==Script==

Welcome to the spoken tutorial on Getting Started with Scilab. We will learn some of the basic uses of Scilab in this tutorial.

To practice this tutorial, you need to have Scilab installed on your system.

Double click on the Scilab shortcut icon on your Desktop or select it from the start menu. This will open the Scilab console window. Notice that the cursor is on the command prompt.

I have a notepad file open with all the commands i will be using in this tutorial.I will cut and paste this commands in the console rather than typing then to save time.

Scilab can be used as a calculator. On entering 42+4^2-64/4, gives ans = 42

```
-->42+4^2-64/4

 ans = 42.
```

Note that the answer 42 is stored in the default variable answer, abbreviated as ans. From now on, we will show the command and the resulting answer.

In the above calculation, we stored the result in the default variable ans. We can also create variables for this purpose:

```
-->a=1, b=2, c=3

 a = 1.

 b = 2.

 c = 3.
```

a=1,b=2,c=3 gives 1,2,3 values stored in variables a,b,c respectively.

Now we will perform some mathematical operations using these variables. For example,

```
-->a+b+c gives
  ans = 6. also
-->a*(b+c) gives
  ans = 5.
```

We can also direct the answer into another variable say 'd' by typing

```
-->d = a+b+c
```

```
 d = 6.
```

We can check the values in the variables by typing the names of the variables separated by commas on the command line as

```
 -->a,b,c,d gives
 a = 1.
 b = 2.
 c = 3. and
 d = 6.
```

We will briefly interrupt the flow here and introduce an important command to remember previously issued commands and the resulting output. Let us first type the command cd:

```
 -->cd
 ans  =
 /Users/guest
```

This is the present working directory. Now issue the diary command by typing:

```
 -->diary('myrecord.txt') press enter
```

This command will create a file with the name "myrecord.txt" in the current working directory. A transcript of the current Scilab session will be saved in this file. Let us demonstrate its usefulness by typing some more commands using Scilab.

Now lets see how Scilab handles complex numbers. Recall that complex numbers are denoted as i. Let us first see what value is assigned to i:

```
 -->i
   !--error 4
 Undefined variable: i
```

It gives an error

In Scilab, predefined numerical constants are accessed by prefixing them with a percentage sign, as we see now:

```
 -->%i
 %i  =
   i
```

We can verify that this i is indeed a complex number by taking its square:

```
 -->%i * %i
   ans  =
   - 1.
```

The answer is -1

Finally, by taking the square root of -1, we get answer = i

```
-->sqrt(-1)
ans =
i
```

Let us now do some calculations using i:

```
-->5*%i gives
ans =
5.i

-->(10+5*%i)*(2*%i) gives us
ans =
- 10. + 20.i
```

Let us see some more predefined numerical constants available in Scilab. As in the case of i, their names also start with the % sign:

```
-->%pi
%pi = 3.1415927
```

The value pf pi is as shown.

We will demonstrate the use of pi using a few built in trigonometric functions as follows.

```
-->sin(%pi/2),and cos(%pi/2) enter
ans = 1.
ans = 6.123D-17
```

Notice that the second answer is zero for all practical purposes. This is related to a number known as "machine epsilon", which will be covered in another spoken tutorial.

The base of the natural logarithm is another predefined numerical constant:

```
-->%e
%e = 2.7182818
```

The value of e is as shown.

We can achieve the same result with the exponential function, exp of 1:

```
-->exp(1)
ans =
```

2.7182818

you can see both the answers are same

Of course,

-->exp(2) gives the following answer
ans =
7.3890561

which can also be achieved by %e square

-->%e^2
ans =
7.3890561

Recall that we invoked a recording of transactions into the file myrecord.txt through the diary command. Let us now close this diary file as follows:

-->diary(0) diary of zero

This command will close and save the file myrecord.txt. Also recall that this file was created in current working directory Let us open and browse this file.
C /Users/anuradha/myrecord.txt
Note that all transactions, both commands and the corresponding answers given by Scilab, have been saved into this file.

While a program is being developed, one experiments a lot before arriving at a suitable code. On an average, about 5% of the attempted code is useful. If the diary command is not used, it is difficult to keep a track of all the commands that did or didn't work. Thus, the diary is the best way to address this issue. It is possible to edit the file created by using the diary command and delete all the unwanted lines to produce a working code.

No further transactions will be saved because we had closed the file using the command diary(0). If we need to save the session once again, we need to issue the diary command again. Remember, that the diary command will overwrite the file. So, if the file contains some useful information, then one should use some other file name in the diary command.

Let me close the file, the browser
Scroll down the note pad file

It is extremely easy to create matrices in Scilab. Let us define a 3x3 matrix as follows:

-->A = [1 2 3;4 5/2 6;7 8+5*%i 0]

```
 A  =
  1.    2.          3.
  4.    2.5         6.
  7.    8. + 5.i    0
```

Note that a semicolon is used to terminate the row.
The beauty of Scilab is that we do not have to specify the dimensions of matrix A nor define its type.

One can also enter matrices row by row as follows:

```
 -->C = [1 2 enter
 -->3 4] press enter
  C  =
    1.    2.
    3.    4.
```

Semicolon is used while creating matrices as above. It serves another important purpose, not echoing the results of a calculation. For example:

```
 -->B = 1+2; enter
```

We see that the result of B is not echoed. This is because of the semicolon. We can see the result of B by typing it explicitly as

```
 -->B enter
  B  =

     3.
```

Note that this use of semicolon is extremely popular as it can be used to suppress voluminous outputs.

We postpone the discussion on matrix calculations to another spoken tutorial.

At some point, if you need help, then the 'help' command from the Scilab menu bar can be used.

Let me close the help browser.

The clc command clears the screen of Scilab as shown below:

```
 -->clc
```

The up - down arrow keys can been used to see the previously executed commands. While using the up - down arrows, you can stop at any command,and press the return key to execute it.

You can edit the commands if necessary.

Use the tab key to auto-completes the command and gives us all the available options to choose from.

This brings us to the end of this spoken tutorial on Getting Started with Scilab. There are many other functions in Scilab which will be covered in other spoken tutorials. Keep watching the Scilab links.

Spoken Tutorials on Scilab are part of the Talk to a Teacher project, supported by the National Mission on Education through ICT. More information on the same is available at the following link
http://spoken-tutorial.org/NMEICT-Intro.

Thanks for joining. Goodbye.