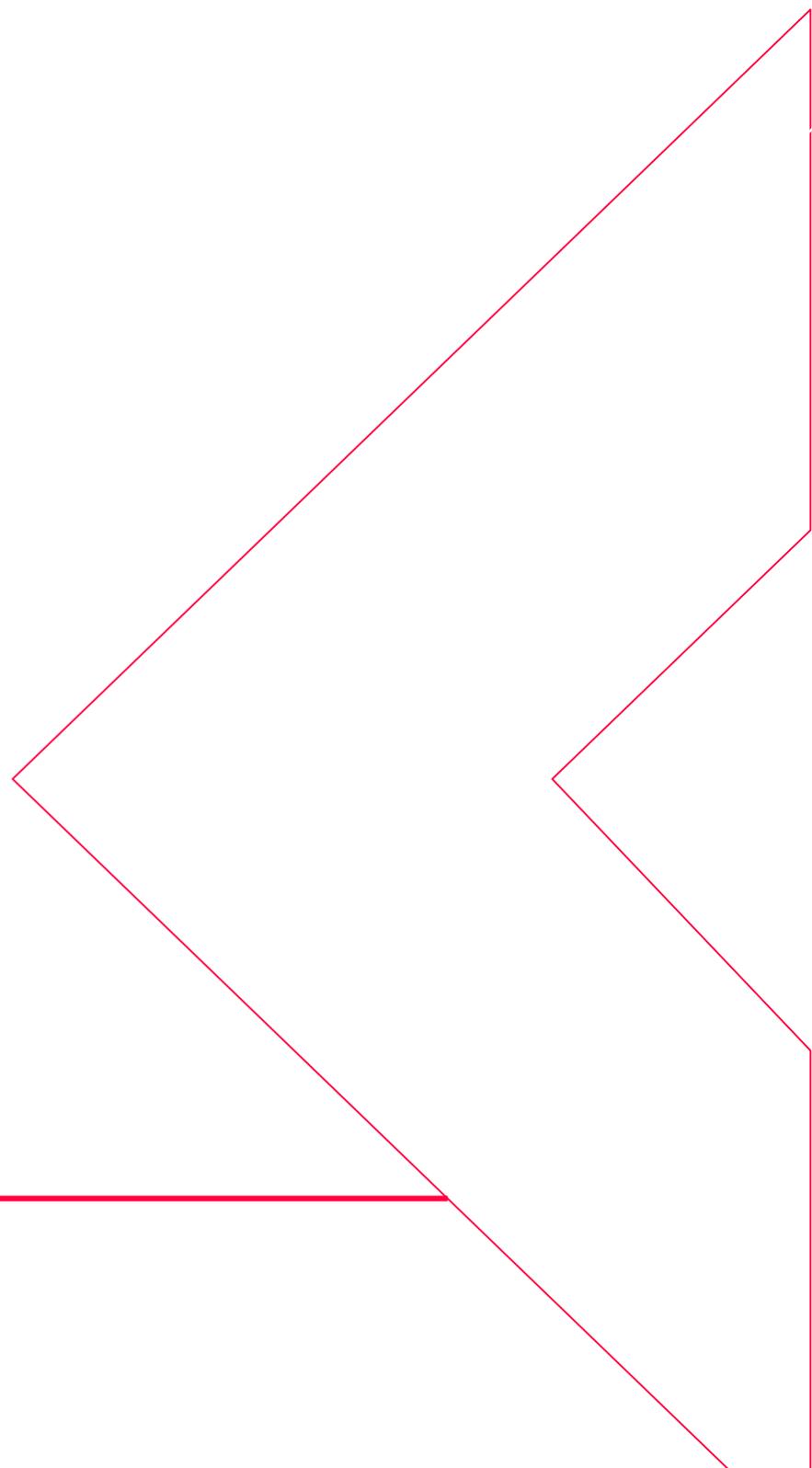


```
pythonProject3 - main.py
pythonProject3 > main.py
Project
main.py
1 # создание списка
2 elements = ['Андрей', 'Макар', 'Олег']
3 # реализация цикла 'while'
4 i = 0
5 while i < len(elements):
6     print(elements[i])
7     i += 1
while i < len(elements)
Run: main
C:\Users\user\PycharmProjects\pythonProject3\venv\Scripts
Андрей
Макар
Олег
Process finished with exit code 0
```

