Anwendung von (Mathematica und) Matlab in der Physikalischen Chemie

3. Interaktive Kommandozeile

UNI FREIBURG

Albert-Ludwigs-Universität Freiburg

Dr. Till Biskup

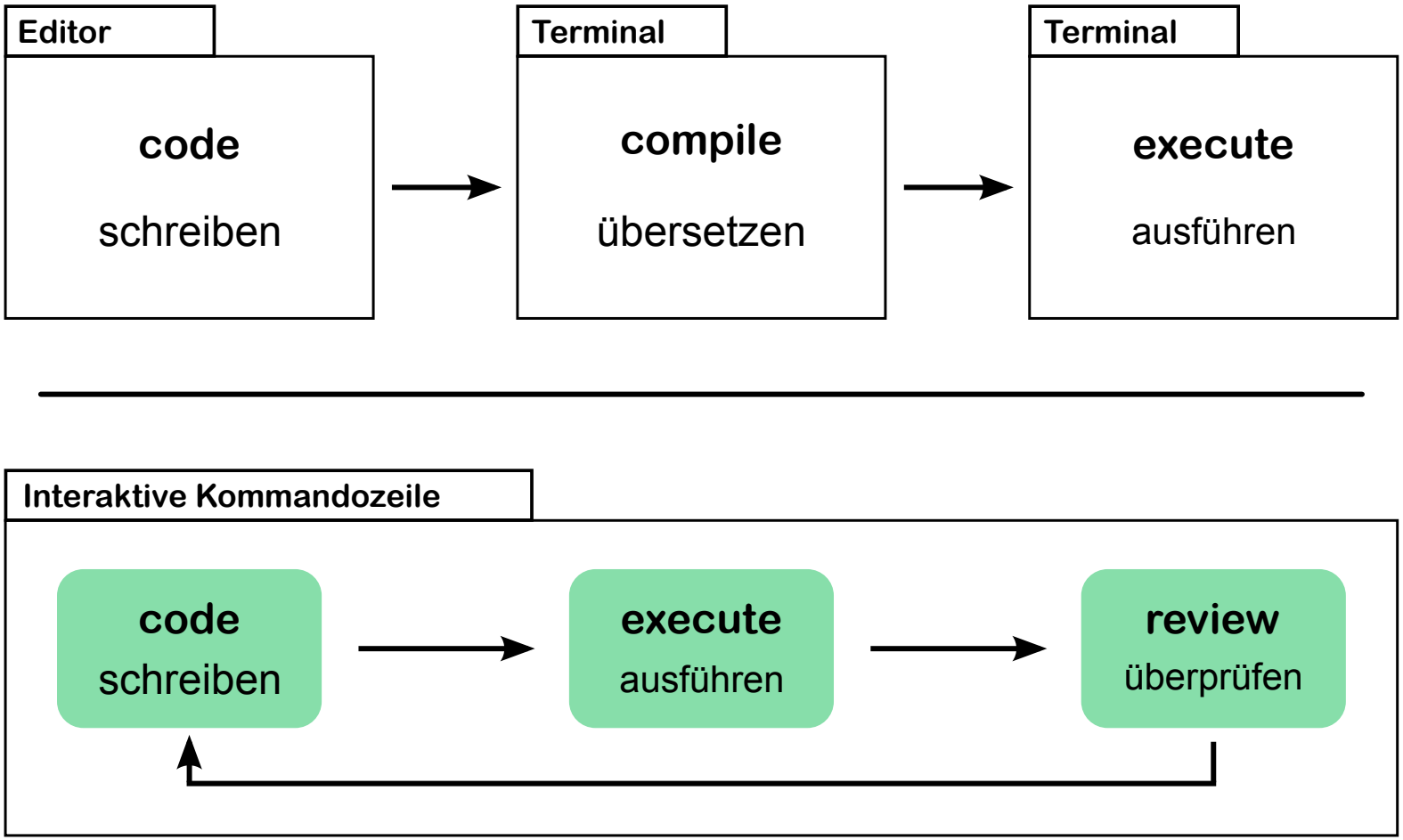
Institut für Physikalische Chemie
Albert-Ludwigs-Universität Freiburg
Wintersemester 2016/2017

