Functions Revisited

On the surface, there doesn't appear to be much different between a script that takes no inputs and produces no outputs, and a function that takes no inputs and produces no outputs. However, recall that there is a *second* way in which functions differ from scripts — a function operates in its own "private" workspace. So any local variables created within a function are not visible from the outside. A script on the other hand shares its workspace with the calling context. To appreciate the difference between the two, create two M-files with the code shown below:

```
x = 3;
y = 5;
```

fooScript.m

```
function fooFunction
x = 3;
y = 5;
end
```

fooFunction.m

Now run the `fooFunction` from the Command window. There are no left-over variables in the Workspace, once the function finishes. Next, run `fooScript`. Now you will observe that there are variables named `x` and `y` in the Workspace when the script terminates. Thus, though `fooFunction` and `fooScript` appear to have identical specifications, there is fundamental difference in the way that the two are executed.