Рисунок 13



Сравнение СПИСКОВ



Программа

с одинаковыми списками



```
pythonProject3 - main.py
pythonProject3 main.py
1 # создание списков
2 z_1 = [1, 3, 5, 7, 9]
3 z_2 = list([1, 3, 5, 7, 9])
4 # реализация логической конструкции if - else
5 if z_1 == z_2:
6     print('Список z_1 соответствует списку z_2')
7 else:
8     print('Список z_1 не соответствует списку z_2')

Run: main x
C:\Users\user\PycharmProjects\pythonProject3\venv\Scripts\python.exe
Список z_1 соответствует списку z_2
Process finished with exit code 0
```

Рисунок 14

Программа с разными списками



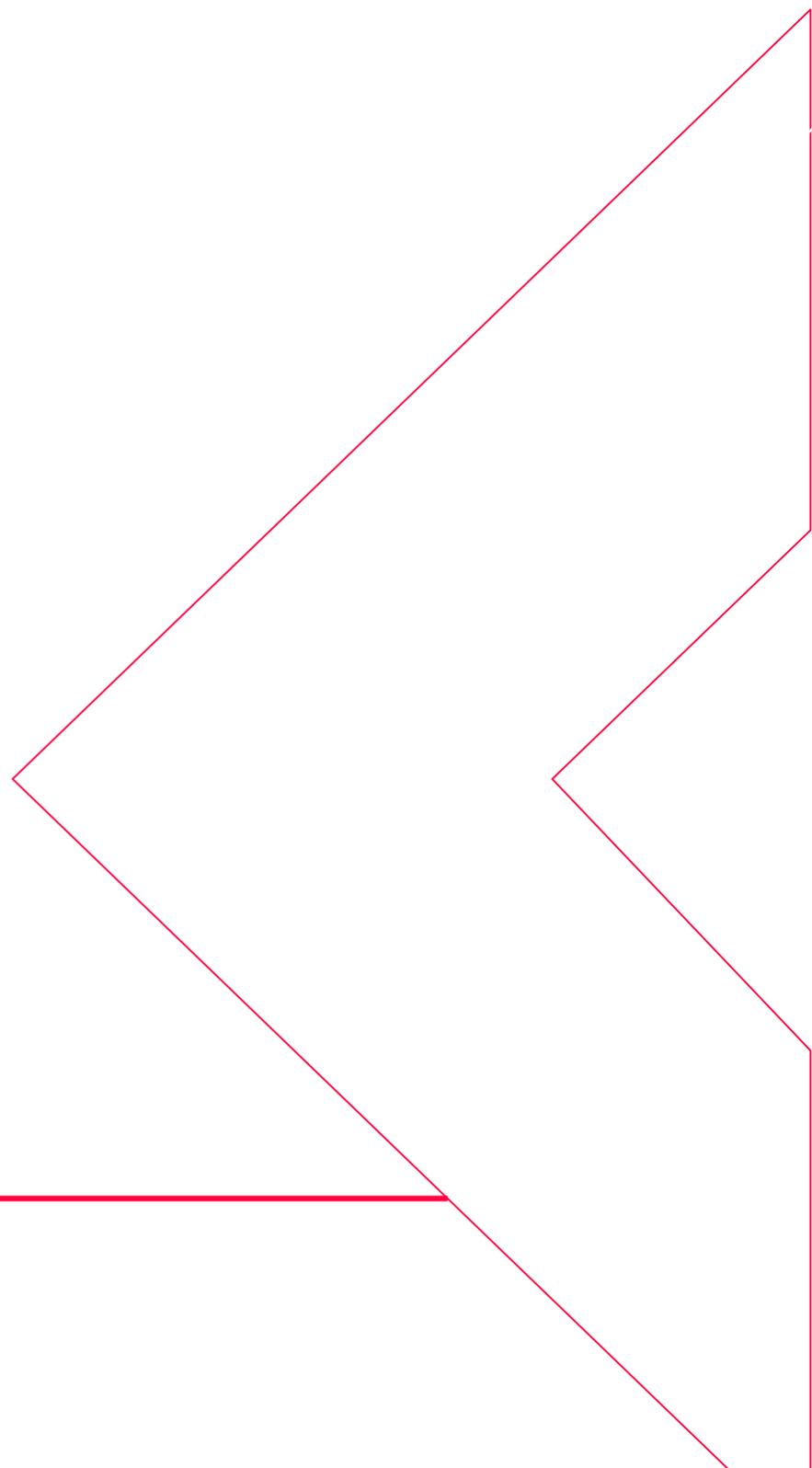
```
pythonProject3 - main.py
pythonProject3 main.py
1 # создание списков
2 z_1 = [1, 3, 5, 7, 9]
3 z_2 = list([1, 3, 5, 7, 9])
4 # реализация логической конструкции if - else
5 if z_1 == z_2:
6     print('Список z_1 соответствует списку z_2')
7 else:
8     print('Список z_1 не соответствует списку z_2')

Run: main x
C:\Users\user\PycharmProjects\pythonProject3\venv\Scripts\python.exe
Список z_1 соответствует списку z_2
Process finished with exit code 0
```

