

# Eine kleine Einführung in GNU Emacs

Philip K.

<https://wwwcip.cs.fau.de/~oj14ozun>

15. Juli 2019

# Was ist Emacs?

*GNU Emacs, the GNU incarnation of the advanced, self-documenting, customizable, extensible Editor.*

# Was ist Emacs?

*GNU Emacs, the GNU incarnation of the **advanced**, self-documenting, customizable, extensible Editor.*

# Was ist Emacs?

*GNU Emacs, the GNU incarnation of the advanced, **self-documenting**, customizable, extensible Editor.*

# Was ist Emacs?

*GNU Emacs, the GNU incarnation of the advanced, self-documenting, **cus-tomizable**, extensible Editor.*

# Was ist Emacs?

*GNU Emacs, the GNU incarnation of the advanced, self-documenting, customizable, **extensible** Editor.*

# Was ist Emacs?

*Oder* ein Lisp Interpreter dem Nebeneffekt  
“Text Editor”.

# Ziele dieser Präsentation



- Grundlagen von Emacs vorstellen



# Ziele dieser Präsentation



- Grundlagen von Emacs vorstellen
- Selbstständiges Nutzen Vermitteln

# Ziele dieser Präsentation



- Grundlagen von Emacs vorstellen
- Selbstständiges Nutzen Vermitteln
- Konfigurieren Lernen

# Ziele dieser Präsentation



- Grundlagen von Emacs vorstellen
- Selbstständiges Nutzen Vermitteln
- Konfigurieren Lernen
- Kurzes kennenlernen populärer Pakete







# Nicht-Ziele dieser Präsentation

- Einführung in Emacs Lisp
- Einführung in {Org-Mode, AUCTeX, GUD, Gnus, ...}
- Geschichte von Emacs
- Argumente für Emacs über {Vi, Vim, Eclipse, Acme, ...}

```
File Options Buffers Tools De Mark Regeep Multiple Single Help

/usr/bin
total used in directory /usr/bin available 307680
d-w-r-xr-x  2 root  406 04-03 18:28 bin/
d-w-r-xr-x  12 root 120 05-29 13:58 include/
d-w-r-xr-x  22 root  88 04-03 18:28 lib/
d-w-r-xr-x  35 root 336 05-28 11:59 lib32/
d-w-r-xr-x  11 root  3 05-28 04:44 lib64 -> lib/
d-w-r-xr-x  15 root 408 05-22 04:23 local/
d-w-r-xr-x  10 root  16 04-03 18:28 locale/
d-w-r-xr-x  271 root 120 04-03 18:28 src/
d-w-r-xr-x  10 root 408 05-22 13:20 src/
d-w-r-xr-x  10 root 408 05-22 09:18 src/

/usr/bin
total used in directory /usr/bin available 307680
d-w-r-xr-x  2 root  406 04-03 18:28 bin/
d-w-r-xr-x  12 root 120 05-29 13:58 include/
d-w-r-xr-x  22 root  88 04-03 18:28 lib/
d-w-r-xr-x  35 root 336 05-28 11:59 lib32/
d-w-r-xr-x  11 root  3 05-28 04:44 lib64 -> lib/
d-w-r-xr-x  15 root 408 05-22 04:23 local/
d-w-r-xr-x  10 root  16 04-03 18:28 locale/
d-w-r-xr-x  271 root 120 04-03 18:28 src/
d-w-r-xr-x  10 root 408 05-22 13:20 src/
d-w-r-xr-x  10 root 408 05-22 09:18 src/

 Emacs: visit this file in other window.
```

# Verschiedene Arten Emacs zu Nutzen

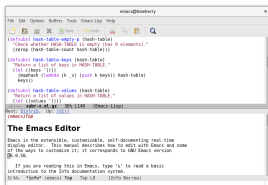




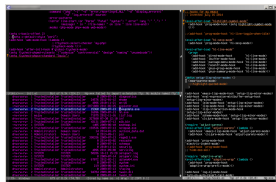




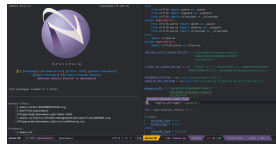
# Verschiedene Arten Emacs zu Nutzen



Ein Notepad mit **merkwürdigen** Tastenkombos und ein paar **koolen** Tricks



**Portabler** Text Editor das mit normalen "CUA" Tastenkombos (→ cua-mode)



**Vim-artiger** Editor (→ Evil/Spacemacs)

**Mein Fokus:** Allgemeine Arbeits- und Programmierumgebung basieren auf **Freier Software**

Teil I

# Emacs nutzen als Text Editor

**Annahme:** Jeder hat einen Text Editor für *längere* Zeit genutzt.

Jeder kann daher *grundsätzlich*  
Emacs nutzen!

