

Практическая работа 8 Функции и процедуры в Python

Цель работы: изучение процедур и функций в Python.

знать - синтаксис процедур и функций, процедура с параметром, локальные и глобальные переменные;

уметь - применять синтаксис процедур и функций при составлении программы;

владеть - основными навыками работы с функциями и процедурами.

Подпрограмма - это именованный фрагмент программы, к которому можно обратиться из другого места программы

Подпрограммы делятся на две категории: процедуры и функции.

1. Процедуры.

Рассмотрим синтаксис процедуры:

def имя процедуры(Список параметров):

 Система команд

Для определения процедуры используется ключевое слово `def`, затем указывается имя процедуры и в скобках её формальные параметры, если они присутствуют. После ставится двоеточие и со следующей строки с отступом в 4 пробела указываются команды.

Процедура — вспомогательный алгоритм, выполняющий некоторые действия. Процедура должна быть определена к моменту её вызова. Определение процедуры начинается со служебного слова `def`.

Вызов процедуры осуществляется по ее имени, за которым следуют круглые скобки, например, `Err()`.

В одной программе может быть сколько угодно много вызовов одной и той же процедуры.

Использование процедур сокращает код и повышает удобочитаемость.

Процедура с параметрами.

Как используются в Python параметры процедуры, рассмотрим на примере.

Пример.

Написать процедуру, которая печатает раз указанный символ (введенный с клавиатуры), каждый с новой строки.

```
def printChar(s):  
    print (s)  
sim = input('введите символ')  
printChar(sim) # первый вызов, вывод введенного символа  
printChar('*') # второй вызов, вывод *
```

```
def printChar(s):  
    print (s)  
sim = input('введите символ: |')  
printChar(sim) # первый вызов, вывод введенного символа  
printChar('*') # второй вызов, вывод *
```

```
>>>  
введите символ: 41  
41  
*
```

Глобальная переменная — если ей присвоено значение в основной программе (вне процедуры).

Локальная переменная (внутренняя) известна только на уровне процедуры, обратиться к ней из основной программы и из других процедур нельзя.

Параметры процедуры — локальные переменные.

2. Примеры использования локальных и глобальных переменных.

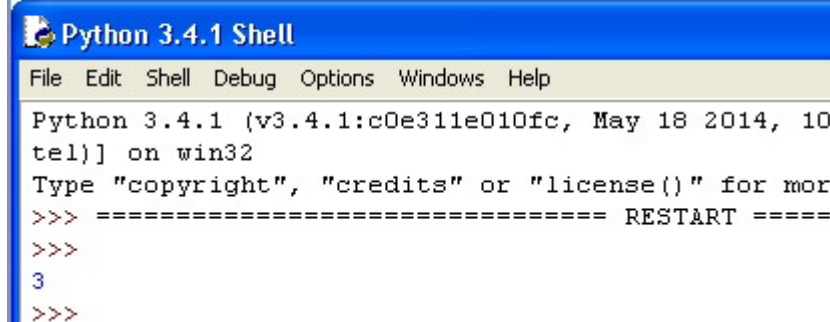
Пример 1.

x = 3 # глобальная переменная

pr(): # процедура без параметров

print (x) # вывод значения глобальной переменной pr()

```
x = 3 # глобальная переменная
def pr(): # процедура без параметров
    print (x) # вывод значения глобальной переменной
pr()
```



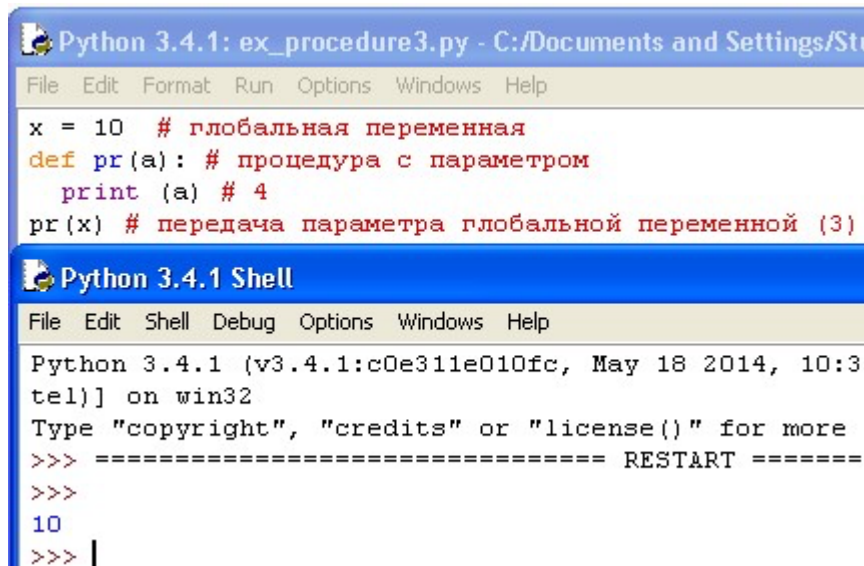
Python 3.4.1 Shell

File Edit Shell Debug Options Windows Help

Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 10:14:14) on win32
Type "copyright", "credits" or "license()" for more
>>> ===== RESTART =====
>>>
3
>>>

