

# TUSTEP

Konzept und ursprüngl. Fassung: *Heidi Hein* (Heidelberg),  
heidi.hein@adw.uni-heidelberg.de

Neue Parameter: *Matthias Schneider* (Trier),  
mail@m-schneider.eu

Stand: 04.10.2013

# Allgemeines zu Dateitypen

Texte, Programme, Segmente

# DATEIEN

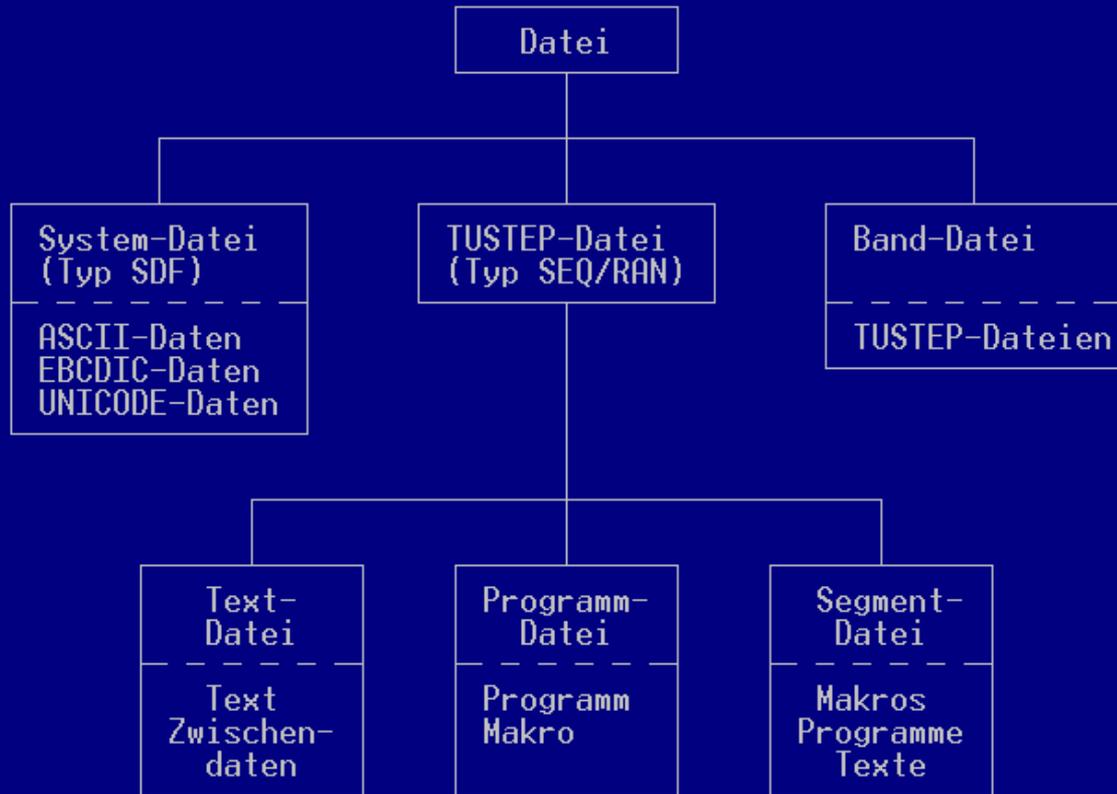
```
graph TD; A[DATEIEN] --> B[Permanente Dateien:]; A --> C[Scratch-Dateien:];
```

## Permanente Dateien:

- 1-12 Zeichen (Buchstaben, Ziffern, Minuszeichen und „\_“), müssen mit einem Buchstaben beginnen, können eine Extension haben (1-4 Zeichen)
- Bleiben erhalten
- Sind im Explorer unter diesem Namen sichtbar

## Scratch-Dateien:

- Bleiben nicht erhalten, sondern werden nach dem Beenden einer Sitzung gelöscht
- erhalten einen windowsinternen Standardnamen (8 Ziffern.TSF)
- können für Testversionen, Zwischenergebnisse u.ä. verwendet werden



