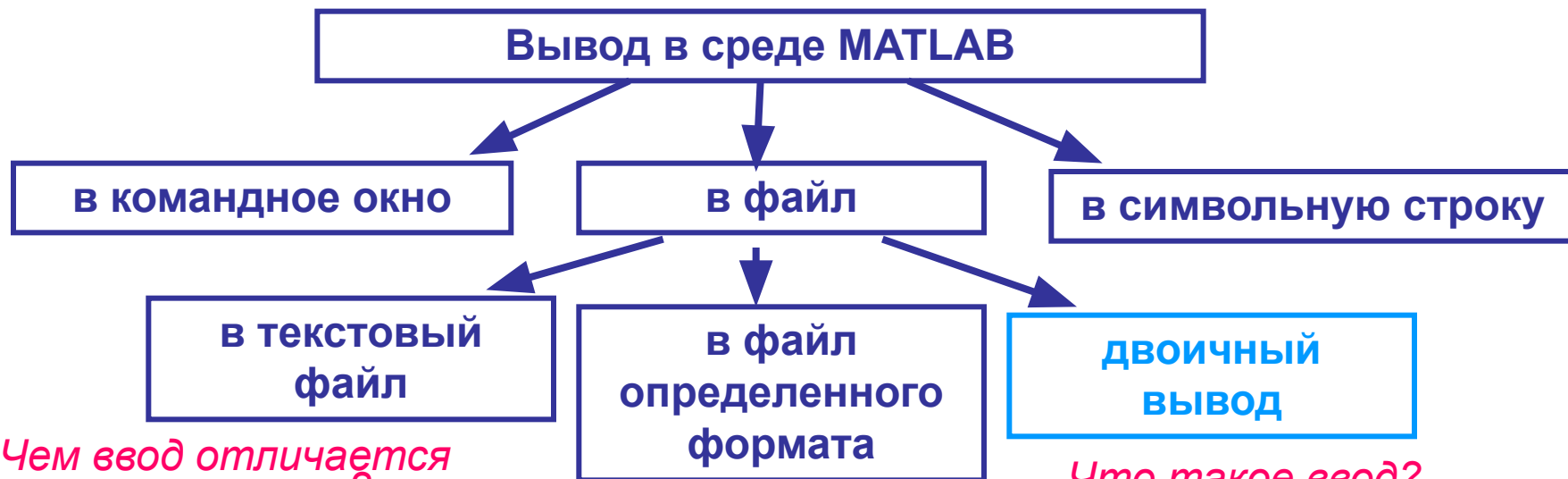
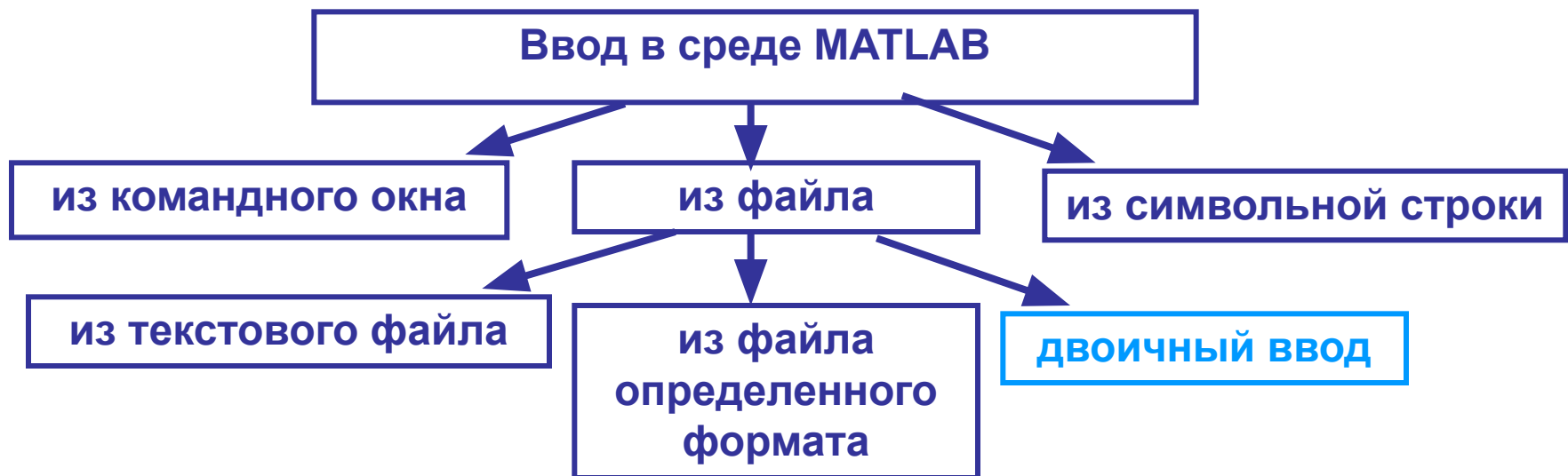


Некоторые возможности ввода и вывода в MATLAB

Лекция 9



Чем ввод отличается от присваивания?

Что такое ввод? Вывод?

Ввод из командного окна: функция `input`

`имя_переменной=input('приглашение к вводу');`

- Можно вводить выражение – переменной будет присвоен его результат.
- Если второй аргумент 's', то ввод строки.

Вывод в командное окно:

1. `>> имя_переменной`
`>> выражение` } без ;

2. Функция **display**:

`display(имя_переменной_или_выражение)`

в случае п.1 неявно вызывается `display`

3. Функция **disp**:

отличается от `display` отсутствием пояснения

`ans=`

или `имя_переменной=`

`display` запрограммирована на основе `disp`.

Примеры ввода-вывода с командным окном

файл primer_input.m

```
PR=input('Введите процентную ставку, PR=')  
V=input('Введите начальное значение вклада, V='); %Обратите внимание на;
```

```
>> primer_input  
Введите процентную ставку, PR=0.5  
  
PR =  
  
0.5000  
  
Введите начальное значение вклада, V=1000
```

```
>> A  
  
A =  
  
3 2 1  
1 1 -1  
1 -2 1  
>> display(A)
```

```
A =  
  
3 2 1  
1 1 -1  
1 -2 1  
>> disp(A)  
3 2 1  
1 1 -1  
1 -2 1
```

```
>> zagolovok=' t Vklad';  
>>for t=1:5  
V(1,:)= [1,1000];  
V(2,:)= [2,1230];  
V(3,:)= [3,1350];  
V(4,:)= [4,1397];  
V(5,:)= [5,1438];  
end  
>> disp(zagolovok);disp(V);  
t Vklad  
1 1000  
2 1230  
3 1350  
4 1397  
5 1438
```

Команда `format` – устанавливает формат вывода чисел в командное окно среды MATLAB

>>`format название_формата %` устанавливает новый формат

>>`help format %` выводит информацию о форматах

>>`format %` устанавливает формат по умолчанию
(для внутреннего представления с плавающей точкой – `short`)

Некоторые форматы вывода в командное окно среды MATLAB

Название формата	Смысл
short	Числовой формат с фиксированной точкой (если число не очень большое и не очень маленькое), 4 десятичных цифры после точки
long	То же, что и short, но 15 цифр после точки
short e	Числовой формат с плавающей точкой (научная нотация), 4 десятичных цифры после точки
long e	То же, что и short e, но 15 цифр после точки
bank	Числовой формат: две десятичных цифры после точки – денежный формат
rational	Вывод числа в виде дроби
loose	Вывод с большим межстрочным интервалом
compact	Вывод с уменьшенным межстрочным интервалом

