



ЦИКЛЫ

Занятие №4

Проверка пройденного на занятии №3

1. Дайте определение строке.
2. Какие три способа создания строки Вы знаете?
3. Расскажите про базовые операции над строками.
4. Дайте определение срезу и индексу.
5. Строка – это изменяемый тип данных?
6. Расскажите про первую форму среза.
7. Расскажите про вторую форму среза.
8. Расскажите про третью форму среза.
9. Расскажите про методы работы с регистром строки.
10. Расскажите про методы работы с объединением и разбивкой строк.
11. Расскажите про методы работы с поиском и заменой внутри строки.
12. В чём разница find и index?

Проверка домашнего задания

Задача №1:

Дана строка.

Сначала выведите третий символ этой строки.

Во второй строке выведите предпоследний символ этой строки.

В третьей строке выведите первые пять символов этой строки.

В четвертой строке выведите всю строку, кроме последних двух символов.

В пятой строке выведите все символы с четными индексами (считая, что индексация начинается с 0, поэтому символы выводятся начиная с первого).

В шестой строке выведите все символы с нечетными индексами, то есть начиная со второго символа строки.

В седьмой строке выведите все символы в обратном порядке.

В восьмой строке выведите все символы строки через один в обратном порядке, начиная с последнего.

В девятой строке выведите длину данной строки.

Решение

```
1 s = input()
2
3 print(s[2]) # Сначала выведите третий символ этой строки.
4 print(s[-2]) # Во второй строке выведите предпоследний символ этой строки.
5 print(s[:4]) # В третьей строке выведите первые пять символов этой строки.
6 print(s[:-2]) # В четвертой строке выведите всю строку, кроме последних двух символов.
7 print(s[::2]) # В пятой строке выведите все символы с четными индексами
8               # (считая, что индексация начинается с 0, поэтому символы выводятся начиная с первого)
9 print(s[1::2]) # В шестой строке выведите все символы с нечетными индексами, то есть начиная со
10               # второго символа строки.
11 print(s[::-1]) # В седьмой строке выведите все символы в обратном порядке.
12 print(s[::-2]) # В восьмой строке выведите все символы строки через один в обратном порядке, начиная
13               # с последнего.
14 print(len(s)) # В девятой строке выведите длину данной строки.
```

План занятия

- 1) Введение в циклы.
- 2) Цикл for.
- 3) Модуль array. Массивы в python

Каждый язык программирования, содержит какую-нибудь конструкцию цикла. В большей части языков есть больше одной такой конструкции. В мире Python есть два типа циклов:

- Цикл **for**
- Цикл **while**

Циклы используются в тех случаях, когда нам нужно сделать что-нибудь много раз. Нередко вам придется выполнить какую-нибудь операцию (или ряд операций) в части данных снова и снова.

Тут то и вступают в силу циклы. Благодаря им становится возможно максимально упростить данный вопрос.

Цикл **for**

Цикл **for**, также называемый циклом с параметром, в языке Питон богат возможностями. В цикле **for** указывается переменная и множество значений, по которому будет пробегать переменная.

Для повторения цикла некоторое заданное число раз n используют цикл **for** вместе с функцией **range**:

```
for i in range(4):  
    print(i)
```

range(4) - формирует диапазон от 0 до 4, не включая 4 = (0,1,2,3), а переменная **i** самостоятельно перебирает этот диапазон и с помощью команды **print(i)**, данная переменная выводится на экран.

```
0
```

```
1
```

```
2
```

```
3
```

```
Process finished with exit code 0
```


Так же в `range` можно передать два или три параметра:

`range(4, 8)` - формирует диапазон от 4 до 8, не включая 8 = (4,5,6,7)

`range(4, 8, 1)` - **1** - это шаг, шаг может быть и отрицательным.

Например:

`range(4, 8, 1)` = (4,5,6,7)

`range(1, 9, 3)` = (1,4,7)

`range(10, 5, -2)` = (10, 8, 6)

Разберем следующую программу:

```
for i in "Я учу Python":  
    print(i)
```

Переменная `i` пробегает по строке «Я учу Python» и на каждой итерации цикла, переменной `i` присваивается следующий символ строки. Поэтому на экран и вывелась наша строка, но посимвольно.

Я

у

ч

у

Р

у

т

h

o

n

Process finished with exit code 0

Разберем задачу: Необходимо вывести числа от 1 до 15 в порядке убывания.

```
for i in range(15, 0, -1):  
    print(i)
```

В **range** мы передаем три параметра 15, 0, -1. **range** формирует диапазон от 15 до 0 с шагом -1, не включая 0 = (15,14.....1) и далее мы просто выводим на экран нашу переменную **i**, которая каждую новую итерацию берет следующее число из нашей последовательности.

```
15  
14  
13  
12  
11  
10  
9  
8  
7  
6  
5  
4  
3  
2  
1
```

```
Process finished with exit code 0
```

Практическое задание:

Задание №1

Пользователь вводит строку и один символ. Программа должна вывести на экран строку без этого символа.

