

Beispiel-Glossar zu Octave

Befehl

Erklärung und Beispiele

abs

Absoluter Betrag: $\text{abs} = |Z| = \sqrt{(x^2+y^2)}$

```
>> abs(i)
```

```
ans = 1
```

```
>> abs(-56)
```

```
ans = 56
```

```
>> abs(3+5i)
```

```
ans = 5.8310
```

```
>>abs([3 -1.5])
```

```
ans = 3.0000 1.5000
```

a'

Transponieren

```
>>y=[1 2 3]'
```

```
y =
```

```
1
```

```
2
```

```
3
```

```
>>x=[1;2;3]'
```

```
x =
```

```
1 2 3
```

zeros

Matrix/Vektor aus Nullen

```
>>zeros(1,2)
```

```
ans =
```

```
0 0
```

```
>>zeros(2,2)
```

```
ans =
```

```
0 0
```

```
0 0
```

ones

Matrix/Vektor aus Einsen

```
>>ones(3,1)
```

```
ans =
```

```
1
```

```
1
```

```
1
```

```
>>ones(2,3)
```

```
ans =
```

```
1 1 1
```

```
1 1 1
```

a:b

Doppelpunkt-Operator:grenzt einen gewählten Bereich ein und legt Schrittweite fest (ohne Angabe ist Schrittweite=1)

```
>>a=[2:6]
```

```
a =
```

```
2 3 4 5 6
```

```
>>b=[2:2:6]
```

```
b =
```

```

2     4     6
>>c=[6:-1:2]
c =
6     5     4     3     2

```

a.^2

Punkt-Operator: Anwendung wird elementweise ausgeführt

```

>>a=[6 0 4];
    >> b=a.^2
    b =
    36  0  16

```

dot(a,b)

Skalarprodukt

```

>> a=[1 2 3];
>> b=[4 5 6];
>> dot(a,b)
ans = 32

```

cross(a,b)

Kreuzprodukt

```

>> a=[1 2 3];
>> b=[4 5 6];
>> cross(a,b)
ans =
-3  6 -3

```

a/b

Division: $a/b = a * b^{-1}$; $x * b = a$

```

>> a=[1 2 3];
>> b=a/2
b =
0.50000  1.00000  1.50000

```

```

>> a=[1 2 3];
>> b=[4 5 6];
>> c=a/b
c = 0.41558

```

a\b

Division: $a\b = a^{-1} * b$; $b*x = a$

```

>> a=[1 2 3];
>> b=a\b
b =
0.14286
0.28571
0.42857

```

```

>> a=[1 2 3];
>> b=[4 5 6];
>> c=a\b
c =
0.28571  0.35714  0.42857
0.57143  0.71429  0.85714
0.85714  1.07143  1.28571

```

input('a')

Benutzer_innen-Eingabe: Benutzer_in kann (auf Aufforderung) etwas auf dem Bildschirm eingeben

```

>> x=input('Geben Sie eine Zahl ein:')
    Geben Sie eine Zahl ein:2016
    x = 2016

```

disp(a) **Benutzer_innen-Ausgabe:** Benutzer_in wird etwas auf dem Bildschirm ausgegeben
x = 2016
>> y=disp(x)
y = 2016

fprintf('a',b) **Formatierte Ausgabe:** zum Ausgeben auf dem Bildschirm einer Zahl, eines Textes, eines Zeichens etc.
>> zahl=359;
>> fprintf('Die Zahl lautet: %d\n', zahl)
Die Zahl lautet: 359
>> fprintf('Die Zahl lautet: %7.3f\n', zahl)
Die Zahl lautet: 359.000

d= Ganzzahl; 7.3f= Gleitkommazahl mit insgesamt 7 Stellen, 3 davon als Nachkommastelle

eye(a) **Einheitsmatrix**
>> I=eye(2)
I =
Diagonal Matrix
 1 0
 0 1

>> I2=eye(2,3)
I2 =
Diagonal Matrix
 1 0 0
 0 1 0

diag([a b]) **Matrix mit wählbarer Diagonale**
>> D=diag([1 5])
D =
Diagonal Matrix
 1 0
 0 5

