

### 7.7 COMPARISON BETWEEN SCRIPT FILES AND FUNCTION FILES

Students who are studying MATLAB for the first time sometimes have difficulty understanding exactly the differences between script and function files, since for many of the problems that they are asked to solve using MATLAB, either type of file can be used. The similarities and differences between script and function files are summarized below.

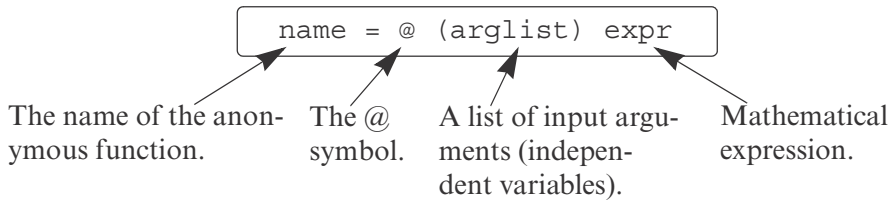
- Both script and function files are saved with the extension .m (that is why they are sometimes called M-files).
- The first executable line in a function file is (must be) the function definition line.
- The variables in a function file are local. The variables in a script file are recognized in the Command Window.
- Script files can use variables that have been defined in the workspace.
- Script files contain a sequence of MATLAB commands (statements).
- Function files can accept data through input arguments and can return data through output arguments.
- When a function file is saved, the name of the file should be the same as the name of the function.
- A user-defined function is used in the same way as a built-in function. It can be used (called) in the Command Window, in a script file, or in another function.

### 7.8 ANONYMOUS FUNCTIONS

User-defined functions written in function files can be used for simple mathematical functions, for large and complicated math functions that require extensive programming, and as subprograms in large computer programs. In cases when the value of a relatively simple mathematical expression has to be calculated many times within a program, MATLAB provides the option of using anonymous functions. An anonymous function is a user-defined function that is defined and written within the computer code (not in a separate function file) and is then used in the code. Anonymous functions can be defined in any part of MATLAB (in the Command Window, in script files, and inside regular user-defined functions).

An anonymous function is a simple (one-line) user-defined function that is defined without creating a separate function file (m-file). Anonymous functions can be constructed in the Command Window, within a script file, or inside a regular user-defined function.

An anonymous function is created by typing the following command:



A simple example is `cube = @ (x) x^3`, which calculates the cube of the input argument.

- The command creates the anonymous function and assigns a handle for the function to the variable name on the left-hand side of the = sign. (Function handles provide means for using the function and passing it to other functions; see Section 7.9.1.)
- The `expr` consists of a single valid mathematical MATLAB expression.
- The mathematical expression can have one or several independent variables. The independent variable(s) is (are) entered in the `(arglist)`. Multiple independent variables are separated with commas. An example of an anonymous function that has two independent variables is: `circle = @ (x,y) 16*x^2+9*y^2`.
- The mathematical expression can include any built-in or user-defined functions.
- The expression must be written according to the dimensions of the arguments (element-by-element or linear algebra calculations).
- The expression can include variables that are already defined when the anonymous function is defined. For example, if three variables `a`, `b`, and `c` are defined (have assigned numerical values), then they can be used in the expression of the anonymous function `parabola = @ (x) a*x^2+b*x+c`.

**Important note:** MATLAB captures the values of the predefined variables when the anonymous function is defined. This means that if new values are subsequently assigned to the predefined variables, the anonymous function is not changed. The anonymous function has to be redefined in order for the new values of the predefined variables to be used in the expression.

#### Using an anonymous function:

- Once an anonymous function is defined, it can be used by typing its name and a value for the argument (or arguments) in parentheses (see examples that follow).
- Anonymous functions can also be used as arguments in other functions (see Section 7.9.1).

**Example of an anonymous function with one independent variable:**

The function  $f(x) = \frac{e^{x^2}}{\sqrt{x^2+5}}$  can be defined (in the Command Window) as an anonymous function for  $x$  as a scalar by:

```
>> FA = @ (x) exp(x^2)/sqrt(x^2+5)
FA =
    @(x) exp(x^2)/sqrt(x^2+5)
```

If a semicolon is not typed at the end, MATLAB responds by displaying the function. The function can then be used for different values of  $x$ , as shown below.

```
>> FA(2)
ans =
    18.1994
>> z = FA(3)
z =
    2.1656e+003
```

If  $x$  is expected to be an array, with the function calculated for each element, then the function must be modified for element-by-element calculations.

```
>> FA = @ (x) exp(x.^2)./sqrt(x.^2+5)
FA =
    @(x) exp(x.^2)./sqrt(x.^2+5)
>> FA([1 0.5 2])
ans =
    1.1097    0.5604    18.1994
```

Using a vector as input argument.

**Example of an anonymous function with several independent variables:**

The function  $f(x, y) = x^2 - 4xy + y^2$  can be defined as an anonymous function by:

```
>> HA = @ (x,y) 2*x^2 - 4*x*y + y^2
HA =
    @(x,y) 2*x^2-4*x*y+y^2
```

Then the anonymous function can be used for different values of  $x$  and  $y$ . For example, typing `HA(2, 3)` gives:

```
>> HA(2,3)
ans =
    -7
```