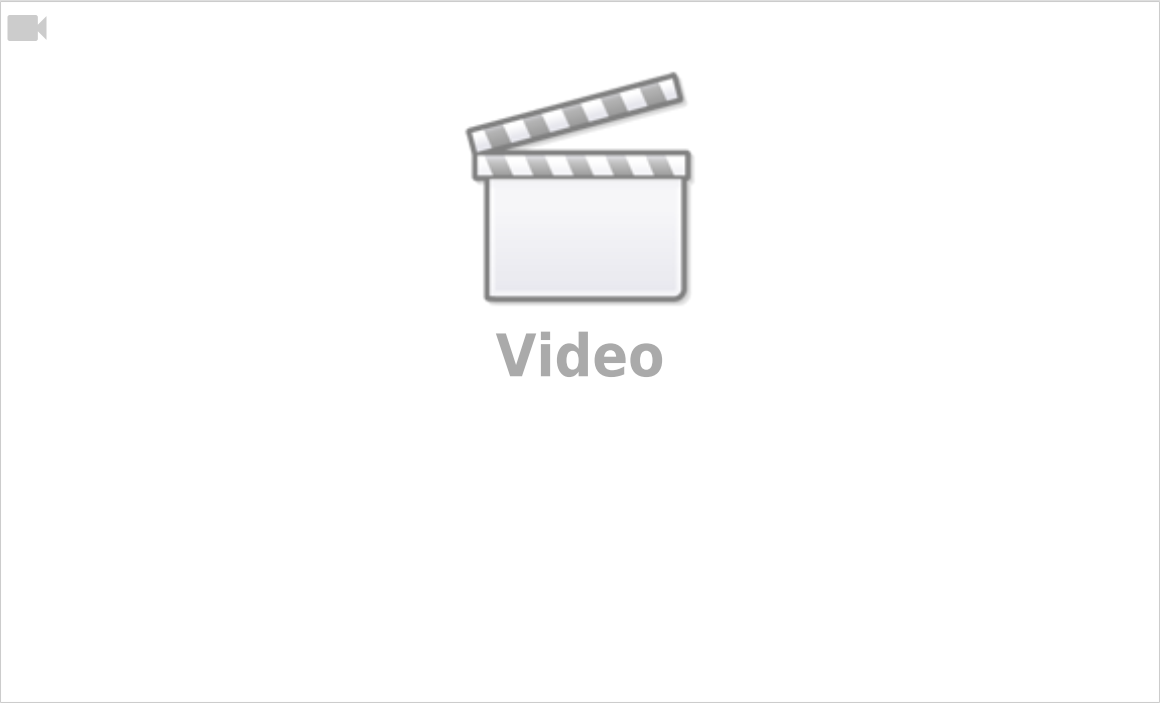
Урок 18. Область видимости



До сих пор при разработке скриптов мы не учитывали одну из важных и в тоже время довольно простых деталей: область видимости.

Разберёмся что это и для чего нужно. Напишем функцию, в которой определим переменную и будем прибавлять к ней значение (5) и возвращать из функции полученную сумму. Затем выведем сумму прибавления этого значения к 10.

```
# функция добавления к числу нового числа
def add_value(val):
    a = 5
    return a + val

b = add_value(10)
print(b)
```

Если запустить этот скрипт, мы увидим в терминале число 15.

А теперь попробуем добавить число к переменной (a):

```
# функция добавления к числу нового числа
def add_value(val):
    a = 5
    return a + val

b = add_value(a)
print(b)
```

Попробуйте запустить этот скрипт на исполнение. В терминале Вы получите ошибку: *NameError: name 'a' isn't defined*. Она говорит о том что переменную “a” вы не определили. Хотя

она вроде у нас определена в функции `add_value`?

Здесь и нужно поговорить об области видимости. Так как переменная определена внутри функции, она “видна” только внутри функции. Как только функция закончилась - переменную больше “не видно” остальной части программы. То же самое касается любых вложенных конструкций.

Например, если объявить переменную внутри цикла, то она “действительна” только в этом цикле. Если в цикл вложено условие, и переменную объявить внутри условия, то переменная будет “видна” только в условной конструкции, но не во всём цикле или программе.

Область видимости имеется практически в любом языке программирования, и определяется почти везде схожими условиями. Тоже самое относится и к объявлению функций и прочих синтаксических конструкций. Область видимости нужно всегда учитывать при написании скриптов.

Запомнить:

- Область видимости - это та область, в которой переменная видна.
- Если переменная объявлена внутри какой-либо конструкции (например, внутри функции) то “видна” она только внутри этой конструкции.
- Избегайте одинаковых названий переменных внутри разных конструкций во избежание путаницы.

Предыдущий урок

Следующий урок