

Методические указания

Урок 9.1. Функции

Задачи урока:

- Изучить понятие локальных и глобальных переменных и научиться писать функции возвращающие значения
- lambda функции

0. Подготовка к уроку

До начала урока преподавателю необходимо:

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

1. Локальные и глобальные переменные

Учитель: С понятием функций в языках программирования тесно связано понятие локальных и глобальных переменных.

Переменная, доступ к которой можно получить из любого места в коде, называется глобальной. Ее можно определить вне блока. Другими словами, глобальная переменная, объявленная вне функции, будет доступна внутри нее.

С другой стороны, переменная, объявленная внутри определенного блока кода, будет видна только внутри этого же блока — она называется локальной.

Давайте рассмотрим это на примерах

```
x = 5
def func():
    print(x)

func()
```

В данном примере переменная `x` - глобальная. Она создана вне функции, а в теле основного файла программы. Данная переменная отлично видится и внутри функции. Теперь попробуем изменить значение переменной из функции

```
x = 5
def func():
    x += 1

func()
print(x)
```

Результатом выполнения данного кода будет ошибка `UnboundLocalError: local variable 'x' referenced before assignment`. Для того, чтобы изменить глобальную переменную из функции необходимо дописать в функции *global x*

```
x = 5
def func():
    global x
    x += 1

func()
print(x)
```

Тем самым, мы указали, что переменная является глобальной. Если же у нас функция вложенная в другую функцию, то для изменения переменной во внешней функции из внутренней необходимо указать `nonlocal`.

Также давайте рассмотрим другой пример

```
def func():
    x = 5

func()
print(x)
```

Тут уже мы получим другую ошибку `NameError: name 'x' is not defined`, что означает, что программа не видит данную переменную и она является локальной для данной функции. Получается если мы рассчитываем в нашей функции какие то значения, то мы не можем передать их обратно в программу? Можем! Для этого нам понадобится функция, возвращающая значения

2. Функции, возвращающие значения

Учитель: До этого мы рассмотрели два типа функций: одни не принимают, другие принимают параметры. Кроме того, функции могут возвращать значения.

Для того, чтобы указать, что функция будет возвращать какое то значение или значения, необходимо указать оператор возврата return.

```
def func():  
    x = 5  
    x += 10  
    return x  
  
result = func()  
print(result)
```

В данном случае функция вернет значение в переменную result, либо же если нам необходимо просто вывести значение - использовать print(func())

Если необходимо вернуть более одного значения, то возвращаемые значения, необходимо указать через запятую.

```
def func():  
    x = 5  
    y = 10  
    return x, y  
  
result1, result2 = func()  
print(result1, result2)
```

Помимо возврата значения, оператор return работает как break для цикла, т.е весь код после оператора return выполнен не будет

```
def func():  
    x = 5  
    return x  
    print('hello')  
  
print(func())
```

Функции могут быть как принимающими или возвращающими, так и принимающими и возвращающими одновременно

3. Lambda функции

Lambda функции или анонимные функции отличаются от обычных.

1. Они не имеют имени
2. Содержат в себе только одну строку кода

3. При создании не используется def

```
result = lambda x: x + 5  
print(result(4))
```

После слово lambda указывается принимаемое значение, а после : что функция возвращает. Принимаемых значений может быть также более 1.

```
result = lambda x, y: x + y  
print(result(4, 5))
```

4. Решение задач

Задача 1

Написать программу, которая принимает от пользователя год и возвращает високосный год или нет. Високосный год - год, который кратен 4, но не кратен 100 или кратен 400

Решение

```
def get_year(n):  
    if (n % 4 == 0 and n % 100 != 0) or n % 400 == 0:  
        return 'Високосный'  
    else:  
        return 'Обычный'  
print(get_year(2000))
```

Дополнительно

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

Домашняя работа

Задача 1

Написать функцию isPrime, которая принимает число и возвращает простое число или нет

Решение

```
def isPrime(n):  
    for i in range(2, n // 2 + 1):  
        if n % i == 0:  
            return False  
    return True
```

```
print(isPrime(11))
```