Python-3
Строки. Циклы

Основные методы Работы со строками в Python.

Метод lower(): переводит строку в нижний регистр.
Метод upper(): переводит строку в верхний регистр.
Метод title(): начальные символы всех слов в строке переводятся в верхний регистр.

Метод lstrip(): удаляет начальные пробелы из строки.
Метод rstrip(): удаляет конечные пробелы из строки.
Метод strip(): удаляет начальные и конечные пробелы из строки.

find(str[, start [, end]): возвращает индекс подстроки в строке. Если подстрока не найдена, возвращается число -1.

replace(old, new[, num]): заменяет в строке одну подстроку на другую.

split([delimeter[, num]]): разбивает строку на подстроки в зависимости от разделителя.

join(strs): объединяет строки в одну строку, вставляя между ними определенный разделитель

Пример (работа со строками).

main.py > ...

```python
s = "Hello, World!"
print(s.lower())  # hello, world!
print(s.upper())  # HELLO, WORLD!
print(s.replace('World', 'Python'))  # Hello, Python!

```

```python
string1 = "Длинное предложение..."
print(len(string1)) # длина строки (сколько символов в строке)
```

Интересный Пример.

Поиск самого длинного слова в строке

```
user_input = input("Введите строку: ")

# Делим строку на слова
words = user_input.split()

# Находим самое длинное слово
longest_word = max(words, key=len)

# Вывести результат
print(f"Самое длинное слово: {longest_word}")
```

Практические задачи на строки.

1. Запросить у пользователя строку (предложение например). Вывести длину строки.
2. Запросить у пользователя строку (предложение например). Применить верхний регистр ко всей строке. Вывести новую строку на экран.
3. Запросить у пользователя строку (предложение например). Удалить начальные пробелы и вывести новую строку.
4. Запросить у пользователя строку (предложение например). Потом запросить слово на замену. Произвести операцию замены слов в строке и вывести новую строку на экран.

Три важных оператора. Break, Continue, Return.

break *полностью выходит из цикла.*
continue *переходит к началу цикла (от слова продолжить)*
return *- оператор возврата чего либо (значений, переменных и тп) из функции или его также можно использовать как жесткий выключатель цикла)*

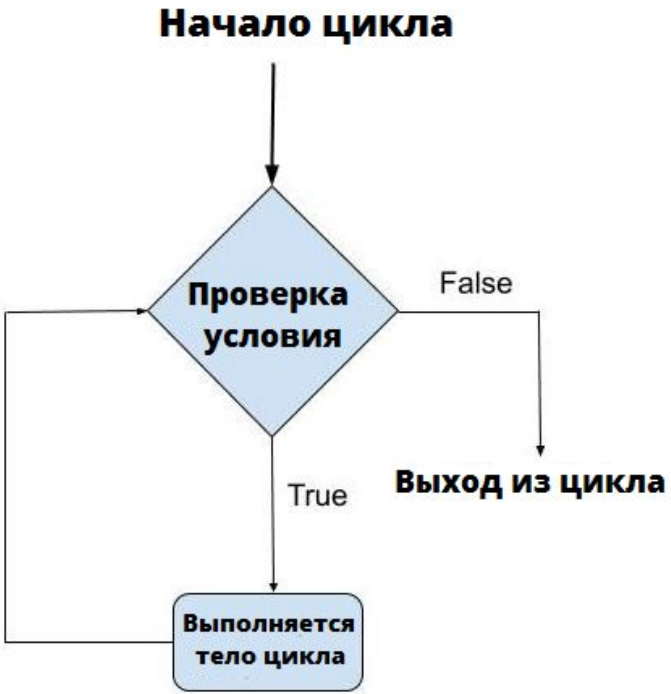
Циклы в Python

Цикл в программировании - это инструкция, которая говорит компьютеру повторять выполнение определенного блока кода несколько раз. Мы используем циклы, чтобы сделать программы более эффективными и чтобы избежать повторения одних и тех же команд.

В Python есть два основных цикла:

- Цикл **while** выполняет блок кода (*набор команд*), пока условие истинно. Как только условие становится ложным, цикл завершается.
- Цикл **for** используется для перебора элементов в последовательности *(например, в списке*). Программа будет выполнять определенные действия для каждого элемента в последовательности.

