

Lec12 IAM550 J. Raeder 10/03/2019 More on functions

- Define functions within a main script.

```
x = 1:10;  
n = length(x);  
avg = mymean(x,n);  
med = mymedian(x,n);
```

```
function a = mymean(v,n)  
% MYMEAN Example of a local function.
```

```
    a = sum(v)/n;  
end
```

```
function m = mymedian(v,n)  
% MYMEDIAN Another example of a local function.
```

```
    w = sort(v);  
    if rem(n,2) == 1 % remainder  
        m = w((n + 1)/2);  
    else  
        m = (w(n/2) + w(n/2 + 1))/2;  
    end  
end
```

- Or define functions in separate files:
- MATLAB demands that each function is in a separate file (not good).
- MATLAB also demands that the file name is the function name (really ugly, but we have to live with it). It does not even matter what name you give the function in the m-file, but it is good practice to keep them the same.
- So, you cannot group several functions into one m-file, with one exception: if the main function uses other functions *exclusively* they can also be in the same m-file, but they remain invisible to the calling function or anything in any other file. Like:

```
function m = myfunc1(v,n)  
% MYMEDIAN Another example of a local function.  
S=myfunc2(v)  
T=myfunc3(n)  
m=S*T  
end  
function m = myfunc2(x)  
m=sin(x)+tanh(x);  
end  
function z = myfunc3(y)  
z=y*log(y)-22;  
end
```

- myfunc2 and myfunc3 remain invisible everywhere else.
- You can have unlimited function m-files. → draw tree

- When a function is called a second time, all variable content is lost. This allows for recursive functions → demonstrate recursion with factorial.
- Persistent variable: may conflict with recursion, but useful in other circumstances: check if function was called before → run demo
- Function handles → pass a function to another function. Also useful to rename functions.
- Special function handle → anonymous functions `F=@(arglist) expression;` for example `sinsqr=@(t) sin(sqrt(t)); result=sinsqr(3)` Basically the same as an inline function.
- Global variables: by default, all variables have a local scope → each variable is only defined within its own unit. One can use the same variable name in the main program and in functions and they do not interfere with each other. → demonstrate
- Statement: `global var1 var2 var3 ...`(no commas!)
- Variables can be made global, so their value can be shared → demo
- Global variables are initialized when first declared
- Does 'clear' clear it? → yes
- Global variables are another way to exchange parameters and return values with functions.
- Global variables in MATLAB are slow!
- Do not use global variables → unintended side effects. Maybe to share commonly used variables that are never changed. But you may see MATLAB code with global variables.
- In other languages: C: 'external' declaration; Fortran: 'common blocks, modules' statement; Perl: variables are global by default, need to declare local(); Python: local by default