Jeder kann daher *grundsätzlich*  
Emacs nutzen!

Jetzt aber ein paar Besonderheiten:



# Bewegen im Text

# Bewegen im Text

C-f/C-b (für “forward”,  
“backward”) um sich im Text  
**um je ein Charakter** zu  
bewegen.

C-n/C-p (für “next”,  
“previous”) um sich im Text  
**um je ein Zeile** zu bewegen.

→ Wie Pfeiltasten

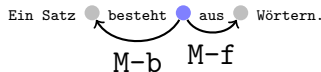
# Bewegen im Text

C-f/C-b (für “forward”,  
“backward”) um sich im Text  
**um je ein Charakter** zu  
bewegen.

C-n/C-p (für “next”,  
“previous”) um sich im Text  
**um je ein Zeile** zu bewegen.

→ Wie Pfeiltasten

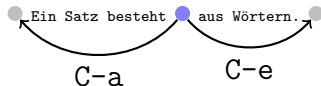
M-f/M-b (M steht für “Meta”  
↔ Meist *Alt*) um sich im Text  
**um je eine Wort** zu bewegen.



# Bewegen im Text II

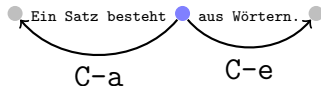
# Bewegen im Text II

C-a/C-e um sich im Text an  
Anfang und das Ende einer  
**Zeile** zu bewegen.

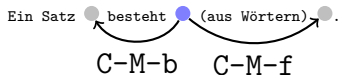


# Bewegen im Text II

C-a/C-e um sich im Text an Anfang und das Ende einer **Zeile** zu bewegen.



C-M-f/C-M-b um sich im Text **um je eine Struktureinheit** zu bewegen.



**Wichtig:** Man kann aber muss die  
Keybindings **nicht** benutzen.

Es gibt aber Unterschiede:

Text Bearbeitung ist nicht wie  
gewöhnlich!



# Copy-Paste in Emacs

Operation	CUA	Emacs
Undo	Ctrl-z	C-/, C-_, C-x u
Copy	Ctrl-c	M-w
Cut	Ctrl-x	C-w
Paste	Ctrl-p	C-y
Pop	N/A	M-y

**Table:** Übersetzungstabelle von CUA nach Emacs (in Region)

# Copy-Paste in Emacs

Operation	CUA	Emacs
Undo	Ctrl-z	C-/, C-_, C-x u
Copy	Ctrl-c	M-w
Cut	Ctrl-x	C-w
Paste	Ctrl-p	C-y
Pop	N/A	M-y

**Table:** Übersetzungstabelle von CUA nach Emacs (in Region)

*Alles was gelöscht wird weiterhin gespeichert und kann wieder eingefügt ("yanking") werden.*

# Mehr Löschen (“Killing”)

**C-d** Lösche Buchstabe

**M-d** Lösche Wort

**C-M-d** Lösche  
Struktureinheit

**C-k** Lösche Rest der  
Zeile

# Mehr Löschen (“Killing”)

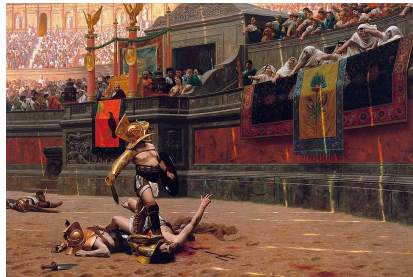
**C-d** Lösche Buchstabe

**M-d** Lösche Wort

**C-M-d** Lösche  
Struktureinheit

**C-k** Lösche Rest der  
Zeile

→ Landet alles im **Kill Ring**

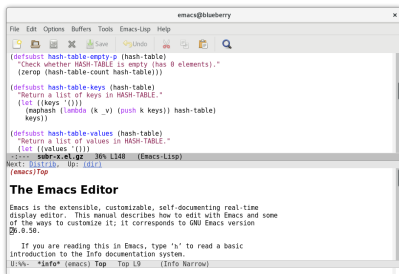


# “Fachbegriffe”: **Buffer** und **Windows**.

# “Fachbegriffe”: **Buffer** und **Windows**.

→ Genutzt um mehreren “Sachen” gleichzeitig zu arbeiten.

