

- 一、在文件中创建函数.....1
  - 1.1 从 R2016b 开始，用于存储函数的另一个选项是将函数包含在脚本文件的末尾。 ..... 1
  - 1.2 以下为脚本内容： ..... 1
  - 1.3 函数和文件的内容.....2
  - 1.4 End 语句..... 2
- 二、函数类型..... 2
  - 2.1 局部函数..... 2
  - 2.2 嵌套函数..... 3
  - 2.3 私有函数..... 3
  - 2.4 匿名函数..... 3

一、在文件中创建函数

脚本和函数都允许您通过将命令序列存储在程序文件中来重用它们。脚本是最简单的程序类型，因为它们存储命令的方式与您在命令行中键入命令完全相同。函数提供的灵活性更大，主要因为您可以传递输入值并返回输出值。例如，名为 **fact** 的以下函数用于计算某个数 (**n**) 的阶乘并返回结果 (**f**)。

```
function f = fact(n)
    f = prod(1:n);
end
```

此类型的函数必须在文件中而不是在命令行中定义。通常，您可以将函数存储在其自己的文件中。在这种情况下，最佳做法是对函数和文件使用相同的名称（此示例中为 **fact.m**），因为 **MATLAB®** 将程序与文件名相关联。将文件保存在当前文件夹中，或者保存在 **MATLAB** 搜索路径上的某个文件夹中。

您可以使用应用于随 **MATLAB** 一起安装的函数的相同语法规则从命令行调用函数。例如，计算 **5** 的阶乘。

```
x = 5;
y = fact(x) % 已在同一目录下定义了fact()求阶乘函数。

y = 120
```

1.1 从 R2016b 开始，用于存储函数的另一个选项是将函数包含在脚本文件的末尾。

例如，创建一个名为 **mystats.m** 并包含一些命令和两个函数 **fact** 和 **perm** 的文件。该脚本用于计算 (3,2) 的置换。

输入一下命令直接运行mystats.m脚本文件：

```
mystats

z = 6
```

1.2 以下为脚本内容：

```
x = 3;
```

```

y = 2;
z = perm(x,y);
function p = perm(n,r) % 脚本中的函数必须位于脚本的结尾，放在前面不行;
    p = fact(n)*fact(n-r);
end

function f = fact(n)
    f = prod(1:n);
end

```

## 1.3 函数和文件的内容

函数主体可以包括有效的 **MATLAB** 表达式、控制流语句、注释、空白行和嵌套函数。您在函数内创建的任何变量都存储在特定于该函数的工作区内，该工作区独立于基础工作区。

程序文件可以包含多个函数。如果文件仅包含函数定义，则第一个函数是主函数，也是 **MATLAB** 与文件名关联的函数。主函数或脚本代码后面的函数称为局部函数。局部函数只能在文件内使用。

## 1.4 End 语句

函数以 **end** 语句、文件末尾或局部函数的定义行结束，以先出现的为准。**end** 语句在以下情况下是必需的：

- 文件中的任何函数都包含嵌套函数（完全包含在其父级内的函数）。
- 该函数是函数文件中的局部函数，并且文件中有局部函数使用 **end** 关键字。
- 该函数是脚本文件内的局部函数。

虽然它有时是可选的，但使用 **end** 可提高代码可读性。

# 二、函数类型

## 2.1 局部函数

是在同一文件中可用的子例程。局部函数是拆分编程任务的最常见方法。在仅包含函数定义的函数文件中，局部函数可以任意顺序出现在文件中主函数的后面。在包含命令和函数定义的脚本文件中，局部函数必须位于文件末尾。（R2016b 或更高版本支持脚本中的函数。）

例如，创建一个名为 **myfunction.m** 的函数文件，其中包含主函数 **myfunction** 以及两个局部函数 **squareMe** 和 **doubleMe**：

```

function b = myfunction(a)
    b = squareMe(a)+doubleMe(a);
end
function y = squareMe(x) %后面两个为局部函数，仅限于 myfunction主函数使用；不可直接调用
    y = x.^2;
end
function y = doubleMe(x)
    y = x.*2;
end

```

可以从命令行或另一程序文件中直接调用主函数，但局部函数仅适用于 **myfunction**。

## 2.2 嵌套函数

嵌套函数完全包含在另一函数内。嵌套函数与局部函数的主要区别是，嵌套函数可以使用在父函数内定义的变量，无需将这些变量作为参数显式传递。

```
function myslider
value = 0;
f = figure;
s = uicontrol(f,'Style','slider','Callback',@slider);
e = uicontrol(f,'Style','edit','Callback',@edittext,...
    'Position',[100,20,100,20]);

function slider(obj,~)
    value = obj.Value;
    e.String = num2str(value);
end
function edittext(obj,~)
    value = str2double(obj.String);
    s.Value = value;
end

end
```

创建一个函数，该函数允许您使用滑块或可编辑的文本框设置介于 0 与 1 之间的一个值。如果您将嵌套函数用于回调，滑块和文本框可以共享值和彼此的句柄，无需显式传递它们：

## 2.3 私有函数

- 子文件夹中的私有函数与局部或嵌套函数一样，私有函数仅供特定位置的函数访问。但是，私有函数与可以调用它们的函数不在同一个文件中。它们位于名称为 **private** 的子文件夹中。仅 **private** 文件夹紧邻的上一级文件夹内的函数可使用私有函数。使用私有函数将代码分割为不同的文件，或在多个相关函数间共享代码。

## 2.4 匿名函数

- 无需文件的匿名函数只要函数包含一个语句，匿名函数即允许您定义该函数而不必创建程序文件。匿名函数通常用于定义数学表达式，然后使用 **MATLAB®** 复合函数（即接受函数句柄用作输入的函数）基于某个值范围计算该表达式。例如，以下语句为匿名函数创建名称为 **s** 的函数句柄：

```
s = @(x) sin(1./x);
```

- 该函数具有一个输入 **x**。**@** 运算符创建函数句柄。您可以使用该函数句柄针对特定值计算函数，例如

```
y = s(pi);
```

- 您也可以将该函数句柄传递给在某一值范围内计算的函数，例如 **fplot**：

```
range = [0.01,0.1];  
fplot(s,range)
```