

Matlab Tutorials

Das *LiveScript* zu dem Tutorial 1.

Aufgaben auch bei den [Matlab Tutorials von MathWorks.](#),

Tutorial 1.1

- Fragen & Anmerkungen: julian.kahnert@jade-hs.de
- Aufbau Matlab
- Variablen

```
3
```

```
ans = 3
```

```
3+4
```

```
ans = 7
```

```
m = 3 + 4
```

```
m = 7
```

```
m/2
```

```
ans = 3.5000
```

```
y = m/2
```

```
y = 3.5000
```

y = m/2 wird berechnet und zum aktuellen Zeitpunkt abgespeichert.

```
ans * 5    % man kann mit "ans" arbeiten, sollte jedoch vermieden werden.
```

```
ans = 17.5000
```

```
m = m+1
```

```
m = 8
```

Nur die Variable **m** wird verändert, das zuvor abgespeicherte **y** bleibt gleich!

Tutorial 1.2

- Variablen
- Funktionen: `log()`, `sin()`, `sqrt()`, `exp()`, ...

```
y = y+1;
```

```
y = m + 4;
5ha = 4;    % Fehler!
ha_5 = 4;
```

Der Variablenname *muss* mit einem Buchstaben beginnen, anschließend können Buchstaben, Zahlen und Unterstriche verwendet werden.

```
clear      % löscht den "Workspace"
clc        % leert das "Command Window"

x = 3 + 5;
pi        % die Variable pi ist bereits in Matlab gespeichert
pi / 2

sin( pi/2 )
abs(-5)   % Betrag
sqrt(4)   % Quadratwurzel
exp(10)   % e^( )
log10(100) % Zehner-Logarithmus,
```

Unexpected MATLAB expression.

Tutorial 1.3

- Vektoren
- Arrays

```
clc
clear
```

```
x = 4
```

```
x = 4
```

```
x = [2 4]    % Zeilenvektor
```

```
x = 1x2 double
```

```
    2    4
```

```
y = [2; 4]   % Spaltenvektor
```

```
y = 2x1 double
```

```
    2
    4
```

```
[1 2 3 4 5]
```

```
ans = 1x5 double
```

```
    1    2    3    4    5
```

Zeilenvektoren (1 x n): Elemente müssen durch Leerzeichen oder Kommas getrennt werden.

Spaltenvektoren (n x 1): Elemente müssen durch Semikolons getrennt werden.

```
m = [1 2 3; 4 5 6]
```

```
m = 2x3 double
```

```
    1    2    3
    4    5    6
```

```
m = [1 2 3; 4 5 6 7]    % Fehler! Dimensionen der Matrix nicht konsitent
```

```
Dimensions of matrices being concatenated are not consistent.
```

Matrix (n x m): Alle Elemente müssen einen Wert enthalten, d.h.: alle Zeilen einer Matrix müssen die selbe Anzahl an Elementen enthalten.

Tutorial 1.4

- Arrays
- Funktion: linspace()
- Documentation: help, doc

```
a = 111;
b = 222;
m = [1 2 b a; 4 a 6 4]
```

```
m = 2x4 double
```

```
    1    2   222   111
    4   111    6    4
```

```
v = [abs(-5) sqrt(9) 2^4]
```

```
v = 1x3 double
```

```
    5    3   16
```

Unterschiedliche Möglichkeiten um einen Zeilenvektor zu erstellen:

```
v = [1 2 3 4 5 6]
```

```
v = 1x6 double
```

```
    1    2    3    4    5    6
```

```
v = 1:6    % Schrittweite: 1
```

```
v = 1x6 double
```

```
    1    2    3    4    5    6
```

```
v = 1:2:6    % Schrittweite: 2
```

```
v = 1x3 double
```

```
1    3    5
```

```
x = linspace(10, 20, 4) % Vektor mit 4 Elementen
```

```
x = 1x4 double
```

```
10.0000    13.3333    16.6667    20.0000
```

Dokumentation von Funktionen, hier am Beispiel von `linspace()`:

```
help linspace % kurzer Ausschnitt in der "Command Line"
```

```
linspace Linearly spaced vector.
```

```
linspace(X1, X2) generates a row vector of 100 linearly  
equally spaced points between X1 and X2.
```

```
linspace(X1, X2, N) generates N points between X1 and X2.  
For N = 1, linspace returns X2.
```

```
Class support for inputs X1,X2:  
float: double, single
```

```
See also logspace, colon.
```

```
Reference page for linspace  
Other functions named linspace
```

```
doc linspace % vollständige Dokumentation mit Beispielen
```

Tutorial 1.5

- Arbeiten mit Matrizen

```
x = linspace(10, 20, 4)
```

```
x = 1x4 double
```

```
10.0000    13.3333    16.6667    20.0000
```

```
x = x.' % transponieren, z.B.: Spaltenvektor <=> Zeilenvektore
```

```
x = 4x1 double
```

```
10.0000  
13.3333  
16.6667  
20.0000
```

```
y = zeros(3); % 3x3 Matrix, die Nullen enthält
```

Spaltenweise anhängen:

```
m = rand(3, 2); % Dimensionen: 3x2  
n = ones(3, 2); % Dimensionen: 3x2  
[m n] % Die Anzahl der *Zeilen* muss übereinstimmen! => 3x4
```

```
ans = 3x4 double
```

```
0.7922    0.0357    1.0000    1.0000  
0.9595    0.8491    1.0000    1.0000  
0.6557    0.9340    1.0000    1.0000
```

```
%[m ones(2, 2)]    % Fehler! m enthält nur 3 Zeilen
```

Zeilenweise anhängen:

```
[m; ones(5, 2)]    % Die Anzahl der *Spalten* muss übereinstimmen! => 8x2
```

```
ans = 8x2 double
```

```
0.7922    0.0357  
0.9595    0.8491  
0.6557    0.9340  
1.0000    1.0000  
1.0000    1.0000  
1.0000    1.0000  
1.0000    1.0000  
1.0000    1.0000
```