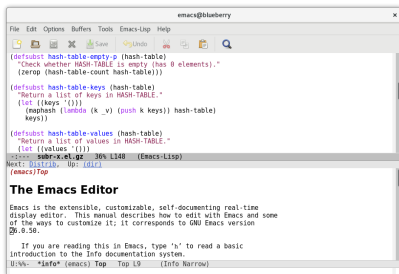
# Anatomie von Emacs



The screenshot shows the Emacs editor window titled 'emacs@blueberry'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Emacs-Lisp', and 'Help'. The toolbar contains icons for file operations and search. The main text area contains Lisp code defining hash-table functions: `(defsubst hash-table-empty-p (hash-table) ...)`, `(defsubst hash-table-keys (hash-table) ...)`, and `(defsubst hash-table-values (hash-table) ...)`. Below the code, a status line shows '---- subr-x.el.gz 30% 1348 (Emacs-Lisp)'. The bottom section of the window displays the title 'The Emacs Editor' followed by introductory text about Emacs and instructions on how to read the documentation.

- 1 Ein **Frame** (Fenster) enthält *mehere...*

# Anatomie von Emacs



```
emacs@blueberry
File Edit Options Buffers Tools Emacs-Lisp Help
[Icons: Save, Undo, Search]
(defun hash-table-empty-p (hash-table)
  "Check whether HASH-TABLE is empty (has 0 elements)."
  (zerop (hash-table-count hash-table)))

(defun hash-table-keys (hash-table)
  "Return a list of keys in HASH-TABLE."
  (let ((keys {}))
    (maphash (lambda (k _v) (push k keys)) hash-table)
    keys))

(defun hash-table-values (hash-table)
  "Return a list of values in HASH-TABLE."
  (let ((values {}))
    (maphash (lambda (k _v) (push _v values)) hash-table)
    values))

----- subr-x.el.gz 30% L148 (Emacs-Lisp)
Next: Distrib, Up: dir
(emacs)Top

The Emacs Editor

Emacs is the extensible, customizable, self-documenting real-time
display editor. This manual describes how to edit with Emacs and some
of the ways to customize it; it corresponds to GNU Emacs version
26.0.50.

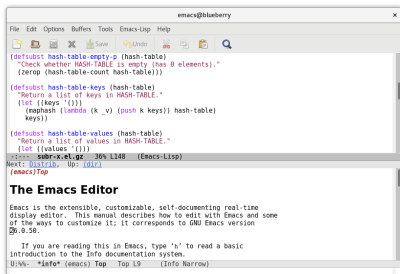
If you are reading this in Emacs, type 'h' to read a basic
Introduction to the Info documentation system.

U:%% *Info* (emacs) Top Top 1/9 (Info Narrow)
```

- 1 Ein **Frame** (Fenster) enthält *mehere...*
- 2 **Windows** (Rahmen) mit je *einem...*



# Anatomie von Emacs



The screenshot shows the Emacs editor window titled 'emacs@blueberry'. The menu bar includes File, Edit, Options, Buffers, Tools, Emacs-Lisp, and Help. The toolbar contains icons for opening files, saving, undo, redo, and search. The main text area contains Lisp code defining hash-table functions: `(defsubst hash-table-empty-p (hash-table) "Check whether HASH-TABLE is empty (has 0 elements)."`, `(defsubst hash-table-keys (hash-table) "Return a list of keys in HASH-TABLE."`, and `(defsubst hash-table-values (hash-table) "Return a list of values in HASH-TABLE."`. Below the code, a status line shows `----- subr-x.el.gz 30% 1348 (Emacs-Lisp)`. The bottom section is titled 'The Emacs Editor' and contains introductory text about Emacs, including the instruction 'If you are reading this in Emacs, type 'h' to read a basic Introduction to the Info documentation system.' and the status bar `U:%% *Info* [emacs] Top Top 19 (Info Narrow)`.

- 1 Ein **Frame** (Fenster) enthält *mehere...*
- 2 **Windows** (Rahmen) mit je *einem...*
- 3 **Buffer**, mit dem Text.

