Локальные и глобальные переменные

PYTHON

Глобальные или локальные?

```
def any():
    print(v
+ 1)

v = 1
any()
print(v)
```

```
2
1
```

1. В Python, переменные, которые используются в основном коде, а затем внутри функции на них **только ссылаются** (нет присваивания для этих переменных в теле функции), считаются глобальными по умолчанию.
Например, на момент вызова функции **any()**, основной код знает переменную **v** и функция может использовать это значение.

```
def any():
    v = 5
    print(v
+ 1)

v = 1
any()
print(v)
```

```
6
1
```

2. Но, как только внутри функции произойдет **новое присваивание** этой переменной **v**, она становится локальной.

3. Если все-таки необходимо и учесть внешнее значение переменной, и при этом внутри функции присвоить новое значение, то необходимо явно указать переменную глобальной с помощью оператора **global**.

```
def any():

    global v
        v +=
    5

    print(v +
    1)

v = 1
any()
print(v)
```

```
7
6
```

Хитрая задача

```
def
func(a):
    return
a**b

a = 100
b = 10
def
print(func(
func(a):
    b = 2

return
a**b

a = 100
b = 10
print(fun
c(b))
```

Получаем 10^{10}

```
10000000000
```

Почему здесь функция **func** знает переменную **b**, когда ее вызывают?

Потому что, в теле функции идет только ссылка на переменную **b** и в теле функции нет присваивания нового значения **b**, значит она воспринимается как глобальная.

Получаем 10^{2}

```
100
```

Если бы в теле функции переменной **b** присвоили новое значение, то она стала бы локальной.

Выводы

1. Если в функции переменную не передавали как параметр, она имеет в теле функции такое же имя,
как и в основном коде и в теле функции НЕТ оператора присваивания для этой переменной, то функция ЗНАЕТ и использует это значение переменной из основного кода,
т.е. переменная **глобальная по умолчанию**.

2. Если в функции переменную не передавали как параметр, она имеет в функции такое же имя,
как в основном коде и в теле функции ЕСТЬ оператор присваивания для этой переменной, то переменная становится **локальной** и всё, что с ней происходит внутри функции НЕ передается
в основной код.

3. Если в функции переменную не передавали как параметр, она имеет в функции такое же имя,
как в основном коде и в теле функции ЕСТЬ оператор присваивания для этой переменной, а изменения этой переменной НАДО передать в основной код, то в теле функции объявите её
с помощью оператора **global**.