Scilab: Programming Quick Reference

Variable Names

- Case sensitive
- Cannot contain spaces
- May begin with: letters, %, –, #, $, ?
- May not begin with number
- 24 character maximum
- Will overwrite functions with same name, including predefined

Comments

```
// This is a one line comment

/* This is a
   multiline comment */
```

Commands

`clf`	Clear graph window
`clear`	Clear all variables
`clear x`	Clear variable x
`clearglobal`	Clear global variables
`who`	List of Scilab variables
`who_user`	List of user variables
`exit`	End Scilab session
`abort`	Stop computation
`pause`	Pause computation, ask for input

Output

`disp(x)`	Display string or value x
`string(x)`	Converts value x to string
`disp(s,t)`	Display strings as list
`disp(s + t)`	Display concatenation of strings

Testing

`x == y`	Returns true if $x = y$
`x ~= y`	Returns true if $x \neq y$
`x < y`	Returns true if $x < y$
`x > y`	Returns true if $x > y$
`x <= y`	Returns true if $x \leq y$
`x >= y`	Returns true if $x \geq y$
`isdef(x)`	Returns true if variable x is defined
`isempty(x)`	Returns true if x is empty matrix or list
`isequal(x,y)`	Returns true if $x = y$
`isvector(x)`	Returns true if x is vector
`isreal(x)`	Returns true if x is real number
`isinf(x)`	Returns true if x is infinite
`isnan(x)`	Returns true if x is “Not A Number”

Variable Types

```
type(x)      // returns object type of variable
typeof(x)    // returns object type as string
```

constant	polynomial	boolean	sparse
int8	int16	int32	boolean sparse
handle	string	function	library
list	tlist	st	mlist
pointer	size implicit	fptr	

Logic

`A & B`	And; true if both A and B are true
`A | B`	Or; true if either A or B is true

Functions

```
--> function functionName( x )
      // function contents
    endfunction
```

```
--> function y = functionName( x )
      // function contents, returned
      // to variable y
    endfunction
```

If-Then-Else

```
--> if ( condition ) then
      // block to execute if condition is true
    end
```

```
--> if ( condition ) then
      // block to execute if condition is true
    else
      // block to execute if condition is false
    end
```

```
--> if ( condition1 ) then
      // block to execute if condition1 is true
    elseif ( condition2 ) then
      // block to execute if condition1 is false
      // and condition2 is true
    else
      // block to execute if condition1 and
      // condition2 are both false
    end
```

Select, Cases

```
--> select x
    case (possible value of x) then
    // commands to execute in this case
    case (another possible value of x) then
    // commands to execute in this case
    else
    // commands to execute if all cases fail
    end
```

For

```
--> for i = s:e
      // block of commands to execute based on
      // value of i, with i ranging from s to e
    end
```

While

```
--> while ( condition )
      // block of commands to execute if
      // condition is true
    end
```

Data Structures

`t=struct('field1', val1, 'field2', val2)`	Defines structure
`fieldnames(t)`	List of field names
`t.field1`	Returns val1

```
--> Damien = struct('breed', 'papillon', 'age', 12,
    'toys', ['elephant','raccoon']);

--> Damien.breed
 ans  =
 papillon

--> Damien.toys
 ans  =
!elephant  raccoon  !
```

Updated January 27, 2019
Dr. Lauren Williams
Mercyhurst University
math.mercyhurst.edu
Department of Mathematics