



Запись вспомогательных алгоритмов на языке Python

НАЧАЛА ПРОГРАММИРОВАНИЯ

8 класс

Ключевые слова

- **Процедуры**
- **Функции**



Процедура - подпрограмма, выполняющая некоторые действия; она может иметь произвольное количество входных параметров.

Описание процедуры имеет вид:

def <имя процедуры> () :

<операторы>

def (от англ. **define** – определять)

Операторы, которые входят в тело процедуры, записываются с отступом, тем самым показывая какие команды входят в процедуру.

Для того, чтобы процедура заработала, её необходимо вызвать по имени: причем таких вызовов может быть сколько угодно. Процедура должна быть определена к моменту её вызова, т.е. должна быть выполнена команда `def`, создающая объект-процедуру в памяти. Если процедура вызывается из основной программы, то нужно поместить определение процедуры раньше точки вызова.

Пример 1. Вывести четыре строки, каждая из которых состоит из семи единиц.

```
def digit():
```

```
    print (1111111)
```

```
print('Четыре строки из семи единиц')
```

```
digit()
```

```
digit()
```

```
digit()
```

```
digit()
```

Пример 2. Вывести четыре строки, каждая из которых состоит из 7, 8, 9, 10 единиц.

```
print('1'*n) #выводит единицу n раз
```

```
def digit(n):
```

```
    print('1'*n )
```

```
print('Четыре строки из 7,8,9,10 единиц')
```

```
digit(7) #выводит единицу 7 раз
```

```
digit(8) #выводит единицу 8 раз
```

```
digit(9) #выводит единицу 9 раз
```

```
digit(10) #выводит единицу 10 раз
```

Пример 3. Можно менять не только длину строки, но и цифра из кот. строится эта строка

```
def digit(d,n):  
    print (d*n )  
x = input('Введите цифру')  
y= int(input('Введите длину строки'))  
digit(x,y)
```

Переменные **d** и **n** - **локальные** переменные; они введены и используются внутри процедуры **digit**.

Обращаться к ним вне этой процедуры нельзя. Как только работа процедуры будет закончена, все локальные переменные удалятся из памяти.

В тех случаях, когда значение переменной, полученное в подпрограмме, должно быть использовано в основной программе, эту переменную следует объявить как **глобальную**.

Пример 4. Алгоритм Евклида для нахождения наибольшего общего делителя (НОД) двух чисел.

```
def nod(a, b):  
    global x  
    while a!=b:  
        if a >b:  
            a = a - b  
        else:  
            b = b - a  
  
    x = a
```

Функция - подпрограмма, имеющая единственный результат, записываемый в ячейку памяти. В отличие от процедуры, функция не только выполняет какие-то команды, но и возвращает результат в виде числа, символьной строки и др.

Описание функции имеет вид:

```
def <имя функции> ():
```

```
    <операторы>
```

```
        return <результат>
```

```
def (от англ. define – определять)
```

Операторы, которые входят в тело функции, записываются с отступом. После оператора `return` записывается результат, который возвращает функция.

В языке Python есть встроенная функция **max**, вычисляющая максимальное значение.

Пример 5. Записать функцию, которая возвращает значение наибольшего из двух чисел.

```
def max(a, b):  
    if a > b:  
        m = a  
    else:  
        m = b  
    return m
```

Пример 6. Написать программу нахождения максимального из 4 целых чисел, использующую функцию поиска максимального из 2-х чисел.

```
def max(a, b):  
    if a > b:  
        m = a  
    else:  
        m = b  
    return m
```

```
a, b, c, d = map(int, input().split())  
f = max(max(a, b), max(c, d))  
print('f=', f)
```

Вопросы и задания

В январе Саше подарили пару новорожденных кроликов. Через два месяца они дали первый приплод – новую пару кроликов. А затем давали приплод по паре кроликов каждый месяц. Каждая новая пара также дает первый приплод (пару кроликов) через два месяца, а затем – по паре кроликов каждый месяц. Сколько пар кроликов будет у Саши в декабре?

Решение:

$f(n)$ – кол-во пар кроликов в месяце с номером n .

$f(1) = 1, f(2) = 1, f(3)=2$. Из двух пар в марте, дать приплод может только одна. Аналогично в апреле. $F(4)=2$ / В мае, только родившиеся в марте и ранее $f(5) = f(4)+f(3) = 5$

Таким образом, получает последовательность Фибоначчи

