



TECHNISCHE
UNIVERSITÄT
DRESDEN

Introduction to Matlab

Conditionals and loops

Dario Cuevas and Vahid Rahmati

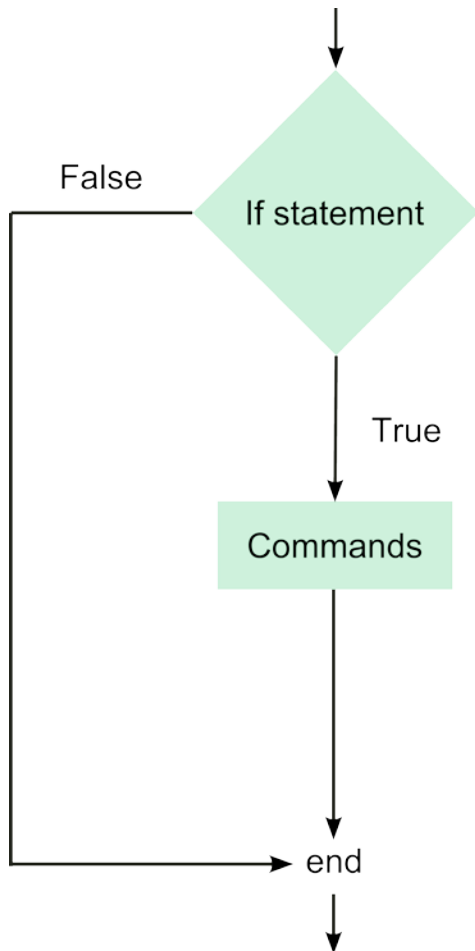


DRESDEN
concept
Exzellenz aus
Wissenschaft
und Kultur

Exercises

1. Create a function called `MyConcatenation`, that takes as input two matrices. It should concatenate these two matrices one next to the other, in the order they are given. Additionally, these two lines of code should be included:
`figure`
`imagesc(X)`
where `x` is the result of the concatenation. The function should also output `X`, the concatenation of the matrices. Run your function with the following matrices:
`A = ones(5,1), B = magic(5)`
2. Using `MyConcatenation`, concatenate the output of the previous exercise with a matrix `C = zeros(5,1)`.
3. Write a function `MyElimination` that removes the last 3 columns of the output of the last exercise
4. Write a script that runs the last 3 exercises together.