Übersicht

UNI FREIBURG

Interaktive Kommandozeile

Vergleich zur konventionellen Programmierung



Interaktive Kommandozeile

Vorteile (und Nachteile)

UNI FREIBURG

Vorteile

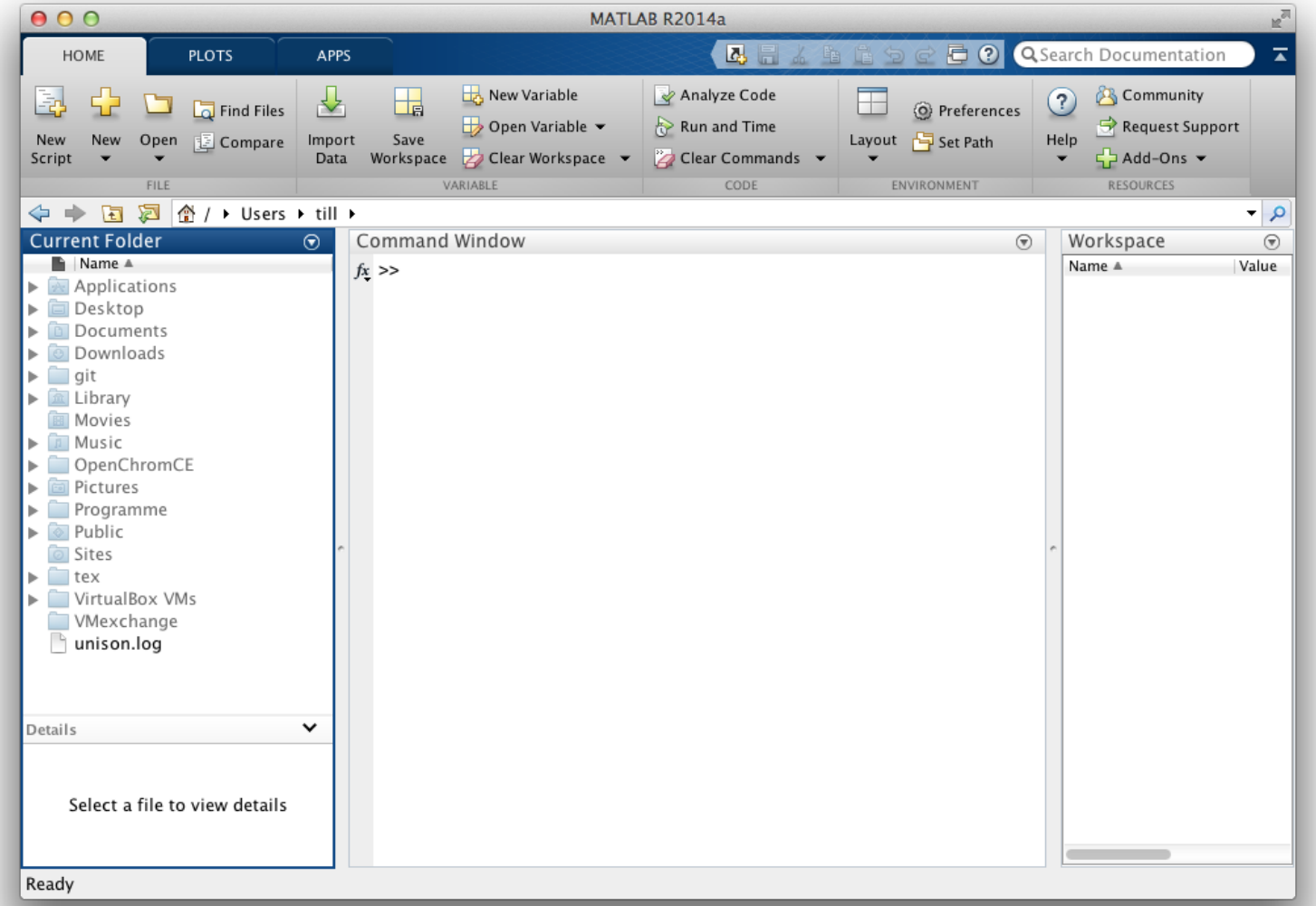
- Direkte Rückmeldung
  - Der Nutzer sieht sofort das Ergebnis eines Befehls
- „Rapid Prototyping“
  - Gut geeignet, um „mal schnell“ etwas auszuprobieren
  - Kurzer Zyklus zwischen Befehlseingabe und Ergebnis