Пример 2.

```
x = 3 # глобальная переменная
def pr(a): # процедура с параметром
    print (a) # 4
pr(x) # передача параметра глобальной переменной (3)
```



Python 3.4.1: ex_procedure3.py - C:/Documents and Settings/St...

File Edit Format Run Options Windows Help

```
x = 10 # глобальная переменная
def pr(a): # процедура с параметром
    print (a) # 4
pr(x) # передача параметра глобальной переменной (3)
```

Python 3.4.1 Shell

File Edit Shell Debug Options Windows Help

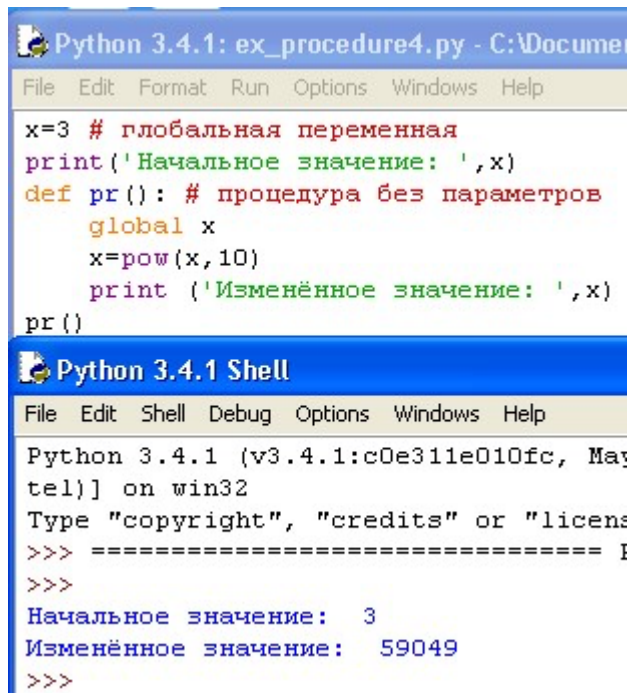
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 10:34:14) on win32
Type "copyright", "credits" or "license()" for more
>>> ===== RESTART =====
>>>
10
>>> |

Существует возможность изменить значение глобальной переменной (не создавая локальную). В процедуре с помощью слова `global`:

```
x = 3 # глобальная переменная
def pr(): # процедура без параметров
    global x
    x = pow(x,10)

    print (x) # вывод измененного значения глобальной переменной
```

pr()



The screenshot shows two windows from a Python 3.4.1 IDE. The top window, titled 'Python 3.4.1: ex_procedure4.py - C:\Docume', contains the following Python code:

```
x=3 # глобальная переменная
print('Начальное значение: ',x)
def pr(): # процедура без параметров
    global x
    x=pow(x,10)
    print('Изменённое значение: ',x)
pr()
```

The bottom window, titled 'Python 3.4.1 Shell', shows the output of running the script:

```
Python 3.4.1 (v3.4.1:c0e311e010fc, May
tel)] on win32
Type "copyright", "credits" or "licens
>>> ===== I
>>>
Начальное значение:  3
Изменённое значение:  59049
>>>
```

3. Функции.

Функция - подпрограмма, к которому можно обратиться из другого места программы. Для создания функции используется ключевое слово `def`, после которого указывается имя и список аргументов в круглых скобках. Тело функции выделяется также как тело условия (или цикла): четырьмя пробелами.

Рассмотрим синтаксис функции:

`def имя функции(Список параметров):`

Система команд `return выражение`

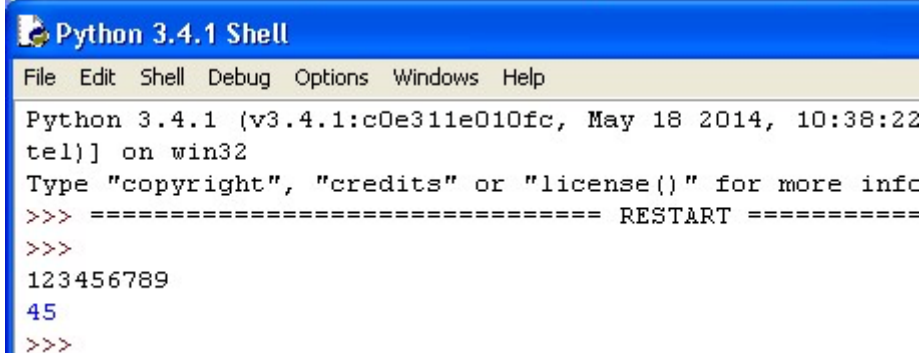
Часть функций языка Python являются встроенными функциями, которые обеспечены синтаксисом самого языка. Например, `int`, `input`, `randint`. Рассмотрим пример создания пользовательских функций.