Например: «Я учу программирование» символ «о»

Результат «Я учу прграммирвание».

Примечание: напоминаю, что строки можно складывать

«Я учу » + «программирование» = «Я учу программирование»

Решение и результат :

```
a = input('Введите строку: ')\nb = input('Введите символ: ')\nc = ''\nfor i in a:\n    if i != b:\n        c += i\nprint('Результат: ', c)
```

Введите строку: Я учу программирование

Введите символ: o

Результат: Я учу прграммирвание

Process finished with exit code 0

Задание №2

Вводится начало, конец и шаг последовательности, нужно вывести на экран данную последовательность чисел.

Решение и результат :

```
a = int(input('Введите начало: '))  
b = int(input('Введите конец : '))  
c = int(input('Введите шаг : '))  
for i in range(a, b, c):  
    print(i)
```

```
Введите начало: 1  
Введите конец : 7  
Введите шаг : 1  
1  
2  
3  
4  
5  
6
```

Process finished with exit code 0

Задание №3

Вывести на экран все числа в диапазоне от 54 до 9184 кратные 5.

Решение:

```
for i in range(94, 9184, 1):  
    if i % 5 == 0:  
        print(i)
```

Модуль array. Массивы в python

Одномерные массивы в Python представляют собой список элементов. Значения указываются внутри квадратных скобок, где перечисляются через запятую. Как правило, любой элемент можно вызвать по индексу и присвоить ему новое значение.

Пустой список:

```
arr = []
```

Массив строк в **Python**:

```
arr = ['string1', 'string2', 'string3']
```

Чтобы вернуть число элементов внутри массива, используют функцию **len()**:

```
arr = ['string1', 'string2', 'string3']  
l = len(arr)  
print(arr, 'Длина: ', l)
```

Когда нужно перечислить элементы массива, применяют цикл **for**.

Записывается он в следующем виде:

[**for** переменная **in** массив]

```
arr = [1, 7, 9, 10]
```

```
for i in arr:  
    print(i)
```

```
1
```

```
7
```

```
9
```

```
10
```

```
Process finished with exit code 0
```

Оператор `break`

Оператор **`break`** досрочно прерывает цикл, либо условие.

```
arr = [1, 7, 9, 10]

for i in arr:
    print(i)
    if i == 9:
        break
```

1

7

9

Process finished with exit code 0

Оператор `continue`

Оператор **`continue`** начинает следующий проход цикла, минуя оставшееся тело цикла.

```
arr = [1, 7, 9, 10]

for i in arr:
    if i == 9:
        continue
    print(i)
```

```
1
7
10
```

```
Process finished with exit code 0
```


Добавление элемента

Добавление элемента в конец массива в python.

Имя массива.append(значение)

```
a = [10, 2, 3]
print(a)
a.append(7)
print(a)
```

```
[10, 2, 3]
```

```
[10, 2, 3, 7]
```

```
Process finished with exit code 0
```

Методы массивов (array) в python

array.append(x) - добавление элемента в конец массива.

array.count(x) - возвращает количество вхождений x в массив.

array.index(x) - номер первого вхождения x в массив.

array.pop(i) - удаляет i-ый элемент из массива и возвращает его. По умолчанию удаляется последний элемент.

array.remove(x) - удалить первое вхождение x из массива.

array.reverse(x) - обратный порядок элементов в массиве.

Задание №4

У вас есть массив, в котором его элементы — это названия блюд на ужин.

Вам нужно перебрать этот массив и если название одного из блюд равняется тому, что вы укажете сами. Например, вывести сообщение: “Я не ем ...” и закончить программу.

Решение и результат :

```
food = ["отбивные", "пельмени", "яйца", "орехи"]
a = input("Введите блюдо: ")
for i in food:
    if i == a:
        print("Я не ем пельмени!")
        break
    print("Отлично, вкусные " + i)
else:
    print("Хорошо, что не было пельменей!")

print("Ужин окончен.")
```

```
Введите блюдо: пельмени
Отлично, вкусные отбивные
Я не ем пельмени!
Ужин окончен.
```

```
Process finished with exit code 0
```

Задание №5

Массив чисел. Найти их сумму и произведение.

Решение и результат :

```
arr = [1, 2, 5, 6, 7, 8, 9]
sum = 0
pr = 1
for i in arr:
    sum += i
    pr *= i
print("Сумма", sum)
print("Произведение", pr)
```

Сумма 38

Произведение 30240

Process finished with exit code 0

Задание №6

Таблица умножения от 1 до 9.

Решение и результат :

```
# Практическое 6

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

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

Process finished with exit code 0

Домашнее задание.

1. Перемножить все нечётные значения в диапазоне от 1 до 100.
2. Записать в массив все числа в диапазоне от 1 до 500 кратные 5.
3. Вывести на экран все чётные значения в диапазоне от 1 до 497.
4. Дан массив чисел. Если число встречается более двух раз, то добавить его в новый массив.