Nachteile

- Ungeeignet für komplexere Abläufe
  - Schleifen können eingesetzt werden... aber mühsam
- Normalerweise keine Speicherung
  - Die Befehlshistorie ist begrenzt.

Interaktive Kommandozeile

Matlab: Ein erster Eindruck



Interaktive Kommandozeile

Zusätzliche Vorteile von Matlab & Co.

UNI FREIBURG

- Integrierte grafische Nutzeroberfläche
  - Übersicht über die definierten Variablen
  - Zugriff auf die Hilfe (in eigenem Fenster)
  - Übersicht über die Befehlshistorie
  - Plots funktionieren direkt aus der Kommandozeile
  - Computermaus als intuitive Bedienhilfe
- Alle Befehle auf der Kommandozeile erreichbar
  - Matlab: Alles, was im Matlab-Suchpfad liegt
  - Andere Sprachen: ggf. mit „import“ o.ä. arbeiten
- Ausgabeformat steuerbar
  - Z.B.: Darstellung von Fließkommazahlen

☛ Die einfach bedienbare grafische Oberfläche war wesentlich am Erfolg von Matlab beteiligt

Grundaspekte der Programmierung

Matlab ist eine Programmiersprache

UNI FREIBURG

- Matlab ist eine Programmiersprache
  - Große Ähnlichkeit mit C, Pascal, ...
  - Nimmt dem Nutzer viele Programmierdetails ab
  - Fokus auf Anwendung (und Mathematik)
- Matlab versucht nicht, seinen grundlegenden Charakter als Programmiersprache zu verbergen.
  - Ursprung als einfache Schnittstelle zu Fortran-Routinen
  - Viel näher an „normalen“ Programmiersprachen als beispielsweise Mathematica

☛ Grundaspekte der Programmierung notwendig, um mit Matlab auf der Kommandozeile zu arbeiten.

Allgemeine Aspekte

- Das Gleichheitszeichen dient der Zuweisung
  - „Zuweisungsoperator“
  - Mathematische Gleichheit wird durch „==“ abgefragt
- Zuweisungen werden von rechts nach links gelesen
  - Der Ausdruck rechts des Gleichheitszeichens wird der Variable links davon zugewiesen
  - Mehrfache Zuweisungen sind (in Matlab) nicht erlaubt

Matlab-Spezifika

- Variablen müssen nicht vordefiniert werden
- Variablen sind Matrizen (wenn nicht anders angegeben)

Matlab als Taschenrechner

Ein allererstes Beispiel

UNI FREIBURG

Listing 1: Die interaktive Kommandozeile in Matlab

```
>> 1+2

ans =

     3

>> a=5; b=3;
>> c=a*b

c =

    15

>>
```

- Das letzte Ergebnis wird immer in „`ans`“ gespeichert.
- Ein Semikolon unterdrückt die Ausgabe.

Matlab als Taschenrechner

Grundlegende mathematische Operationen

UNI FREIBURG

Operation	Matlab-Befehl
Grundrechenarten	`+, -, *, /`
Potenz	`^`
Quadratwurzel	`sqrt`
n-te Wurzel	`nthroot`

Anmerkungen

- Der Multiplikationsoperator „`*`“ muss immer explizit ausgeschrieben werden.
- Matlab beherrscht grundlegende Operratorrangfolge („Punkt vor Strich“)
- Terme können durch *runde* Klammern gruppiert werden