Пример 1.

Вычислить сумму цифр числа. `def sumD(n): #`
определение функции с параметром `sumD = 0`
`while n!= 0:`
 `sumD += n % 10`
 `n = n // 10`

```
    return sumD # возврат значения функции
# основная программа
print (sumD(int(input()))) # вызов функции с параметром
```

```
def sumD(n): # определение функции с параметром
    summa = 0
    while n!= 0:
        summa += n % 10
        n=n//10
    return summa # возврат значения функции
# основная программа
print (sumD(int(input()))) # вызов функции с параметром
```



Python 3.4.1 Shell

File Edit Shell Debug Options Windows Help

Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 10:38:22
tel)] on win32
Type "copyright", "credits" or "license()" for more infc
>>> ===== RESTART =====
>>>
123456789
45
>>>

Вариант 0.

1. Определить, являются ли три треугольника равновеликими. Длины сторон вводить с клавиатуры. Для подсчёта площади треугольника использовать формулу Герона. Вычисление площади оформить в виде функции с тремя параметрами.

Формула Герона:

$$S = \sqrt{p(p-a)(p-b)(p-c)},$$
$$\text{где } p = \frac{a+b+c}{2}$$

Решение:

```

import math
def s(x,y,z):
    p=(x+y+z)/2
    s=math.sqrt(p*(p-x)*(p-y)*(p-z))
    return s
A=[]
for i in range(3):
    print('Введите стороны ',i,'-го треугольника:')
    a=int(input('a:'))
    b=int(input('b:'))
    c=int(input('c:'))
    A.append(s(a,b,c))
for i in range(3):
    print('Площадь ',i,'-го треугольника {:.2f}'.format(A[i]))
if A[0]==A[1]:
    if A[0]==A[2]:
        print('Треугольники равновеликие')
else: print('Треугольники не равновеликие')

```

```

Введите стороны  0 -го треугольника:
a:3
b:4
c:5
Введите стороны  1 -го треугольника:
a:6
b:7
c:8
Введите стороны  2 -го треугольника:
a:9
b:10
c:11
Площадь  0 -го треугольника 6.00
Площадь  1 -го треугольника 20.33
Площадь  2 -го треугольника 42.43
Треугольники не равновеликие

```

2. Ввести одномерный массив A длиной m. Поменять в нём местами первый и последний элементы. Длину массива и его элементы ввести с клавиатуры. В программе описать процедуру для замены элементов массива. Вывести исходные и полученные массивы.

Решение:

```

def zam(X):
    tmp=X[0]
    X[0]=X[len(X)-1]
    X[len(X)-1]=tmp
A=[]
m=int(input('Введите длину массива:'))
for i in range(m):
    print('Введите ',i,'элемент массива')
    A.append(int(input()))
print(A)
zam(A)
print(A)

```

```

Введите длину массива:5
Введите  0 элемент массива
0
Введите  1 элемент массива
1
Введите  2 элемент массива
2
Введите  3 элемент массива
3
Введите  4 элемент массива
4
[0, 1, 2, 3, 4]
[4, 1, 2, 3, 0]

```

Вариант 1.

1. Составить программу для вычисления площади разных геометрических фигур.
2. Даны 3 различных массива целых чисел (размер каждого не превышает 15). В каждом массиве найти сумму элементов и среднеарифметическое значение.

Вариант 2.

1. Вычислить площадь правильного шестиугольника со стороной а, используя подпрограмму вычисления площади треугольника.
2. Пользователь вводит две стороны трех прямоугольников. Вывести их площади.

Вариант 3.

1. Даны катеты двух прямоугольных треугольников. Написать функцию вычисления длины гипотенузы этих треугольников. Сравнить и вывести какая из гипотенуз больше, а какая меньше.
2. Преобразовать строку так, чтобы буквы каждого слова в ней были отсортированы по алфавиту.

Вариант 4.

1. Даны две дроби A/B и C/D (A, B, C, D — натуральные числа). Составить программу деления дроби на дробь. Ответ должен быть несократимой дробью. Использовать подпрограмму алгоритма Евклида для определения НОД.
2. Задана окружность $(x-a)^2 + (y-b)^2 = R^2$ и точки $P(p_1, p_2)$, $F(f_1, f_1)$, $L(l_1, l_2)$. Выяснить и вывести на экран, сколько точек лежит внутри окружности. Проверку, лежит ли точка внутри окружности, оформить в виде процедуры.