Рисунок 15



Ключевое слово *pass*





Программа с использованием ключевого слова `pass`

```
pythonProject3 - main.py
pythonProject3 main.py
1 # реализация цикла 'for'
2 z = 0
3 for z in range(6):
4     if z == 4:
5         pass
6     print('Значение z: ', z)
7
8 print('Конец работы цикла for')
```

Run: main

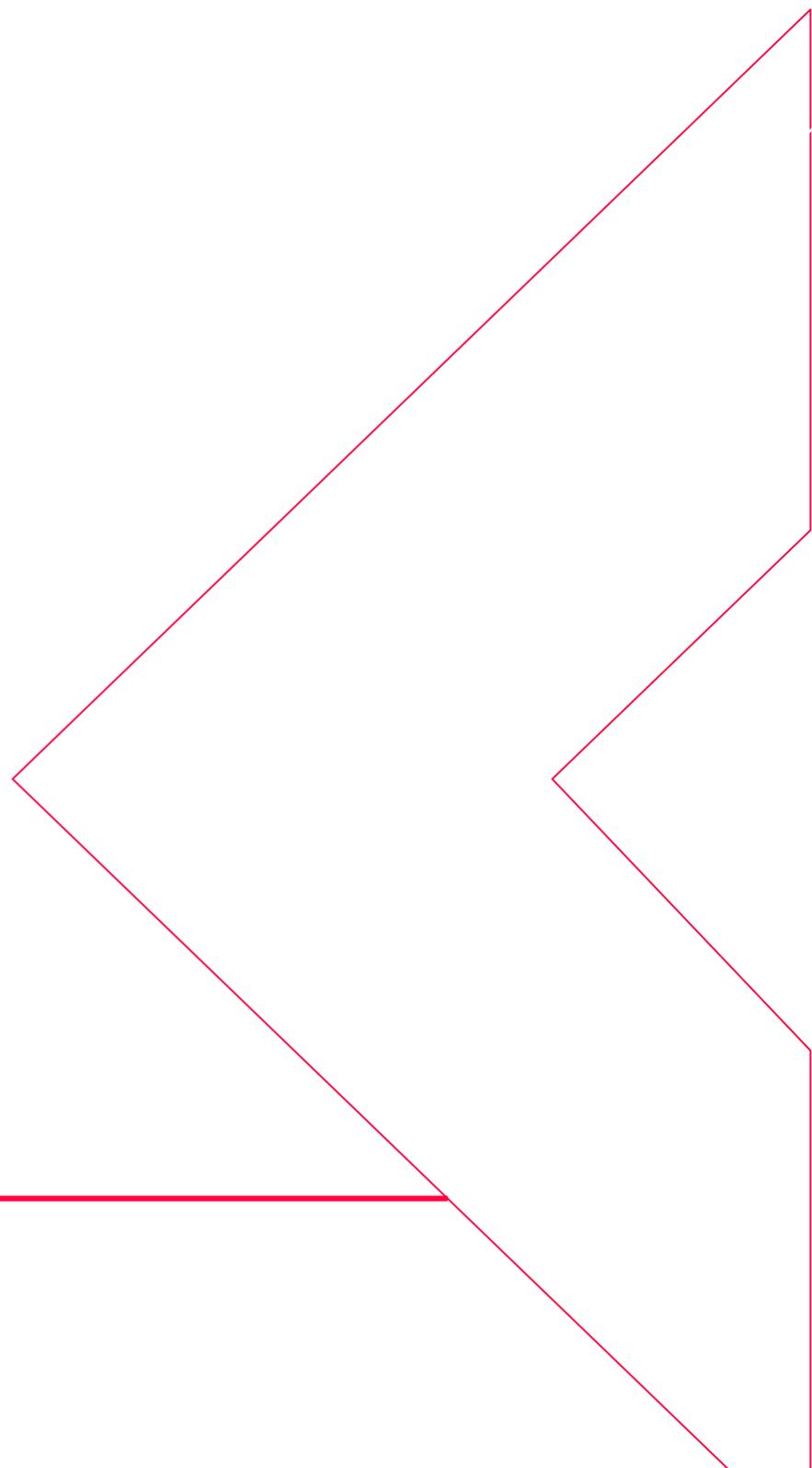
```
Значение z: 0
Значение z: 1
Значение z: 2
Значение z: 3
Значение z: 4
Значение z: 5
Конец работы цикла for

Process finished with exit code 0
```