Matlab als Taschenrechner

Mathematische Funktionen

UNI FREIBURG

Funktion	Matlab-Befehl
trigonometrisch	`sin`, `cos`, `tan`
	`asin`, `acos`, `atan`
hyperbolisch	`sinh`, `cosh`, `tanh`
	`asinh`, `acosh`, `atanh`
Exponentialfunktion	`exp`
Logarithmen	`log`, `log10`, `log2`
Vorzeichen	`sign`

Anmerkungen

- Argumente für die trigonometrischen Funktionen in Radians

Matlab als Taschenrechner

Funktionen für ganze Zahlen

UNI FREIBURG

Funktion	Matlab-Befehl
Runden auf nächste ganze Zahl	`round`
Runden in Richtung 0	`fix`
Runden in Richtung $-\infty$	`floor`
Runden in Richtung $+\infty$	`ceil`
Rest (Vorzeichen des Zählers)	`rem`
Rest (Vorzeichen des Nenners)	`mod`
größter gemeinsamer Teiler	`gcd`
kleinstes gemeinsames Vielfaches	`lcm`

Matlab als Taschenrechner

Funktionen für komplexe Zahlen

UNI FREIBURG

Funktion	Matlab-Befehl
Betrag	`abs`
Argument (Winkel der Polarkoordinaten)	`angle`
Realteil	`real`
Imaginärteil	`imag`
komplex-konjugierte Zahl	`conj`

Anmerkungen

- Komplexe Zahlen werden mittels „`i`“ (bzw. „`j`“) angegeben
- Niemals „`i`/`j`“ als Laufvariable in Schleifen verwenden!

Matlab als Taschenrechner

Vordefinierte Konstanten

UNI FREIBURG

Konstante	Matlab-Befehl
komplexe Zahl (i)	`i, j`
Kreiszahl (π)	`pi`
Maschinengenauigkeit (ε)	`eps`

Anmerkungen

- Die Schreibweise der komplexen Zahl als „j“ kommt aus der Elektrotechnik.
- `pi` ist nur so exakt wie die numerische Genauigkeit (ε).
  - `sin(pi)` gibt deshalb in Matlab *nicht* 0 zurück...

- Mehr zur numerischen Genauigkeit (ε) später

Matlab als Taschenrechner

Vektoren und Matrizen

UNI FREIBURG

- Alle Variablen sind für Matlab Matrizen.
  - Solange man nicht explizit etwas anderes sagt...
  - Matlab = MATrix LABoratory
- Matrixoperationen sind in Matlab schnell, der Rest nicht.
- Eckige Klammern dienen der Definition:

Listing 2: Definition von Matrizen und Vektoren in Matlab

```
% Ein Zeilenvektor
a = [1 2 3]
% Ein Spaltenvektor
b = [1; 2; 3]
% Eine Matrix
c = [1 0; 0 1];
```

- Matrizen werden in der Reihenfolge Zeile-Spalte indiziert
- Zugriff auf Elemente über runde Klammern
  - Alle Elemente einer Zeile/Spalte über „:“

Listing 3: Zugriff auf Elemente von Matrizen und Vektoren in Matlab

```
% Eine 2x3-Matrix
m = [1 2 3; 4 5 6];

% Zugriff auf das zweite Element der ersten Zeile
m(1,2)

% Zugriff auf die zweite Zeile
m(2,:)

% Zugriff auf die erste Spalte
m(:,1)
```

Operatoren

Operator	Bedeutung
`+, -, *, /, ^`	Matrix-Operationen
`.+, .-, .*, ./, .^`	elementweise Operationen
`\`	Division von links
`:`	Zugriff auf Bereiche (Spalte/Zeile)
`'`	Adjungieren
`.'`	Transponieren

Anmerkungen

- „`.+`“ und „`.-`“ sind identisch mit „`+`“ und „`-`“.
- Arithmetische Operatoren führen Matrix-Operationen aus

Funktionen

Funktion	Bedeutung
`cross`	Kreuzprodukt
`dot`	Punktprodukt
`kron`	Kronecker-Tensorprodukt
`eig, eigs`	Eigenwerte
`diag`	Diagonale
`transpose`	Transponieren

Anmerkungen

- „`*`“ führt eine Matrixmultiplikation durch.

