
The Black Book of MATLAB

Release 0.0.1

Mohsen Nova

April 26, 2014

1	Empty arrays	3
1.1	Comparison	4
1.2	For statement	4
2	Indices and tables	5

Contents:

Empty arrays

Empty arrays in MATLAB come in different sized and number of dimensions. Even though the commonly known notation for empty arrays are

- `[]` for matrices
- `{}` for cell arrays
- `struct([])` for structure arrays

these arrays are all 2-dimensional of size 0-by-0:

```
>> size([])
ans =
     0     0
>> size({})
ans =
     0     0
>> size(struct([]))
ans =
     0     0
```

Empty arrays can come in as many number of dimensions or dimensions as long as at least one dimension is zero. Any of MATLAB function that create a special array by taking the dimensions of it as input arguments can be used to create such an empty array:

```
>> ones(1,2,0)
ans =
    Empty array: 1-by-2-by-0
>> zeros(5,0,2,4)
ans =
    Empty array: 5-by-0-by-2-by-4
>> cell(3,5,8,13,0)
ans =
    Empty cell array: 3-by-5-by-8-by-13-by-0
>> struct(rand(1,2,0))
ans =
    1x2x0 struct array with no fields.
```

Warning: These functions silently ignore negative arguments and replace them with zero. That means `ones(-10,2)` is exactly the same as `ones(0,2)`. This can be an issue if the arguments are calculated using other variables, e.g., before using `ones(m, n-m)` one needs to explicitly check if $n \geq m$.

A common way of encountering empty arrays is by all-false logical indexing in a non-empty array. However note that using a full-size logical matrix for indexing always leads to a 0-by-1 empty array.

```
>> A=rand(2,4)
A =
    0.0759    0.5308    0.9340    0.5688
    0.0540    0.7792    0.1299    0.4694
>> A(A(:,1)>1, :)
ans =
    Empty matrix: 0-by-4
>> A(A>1)
ans =
    Empty matrix: 0-by-1
```

1.1 Comparison

Empty arrays follow the same rules when it comes to comparison operator `==`. They can either be compared with an array of the same size or a scalar. That the result is always an empty logical array of the same size.

```
>> a=[]
a =
    []
>> a==[]
ans =
    []
>> class(a==[])
ans =
logical
>> (1==[])
ans =
    []
>> class(1==[])
ans =
logical
```

1.2 For statement

MATLAB `for` statement `for I=M` iterates over columns of `M` however it doesn't check if `M` is an empty array or not, therefore, following code

```
F = rand(4);
M = F(F(:,1)>1, :);
for I=M,
    disp('no way!')
end
```

will execute the inner loop 4 times, which in most cases is not a desirable outcome.

[being edited here](#)

Indices and tables

- *genindex*
- *modindex*
- *search*