Примеры влияния форматов на вид выводимой информации

```
format short
>> pi

ans =

    3.1416
>> realmin

ans =

    2.2251e-308

>> format long
>> pi

ans =

    3.14159265358979
>> format short e
>> pi

ans =

    3.1416e+000
```

```
>> format long e
>> pi

ans =

    3.141592653589793e+000

>> format rational
>> 3/4

ans =

    3/4

>> pi

ans =

    355/113
>> format bank
>> pi

ans =

    3.14
```

```
>> format compact
>> pi
ans =
    3.14
>> 3/4
ans =
    0.75
>> format loose
>> pi

ans =

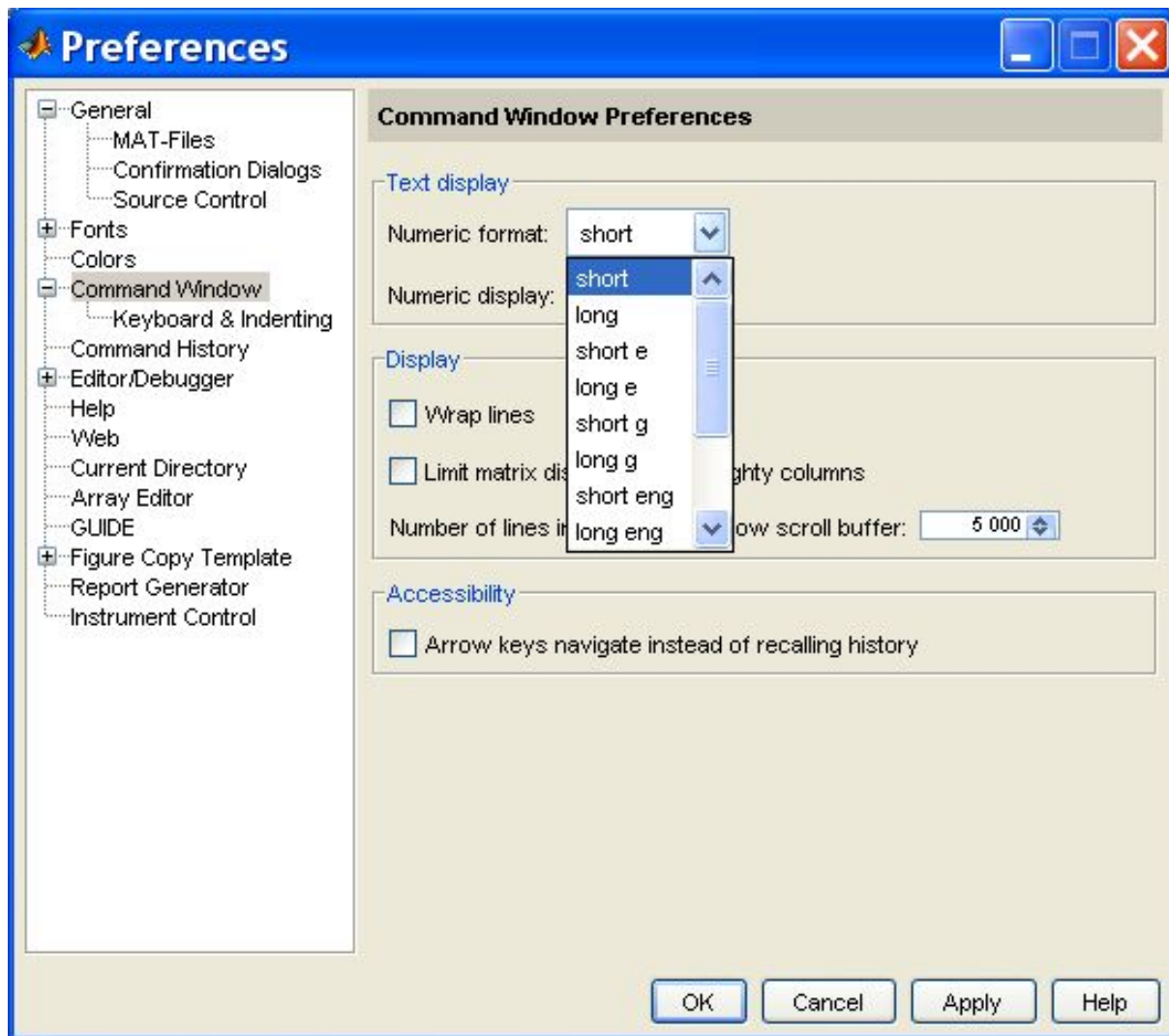
    3.14

>> 3/4

ans =

    0.75
```

Форматы также можно устанавливать с помощью окна пункта Preferences Главного меню MATLAB



Ввод из файла: функция load

1. Знаем: `load (имя_файла.txt)` – считывает содержимое текстового файла `имя_файла.txt` в переменную с именем `имя_файла`. Содержимое файла – обязательно прямоугольная таблица чисел (матрица), разделенных пробелами; в каждой строке файла должно быть одинаковое количество чисел. Файл обязательно текстовый (кодировка ASCII), расширение может быть отличным от `.txt`, **но обязательно непустое и не `.mat`**.
Пример: `load(A.txt)` – содержимое текстового файла считывается в переменную `A`.
2. `load (имя_файла.расширение)` – если расширение пустое или `.mat`, то файл считается файлом среды MATLAB, иначе (при любом другом расширении) текстовым (ASCII) файлом.
3. `Имя_переменной=load(Имя_файла.txt)` – содержимое файла `Имя_файла.txt` считывается в переменную с именем `Имя_переменной`. Требования к содержимому файла – как в п.1.
Пример: `X=load(Y.txt)` – содержимое файла `Y.txt` считывается в переменную `X`.
4. `load -ascii имя_файла.расширение` – считывает данные из файла, в предположении, что это текстовый файл, независимо от его расширения; если файл содержит данные, отличные от текстового представления чисел, то выдается сообщение об ошибке.
Пример: `load -ascii A.data` – считывается содержимое файла `A.data` в переменную `A`.
5. `load -mat имя_файла.расширение` – считывает данные из файла, в предположении, что это файл формата `.mat`, независимо от его расширения; если это файл другого формата, то выдается сообщение об ошибке.

