

# Comando `function`

- Deve stare nella *prima riga* di un m-file che porta lo *stesso nome* della funzione
- Sintassi: `function [out1,out2,...] = nome_fun (param1,param2,...)`

## Esempi

sum\_pos.m

```
function S = sum_pos(x)

S = 0;
for i = 1:length(x)
    if x(i)>0
        S = S+x(i);
    end
end
```

elimina\_val.m

```
function [y,quanti] = elimina_val(x,val)

y = x;
quanti = 0;
for i = length(x):-1:1
    if x(i)==val
        y(i) = [];
        quanti = quanti+1;
    end
end
```

Shell (oppure lab2.m)

```
t = -10:0.1:10;
somma = sum_pos(t);

somma = sum(t(t>0));
```

```
x = [1:10 1:10];
[v,q] = elimina_val(x,5)
```

# Comando function

- Il comando `return` esce dalla funzione

## Esempio

scomp.m

```
function divisori = scomp(x)

divisori = [];
x = int32(x);
fprintf('Searching the decomposition of %d...\n',x);
if x<=1
    fprintf('Warning! The argument is <=1, returning empty vector...\n');
    return;
end

primi = primes(double(x)/2);
for i = 1:length(primi)
    while ~mod(x,primi(i))
        divisori = [divisori primi(i)];
        x = x/primi(i);
    end
    if x==1
        return;
    end
end
end
```

Shell (oppure lab2.m)

```
vett_scomp = scomp(831096)
```