# TUSTEP-Dateien

- Textdatei:
  - enthält Texte, Zwischendaten
  - ist im Textmodus nummeriert
- Programmdatei:
  - enthält eine Kommandofolge oder ein Kommandomakro
  - ist im Programm-Modus nummeriert
- Segmentdatei:
  - enthält einzelne Dateien (= Segmente)
  - am Anfang der Datei steht ein Inhaltsverzeichnis
  - Beispiele: pr.ueb: enthält Übungsprogramme  
tx.ueb: enthält Textdateien als Übungsmaterial

nie in Segmentdateien direkt arbeiten,  
sondern  
Segment rausholen, bearbeiten, zurückkopieren!!!

# Die Kommando-Ebene

# TUSTEP-Kommandos

## Textdatenverarbeitung:

- Edieren: #ediere, #kausfuehre
- Bearbeiten: #kopiere, #einfuege, #nummeriere
- Vergleichen: #vergleiche, #vaufbereite
- Register vorbereiten, aufbereiten: #rv, #ra
- Sortieren: #sv, #so, #spruefe
- Drucken vorbereiten, drucken: #formatiere, #dv, #dr
- Setzen: #satz

## Organisatorische Leistungen:

- Dateiverwaltung: #datei, #anmelde, #abmelde, #aendere, #loesche
- Ablaufsteuerung: #tue, #makro
- Archivierung, Sicherung: #mbeingabe, #mbausausgabe ...
- Datenaustausch/Konvertierungen: #umwandle

- # ist bei der Eingabe auf Kommandoebene nicht notwendig, innerhalb einer Programmroutine aber immer
- Abkürzungen möglich, sobald sie eindeutig sind (Ausnahme: #ediere)

# TUSTEP-Standardmakros:

- Aufruf auf der Kommandoebene, Kennzeichen \*
- vorgefertigte Abläufe, denen man bestimmte Spezifikationen zuordnen muss

Beispiele:

- `#*desi`: **D**efiniere **S**itzung
- `#*d / #*e`: **D**ateien einrichten, anmelden, abmelden = „kleiner **E**xplorer“
- `#*drube`: **D**rucke **B**eschreibung
- `#*tustep2stick`: Tustep auf einen USB-Stick kopieren
- `#*psaus`: **PS**-Ausgabe als Abschluss eines Satzlaufes
- `#*silmarke`: automatisches Eintragen von **Silbentrennungsmarkierungen** anhand einer eigenen Liste
- `#*satz`

Liste aller Standardmakros: `#informiere, ma=-std-`

# Die wichtigsten Kommandos

#<kommando> , <spezifikation1> , <spezifikation2> , <spezifikation3> ,  
    <spezifikation4> ...

## ➤ **Dateien kreieren/einrichten:**

#da , <name>: Einrichten einer Scratch-Datei

#da , <name> , seq-ap: Einrichten einer TUSTEP-Datei mit sequentiellem  
Zugriff, anmelden, falls vorhanden, permanent

## ➤ **Dateien anmelden:**

#an , <name>: Zum Lesen anmelden

#an , , <name>: Zum Schreiben anmelden

#an , , +: alle Dateien im Verzeichnis zum Schreiben anmelden

#an , , + , po= | txt |: alle Dateien, die im Dateinamen „txt“ enthalten, zum  
Schreiben anmelden

## ➤ **Dateien abmelden:**

#ab , <name>

## ➤ **Dateien löschen:**

#lo , <name>: labelt die Datei, d.h. ihr Inhalt wird gelöscht, nicht aber die  
Datei selbst

#lo , , <name>: Datei wird gelöscht

# Die wichtigsten Kommandos

#<kommando>, <spezifikation1>, <spezifikation2>, <spezifikation3>,  
<spezifikation4> ...