Считывание файлов формата `.mat` с помощью функции `load`

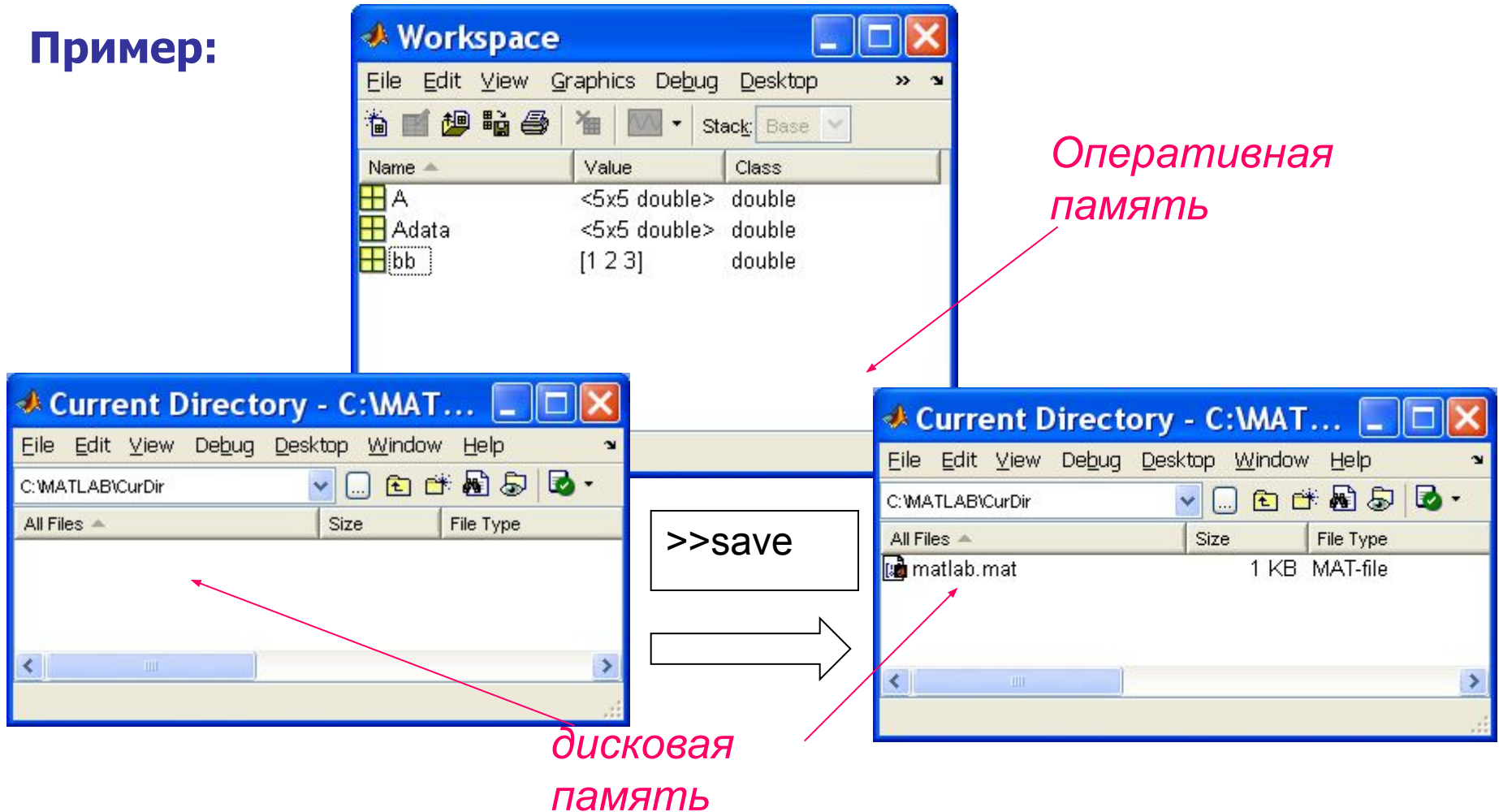
1. `load` без параметров загружает (считывает) **все переменные** из файла с именем `matlab.mat`. Этот файл ранее должен быть создан с помощью команды `save`. Как правило, это все переменные рабочей области.
2. `load имя-файла X Y Z ...` считывает значения **переменных** `X`, `Y`, `Z` и т. д. из МАТ-файла. Вместо имени переменной может быть использовано обобщенное имя, например, `A*`.

Далее надо разобраться, как сохраняются файлы в формате `.mat` и других форматах.

Вывод в файл: функция save

save без параметров сохраняет все переменные рабочей области в файле matlab.mat текущей директории.

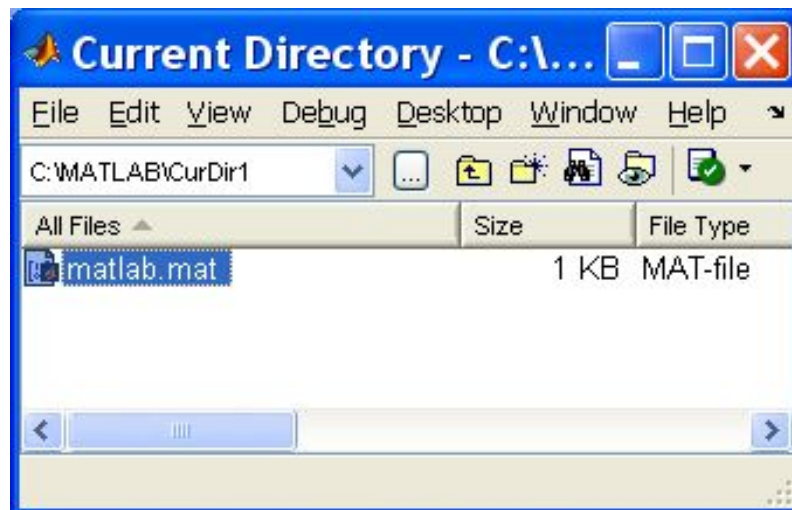
Пример:



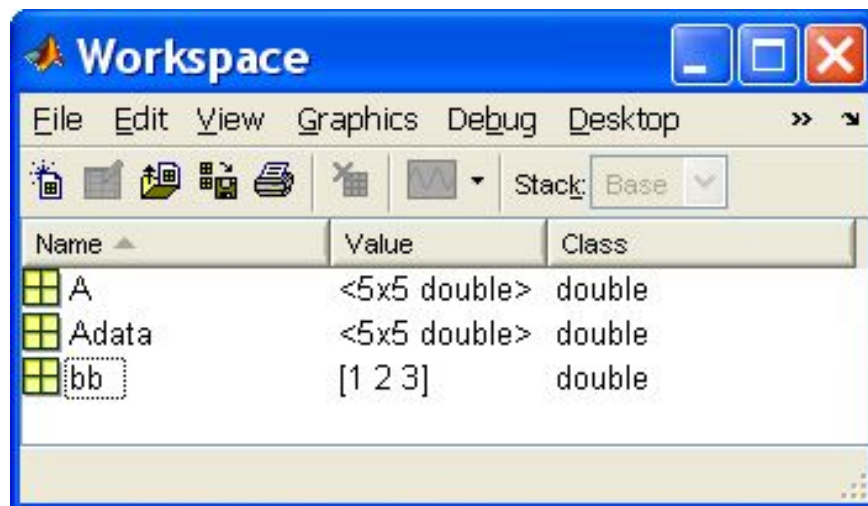
Далее файл matlab.mat можно перенести на другой компьютер или в другую папку этого компьютера и считать из него переменные.

Продолжение примера

Перенесли файл matlab.mat в другую папку, сделали ее текущей директорией:



и применили >>load



Вывод в файл: функция `save`

`save имя_файла` сохраняет все переменные рабочей области в файле `имя_файла.mat` текущей директории.

`save имя_файла имя_переменной1 имя_переменной2 ...` сохраняет указанные переменные в файле `имя_файла.mat` текущей директории.

Сохранение предполагает перезапись файла, предыдущее содержимое не сохраняется. Для добавления новых данных и сохранения их в других форматах (не `.mat`) следует указать дополнительные параметры команды `save`.