>> D2=diag([1 6 28])
D2 =
Diagonal Matrix
 1 0 0
 0 6 0
 0 0 28

length(a) **Länge einer Matrix**
>> A=[52 96 12; 26 48 3];
>> length(A)
ans = 3

>> A=[52 96 12; 26 48 3; 4 2 6];
>> length(A)
ans = 3

det(a) **Determinante**
>> A=[52 96 12; 26 48 3; 4 2 6];

```

>> det(A)
ans = -840

inv(a)      Inverse Matrix
>> A=[52 96 12; 26 48 3; 4 2 6];
>> inv(A)
ans =
    -0.33571    0.65714    0.34286
     0.17143   -0.31429   -0.18571
     0.16667   -0.33333    0.00000

rank(a)     Rang einer Matrix
>> A=[52 96 12; 26 48 3; 4 2 6];
>> rank(A)
ans = 3

eig(a)      Eigenwerte der Matrix
>> A=[52 96 12; 26 48 3; 4 2 6];
>> eig(A)
ans =
    100.4831
     -1.2376
     6.7546

polyval(p,x) Auswertung eines Polynoms
>> p=[5 6 2];
>> x=3;
>> polyval(p,x)
ans = 65

roots(p)    Nullstellen bestimmen
>> p=[5 6 2]
>> x=3;
>> roots(p)
ans =
    -0.60000 + 0.20000i
    -0.60000 - 0.20000i

poly([a b]) Polynom aus Nullstellen bestimmen
>> p2=poly([5 3])
p2 = 1 -8 15

polyder(p)  Polynom ableiten
>> p2=[1 -8 15]
>> polyder(p2)
ans =
     2 -8

>> p=[5 6 2]
>> polyder(p)
ans =
    10 6

conv(p1, p2) Polynom-Multiplikation
>> p=[5 6 2]
>> p2=[1 -8 15]
>> conv(p,p2)

```

```

ans =
    5 -34 29 74 30
deconv(p1, p2) Polynom-Division
>> p=[5 6 2]
>> p2=[1 -8 15]
>> deconv(p,p2)
ans = 5
x=linspace(a,b,c) Erzeugt zwischen a und b c äquidistante Zahlen
>> x=linspace(1, 10, 4)
x =
    1    4    7   10
plot(x,y) Stellt y gegen x graphisch dar
>> x=linspace(0,10,100);
>> y=sin(x);
>> plot(x,y);
xlabel('Name') Beschriftung der x- und y-Achse
ylabel('Name') >> xlabel('Zeit');
>> ylabel('Amplitude');
title('Name') Titel des Plots:
>> title('Erster Plot');
semilogy(x,y) Logarithmische Einteilung einzelner und beider Achsen
semology(x,y) >> x=linspace(0,10,100);
loglog(x,y) >> y=exp(0.5*x);
>> plot(x,y);
>> semilogy(x,y);
y-Achse wird logarithmisch dargestellt

function function: allgemein:
    function [output-Variable] = Name (input-Variablen)
        %Hier wird reingeschrieben, was die Funktion machen soll.
    endfunction
Beispiel:
function [out1, out2] = ErsteFunktion (x)
    out1 = x^2;
    out2 = out1*x;
endfunction

Aufruf im Befehlsfenster:
>> [x1, x2] = ErsteFunktion(3)
x1 = 9
x2 = 27

if if-Bedingung: zum Prüfen einer Bedingung, Ausführung, wenn Bedingung
wahr
x=10;
y=20;
if x == 10
    disp(x);
end

```

elseif	<p>elseif-Bedingung: zur Auswahl zwischen mehreren Alternativen. Trifft keine der vorher festgelegten Bedingungen zu, wird die unter else stehende Bedingung ausgeführt</p> <pre>a=15; b=65; if a==65 disp('a=10'); elseif b==65 disp('b=65');</pre>
for	<p>For-Schleife</p> <pre>for i=0:5 disp(i); i=i+1; end</pre>
while	<p>while-Schleife</p> <pre>a=1; i=1 while i<10 a=a+i; i=i+3; disp(a); end</pre>
fopen('a','b') fclose(a)	<p>Aufrufen/Importieren einer Datei</p> <pre>a=fopen('OctaveTest1.txt', 'rt'); A=fscanf(a, '%c', [1, Inf]); fclose(a);</pre> <p>allgemein: name = fopen('Dateiname.format', 'rt') (rt= read text); Arbeitsanweisung; fclose(name);</p>
xlsread('Name')	<p>Excel-Datei lesen</p> <pre>x = xlsread('OctaveTest2.xlsx');</pre> <p>allgemein: name = xlsread('Dateiname.xlsx');</p>
fscanf(a,'%b',[])	<p>Datei lesen</p> <pre>a=fopen('OctaveTest1.txt', 'rt'); A=fscanf(a, '%c', [1, Inf]); fclose(a);</pre> <p>allgemein: Öffnen der Datei name2 = fscanf(name1, '%Format', [Lesebereich]); Schließen der Datei; name1 steht für den Namen der geöffneten Datei; c bedeutet Character; Inf bedeutet, dass bis zum Ende der Datei gelesen wird</p>
csvwrite('a', b)	<p>Csv-Datei beschreiben/speichern (in Excel-Datei umwandelbar)</p> <pre>B=rand(100,100); csvwrite('Zufall.csv', B)</pre> <p>allgemein: csvwrite('Dateiname.format', name1); name1 steht für den zu speichernden Programmabschnitt</p>
fputs(a,b)	<p>Datei speichern</p> <pre>M = 'Mal gucken, ob es klappt.'; d = fopen('Test.txt', 'w');</pre>

fputs(d, M);

fclose (d);

allgemein: Öffnen der Datei

fputs(name1, name2);

Schließen der Datei

name1 steht für die geöffnete Datei, name2 für den zu speichernden Programmabschnitt

system(a)

Kontrolle von Subprozessen

allgemein: Mit diesem Prozess können bpsw. Externe Programme gestartet und gesteuert werden. In die Klammern werden die Befehle für das externen Programm geschrieben.