Spezielle Matrizen

Matrix	Bedeutung
`ones`	Matrix aus Einsen
`zeros`	Matrix aus Nullen
`eye`	Einheitsmatrix
`diag`	Diagonalmatrix
`rand`	Matrix aus Zufallszahlen

Anmerkungen

- „`eye`“ ist ein Wortspiel für „`I`“.
  - Anfangs unterschied Matlab nicht zwischen Groß- und Kleinschreibung, und „`i`“ war vergeben (imaginäre Zahl).

Inline-Funktionen

Listing 4: Definition einer Inline-Funktion

```
>> fun = @(x)3*x^3+5*x^2+8*x+17;
>> fun(3)

ans =

   167
```

- Für wiederkehrende mathematische Zusammenhänge
  - Auch in Abhängigkeit mehrerer Variablen.
  - Einfache Zusammenhänge, in einer Zeile formulierbar
- Definition über `@`-Zeichen
  - Variablen der Funktion nach dem `@` in runden Klammern
  - Variablenname für die Zuweisung beliebig

Plots

Listing 5: Einfaches Beispiel eines Plot-Befehls

```
x=0:0.1:10; y=sin(x)
plot(x,y);
```

- Intuitiver Befehl für Vektoren/Funktionen einer Variablen
  - Vektoren (`x`, `y`) müssen gleich lang sein.
  - Bei nur einem Vektor wird gegen dessen Index geplottet.
- Jeder `plot`-Befehl überschreibt das aktuelle Grafikfenster.
  - Abhilfe (I): Paare von Vektoren in einem `plot`-Befehl.
  - Abhilfe (II): `hold on`, `hold off`

☛ Details ausführlicher in einer späteren Lektion

Und Vieles mehr...

- Es gibt noch viele weitere Aspekte.
  - Der Kurs erhebt keinerlei Anspruch auf Vollständigkeit.
  - Die Auswahl ist (notwendigerweise) subjektiv.
- Die Matlab-Hilfe ist ein guter Startpunkt.
  - Einführende Kapitel geben einen guten Überblick.

- Vieles läuft über (ernsthaftere) Programmierung.
  - Dazu kommen wir noch später.
  - Hier geht es erstmal um einen ersten Eindruck.
  - Matlab taugt auch als komfortabler „Taschenrechner“.

Kosmetik und Komfort

Ausgabeformat auf der Kommandozeile

UNI FREIBURG

Listing 6: Standard-Ausgabeformat der Matlab-Kommandozeile

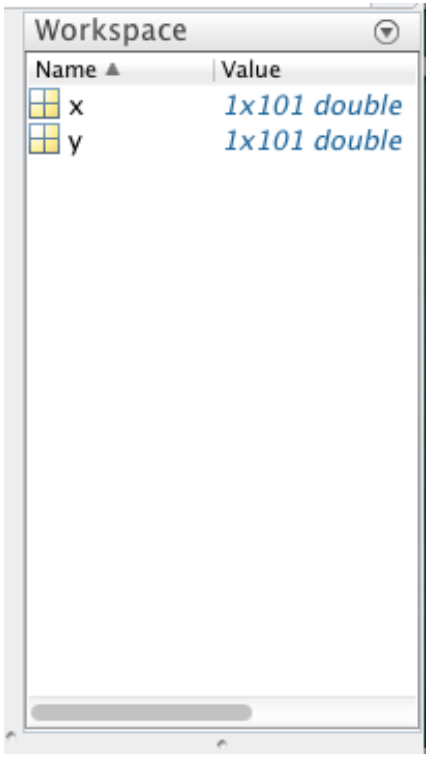
```
>> pi

ans =

    3.1416
```

- Normalerweise werden nur vier Dezimalstellen angezeigt.
  - Die interne Rechengenauigkeit ist natürlich viel höher.
  - Die Ausgabe rundet die letzte angezeigte Dezimale...
- Befehl zur Kontrolle der Anzeige: `format`
  - Optionen (u.a.): `long`, `short`, `rat`, `compact`