Некоторые значения параметров команды **save**

Параметр	Как параметр влияет на сохранение
-append	Добавление данных в конец файла. Возможно только для MAT-файлов
-ascii	Текстовый файл, 7 десятичных цифр после десятичной точки
-ascii -double	Текстовый файл, 16 десятичных цифр после десятичной точки
-ascii -tabs	Текстовый файл, 7 десятичных цифр после десятичной точки, числа в строке разделяются символом табуляции
-ascii -double -tabs	Текстовый файл, 16 десятичных цифр после десятичной точки, числа в строке разделяются символом табуляции
-mat	MAT-файл, значение параметра по умолчанию
-v4	Сохранение в MAT-формате версии 4
-v6	Сохранение в MAT-формате версии 6

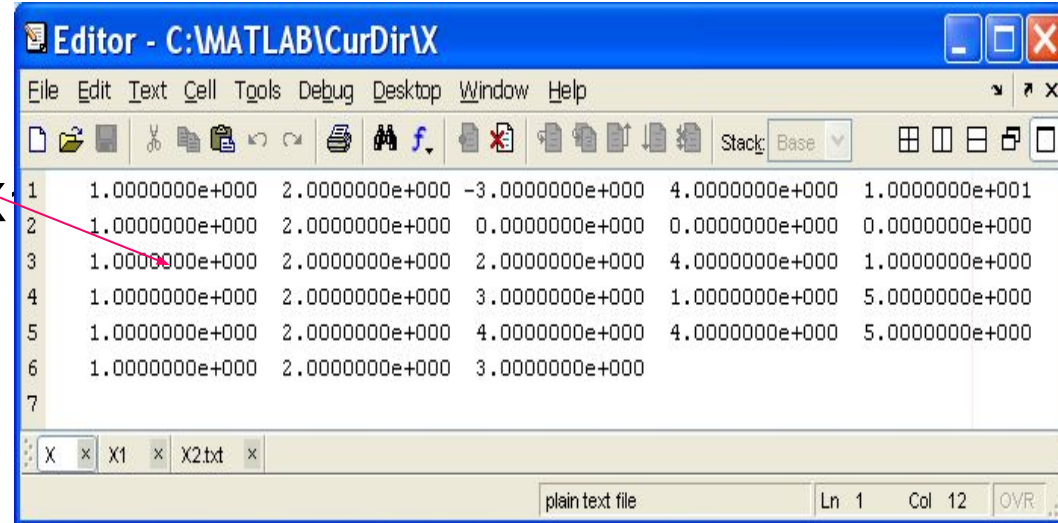
Примеры работы save

>> save X A bb %Файл .mat, в редакторе не посмотришь! **А как посмотреть ?**

>> save X A bb -ascii

>> save X1 A bb -ascii -double

>> save X2.txt A bb -ascii%Как X

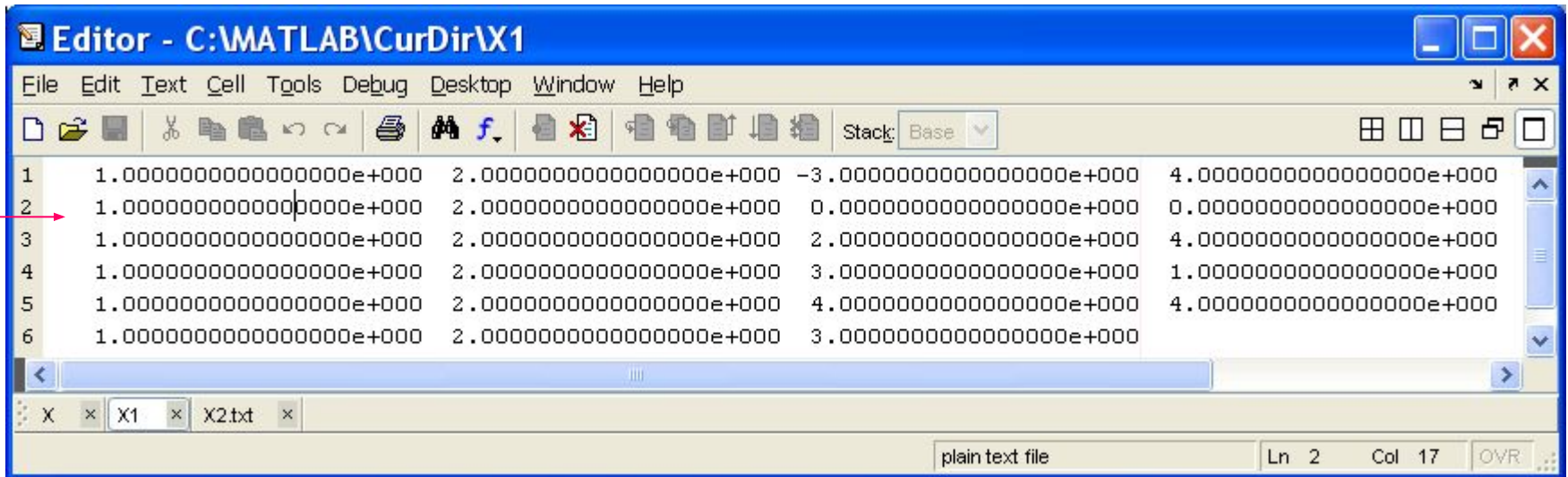


Editor - C:\MATLAB\CurDir\X

```
File Edit Text Cell Tools Debug Desktop Window Help
[Icons] Stack: Base [Layout Icons]
1 1.0000000e+000 2.0000000e+000 -3.0000000e+000 4.0000000e+000 1.0000000e+001
2 1.0000000e+000 2.0000000e+000 0.0000000e+000 0.0000000e+000 0.0000000e+000
3 1.0000000e+000 2.0000000e+000 2.0000000e+000 4.0000000e+000 1.0000000e+000
4 1.0000000e+000 2.0000000e+000 3.0000000e+000 1.0000000e+000 5.0000000e+000
5 1.0000000e+000 2.0000000e+000 4.0000000e+000 4.0000000e+000 5.0000000e+000
6 1.0000000e+000 2.0000000e+000 3.0000000e+000
7
```

X X1 X2.txt

plain text file Ln 1 Col 12 OVR



Editor - C:\MATLAB\CurDir\X1

```
File Edit Text Cell Tools Debug Desktop Window Help
[Icons] Stack: Base [Layout Icons]
1 1.0000000000000000e+000 2.0000000000000000e+000 -3.0000000000000000e+000 4.0000000000000000e+000
2 1.0000000000000000e+000 2.0000000000000000e+000 0.0000000000000000e+000 0.0000000000000000e+000
3 1.0000000000000000e+000 2.0000000000000000e+000 2.0000000000000000e+000 4.0000000000000000e+000
4 1.0000000000000000e+000 2.0000000000000000e+000 3.0000000000000000e+000 1.0000000000000000e+000
5 1.0000000000000000e+000 2.0000000000000000e+000 4.0000000000000000e+000 4.0000000000000000e+000
6 1.0000000000000000e+000 2.0000000000000000e+000 3.0000000000000000e+000
```

X X1 X2.txt

plain text file Ln 2 Col 17 OVR

Просмотр содержимого MAT-файла

`whos -file имя_файла.mat`

Пример:

```
>>whos -file matlab.mat
```

Name	Size	Bytes	Class
A	5x5	200	double array
Adata	5x5	200	double array
bb	1x3	24	double array

```
Grand total is 53 elements using 424 bytes
```

`whos` (без параметров) – выдает список в алфавитном порядке всех переменных рабочей области с указанием их размера и типа.

`whos global` - выдает список в алфавитном порядке всех глобальных переменных с указанием их размера и типа.