If conditionals

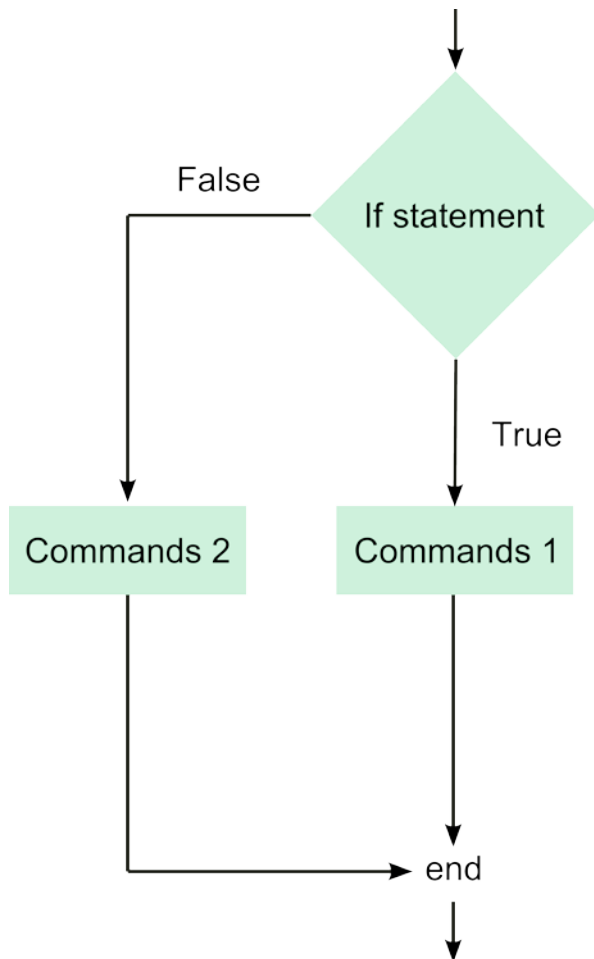


`if condition`
`commands`
`end`

Example:

```
If a<5  
fprintf('a is smaller than 5');  
end
```

If else

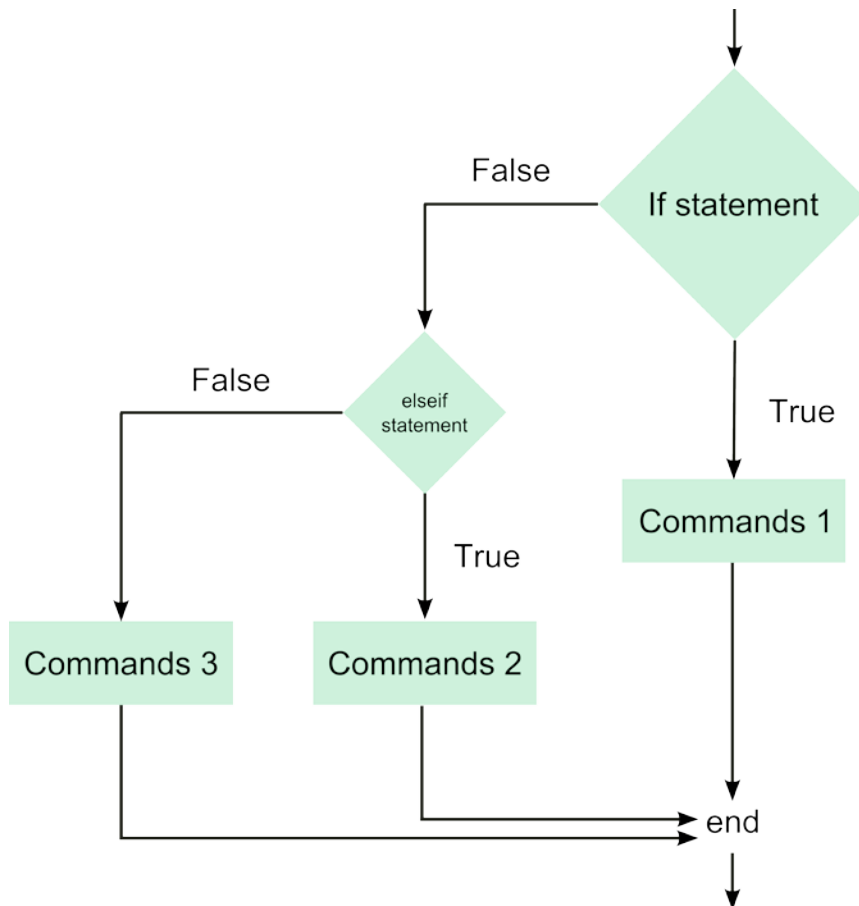


```
if condition  
  commands  
else  
  commands  
end
```

Example:

```
if a<5  
  printf('a is smaller than 5');  
else  
  printf('a is bigger or equal  
  than 5');  
end
```

If elseif



```
if condition1  
  commands  
elseif condition2  
  commands  
end
```

Example:

```
if a<5  
  fprintf('a is smaller than 5');  
elseif a>5  
  fprintf('a is bigger than 5');  
else  
  fprintf('a is 5');  
end
```

Conditionals

There are many conditionals you can use with the if statement:

- `<` (smaller than)
- `>` (bigger than)
- `==` (equal to)
- `<=` (smaller or equal to)
- `>=` (bigger or equal to)
- `!=` (different from)
- `0` (true)
- `1` (false)

There are combination operators:

- `&&` (and)
- `||` (or)
- `~` (negation)

If it is True, it will return 1. If it is False, it returns 0. For example:

$1 < 3 \rightarrow 1$	$1 \&\& 1 \rightarrow 1$	$1 \ \ 1 \rightarrow 1$
$8 > 9 \rightarrow 0$	$1 \&\& 0 \rightarrow 0$	$1 \ \ 0 \rightarrow 1$
$5 == 5 \rightarrow 1$	$0 \&\& 1 \rightarrow 0$	$0 \ \ 1 \rightarrow 1$
$\sim(1 > 2) \rightarrow 1$	$0 \&\& 0 \rightarrow 0$	$0 \ \ 0 \rightarrow 0$
$\sim 6 \rightarrow 0$		

Switch and case

```
switch x  
case value1  
commands  
case value2  
commands  
...  
otherwise  
commands  
end
```

Example:

```
switch x  
case 1  
y = 2;  
x = 2;  
case 2  
y = -2;  
case {3,4}  
y = 9;  
otherwise  
y = 0;  
end
```

for loop

To repeat a block of commands many times, use a for loop.

```
for index = values  
  commands  
end
```

For example:

```
for k = 1:3  
  display(k)  
end
```

Example:

```
for k = [1,5,-1]  
  display(k)  
end
```


Examples

```
for x = 1:10
    if x==5 || x== 7
        display(x);
    end
end
```

```
for k = 1:10
    y(k) = exp(k);
    if y(k)>30
        y(k) = 30;
    end
end
```

```
gamma = 1;
x = 5;
for i = 1:x
    gamma = gamma*i;
end
```

Examples

Example 1:

Write a function that, given a vector `VecA` and a number `NumX`, displays the indices and values of any component of `VecA` that equals `NumX`.

Example 2:

Write a function that, given a vector `VecA` and another vector `VecB` with two entries (i.e. `size(vecB) = 2 1`), finds the elements of `VecA` that equal any of the two elements of `VecB` and displays the indices.

Example 3:

Write a function that, given a matrix `MatA` and a number `NumX`, returns a matrix `MatX` with the indices of the components of `MatA` that equal `NumX`. If `N` components of `MatA` equal `NumX`, then `size(MatX) = N 2`.

Exercises

1. Using for loops, calculate the volumes of cylinders whose radii are $r = \{1, 1.2, 1.3\}$ and whose height is $h = 5$. That is, calculate three volumes (one for each cylinder). Write these volumes to a vector `VolumesCylinder`. The volume of a cylinder is given by $V = \pi r^2 h$.
2. Repeat the previous exercise, but now with $r = \{1, 1.2, 1.3\}$ and $h = \{5, 10, 12\}$. Write the results to a 3x3 matrix. Hint: use two nested for loops.
3. Write a function with two inputs, a vector `VecX` and a number `Y`. The function should search `VecX` and find those elements that equal `Y`. The output of the function is a vector `Z` with those indices. The function must work with any size of vector `VecX`.