☛ Details in der Matlab-Hilfe: `doc format`

Übersicht über die Variablen – grafisch

- Vorteil der grafischen Oberfläche
  - Fenster „Workspace“ mit Variablen
  - Doppelklick auf Variable öffnet einen Variableneditor (wozu auch immer...)
- Übersicht über
  - Name
  - Typ (über Symbol vor dem Namen)
  - Größe („Value“)
  - Minimum und Maximum
- Nachteile
  - Typ nur über Symbol
  - Scrollen notwendig



Übersicht über die Variablen – programmatisch

Listing 7: Übersicht über die definierten Variablen

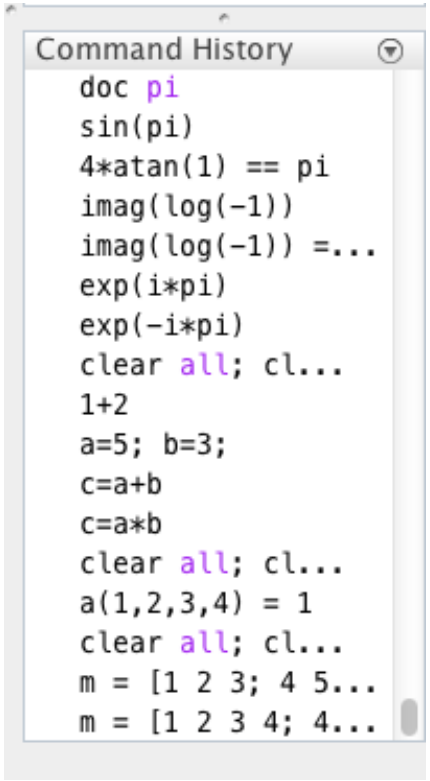
```
>> who

Your variables are:

x  y

>> whos
  Name        Size                Bytes  Class     Attributes

  x           1x101                 808  double
  y           1x101                 808  double
```

- Zwei Befehle
  - `who`, `whos`
- Zusätzliche Optionen für die beiden Befehle

Kosmetik und Komfort

Historie

- Matlab legt eine Befehlshistorie an
  - Für jeden Nutzer separat
  - Standard: letzte 25.000 Befehle
  - Einstellbar ($\Rightarrow$ „Preferences“)
- Anzeige der Historie in der GUI möglich
  - Muss ggf. eingestellt werden
- Historie über Pfeiltasten erreichbar
  -  ruft die Historie auf
  -  und  zum Blättern
  - Anfangsbuchstabe(n) und  für gezielte Suche



☛ Kann das Leben mitunter sehr vereinfachen...

Kosmetik und Komfort

Löschen von Variablen, Schließen von Fenstern, Aufräumen

UNI FREIBURG

Hilfe, mein Matlab müllt zu...

- Problem
  - Alle Variablen auf der Kommandozeile sichtbar/erreichbar
  - Kommandozeile unübersichtlich vollgeschrieben
  - Ggf. unzählige (Grafik-)Fenster offen
- Lösung
  - Gezielt Variablen löschen, Fenster schließen, ...

Matlab-Befehl	Beschreibung
`clear`	Variable(n) löschen
`close`	(Grafik-)Fenster schließen
`clc`	Kommandozeile aufräumen

Kosmetik und Komfort

Löschen von Variablen, Schließen von Fenstern, Aufräumen

UNI FREIBURG

Listing 8: Der ultimative Aufräumbefehl...

```
clear all; close all; clc
```

- Drei auf einen Streich:
  - Löscht alle definierten Variablen (`clear all`).
  - Schließt alle offenen (Grafik-)Fenster (`close all`).
  - Räumt die Kommandozeile auf (`clc`).
- Löscht nicht die Befehlshistorie
- Mit Bedacht einsetzen...
  - Alle Variablen zu löschen, muss keine gute Idee sein.
  - Nicht unbedingt geeignet als Beginn eines Skripts...

Ausblick

Wie geht's weiter?

...Zeit für eigene praktische Arbeit...

Vorschau: Skripte und Funktionen

- Editor
- Befehle/Funktionen