# Bewegung *um* Text

Operation	Keybinding
<i>Vertikal</i> Spalten	C-x 2
<i>Horizontal</i> Spalten	C-x 3
Dieses <i>Windows</i> Löschen	C-x 0
Alle anderen <i>Windows</i> Löschen	C-x 1
Fokussiere nächstes <i>Window</i>	C-x o

**Tabelle:** Wichtige Keybindings für *Window*-management

# Bewegung *um* Text

Operation	Keybinding
<i>Vertikal</i> Spalten	C-x 2
<i>Horizontal</i> Spalten	C-x 3
Dieses <i>Windows</i> Löschen	C-x 0
Alle anderen <i>Windows</i> Löschen	C-x 1
Fokussiere nächstes <i>Window</i>	C-x o

**Tabelle:** Wichtige Keybindings für *Window*-management

→ Interessant für Manche: `windmove`.

# Mit Buffern Arbeiten

C-x b **Wechsele** einen Buffer aus

# Mit Buffern Arbeiten

C-x b **Wechsele** einen Buffer aus

C-x k **Lösche** einen Buffer

# Mit Buffern Arbeiten

C-x b **Wechsele** einen Buffer aus

C-x k **Lösche** einen Buffer

C-x C-b **Übersicht** aller Buffer

# Mit Buffern Arbeiten

C-x b **Wechsele** einen Buffer aus

C-x k **Lösche** einen Buffer

C-x C-b **Übersicht** aller Buffer

C-x C-f **Öffne** Datei im Buffer

# Mit Buffern Arbeiten

C-x b **Wechsele** einen Buffer aus

C-x k **Lösche** einen Buffer

C-x C-b **Übersicht** aller Buffer

C-x C-f **Öffne** Datei im Buffer

C-x C-s **Speichere** Buffer in Datei



Wichtig: Ein **Buffer** ist nicht nur  
Text!

**Wichtig:** Ein **Buffer** ist nicht nur  
Text!

**Major-** und **Minor-Modes** ändern das “Verhalten”  
eines Buffers.

Davor:

**M-x**

Davor:

**M-x**

Ausführen einer **interaktiven Funktion**

# Interaktive Befehle

Keybinding | Funktion

---

# Interaktive Befehle

Keybinding	Funktion
C-a	move-beginning-of-line

# Interaktive Befehle

Keybinding	Funktion
C-a	<code>move-beginning-of-line</code>
M-y	<code>yank-pop</code>

# Interaktive Befehle

Keybinding	Funktion
C-a	move-beginning-of-line
M-y	yank-pop
C-x 2	split-window-below



# Interaktive Befehle

Keybinding	Funktion
<code>C-a</code>	<code>move-beginning-of-line</code>
<code>M-y</code>	<code>yank-pop</code>
<code>C-x 2</code>	<code>split-window-below</code>
<code>M-x</code>	<code>execute-extended-command</code>

# Interaktive Befehle

Keybinding	Funktion
<code>C-a</code>	<code>move-beginning-of-line</code>
<code>M-y</code>	<code>yank-pop</code>
<code>C-x 2</code>	<code>split-window-below</code>
<code>M-x</code>	<code>execute-extended-command</code>
<code>a</code>	<code>self-insert-command</code>

*Ein Buffer hat ein Major-Mode.*

# *Ein Buffer hat ein Major-Mode.*

Beispiele (nutze M-x um zu wechseln):

**c-mode** Für C Programmierung

# *Ein Buffer hat ein Major-Mode.*

Beispiele (nutze M-x um zu wechseln):

`c-mode` Für C Programmierung

`fortran-mode` Für Fortran Programmierung

# *Ein Buffer hat ein Major-Mode.*

Beispiele (nutze M-x um zu wechseln):

`c-mode` Für C Programmierung

`fortran-mode` Für Fortran Programmierung

`javascript-mode` Für Javascript "Programmierung"

# *Ein Buffer hat ein Major-Mode.*

Beispiele (nutze M-x um zu wechseln):

`c-mode` Für C Programmierung

`fortran-mode` Für Fortran Programmierung

`javascript-mode` Für Javascript "Programmierung"

⋮

# *Ein Buffer hat ein Major-Mode.*

Beispiele (nutze M-x um zu wechseln):

`c-mode` Für C Programmierung

`fortran-mode` Für Fortran Programmierung

`javascript-mode` Für Javascript "Programmierung"

⋮

`mail-mode` Für Email-Verfassung



## *Ein Buffer hat ein Major-Mode.*

Beispiele (nutze M-x um zu wechseln):

`c-mode` Für C Programmierung

`fortran-mode` Für Fortran Programmierung

`javascript-mode` Für Javascript "Programmierung"