Вариант 5.

1. Даны две дроби A/B и C/D (A, B, C, D — натуральные числа). Составить программу вычитания из первой дроби второй. Ответ должен быть несократимой дробью. Использовать подпрограмму алгоритма Евклида для определения НОД.
2. Напишите программу, которая выводит в одну строчку все делители переданного ей числа, разделяя их пробелами.

Вариант 6.

1. Составить программу нахождения наибольшего общего делителя (НОД) и наименьшего общего кратного (НОК) двух натуральных чисел $\text{НОК}(A, B) = (A*B)/\text{НОД}(A, B)$. Использовать подпрограмму алгоритма Евклида для определения НОД.
2. Составить программу вычисления площади выпуклого четырехугольника, заданного длинами четырех сторон и диагонали.

Вариант 7.

1. Даны числа X, Y, Z, T — длины сторон четырехугольника. Вычислить его площадь, если угол между сторонами длиной X и Y — прямой. Использовать две подпрограммы для вычисления площадей: прямоугольного треугольника и прямоугольника.
2. Напишите программу, которая переводит переданное ей неотрицательное целое число в 10-значный восьмеричный код, сохранив лидирующие нули.

Вариант 8.

1. Найти все натуральные числа, не превосходящие заданного n , которые делятся на каждую из своих цифр.
2. Ввести одномерный массив A длиной m . Поменять в нём местами первый и последний элементы. Длину массива и его элементы ввести с клавиатуры. В программе описать процедуру для замены элементов массива. Вывести исходные и полученные массивы.

Вариант 9.

1. Из заданного числа вычислить сумму его цифр. Из результата вновь вычислить сумму его цифр и т. д. Через сколько таких действий получится ноль?
2. Даны 3 различных массива целых чисел. В каждом массиве найти произведение элементов и среднеарифметическое значение.

Вариант 10.

1. На отрезке $[100, N]$ ($210 < N < 231$) найти количество чисел, составленных из цифр a, b, c .
2. Составить программу, которая изменяет последовательность слов в строке на обратную.

Вариант 11.

1. Два простых числа называются «близнецами», если они отличаются друг от друга на 2 (например, 41 и 43). Напечатать все пары «близнецов» из отрезка $[n, 2n]$, где n — заданное натуральное число, большее 2..
2. Даны две матрицы A и B . Написать программу, меняющую местами максимальные элементы этих матриц. Нахождение максимального элемента матрицы оформить в виде процедуры.

Вариант 12.

1. Два натуральных числа называются «дружественными», если каждое из них равно сумме всех делителей (кроме его самого) другого (например, числа 220 и 284). Найти все пары «дружественных» чисел, которые не больше данного числа N .
2. Даны длины сторон треугольника a , b , c . Найти медианы треугольника, сторонами которого являются медианы исходного треугольника. Для вычисления медианы проведенной к стороне a , использовать формулу
Вычисление медианы оформить в виде процедуры.

Вариант 13.

1. Натуральное число, в записи которого n цифр, называется числом Армстронга, если сумма его цифр, возведенная в степень n , равна самому числу. Найти все числа Армстронга от 1 до k .
2. Три точки заданы своими координатами $X(x_1, x_2)$, $Y(y_1, y_2)$ и $Z(z_1, z_2)$. Найти и напечатать координаты точки, для которой угол между осью абсцисс и лучом, соединяющим начало координат с точкой, минимальный. Вычисления оформить в виде процедуры.

Вариант 14.

1. Составить программу для нахождения чисел из интервала $[M, N]$, имеющих наибольшее количество делителей.
2. Четыре точки заданы своими координатами $X(x_1, x_2)$, $Y(y_1, y_2)$, $Z(z_1, z_2)$, $P(p_1, p_2)$. Выяснить, какие из них находятся на максимальном расстоянии друг от друга и вывести на экран значение этого расстояния. Вычисление расстояния между двумя точками оформить в виде процедуры.

Вариант 15.

1. Найти все простые натуральные числа, не превосходящие n , двоичная запись которых представляет собой палиндром, т. е. читается одинаково слева направо и справа налево.
2. Четыре точки заданы своими координатами $X(x_1, x_2, x_3)$, $Y(y_1, y_2, y_3)$, $Z(z_1, z_2, z_3)$, $T(t_1, t_2, t_3)$. Выяснить, какие из них находятся на минимальном расстоянии друг от друга и вывести на экран значение этого расстояния. Вычисление расстояния между двумя точками оформить в виде процедуры.