

# Grundlagen der Programmierung

## Matlab Cheat Sheet

### Einfache Verzweigung

```
if bedingung1
    block
elseif bedingung2
    block
elseif bedingung3
    block
...
else
    block
end
```

### Mehrfachverzweigung

```
switch switch_ausdruck
    case case_ausdruck
        block
    case case_ausdruck
        block
    case case_ausdruck
        block
    ...
    otherwise
        block
end
```

### For Schleife

```
for variable = bereich
    block
end
```

### While Schleife

```
while bedingung
    block
end
```

### Funktionsdeklaration

```
function name(parameterliste)
    block
end
```

### Funktion mit Ergebnis

```
function ergebnisvar = name(parameterliste)
    block
end
```

Im Block muss der Ergebnisvariablen ein Wert zugewiesen werden.

### Bezeichner

für Namen von Variablen, Skripte (Matlab-Dateien), Funktionen, ...

- Beginnen mit Buchstaben
- Nur Buchstaben, Zahlen und `_` erlaubt
- Case-sensitiv
- Keine reservierten Worte von Matlab:  
break case catch classdef continue else  
elseif end for function global if otherwise  
parfor persistent return spmd switch try  
while

### Gültigkeitsbereiche

- Jeder Funktionsaufruf hat einen eigenen Bereich zum Speichern von Variablen
- ⇒ Eine Funktion kann nicht auf die Variablen einer anderen Funktion zugreifen.  
Ausnahmen:
  - Deklaration mit `global`: gemeinsamer Speicherbereich für alle Funktionen
  - Deklaration mit `persistent`: gemeinsamer Speicherbereich für alle Aufrufe einer Funktion
  - Variablen im Hauptprogramm teilen sich einen gemeinsamen Speicher ('Workspace'), auch mit der Konsole

## Basics

clc	Lösche Command Window
clear	Lösche Variablen
%	Kommentarzeichen
Strg-n	Neues Skript
F5	Skript starten
Strg-c	Programm abbrechen

## Mac Tastatur

[ ]	⌘+5	⌘+6
{ }	⌘+8	⌘+9
	⌘+7	
\	⌘+⇧+7	
~	⌘+n	

## fprintf

Syntax: `fprintf( zeichenkette, ... )`

\n	Newline
%d	Ganze Zahl
%f	Gleitkommazahl
%.2f	... nur 2 Nachkommastellen
%c	Char (ein Zeichen)
%s	Zeichenkette

## Operatoren

+ - * / ^	arithmetische Op.
+ - .* ./ .^ ' .'	Arrayoperationen
* / \ ^	Matrixoperationen
< <= > >= == ~=	Vergleichsop.
&&    ~	logische Op.
=	Zuweisungsop.

## Vektoren, Matrizen, Arrays

$x = [1 \ 2 \ 3]$	Zeilenvektor erzeugen
$x = s : e$	Zeilenvektor $[s \dots e]$ erzeugen
$x = s : inc : e$	dto., mit Schrittweite $inc$
$x = [1; 2; 3]$	Spaltenvektor erzeugen
$A = [1 \ 2; 3 \ 4]$	Matrix erzeugen
$x(2), A(1,2)$	Zugriff auf Element in $x$ bzw. $A$
$A([1 \ 2], 2)$	Zugriff auf mehrere Zeilen von $A$
$A(:, 2) = []$	Spalte 2 in $A$ löschen

## Zahlenkonvertierungen

### Binär (Basis 2)

Ziffern 0 und 1

### Hexadezimal (Basis 16)

Ziffern: 0, 1, 2, ..., 9, A, B, C, D, E

### Basis $b \rightarrow$ Dezimal

Zahl (Basis  $b$ ):  $z_n z_{n-1} \dots z_1 z_0$

Dezimalwert:  $\sum_{i=0}^n z_i \cdot b^i$

### Dezimal $\rightarrow$ Basis $b$

Teile durch  $b$ ,  
notiere Rest,  
bis Ergebnis = 0

### Hexadezimal $\leftrightarrow$ Binär

4 Ziffern binär  $\hat{=}$  1 Ziffer hexadezimal

### negative Zahlen (Zweierkomplement)

1. Wandle Betrag in Binärzahl um (Anzahl Bits beachten!)
2. Bilde bitweises Komplement (0  $\leftrightarrow$  1)
3. Addiere 1