Рисунок 16



Практические задачи





Задача №1

Создать программу с использованием цикла **while** и оператора **break**.

Цикл **while** работает с переменной **'a'**, значение которой изначально равно 3. Данный цикл выводит значение переменной **'a'** во второй степени. При каждой итерации значение переменной **'a'** увеличивается на единицу. Когда значение переменной становится равным 6, оператор **break** завершает работу цикла **while**

Решение



```
pythonProject3 > main.py
1
2 # реализация цикла 'while' и оператора 'break'
3 a = 3
4 while a < 8:
5     print('Результат: ', a*2)
6     a += 1
7     if a == 6:
8         break

while a < 8 > if a == 6
Run: main x
C:\Users\user\PycharmProjects\pythonProject3\venv
Результат: 9
Результат: 16
Результат: 25
Process finished with exit code 0
```

Напишем код для решения данной практической задачи и посмотрим на вывод:



Задача №2

Создать программу с использованием цикла **for**.

Данный цикл перебирает значения переменной **'i'** от 1 до 5 и выводит на консоль.



Решение

```
pythonProject3 - main.py
pythonProject3 > main.py
1 # реализация цикла 'for'
2 for i in range(1, 6):
3     print('i = ', i)
4
Run: main
C:\Users\user\PycharmProjects\pythonProject3\
i = 1
i = 2
i = 3
i = 4
i = 5
```

Напишем код для решения данной практической задачи и посмотрим на вывод:



Задача №3

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

```
pythonProject3 - main.py
pythonProject3 > main.py
1 # реализация циклов 'while'
2 y = 2
3 while y < 10:
4     print('Значение: ', y**3)
5     y += 2
6
while y < 10
Run: main x
C:\Users\user\PycharmProjects\pythonProject3\venv\
Значение: 8
Значение: 64
Значение: 216
Значение: 512

Process finished with exit code 0
```

Напишем код для решения данной практической задачи и посмотрим на вывод:



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

```
pythonProject3 - main.py
pythonProject3 main.py
Project
  main.py
1
2 # реализация цикла 'while' и оператора 'continue'
3 z = 3
4 while z < 8:
5     print('Результат: ', z**2)
6     z += 1
7     if z == 6:
8         continue

while z < 8 > if z == 6
Run: main
Результат: 9
Результат: 16
Результат: 25
Результат: 36
Результат: 49
Process finished with exit code 0
Version Control Python Packages TODO Python Console Problems Terminal Services
PEP 8: W292 no newline at end of file
```

Напишем код для решения данной практической задачи и посмотрим на вывод:



Вопросы

1. Какую роль выполняет цикл **while** в языке Python?
2. Что такое **программа**?
3. Как можно передавать аргументы циклам?
4. Какую роль выполняет функция **max(x)** в языке Python?
5. Какое **максимальное количество аргументов** можно передать циклу на языке Python за один раз?