

Методические указания

Урок 4.1. Цикл for. Функция range().

Задачи урока:

- Познакомиться с циклом for;
- Познакомиться с функцией range();

План урока:

0. Подготовка к уроку

До начала урока преподавателю необходимо:

- 1) Просмотреть, как ученики справились с домашним заданием
- 2) Прочитать методичку

1. Цикл for

Учитель: Мы уже изучили:

- ввод и вывод данных;
- условный оператор (кстати, одна из важнейших концепций всех языков программирования, без которой тяжело написать сколько-нибудь полезную программу);
- строковый тип данных.

Пришло время поговорить об еще одной концепции, присущей всем языкам программирования - это циклы.

В языке Python существует два типа циклов:

- счетные циклы, когда мы заранее знаем количество повторений;
- условные циклы, когда мы заранее не знаем количество повторений, но знаем условие, согласно которому нужно что-либо повторять.

Счетный цикл в языке Python (как и в большинстве других языков программирования) называется for.

Представьте, что надо написать программу, выводящую на экран слово “Привет” один раз. Очевидно, команда `print()`, запросто справится с поставленной задачей. А что если слово “Привет” нужно вывести 5 раз? Можно написать 5 строк кода. А если 10 раз? Неужели писать 10 одинаковых строк кода? Это достаточно частый сценарий: повторить что-либо определенное количество раз, поэтому можно предположить, что в языках программирования должны быть средства, позволяющие решить поставленную задачу проще и быстрее. И действительно, они есть, и называются циклами.

Рассмотрим код, использующий цикл `for`:

```
for i in range(10):  
    print('Привет')
```

Обратите внимание, программа состоит из 2 строчек кода вместо 10.

Общая структура цикла `for` в Python имеет вид:

```
for название переменной in range(количество повторений):  
    блок кода
```

Мы пишем ключевое слово `for`, затем название переменной цикла (об этом чуть позже), далее слова `in range()` и в скобках указываем количество повторений. Далее следует обязательный символ двоеточия, а затем на новой строке, с отступом - блок кода, который и будет повторяться заданное количество раз.

Обратите внимание: двоеточие (:) в конце строки сообщает Python, что дальше находится повторяющийся блок команд - **тело цикла**.

Похожую структуру имеет и уже знакомый условный оператор `if`.

```
if условие:  
    блок кода
```

Однократное выполнение тела цикла называется **итерацией цикла**;

Учитель: На предыдущих уроках мы писали программы, обрабатывающие числа. С помощью цикла `for` можно писать код для считывания и обработки любого количества чисел.

Эта программа, используя цикл for, считывает 5 чисел и выводит их квадраты с поясняющим текстом:

```
for i in range(5):  
    num = int(input())  
    print('Квадрат вашего числа равен:', num*num)  
print('Цикл завершен')
```

Тело цикла состоит из двух строк кода, они выделены отступом. Не забывайте, что в Python отступы имеют большое значение!

Четвертая строка кода, печатающая текст “Цикл завершен” не содержит отступа, так как не является частью цикла.

Учитель: Давайте закрепим материал. Что будет выведено на экран в результате выполнения следующего кода?

```
print('A')  
print('B')  
for i in range(5):  
    print('C')  
    print('D')  
print('E')
```

Если ученики испытывают трудности, наводящими вопросами помогаем прийти к правильному ответу. Затем подробно комментируем код, отмечая, что его первые две строки находятся вне цикла, и будут выполнены единожды, до цикла.

Тело цикла состоит из двух строк кода, выводящих по одному символу C, D, и повторяется 5 раз. А после цикла есть еще одна команда — печати символа E. Таким образом, результатом выполнения такой программы будет:

A
B
C
D
C
D

C
D
C
D
C
D
E

Учитель: Программа может содержать сколько угодно циклов. Как вы думаете, что будет выведено на экран в результате выполнения следующей программы?

```
print('A')
print('B')
for i in range(5):
    print('C')
for i in range(5):
    print('D')
print('E')
```

После получения правильного ответа учитель комментирует код: программа содержит два цикла, первый печатает 5 раз символ C, второй – 5 раз символ D. Три строки программы не входят в тела циклов.

Учитель: Итак:

1. однократное выполнение тела цикла называется **итерацией цикла**;
2. слово **for** пишется маленькими буквами;
3. первая строка цикла заканчивается двоеточием (:);
4. тело цикла выделяют отступом.