Сохранение всех переменных рабочей области в текстовом формате

`save -ascii имя_файла.расширение`

← любое, кроме *.mat*

Пример:

```
>> save -ascii work_space.txt
```

```
Editor - C:\MATLAB\CurDir\work_space.txt
File Edit Text Cell Tools Debug Desktop Window Help
[Toolbar icons] Stack: Base
1 1.0000000e+000 2.0000000e+000 -3.0000000e+000 4.0000000e+000 1.0000000e+001
2 1.0000000e+000 2.0000000e+000 0.0000000e+000 0.0000000e+000 0.0000000e+000
3 1.0000000e+000 2.0000000e+000 2.0000000e+000 4.0000000e+000 1.0000000e+000
4 1.0000000e+000 2.0000000e+000 3.0000000e+000 1.0000000e+000 5.0000000e+000
5 1.0000000e+000 2.0000000e+000 4.0000000e+000 4.0000000e+000 5.0000000e+000
6 0.0000000e+000 0.0000000e+000 0.0000000e+000 0.0000000e+000 0.0000000e+000
7 1.0000000e+000 2.0000000e+000 0.0000000e+000 0.0000000e+000 0.0000000e+000
8 1.0000000e+000 2.0000000e+000 2.0000000e+000 4.0000000e+000 1.0000000e+000
9 1.0000000e+000 2.0000000e+000 3.0000000e+000 1.0000000e+000 5.0000000e+000
10 1.0000000e+000 2.0000000e+000 4.0000000e+000 4.0000000e+000 5.0000000e+000
11 1.0000000e+000 2.0000000e+000 3.0000000e+000
12
```

Name	Value	Class
A	<5x5 double>	double
Adata	<5x5 double>	double
bb	[1 2 3]	double

Ввод из файла: функция `dlmread`

`dlm` – от англ. *delimiter* - разделитель

`M = dlmread('filename', delimiter)` – считывает числа **из текстового файла**, разделенные указанным разделителем, в матрицу `M`. Если разделитель не указывается, то используется запятая. У функции также могут быть параметры, определяющие границы файла, откуда следует считывать информацию.

Примеры: `A=dlmread('ZZ.txt', ';')`
`B=dlmread('Z.txt', '/t')`

Вывод в файл: функция `dlmwrite`

`dlmwrite('filename', M, delimiter)` – записывает в текстовый файл значения элементов матрицы `M`. Могут быть дополнительные параметры (границы файла, форматы чисел).

Примеры:

`dlmwrite('myfile.txt', M, '\t')`

`dlmwrite('myfile.txt', M, 'delimiter', '\t', 'precision', 6)`

Низкоуровневое программирование работы с файлами – как в алгоритмических языках (Си)

fclose
feof
ferror
fgetl
fgets
fopen
fprintf
fread
frewind
fscanf
fseek
ftell
fwrite

Низкоуровневое программирование:

1. Требуется специальных навыков и знаний.
2. Предоставляет большие возможности ввода-вывода, в том числе большие возможности форматирования вывода.
3. Требуется трудозатрат.

Низкоуровневое программирование рассматривает файл как устройство последовательного доступа.

В среде MATLAB еще много функций для работы с файлами!!!

Работа с файлами других приложений

Функции:

importdata – загружает (считывает) данные из файлов различных типов.

open – открывает файлы различных типов, используя соответствующие редакторы или приложения (к сожалению, круг допустимых приложений ограничен).

uiimport – открывает окно Мастера импорта данных для импортирования данных.

winopen – открывает соответствующее Windows-приложение.

Импорт Excel-файла с помощью функции importdata

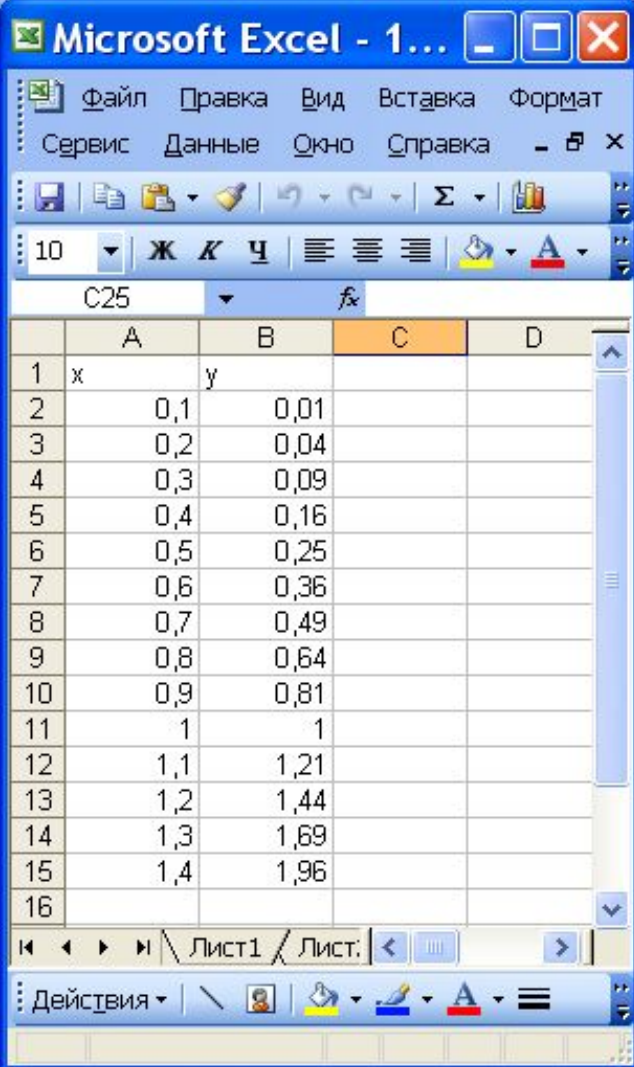
Исходный файл 1.xls

```
>> Q=importdata('1.xls')
```

Q =

```
data: [1x1 struct]  
textdata: [1x1 struct]  
colheaders: [1x1 struct]
```

↓
Надо уметь работать со структурами!



