

---

# Chapter 02 - Basic Commands

## Table of Contents

A Few Basic Commands .....	1
Help! .....	2
Play! .....	2
Precision and Other Display Stuff .....	2

## A Few Basic Commands

Matlab has hundreds of commands for doing various thing. Just to show a few, we can find the cosine of an angle in radians:

```
cos(20)
```

```
ans =
```

```
0.4081
```

We can find the sine of an angle in degrees:

```
sind(82.4)
```

```
ans =
```

```
0.9912
```

We can find the natural logarithm of a value, noting that `log` gives natural logarithm:

```
log(10)
```

```
ans =
```

```
2.3026
```

And the exponential:

```
exp(1.3)
```

```
ans =
```

```
3.6693
```

## Help!

One of the most useful commands is the `help` command which gives you help on another command. It also gives you information on other commands you should check out which are related:

```
help tan
```

```
TAN      Tangent of argument in radians.  
TAN(X) is the tangent of the elements of X.
```

```
See also ATAN, ATAN2, TAND, ATAN2D.
```

```
Reference page in Doc Center  
doc tan
```

```
Other functions named tan
```

```
codistributed/tan    gpuArray/tan    sym/tan
```

## Play!

As an exercise, can you figure out what the following commands do? Try them all!

```
pi  
cscd(200)  
factorial(5)  
round(pi*6)  
primes(40)  
factor(600)  
gcd(120,90)  
date
```

## Precision and Other Display Stuff

Whenever you are performing mathematical computations on a computer, you must realize that some rounding of values is necessarily involved. A computer stores each value in a relatively small *finite* region of memory. How could you possibly store a value like  $\pi$ , which has an infinite non-repeating decimal expansion, in a small space? You can't. So computers will necessarily round messy numbers. Some programming languages round more than others, but calculations in Matlab are handled fairly precisely. What we mean is that each numerical value is stored using a lot of digits. For example, calculations involving the constant `pi` use the value 3.141592653589793 instead of 3.14 or 3.1416.

However, the internal precision of values in Matlab will not always get displayed when Matlab shows you an answer. Try the following:

```
pi  
  
ans =  
  
3.1416
```

This answer might seem like a disappointing level of precision. The good news is that the internal value really is stored very precisely, but Matlab assumes that most people just don't want to see all of those ugly digits, so the value is rounded before it is displayed. If you really want to see all of the digits in your answers to computations, try the following:

```
format long
pi

ans =

    3.141592653589793
```

Aha! There are those digits. If you want to go back to showing fewer digits in Matlab's answers, just use:

```
format short
```

Don't forget that values used in computations are always represented internally using full precision, regardless of whether you specify the long or short format. The format command only affects the way the answer is displayed on the screen.

```
format compact
```

This command suppresses extra line feeds and keeps your Matlab output, from then on, more compact and with no extra spaces between lines. I almost always turn this on since I like to see lots of stuff at once. If you want to return it to normal use

```
format loose
```

*Published with MATLAB® R2018a*