Учитель: Посмотрим еще раз на структуру цикла for в Python:

```
for название переменной in range (количество повторений) :
    блок кода
```

Для чего нужна и как работает **переменная цикла**? Рассмотрим следующий программный код:

```
for i in range(10):  
    print(i)
```

В теле цикла указана команда `print()`, печатающая переменную цикла `i`. Что увидим на экране?

Учитель: Когда цикл начинает работу Python устанавливает значение переменной цикла `i = 0`. Каждая итерация цикла Python увеличивает значение переменной на 1. Таким образом, результатом выполнения такого кода будет:

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9
```

Учитель: Обратите внимание, первое значение 0, а не единица! И последнее число на единицу меньше указанного в скобках. Это особенность языка Python, о которой важно всегда помнить.

Так решили разработчики языка Python. Существуют языки в которых начальное значение начинается с 1, это не принципиально, просто нужно помнить. После решения некоторого количества задач, это станет настолько привычно, что не будет вызывать никаких трудностей.

Учитель: Если все же нам нужно начать с 1, то можно сделать так:

```
for i in range(10):  
    print(i + 1, '-- Привет')
```

В теле цикла печатаем `i+1`, вместо `i`, а также слово Привет. Результатом выполнения такой программы будет:

```
1 -- Привет  
2 -- Привет  
3 -- Привет  
4 -- Привет
```

```
5 -- Привет
6 -- Привет
7 -- Привет
8 -- Привет
9 -- Привет
10 -- Привет
```

Учитель: Поговорим об именовании переменных цикла. Имя переменной должно быть “говорящим”. Например, если она обозначает “имя человека”, то может называться name. Если “число”, то переменная может называться num, number

Для переменных цикла принято использовать буквы **i, j, k**. Однако это не значит, что мы не можем давать более говорящие названия переменным цикла. Ничто не мешает назвать переменную цикла num.

Иногда переменная цикла совсем не используется внутри тела цикла. Например, следующая программа выводит 5 раз текст:

```
for i in range(5):
    print('Python')
```

В теле цикла нет упоминания переменной i, она не используется. Так зачем ей давать название? В Python можно написать так:

```
for _ in range(5):
    print('Python')
```

Мы заменили имя переменной на символ нижнего подчеркивания, тем самым сказав: “нам не нужна переменная цикла внутри тела цикла”. Результатом выполнения такого кода будет:

```
Python
Python
Python
Python
Python
```

И еще раз: помните, что правая граница в цикле for, всегда не включена, а счет начинается с нуля. Забывают об этом часто, это очень распространенная ошибка.

2. Range()

Учитель: Функция `range()`, на самом деле позволяет нам намного больше. В примерах выше мы указывали только одно значение внутри скобок. Давайте рассмотрим другие варианты.

```
for i in range(1, 10):  
    print(i)
```

Данный пример выведет числа от 1 до 10.

```
for i in range(1, 10, 2):  
    print(i)
```

Данный же пример выведет каждое второе число от 1 до 10.

Разберем синтаксис `range()`

`range(начало, конец, шаг)`

3. Перебор последовательности с `for`

Учитель: Давайте представим, что нам необходимо с помощью цикла перебрать, например, строку. Как думаете возможно ли это?

Учитывая, что строка является последовательностью, мы можем обратиться к каждому символу строки(элементу списка) при переборе с помощью цикла.

Давайте рассмотрим пример. Пользователь вводит строку и программа выводит все символы данной строки.

```
word = input()  
for char in word:  
    print(char)
```

В данном случае мы не используем функцию `range()`. Мы условно говорим что у нас создается переменная `char`, внутри цикла, а каждый символ строки (элемент, если это список) перезаписывается поочередно в переменную `char`. Таким образом у нас выведутся поочередно все символы нашей строки (каждый на новой строке).

Также можно перебрать последовательность, используя и функцию `range()`, но для этого в качестве количества повторов цикла, нам надо указать длину последовательности, а обращаться к элементам последовательности по индексу (порядковому номеру, который начинается с 0). Индекс указывается в квадратных скобках.

```
word = input()
for i in range(len(word)):
    print(word[i])
```

4. Решение задач

Задача 1

Вывести на экран циклом пять строк из нулей, причем каждая строка должна быть пронумерована;

Решение

```
for i in range(5):
    print(str(i) + ' 0')
```

Задача 2

Посчитать сумму числового ряда от 0 до 14 включительно. Например, $0+1+2+3+...+14$;

Решение

```
sum = 0
for i in range(14):
    sum += i

print(sum)
```

Дополнительно

Если на уроке остается время, то ученикам можно предложить начать прорешивать домашнее задание.

Домашняя работа