⋮

`mail-mode` Für Email-Verfassung

`"dired-mode"` Für Verzeichnis-Bearbeitung

## *Ein Buffer hat ein Major-Mode.*

Beispiele (nutze M-x um zu wechseln):

`c-mode` Für C Programmierung

`fortran-mode` Für Fortran Programmierung

`javascript-mode` Für Javascript "Programmierung"

⋮

`mail-mode` Für Email-Verfassung

`"dired-mode"` Für Verzeichnis-Bearbeitung

`image-mode` Für Bild-Betrachtung

## *Ein Buffer hat ein Major-Mode.*

Beispiele (nutze M-x um zu wechseln):

`c-mode` Für C Programmierung

`fortran-mode` Für Fortran Programmierung

`javascript-mode` Für Javascript "Programmierung"

⋮

`mail-mode` Für Email-Verfassung

`"dired-mode"` Für Verzeichnis-Bearbeitung

`image-mode` Für Bild-Betrachtung

⋮

*Ein Buffer hat mehrere  
Minor-Modes.*

# *Ein Buffer hat mehrere Minor-Modes.*

Beispiele (einmal M-x für **an**, zweimal für **aus**):

**electric-pair-mode** Ausgleichen von Klammern

# *Ein Buffer hat mehrere Minor-Modes.*

Beispiele (einmal M-x für **an**, zweimal für **aus**):

`electric-pair-mode` Ausgleichen von Klammern

`flycheck-mode` Rechtschreibkorrektur

# *Ein Buffer hat mehrere Minor-Modes.*

Beispiele (einmal M-x für **an**, zweimal für **aus**):

**electric-pair-mode** Ausgleichen von Klammern

**flycheck-mode** Rechtschreibkorrektur

**auto-revert-mode** Automatisches Aktualisieren einer Datei

# *Ein Buffer hat mehrere Minor-Modes.*

Beispiele (einmal M-x für **an**, zweimal für **aus**):

**electric-pair-mode** Ausgleichen von Klammern

**flycheck-mode** Rechtschreibkorrektur

**auto-revert-mode** Automatisches Aktualisieren einer Datei

**abbrev-mode** Verwaltung von Kürzel



## *Ein Buffer hat mehrere Minor-Modes.*

Beispiele (einmal M-x für **an**, zweimal für **aus**):

**electric-pair-mode** Ausgleichen von Klammern

**flycheck-mode** Rechtschreibkorrektur

**auto-revert-mode** Automatisches Aktualisieren einer Datei

**abbrev-mode** Verwaltung von Kürzel

**display-line-numbers** Anzeigen von Zeilennummern

## *Ein Buffer hat mehrere Minor-Modes.*

Beispiele (einmal M-x für **an**, zweimal für **aus**):

**electric-pair-mode** Ausgleichen von Klammern

**flycheck-mode** Rechtschreibkorrektur

**auto-revert-mode** Automatisches Aktualisieren einer Datei

**abbrev-mode** Verwaltung von Kürzel

**display-line-numbers** Anzeigen von Zeilennummern

⋮

## *Ein Buffer hat mehrere Minor-Modes.*

Beispiele (einmal M-x für **an**, zweimal für **aus**):

**electric-pair-mode** Ausgleichen von Klammern

**flycheck-mode** Rechtschreibkorrektur

**auto-revert-mode** Automatisches Aktualisieren einer Datei

**abbrev-mode** Verwaltung von Kürzel

**display-line-numbers** Anzeigen von Zeilennummern

⋮

**menu-bar-mode** Anzeigen des Menu-Balkens

## *Ein Buffer hat mehrere Minor-Modes.*

Beispiele (einmal M-x für **an**, zweimal für **aus**):

**electric-pair-mode** Ausgleichen von Klammern

**flycheck-mode** Rechtschreibkorrektur

**auto-revert-mode** Automatisches Aktualisieren einer Datei

**abbrev-mode** Verwaltung von Kürzel

**display-line-numbers** Anzeigen von Zeilennummern

⋮

**menu-bar-mode** Anzeigen des Menu-Balkens

**global-auto-revert-mode** Aktiviere auto-revert-mode  
für alle Buffer an

## *Ein Buffer hat mehrere Minor-Modes.*

Beispiele (einmal M-x für **an**, zweimal für **aus**):

**electric-pair-mode** Ausgleichen von Klammern

**flycheck-mode** Rechtschreibkorrektur

**auto-revert-mode** Automatisches Aktualisieren einer Datei

**abbrev-mode** Verwaltung von Kürzel

**display-line-numbers** Anzeigen von Zeilennummern

⋮

**menu-bar-mode** Anzeigen des Menu-Balkens

**global-auto-revert-mode** Aktiviere auto-revert-mode  
für alle Buffer an

⋮

Genaueres zu Keybindings findet man  
in **Emacs Manual** → C-h t

## Teil II

# Emacs nutzen als *Emacs*

Emacs *kann* genutzt werden wie es  
geliefert wird.