Работа цикла while



Пример#1 (while)

```
counter = 1

while counter <= 5:

        print(counter)

        counter += 1
```

В этом примере, программа будет выводить числа от 1 до 5, потому что мы увеличиваем counter на 1 после каждого вывода, и цикл продолжает выполняться, пока counter меньше или равен 5.

Пример#2 (while)

Допустим, мы хотим, чтобы пользователь вводил числа, пока не введет число 0, и мы будем выводить сумму всех введенных чисел.

```
sum_of_numbers = 0

print("Введите числа. [0] - чтобы выйти.")

while True:

        user_input = int(input("Введите число: "))

        if user_input == 0:

                break

        sum_of_numbers += user_input

print("Сумма введенных чисел:", sum_of_numbers)
```

В этом примере:

1. Мы создаем переменную sum_of_numbers, которая будет содержать сумму введенных чисел.
2. Мы используем цикл while True, чтобы создать бесконечный цикл (цикл, который будет выполняться бесконечно), и просим пользователя ввести число.
3. Если пользователь вводит 0, мы используем оператор break, чтобы выйти из цикла.
4. В противном случае, число добавляется к сумме.

Работа цикла for

Синтаксис цикла for

```
for переменная in последовательность:
    # Блок кода, который будет выполнен несколько раз
```

- переменная - это переменная, которая будет использоваться для доступа к элемент
- последовательность - это набор элементов, по которым будет выполняться цикл.

Пример#3 (for)

range() - она генерирует список чисел, который обычно используется для работы с циклом for.

range(2) - значит список 0,1

range(5) - значит список 0,1,2,3,4

range(2, 5) - значит уже от 2 начинаем т.е 2,3,4

range(2, 5, 2) - третий аргумент показывает насколько будет увеличена переменная index (стандартно это 1 - и единица не пишется, иными словами это index += 1

Пример кода:

```
for index in range(3):

    print("Привет, мир!")
```

В этом примере:

1. фраза Hello, world! выводится в консоль ровно три раза.
2. range(3) - значит пройтись от позиции 0 до позиции 3 (не включая цифру 3) т.е три раза 0,1,2 кстати это значения index (счетчика) который переходит внутри цикла с каждой итерацией счетчик увел-ся на 1

Практические задачи#1.

1. Используя цикл while, напишите программу, которая будет выводить числа от 1 до 5.
2. Напиши программу, в которой компьютер загадывает число от 1 до 10 (просто создать переменную **value_computer** со значением например 7), а игрок должен угадать это число. Программа продолжает выполняться, пока игрок не угадает число.
3. Используя цикл for, напечатайте таблицу умножения для числа 5 (от 1 до 10). Формат таков:

   5 x 1 = 5

   5 x 2 = 10

   …

4. Используя цикл for, посчитай сумму чисел от 1 до 100000.
5. Используя цикл while, вывести все четные числа от 2 до 10.
6. Запросите у пользователя число в самом цикле while. Если число >0 выйти из цикла, иначе продолжать запрашивать у пользователя число пока оно не окажется положительным.
7. Используя цикл for, напечатай числа от 10 до 1 в обратном порядке.
8. Используя цикл while, найди первые 5 степеней числа 2 (2 в степени 1, 2 в степени 2 и так далее).
9. Используя цикл while, найди сумму цифр числа 123.
10. Используя цикл for, найди факториал числа 5 (5! = 5 * 4 * 3 * 2 * 1).
11. Используя цикл for, вывести числа от 1 до 100. (range)

Практические задачи#2.

1. Есть константа L=50, запросите у пользователя число (целого типа данных int) и с помощью цикла for умножьте это число 5 раз на i (причем каждый раз i в степени i) на значение переменной L.
2. **Треугольник из звезд -** Используя вложенные циклы for, нарисуй простой треугольник из звезд. Например:

a)

b)

```
    *
   ***
  *****
 *******
*********
```

c)

для b), c) стоит учитывать пробелы (spaces) - это подсказка