

Теоретический материал к занятию Функции 2 занятие.

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.

```
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.

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`.

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

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))
```