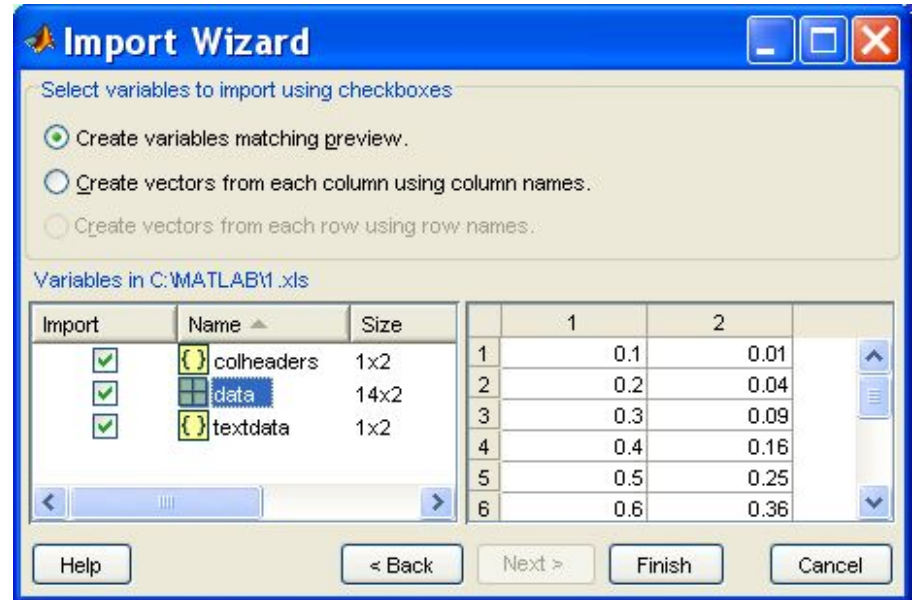
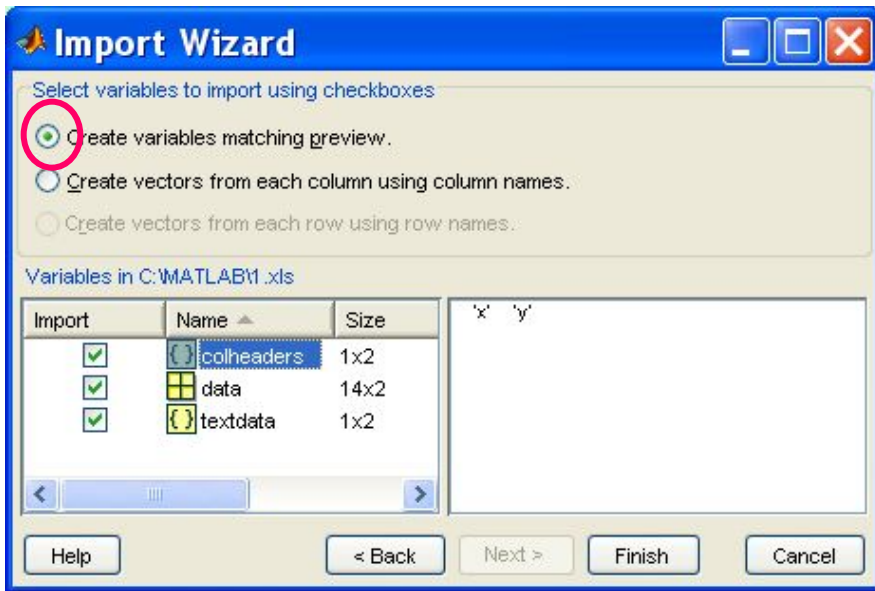
The screenshot shows a Microsoft Excel window titled "Microsoft Excel - 1...". The spreadsheet contains the following data:

	A	B	C	D
1	x	y		
2	0,1	0,01		
3	0,2	0,04		
4	0,3	0,09		
5	0,4	0,16		
6	0,5	0,25		
7	0,6	0,36		
8	0,7	0,49		
9	0,8	0,64		
10	0,9	0,81		
11	1	1		
12	1,1	1,21		
13	1,2	1,44		
14	1,3	1,69		
15	1,4	1,96		
16				

Импорт Excel-файла с помощью Мастера импорта данных

```
>> open('1.xls')
```

```
>> uiimport('1.xls')
```



```
>> data
```

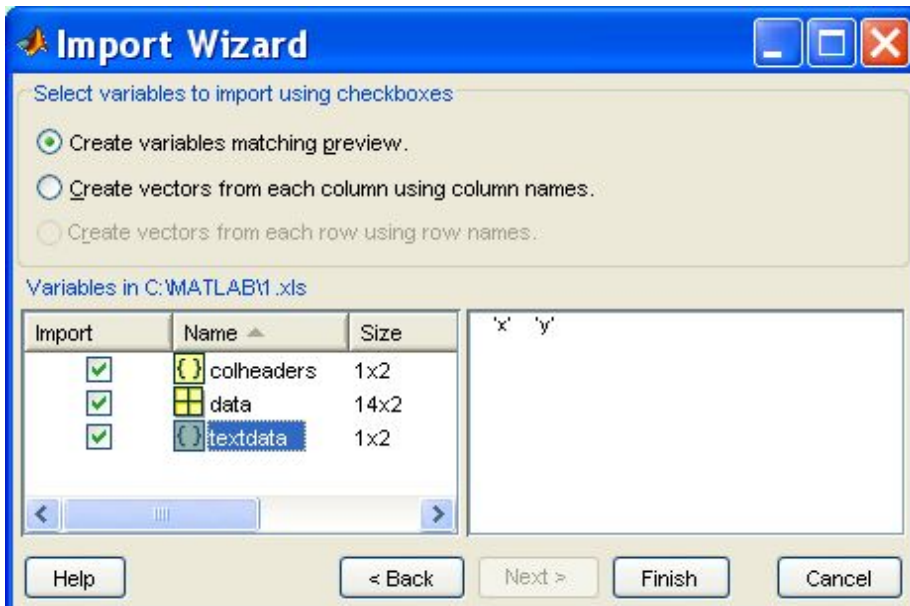
```
data =
```

```
0.1000 0.0100  
0.2000 0.0400  
0.3000 0.0900  
0.4000 0.1600  
0.5000 0.2500  
0.6000 0.3600  
0.7000 0.4900  
0.8000 0.6400  
0.9000 0.8100  
1.0000 1.0000  
1.1000 1.2100  
1.2000 1.4400  
1.3000 1.6900  
1.4000 1.9600
```

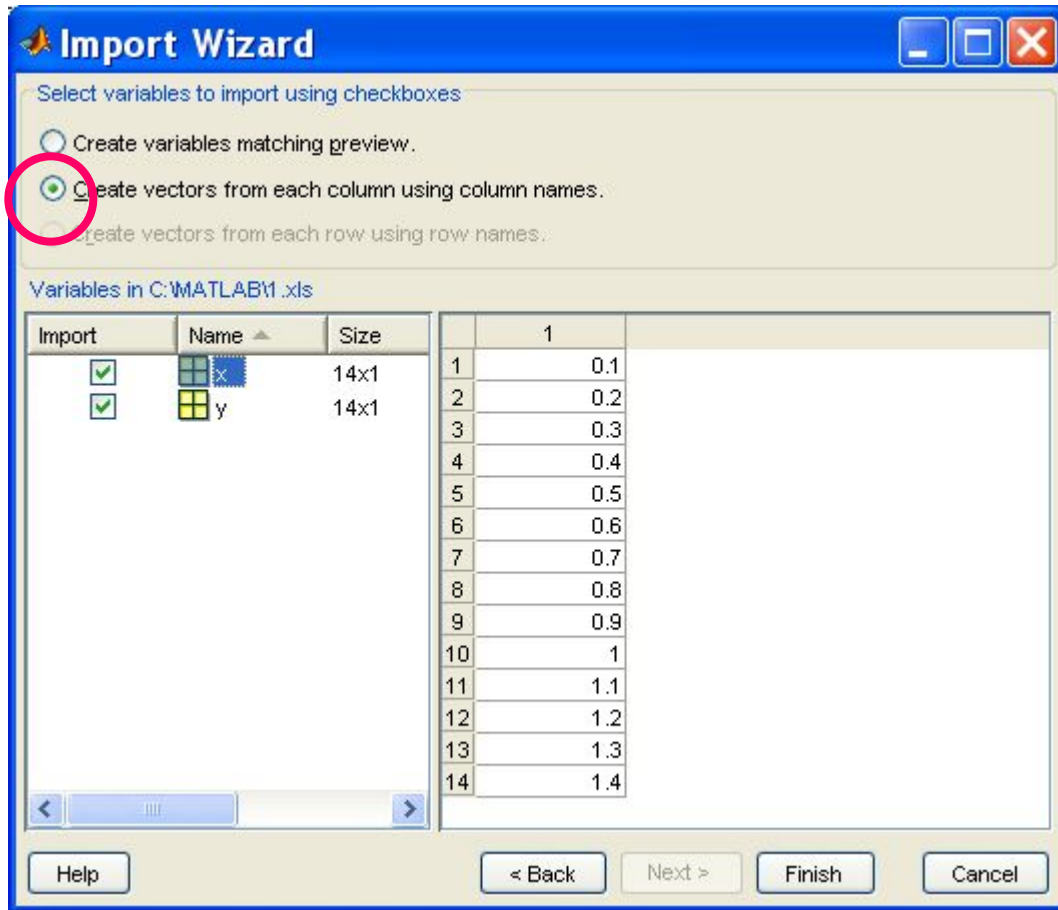
```
>> colheaders
```

```
colheaders =
```

```
'x' 'y'
```



