
Пространство имен и области видимости функций.

Пространство имен (namespace) - это контейнер, в котором хранятся имена переменных и функций в Python. Каждое пространство имен имеет свою область видимости, определяющую, где можно использовать эти имена в программе.

Существуют три основных пространства имен в Python:

1. Пространство имен встроенных имен (Built-in Namespace):

- Содержит все встроенные функции и переменные, предоставляемые Python.
- Эти имена доступны в любой части программы без необходимости импортирования.
- Примеры встроенных имен: `print()`, `len()`, `range()`, `True`, `False` и т.д.

2. Пространство имен глобальных имен (Global Namespace):

- Содержит имена переменных и функций, определенные на верхнем уровне модуля или внутри глобальных блоков кода.
- Эти имена доступны в пределах всего модуля.
- Имена в глобальном пространстве имен могут быть использованы в любом месте в пределах модуля.

3. Пространство имен локальных имен (Local Namespace):

- Содержит имена переменных и функций, определенные внутри функций.
- Эти имена доступны только внутри соответствующей функции и не видны извне.
- После завершения выполнения функции локальное пространство имен уничтожается.

В Python существуют три типа переменных: локальные, глобальные и нелокальные. Эти типы переменных определяют их область видимости и время жизни в программе.

1. Глобальные переменные:

- Глобальные переменные объявляются вне функций и доступны во всей программе.
- Они создаются при первом присваивании значения и существуют до завершения выполнения программы.
- Глобальные переменные могут быть использованы в любой функции или в основном коде программы.

Пример глобальной переменной:

```
y = 20 # y - глобальная переменная

def my_function():
    print("Значение y внутри функции:", y)

my_function()
print("Значение y вне функции:", y)
```

2. Локальные переменные:

- Локальные переменные объявляются внутри функции и доступны только внутри этой функции.
- Они создаются при вызове функции и уничтожаются, когда функция завершает свою работу.
- Локальные переменные не могут быть использованы вне функции, в которой они определены.

Пример локальной переменной:

```
def my_function():  
    x = 10 # x - локальная переменная  
    print("Значение x внутри функции:", x)  
  
my_function()  
# print(x) # Ошибка: переменная x не определена вне функции
```

3. Нелокальные переменные:

- Нелокальные переменные используются во вложенных функциях и находятся между локальными и глобальными переменными.
- Они объявляются с помощью ключевого слова `nonlocal`.
- Нелокальная переменная является переменной внешней функции, к которой внутренняя функция имеет доступ для изменения ее значения.

Пример нелокальной переменной:

```
def outer_function():
    z = 30 # z - локальная переменная внешней функции

    def inner_function():
        nonlocal z # объявляем z как нелокальную переменную
        z += 1
        print("Значение z внутри внутренней функции:", z)

    inner_function()
    print("Значение z внутри внешней функции после вызова внутренней функции:", z)

outer_function()
```

Ключевое слово global

Ключевое слово `global` в Python используется для указания на то, что переменная внутри функции является глобальной. Это означает, что функция может изменять значение этой переменной, которая определена в глобальной области видимости.

Вот пример использования ключевого слова `global`:

```
x = 10 # глобальная переменная

def increment():
    global x # указываем, что переменная x является глобальной
    x += 1

print("Before increment:", x) # выводит 10
increment()
print("After increment:", x) # выводит 11
```

В этом примере мы объявляем переменную `x` как глобальную внутри функции `increment()`, используя ключевое слово `global`. Это позволяет нам изменять значение глобальной переменной `x` внутри функции.

Однако, следует быть осторожным при использовании ключевого слова `global`, так как это может сделать код менее читаемым и управляемым. В большинстве случаев лучше избегать использования глобальных переменных и передавать значения через параметры функций.

Ключевое слово `nonlocal`:

Ключевое слово `nonlocal` в Python используется для указания на то, что переменная внутри вложенной функции является нелокальной, то есть переменной из объемлющей функции, но не является глобальной.

Вот пример использования ключевого слова `nonlocal`:

```
def outer_function():
    x = 10 # локальная переменная

    def inner_function():
        nonlocal x # указываем, что переменная x является нелокальной
        x += 1

    inner_function()
    print("After increment in outer function:", x) # выводит 11

outer_function()
```

В этом примере переменная `x` определена внутри функции `outer_function()`. Затем мы определяем вложенную функцию `inner_function()`, которая использует ключевое слово `nonlocal`, чтобы указать на то, что переменная `x` является нелокальной. Это позволяет нам изменять значение переменной `x` внутри вложенной функции и видеть изменения в объемлющей функции.

Ключевое слово `nonlocal` полезно, когда у вас есть несколько уровней вложенности функций, и вы хотите изменить значение переменной из более глубоко вложенной функции.

Практические задания

1. Функция для расчета площади и периметра прямоугольника: Напишите функцию `calculate_rectangle_properties`, которая принимает длину и ширину прямоугольника в качестве аргументов и возвращает два значения - площадь и периметр этого прямоугольника.
2. Функция для конвертации температуры: Создайте функцию `convert_temperature`, которая принимает температуру в градусах Цельсия и возвращает эту же температуру в градусах Фаренгейта и Кельвина.
3. Напишите программу, которая использует как локальные, так и глобальные переменные. Создайте глобальную переменную `global_var` со значением "global", а затем определите функцию `local_function`, которая создает локальную переменную `local_var` со значением "local". Выведите значения обеих переменных на экран.
4. Напишите функцию `outer_function`, внутри которой определена переменная `x`. Затем внутри `outer_function` определите вложенную функцию `inner_function`, которая пытается изменить значение переменной `x`. Используйте ключевое слово `nonlocal`, чтобы указать, что переменная `x` находится в нелокальной области видимости.
5. Создайте глобальную переменную `x` и функцию `outer_function`, внутри которой определена переменная `y`. Затем внутри `outer_function` определите вложенную функцию `inner_function`, которая пытается изменить значения переменных `x` и `y`. Используйте ключевые слова `global` и `nonlocal` соответственно, чтобы указать, что переменные `x` и `y` находятся в глобальной и нелокальной областях видимости.