Emacs *kann* genutzt werden wie es geliefert wird.

*Üblich* ist es aber Emacs nach Bedürfnissen anzupassen.

Emacs *kann* genutzt werden wie es geliefert wird.

*Üblich* ist es aber Emacs nach Bedürfnissen anzupassen.

→ **Konfiguration**

# Beispielkonfiguration

Siehe:

```
https://wwwcip.cs.fau.de/~oj14ozun/src+etc/init.el
```

Speichere in `$HOME/.emacs.d/init.el`, oder (`find-file` `user-init-file`).

# Beispielkonfiguration

Siehe:

`https://www.cip.cs.fau.de/~oj14ozun/src+etc/init.el`

Speichere in `$HOME/.emacs.d/init.el`, oder (`find-file user-init-file`).

**Features:**

- Merkwürdige Defaults deaktiviert

# Beispielkonfiguration

Siehe:

```
https://wwwcip.cs.fau.de/~oj14ozun/src+etc/init.el
```

Speichere in `$HOME/.emacs.d/init.el`, oder (`find-file` `user-init-file`).

**Features:**

- Merkwürdige Defaults deaktiviert
- Bessern Input (`ido-mode`) aktiviert

# Beispielkonfiguration

Siehe:

<https://wwwcip.cs.fau.de/~oj14ozun/src+etc/init.el>

Speichere in `$HOME/.emacs.d/init.el`, oder (`find-file user-init-file`).

**Features:**

- Merkwürdige Defaults deaktiviert
- Bessern Input (`ido-mode`) aktiviert
- Basiskonfiguration für Dired, Org, GUD/GDB, IRC, Rechtschreibung

# Beispielkonfiguration

Siehe:

<https://wwwcip.cs.fau.de/~oj14ozun/src+etc/init.el>

Speichere in `$HOME/.emacs.d/init.el`, oder (`find-file user-init-file`).

## Features:

- Merkwürdige Defaults deaktiviert
- Bessern Input (`ido-mode`) aktiviert
- Basiskonfiguration für Dired, Org, GUD/GDB, IRC, Rechtschreibung
- `text-mode` und `prog-mode` verbessert

# Beispielkonfiguration

Siehe:

<https://wwwcip.cs.fau.de/~oj14ozun/src+etc/init.el>

Speichere in `$HOME/.emacs.d/init.el`, oder (`find-file user-init-file`).

## Features:

- Merkwürdige Defaults deaktiviert
- Bessern Input (`ido-mode`) aktiviert
- Basiskonfiguration für Dired, Org, GUD/GDB, IRC, Rechtschreibung
- `text-mode` und `prog-mode` verbessert
- Paar nützliche Funktionen gebunden



# Beispielkonfiguration

Siehe:

<https://wwwcip.cs.fau.de/~oj14ozun/src+etc/init.el>

Speichere in `$HOME/.emacs.d/init.el`, oder (`find-file user-init-file`).

## Features:

- Merkwürdige Defaults deaktiviert
  - Bessern Input (`ido-mode`) aktiviert
  - Basiskonfiguration für Dired, Org, GUD/GDB, IRC, Rechtschreibung
  - `text-mode` und `prog-mode` verbessert
  - Paar nützliche Funktionen gebunden
- Frei für verfügbar.

# Elemente des Konfigurierens: setq

Genutzt um Variablenwerte zu setzen

# Elemente des Konfigurierens: setq

Genutzt um Variablenwerte zu setzen

## Beispiel:

```
(setq inhibit-startup-screen t)
(setq sentence-end-double-space nil)
(setq locale-coding-system 'utf-8)
```

# Elemente des Konfigurierens: add-hook

Genutzt um Automatische Aktionen zu bedingen

# Elemente des Konfigurierens: add-hook

Genutzt um Automatische Aktionen zu bedingen

## Beispiel:

```
(add-hook 'text-mode-hook #'flyspell-mode)
(add-hook 'prog-mode-hook #'electric-indent-mode)
(add-hook 'TeX-after-compilation-finished-functions
          #'TeX-revert-document-buffer)
```

# Elemente des Konfigurierens: define-key

Wird genutzt um *Funktionen* an *Tasten* zu binden.