Импорт Excel-файла с помощью Мастера импорта данных



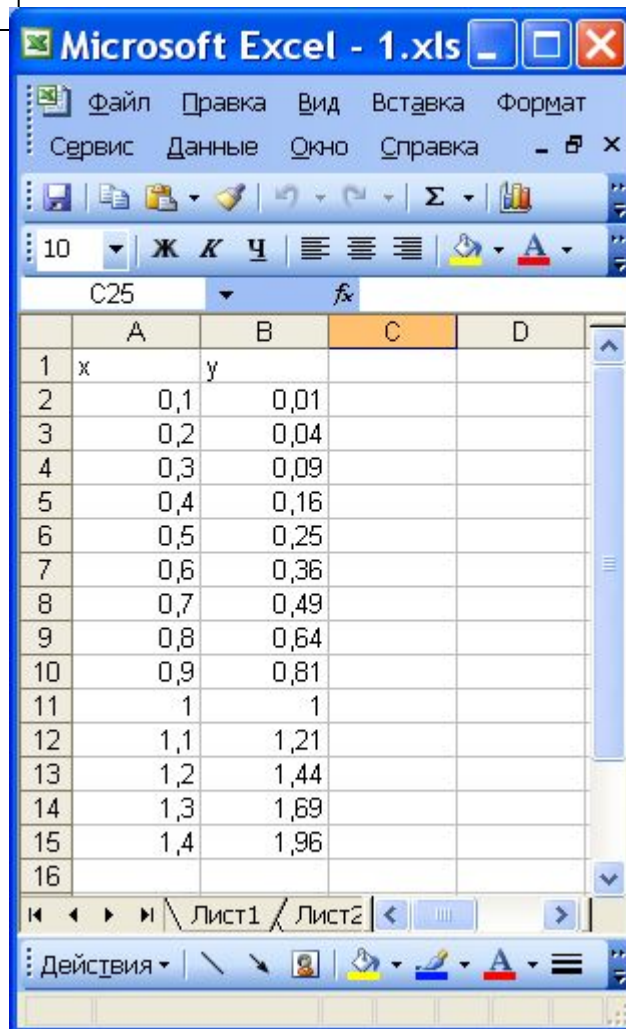
```
>> disp(x)
0.1000
0.2000
0.3000
0.4000
0.5000
0.6000
0.7000
0.8000
0.9000
1.0000
1.1000
1.2000
1.3000
1.4000
```

```
>> disp(y)
0.0100
0.0400
0.0900
0.1600
0.2500
0.3600
0.4900
0.6400
0.8100
1.0000
1.2100
1.4400
1.6900
1.9600
```

Мастер импорта данных можно вызвать также с помощью пункта Import Data главного меню MATLAB.

Работа с файлом Excel из среды MATLAB

```
>> winopen('1.xls')
```



Microsoft Excel - 1.xls

Файл Правка Вид Вставка Формат
Сервис Данные Окно Справка

10 Ж К Ч

C25

	A	B	C	D
1	x	y		
2	0,1	0,01		
3	0,2	0,04		
4	0,3	0,09		
5	0,4	0,16		
6	0,5	0,25		
7	0,6	0,36		
8	0,7	0,49		
9	0,8	0,64		
10	0,9	0,81		
11	1	1		
12	1,1	1,21		
13	1,2	1,44		
14	1,3	1,69		
15	1,4	1,96		
16				

Лист1 / Лист2

Действия

Импорт файла .html- с помощью функции `importdata`

```
>> Q1=importdata('1.html')
```

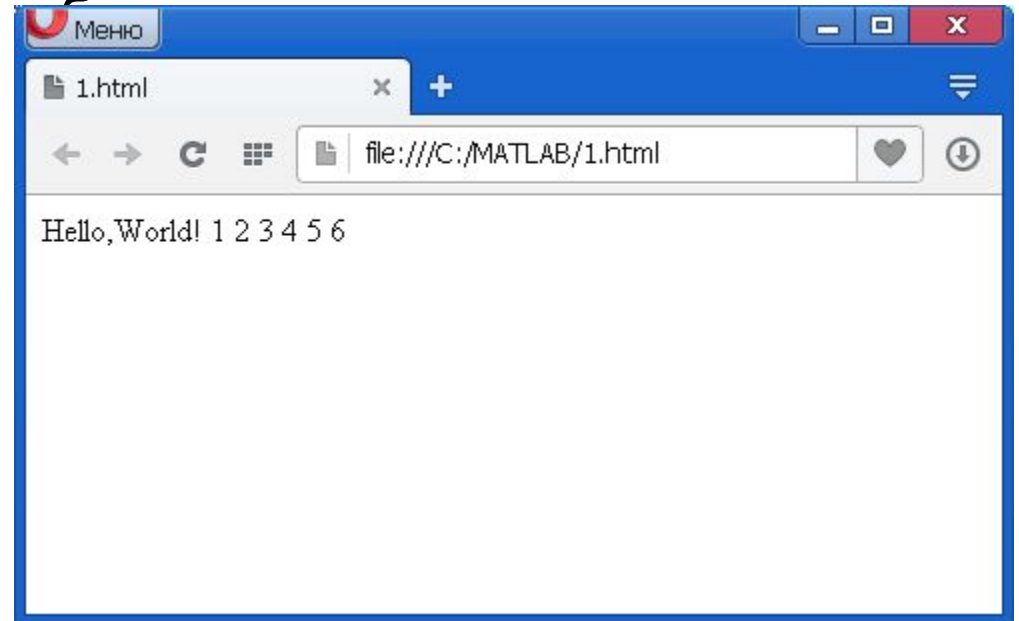
Q1 =

```
data: [1 2 3 4 5 6]
textdata: {'Hello,World!'}
rowheaders: {'Hello,World!'}
```

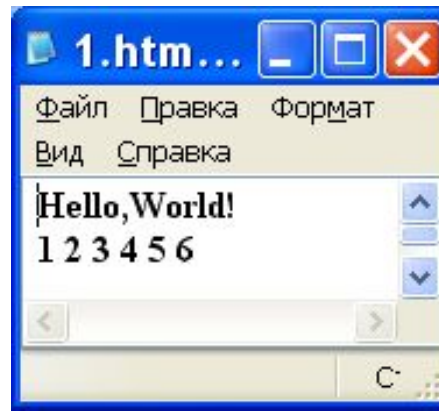


Надо уметь работать со структурами!

