

Floating-point numbers - Matlab related

Ágnes Baran, Csaba Noszály

On-line documentation:

- Octave at www.gnu.org
- Matlab at www.mathworks.com

You can write expressions, statements in the command window:

```
>> 1+1
```

```
ans =
```

```
2
```

```
>> 2*2
```

```
ans =
```

```
4
```

The result will get into the variable named `ans`, unless we used assignment.

We can define our own variables:

```
>> a=2*3  
  
a =  
  
    6  
  
>> b=3; c=a+b;
```

As you see, writing a semicolon after the expression, will turn off echoing the result - although the evaluation will be done. The value of a variable can be accessed by typing its name:

```
>> c  
  
c =  
  
    9
```

Variable names

- For the name of a variable one can use a sequence of characters that begins with a letter (of the english alphabet), and consists only of letters, digits and underscores. It is case sensitive!
- It is forbidden (and impossible) to use for naming variables the so called keywords: `if, for, while, function, ...`. For a full list keywords type `iskeyword`.
- It is strongly discouraged (but possible) to use the names of the so called built-in's: `size, sin, cos, exp, ...`
- You can query the system about the existence of a particular name: `exist cos`
- You can destroy a variable with: `clear yourVariableName` You can destroy all variables in the workspace with: `clear all`
- for further details, see `Variable names`.

Relational operators

The result of a comparison is a logical `1` (=true) or logical `0` (=false).

- `a < b` is true iff. a is less than b
- `a <= b` is true iff. a is less than or equal to b
- `a > b` is true iff. a is greater than b
- `a >= b` is true iff. a is greater than or equal to b
- `a == b` is true iff. a is equal to b
- `a ~= b` is true iff. a is not equal to b

For matrices of the same size the comparison is performed elementwise, i.e. comparing elements in the same location. The result is a logical matrix of the same shape.

script: A series of commands that you write into a file. It can be executed as a complete unit.

- Open in the editor window a new script and write our program here.

Comments: Everything is ignored (not parsed,executed) after the **%** sign.

- Note that each of the statements are executed as it were typed in the command window, so without using **;** the results are printed on the screen.
- Save the file.
- Execute the script: either by pressing the **run** button in the top of editor window, or switching back to the command window by typing the **name** of script (without the .m extension)

The for-loop

```
for variable = vector  
    statements  
end
```

```
s=0;  
for k=1:100  
    s=s+k;  
end
```

```
s=0;  
for k=[ 1 3 -2 5 ]  
    s=s+1/k;  
end
```

```
s=0;  
for k=100:-3:1  
    s=s+k^2  
end
```


The while-loop

```
while logical-expression  
    statements  
end
```

```
s=0; k=1;  
while k<=100  
    s=s+k; k=k+1;  
end
```

```
s=1; k=10;  
while k>1  
    s=s*k; k=k-1;  
end
```

```
s=0; k=100;  
while k>=1  
    s=s+k^2; k=k-3;  
end
```