# Elemente des Konfigurierens: define-key

Wird genutzt um *Funktionen* an *Tasten* zu binden.

## Beispiel:

```
(define-key prog-mode-map (kbd "C-c w")
  #'whitespace-mode)
(define-key calendar-mode-map (kbd "f")
  #'calendar-forward-day)
(global-set-key (kbd "M-/") #'hippie-expand)
(global-set-key (kbd "<f2>") #'man)
```

Emacs hat *viele* Keybindings.



Emacs hat *viele* Keybindings.

→ Lernt nach Bedarf und durch  
Angewöhnung, **nicht** durch  
Auswendigmerken.

# Wo kann man Lernen? C-h!

`C-h f` Suche Dokumentation für Funktion auf.

# Wo kann man Lernen? C-h!

**C-h f** Suche Dokumentation für Funktion auf.

**C-h v** Suche Dokumentation für Variable auf.

# Wo kann man Lernen? C-h!

**C-h f** Suche Dokumentation für Funktion auf.

**C-h v** Suche Dokumentation für Variable auf.

**C-h k** Schau was auf Taste gebunden ist.

# Wo kann man Lernen? C-h!

- C-h f** Suche Dokumentation für Funktion auf.
- C-h v** Suche Dokumentation für Variable auf.
- C-h k** Schau was auf Taste gebunden ist.
- C-h i** Öffne Emacs Benutzeranleitung.

# Wo kann man Lernen? C-h!

**C-h f** Suche Dokumentation für Funktion auf.

**C-h v** Suche Dokumentation für Variable auf.

**C-h k** Schau was auf Taste gebunden ist.

**C-h i** Öffne Emacs Benutzeranleitung.

**C-h t** Starte Emacs Tutorial

# Wo kann man Lernen? C-h!

**C-h f** Suche Dokumentation für Funktion auf.

**C-h v** Suche Dokumentation für Variable auf.

**C-h k** Schau was auf Taste gebunden ist.

**C-h i** Öffne Emacs Benutzeranleitung.

**C-h t** Starte Emacs Tutorial

**C-h m** Zähle alle Modi auf

# Wo kann man Lernen? C-h!

**C-h f** Suche Dokumentation für Funktion auf.

**C-h v** Suche Dokumentation für Variable auf.

**C-h k** Schau was auf Taste gebunden ist.

**C-h i** Öffne Emacs Benutzeranleitung.

**C-h t** Starte Emacs Tutorial

**C-h m** Zähle alle Modi auf

**C-h b** Zähle Keybindings auf



# Wo kann man Lernen? C-h!

- C-h f** Suche Dokumentation für Funktion auf.
- C-h v** Suche Dokumentation für Variable auf.
- C-h k** Schau was auf Taste gebunden ist.
- C-h i** Öffne Emacs Benutzeranleitung.
- C-h t** Starte Emacs Tutorial
- C-h m** Zähle alle Modi auf
- C-h b** Zähle Keybindings auf
- C-h a** Suche nach Funktionen

# Wo kann man Lernen? C-h!

**C-h f** Suche Dokumentation für Funktion auf.

**C-h v** Suche Dokumentation für Variable auf.

**C-h k** Schau was auf Taste gebunden ist.

**C-h i** Öffne Emacs Benutzeranleitung.

**C-h t** Starte Emacs Tutorial

**C-h m** Zähle alle Modi auf

**C-h b** Zähle Keybindings auf

**C-h a** Suche nach Funktionen

⋮

Wenn Quelltext installiert ist, kann man mit `C-h f/C-h v` dieses leicht einsehen.

Wenn Quelltext installiert ist, kann man mit `C-h f/C-h v` dieses leicht einsehen.

Wichtiges Beispiel für **Software Freiheit!**

Emacs wird nicht nur von Nutzer  
selbst, sondern auch anderen  
Erweitert.

Emacs wird nicht nur von Nutzer  
selbst, sondern auch anderen  
Erweitert.