➤ **Dateien bearbeiten:**

z.B.: #ko, <quelle>, <ziel>, mo=+, lo=+, pa=\*

➤ **Dateien vergleichen:**

#ve, <versionA>, <versionB>, mo=Wort, pr=<datei>

➤ **Dateien aufrufen:**

#ed, <datei>, mo=p: Datei im Programm-Modus edieren (nur wichtig bei neu angelegten Dateien)

#ed, <datei>, mo=t: Datei im Textmodus edieren (nur wichtig bei neu angelegten Dateien)



# EDITOR

# Datensatznummer: Seitenzahl und Zeilenzahl

```
TUSTEP dh DH
Datei DH*MARTCAP.QU PCKPH715
26.1 |<nr value = 26> his dictis Virtus praecedentis officio ac
      |Mercurialis virgae perflatione concussa in caelum itura
      |sustollitur. augurales vero alites ante currum Delio
      |constiterunt, uti quis vellet vectus ascenderet; nam futura
      |plerumque conformans his praesagire consueverat. petaso autem
      |ac talaribus concitatis coepit praeire Mercurius; sed scandente
      |Phoebo Musarum pedisecus adhaerens que comitatus candenti
      |canora que alite vehebatur.</nr>
26.2 |
27.1 |<nr value = 27> tum vero conspiceres totius mundi gaudia
      |convenire. nam et Tellus floribus luminata, quippe veris deum
      |conspexerat subvolare Mercurium, et Apolline conspicato aera
      |Temperies sudis tractibus renidebat. superi autem globi orbes
      |que septemplices suavis cuiusdam melodiae harmonicis tinnitibus
      |concinebant ac sono ultra solitum dulciore, quippe Musas
      |adventare praesenserant; quae quidem singillatim circulis
      |quibusque metatis, ubi suae pulsum modulationis agnoverant,
      |constiterunt. nam Vranie stellantis mundi sphaeram extimam
      |continatur, quae acuto raptabatur sonora tinnitu,</nr>
27.2 |

*=27.2 Gib Anweisung >
**;01 ENTER INSERT 15:00
```

Anweisungszeile  
(Command line)

Meldungszeile  
(Message line)

Uhrzeit

Unterscheidungsnummern

Textfeld

The screenshot shows a terminal window titled "TUSTEP dh DH". The main content area has a dark blue background with white text. At the top, a red bar contains "Datei DH\*MARTCAP.QU" on the left and "PCKPH715" on the right. The text in the terminal is as follows:

```
26.1 |<nr value = 26>  
26.1/1 |his dictis Virtus praecedentis officio ac Mercurialis virgae  
|perflatione concussa in caelum itura sustollitur. augurales  
|vero alites ante currum Delio constiterunt, uti quis vellet  
|vectus ascenderet; nam futura plerumque conformans his  
|praesagire consueverat. petaso autem ac talaribus concitatis  
|coepit praere Mercurius; sed scandente Phoebos Musarum  
|pedisequs adhaerens que comitatus candenti canora que alite  
|vehebatur.</nr>  
  
27.1 |  
27.1/1 |<nr value = 27>  
27.1/2 |tum vero conspiceres totius mundi gaudia convenire. nam et  
|Tellus floribus luminata, quippe veris deum conspexerat  
|subvolare Mercurium, et Apolline conspicato aerea Temperies  
|sudis tractibus renidebat. superi autem globi orbes que  
|septemplices suavis cuiusdam melodiae harmonicis tinnitibus  
|concinebant ac sono ultra solitum dulciore, quippe Musas  
|adventare praesenserant; quae quidem singillatim circulis  
|quibusque metatis, ubi suae pulsum modulationis agnoverant,  
|constiterunt. nam Vranie stellantis mundi sphaeram extimam  
|continatur, quae acuto raptabatur sonora tinnitu,</nr>
```

At the bottom, a green status bar contains the text: `*=26.1 Gib Anweisung >` followed by `** :01`, `ENTER`, `INSERT`, and `15:09`.

