function

Declare M−file function

Syntax

```
function [out1, out2, ...] = funname(in1, in2, ...)
```

Description

`function [out1, out2, ...] = funname(in1, in2, ...)` defines function `funname` that accepts inputs `in1`, `in2`, etc. and returns outputs `out1`, `out2`, etc.

You add new functions to the MATLAB vocabulary by expressing them in terms of existing functions. The existing commands and functions that compose the new function reside in a text file called an *M−file*.

M−files can be either *scripts* or *functions*. Scripts are simply files containing a sequence of MATLAB statements. Functions make use of their own local variables and accept input arguments.

The name of an M−file begins with an alphabetic character and has a filename extension of `.m`. The M−file name, less its extension, is what MATLAB searches for when you try to use the script or function.

A line at the top of a function M−file contains the syntax definition. The name of a function, as defined in the first line of the M−file, should be the same as the name of the file without the `.m` extension.

The variables within the body of the function are all local variables.

A *subfunction*,visible only to the other functions in the same file, is created by defining a new function with the `function` keyword after the body of the preceding function or subfunction. Subfunctions are not visible outside the file where they are defined.

You can terminate any function with an `end` statement but, in most cases, this is optional. `end` statements are required only in M−files that employ one or more nested functions. Within such an M−file, *every* function (including primary, nested, private, and subfunctions) must be terminated with an `end` statement. You can terminate any function type with `end`, but doing so is not required unless the M−file contains a nested function.

Functions normally return when the end of the function is reached. Use a `return` statement to force an early return.

When MATLAB does not recognize a function by name, it searches for a file of the same name on disk. If the function is found, MATLAB compiles it into memory for subsequent use. The section Determining Which Function Is Called in the MATLAB Programming documentation explains how MATLAB interprets variable and function names that you enter, and also covers the precedence used in function dispatching.

When you call an M−file function from the command line or from within another

M−file, MATLAB parses the function and stores it in memory. The parsed function remains in memory until cleared with the `clear` command or you `quit` MATLAB. The `pcode` command performs the parsing step and stores the result on the disk as a P−file to be loaded later.

Examples

Example 1

The existence of a file on disk called `stat.m` containing this code defines a new function called `stat` that calculates the mean and standard deviation of a vector:

```
function [mean,stdev] = stat(x)
n = length(x);
mean = sum(x)/n;
stdev = sqrt(sum((x−mean).^2/n));
```

Example 2

`avg` is a subfunction within the file `stat.m`:

```
function [mean,stdev] = stat(x)
n = length(x);
mean = avg(x,n);
stdev = sqrt(sum((x−avg(x,n)).^2)/n);

function mean = avg(x,n)
```

mean = sum(x)/n;

See Also

`nargin`, `nargout`, `pcode`, `varargin`, `varargout`, `what`