Diese nennen wir **Pakete**

# Beispielpakete, Vorinstalliert

**Org** Mächtiges Markup modus für alles von Textverfassung, Präsentationen, bis zu Terminverwaltung.

# Beispielpakete, Vorinstalliert

**Org** Mächtiges Markup modus für alles von Textverfassung, Präsentationen, bis zu Terminverwaltung.

**calc** Vielseitig nutzbarer RPN Taschenrechner



# Beispielpakete, Vorinstalliert

- Org** Mächtiges Markup modus für alles von Textverfassung, Präsentationen, bis zu Terminverwaltung.
- calc** Vielseitig nutzbarer RPN Taschenrechner
- GUD** Text-Graphische (2D) Oberfläche für GDB und andere Debugger

# Beispielpakete, Vorinstalliert

**Org** Mächtiges Markup modus für alles von Textverfassung, Präsentationen, bis zu Terminverwaltung.

**calc** Vielseitig nutzbarer RPN Taschenrechner

**GUD** Text-Graphische (2D) Oberfläche für GDB und andere Debugger

**rcirc/ERC** IRC clients *in* Emacs

# Beispielpakete, Vorinstalliert

**Org** Mächtiges Markup modus für alles von Textverfassung, Präsentationen, bis zu Terminverwaltung.

**calc** Vielseitig nutzbarer RPN Taschenrechner

**GUD** Text-Graphische (2D) Oberfläche für GDB und andere Debugger

**rcirc/ERC** IRC clients *in* Emacs

**Gnus** Email client *in* Emacs

# Beispielpakete, Vorinstalliert

**Org** Mächtiges Markup modus für alles von Textverfassung, Präsentationen, bis zu Terminverwaltung.

**calc** Vielseitig nutzbarer RPN Taschenrechner

**GUD** Text-Graphische (2D) Oberfläche für GDB und andere Debugger

**rcirc/ERC** IRC clients *in* Emacs

**Gnus** Email client *in* Emacs

**TRAMP** Transparentes Einblenden von anderen Dateistrukturen in Emacs

# Beispielpakete, nicht Vorinstalliert

**AucTeX** Ausgezeichneter  $\text{\LaTeX}$  Major-mode

# Beispielpakete, nicht Vorinstalliert

**AucTeX** Ausgezeichneter  $\text{\LaTeX}$  Major-mode

**Magit** Ausgezeichnete Git Oberfläche

# Beispielpakete, nicht Vorinstalliert

**AucTeX** Ausgezeichneter  $\text{\LaTeX}$  Major-mode

**Magit** Ausgezeichnete Git Oberfläche

**Avy** Funktionen um im Text herumzuspringen

# Beispielpakete, nicht Vorinstalliert

**AucTeX** Ausgezeichneter  $\text{\LaTeX}$  Major-mode

**Magit** Ausgezeichnete Git Oberfläche

**Avy** Funktionen um im Text herumzuspringen

**Helm oder Ivy** Verbesserter Input Modus für Mini-Buffer  
Anfragen



# Beispielpakete, nicht Vorinstalliert

**AucTeX** Ausgezeichneter  $\text{\LaTeX}$  Major-mode

**Magit** Ausgezeichnete Git Oberfläche

**Avy** Funktionen um im Text herumzuspringen

**Helm oder Ivy** Verbesserter Input Modus für Mini-Buffer  
Anfragen

**Paredit** Minor-mode für strukturelles Editing

# Beispielpakete, nicht Vorinstalliert

**AucTeX** Ausgezeichneter  $\text{\LaTeX}$  Major-mode

**Magit** Ausgezeichnete Git Oberfläche

**Avy** Funktionen um im Text herumzuspringen

**Helm oder Ivy** Verbesserter Input Modus für Mini-Buffer  
Anfragen

**Paredit** Minor-mode für strukturelles Editing

**Undo-Tree** Baum-Artige Visualisierung der Edit-Geschichte

# Beispielpakete, nicht Vorinstalliert

**AucTeX** Ausgezeichneter  $\text{\LaTeX}$  Major-mode

**Magit** Ausgezeichnete Git Oberfläche

**Avy** Funktionen um im Text herumzuspringen

**Helm oder Ivy** Verbesserter Input Modus für Mini-Buffer  
Anfragen

**Paredit** Minor-mode für strukturelles Editing

**Undo-Tree** Baum-Artige Visualisierung der Edit-Geschichte

**Yasnippet** Sammlung von "Snippets" für verschiedene  
Major-Modes

# Danke für's Zuhören.

---

Andrei Alexandrescu, auf die Frage "What's your favourite Editor:"

*Well, quite honest, I think all of these [...] debates are kind of silly, which is "the" Editor, and stuff. I think, are we in highschool, or what...*

***Of course Emacs is best!***