Исходный файл 1.html в

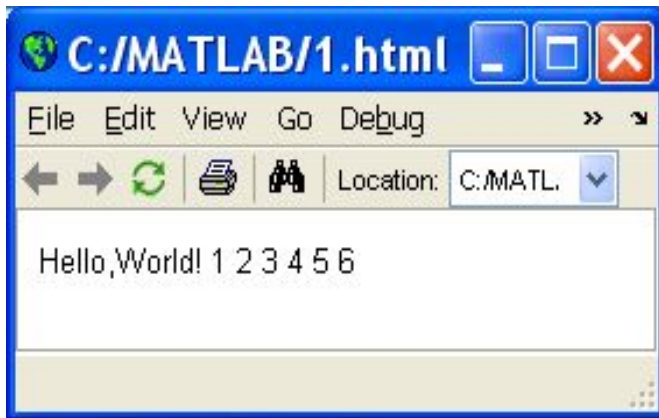


Исходный файл 1.html в блокноте

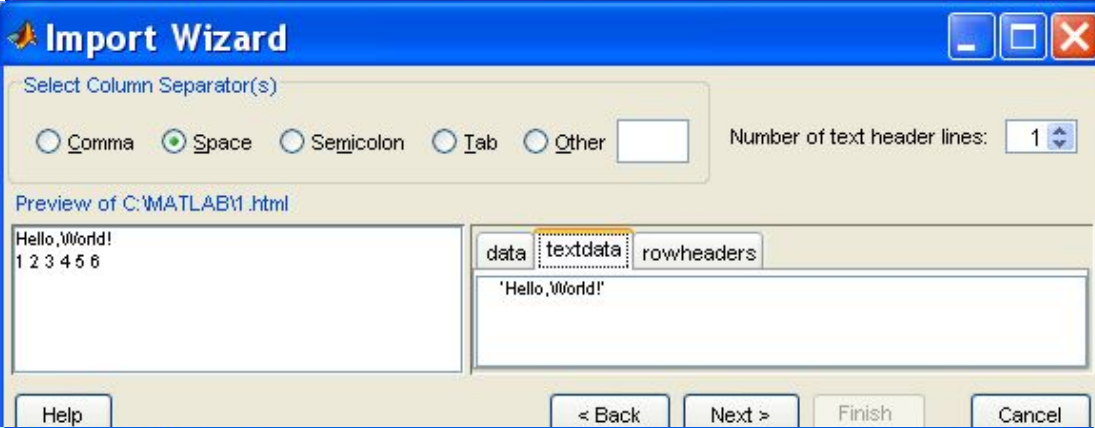
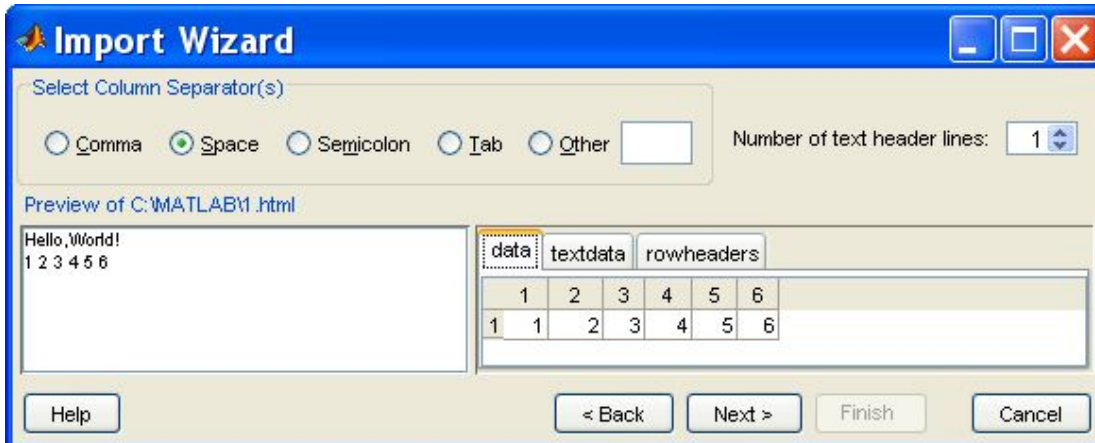


Открытие файла .html с помощью функции open

```
>>open('1.html')
```



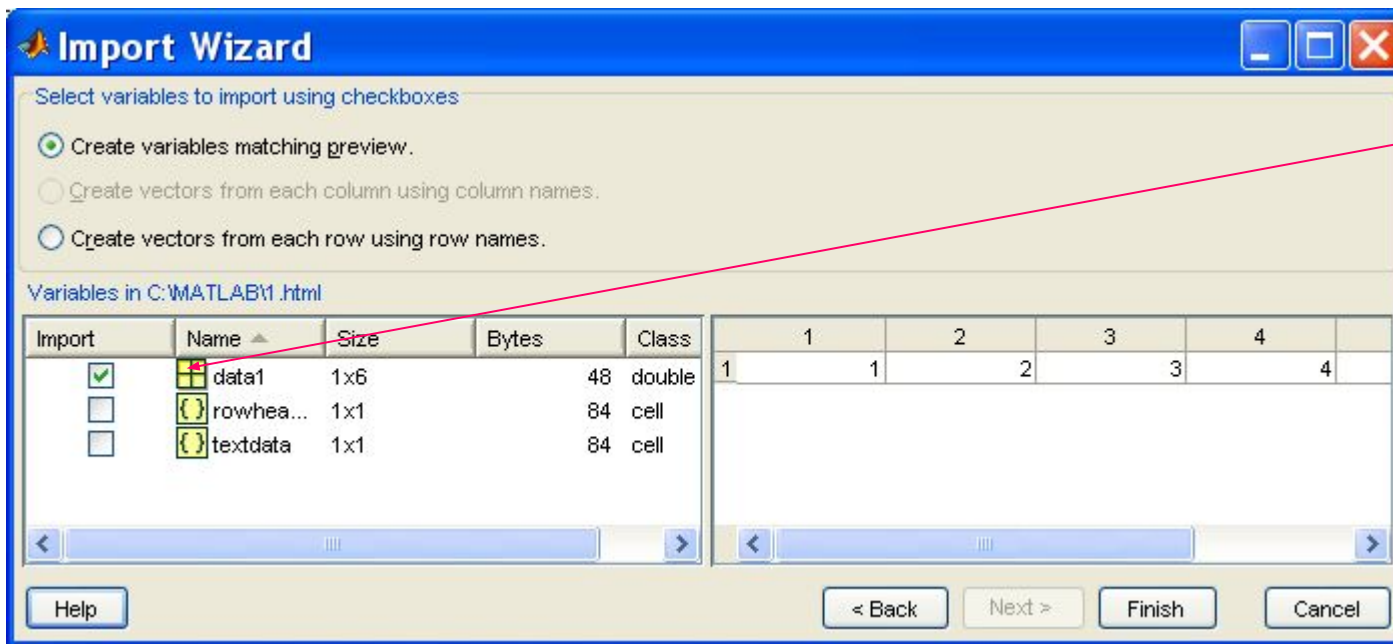
В соответствии с расширением файла
выбрано нужное приложение



>>uimport('1.html')

Функция **uimport**
вызывает Мастера
импорта данных

Продолжение работы Мастера импорта данных



```
>> data1
```

```
data1 =
```

```
1 2 3 4 5 6
```