

Основные  
возможности Scilab,  
используемые при  
работе с  
пользовательскими  
функциями.



## Приёмы, методы, функции при работе со скрипт- файлами.

Обычно пользовательские функции определяются в файлах с помощью редактора и загружаются в Scilab, используя функцию `exes` или через библиотеку `lib` либо `genlib`. Однако, они также могут быть определены во время исполнения программы `deff` либо `function`. Функция определяется двумя составляющими:


Пример пользовательской функции:  
**function y = myfunction ( x )**  
**y = 2 \* x**  
**endfunction**

Способ вызов пользовательской  
функции:  
**outvar = myfunction ( invar )**

В пользовательских скриптах можно использовать любые сторонние функции, входящие в состав самой системы Scilab или ее пакетов-расширений. При этом используются следующие знаки действий: + (сложение), - (вычитание), \* (умножение), / (деление), ^ (возведение в степень), ' (транспонирование). А также любой синтаксис и встроенные функции, которые предоставляет интерпретатор Scilab.

Создать скрипт-файл можно с помощью текстового редактора. Скрипт-файл запускается на исполнение функцией `exes`.

Пример запуска скрипта:  
**exec('script.txt').**





## Приёмы, методы, функции при работе с подпрограммами-функциями.

Определить собственную подпрограмму-функцию пользователь может оформив ее в виде файла с расширением `sci`, используя текстовый редактор. Имя созданного файла и определяет имя функции, которое будет использоваться при ее вызове. В отличие от скрипт-файла подпрограмма-функция должна содержать заголовок и оператор окончания. Заголовок идет в первой строке и начинается оператором **function**. После этого в заголовке указываются имена выходных переменных.

Затем, после знака равенства, указывается имя функции и приводится заключенный в круглые скобки список имен входных переменных, разделенных запятой. После заголовка идет само тело функции, за которым следует оператор окончания **endfunction**.

Пример файла `sci`:

```
function y=func(t)  
y=zeros(t);  
for j=1:length(t)  
if t(j)>0 then  
y(j)=1/%e^t(j);  
else  
y(j)=%e^t(j);  
end,  
end  
endfunction
```

При создании скриптов и функций могут использоваться конструкции циклов и ветвлений для организации различных алгоритмов.



## Трудности при работе с функциями.

При работе с функциями можно столкнуться со следующими трудностями:

- Создание функции с несколькими параметрами осуществляется в квадратных скобках.
- Последовательность вызова внешней функции или подпрограммы предписывается примитивом высокого уровня, который устанавливает аргументы данной внешней функции или подпрограммы.
- Множество специальных команд при работе с функциями которые необходимо знать.