„Sternchenposition“

Status-Zeile

## Die wichtigsten Fakten und Tipps zur Navigation im Editor

- Immer nur „**bildschirmweise**“ denken
- „**abgeschickt ist abgeschickt**“, jedoch STRG+R für Reshow möglich
- Blättern mit Bild-hoch und Bild-runter schickt ab und ist **immer möglich**
- Scrollen mit der Maus schickt ab und ist **immer möglich**

## Die wichtigsten Fakten und Tipps zur Navigation im Editor:

Abschicken (Speichern):	Enter Strg + E Eingabetaste (nur bei Modus ENTER) Sowie F1, F2, F4, F7, F8, F11, F12 (verbunden mit Zeige-Anweisung)
Beenden:	B ESC F3
1 Bildschirm rückgängig:	STRG + R (reshow)

# Die Orientierung in bereits vorhandenen Daten

Anweisung	Taste bzw. Tastaturkombination	Wirkung
ZA	F1	Zeigt den Anfang der Datei
ZE	F2	Zeigt das Ende der Datei
ZU,*	F4	Zeigt die Sternchenposition in der Mitte des Bildschirms
ZB,*	F7	Zeigt die Sternchenposition am Ende des Bildschirms, also den Text BIS zum Sternchen
ZA,*	F8	Zeigt die Sternchenposition am Anfang des Bildschirms
	Bild hoch	Blättert von der Cursorposition nach oben
	Bild runter	Blättert von der Cursorposition nach unten
	Cursortasten, Mausclick	Bewegen des Cursors an die entsprechende Stelle
ZA,3.27		Zeigt den Datensatz 3.27 am Anfang des Bildschirms mit den nachfolgenden Datensätzen an
4.57		Zeigt den Datensatz 4.57 allein am Anfang des Bildschirms an

# Die Eingabe in bereits vorhandene Daten

- Einfaches Reinschreiben mitten in die Daten ist möglich, aber:
- Reinschreiben und den unteren Teil nach unten bzw. den oberen Teil nach oben wegschieben ist nur so lange möglich, wie der Datensatz, in den man gerade schreibt, auf dem Bildschirm komplett zu sehen ist. Dieser bleibt auf dem Bildschirm und kann erst nach dem Abschicken weggeblättert werden.
- Eingabe an den Schluss der Datei mit der Anweisung „ee“ wie „Eingabe am Ende“
- Einfügen der Windows-Zwischenablage möglich (ez), STRG+C und STRG+V möglich, Tustep merkt sich mehrere Zwischenablagen (ALT + C)
- Einfügen der Linux-Zwischenablage ist seit 2013 mittels des Makros `#*CB` möglich: `#*CB,IM[PORT]` , umgekehrt aus TUSTEP heraus: `#CB,EX[PORT]`; Vorgehen: Mittels ez oder STRG+C Zwischenablage Inhalt erfassen + auf Kommandoebene (!) den gewünschten Austausch vornehmen
- Erstellen und Einfügen von Textbausteinen (Plustaste, Mauszwischenablage, Tastaturmakros)

# Datensätze einfügen

## Datensatz splitten (Split):

Eingabetaste drücken (wenn Modus Split)

Cursor an die Trennstelle bewegen: erst Plus, dann Eingabetaste (Split)

## Datensätze verbinden (Join):

Mit Backspace nach oben

ALT + J

Ans Ende des Datensatzes gehen: erst Plus, Plus, dann Eingabetaste

Join hängt die aktuelle Zeile an die Zeile darüber

# Die Plustaste / STRG + B

Plustaste im Nummernblock drücken, loslassen, dann zweite Taste drücken! Doppeltes Plus ist (fast) immer das Gegenstück!

Plus, Eingabetaste	Split, Aufsplitten des Datensatzes
Plus, Plus, Eingabetaste	Join / Verbinden
Plus, Entf	Entfernt den Datensatz
Plus, Plus, Entf	Fügt den mit (Plus, Entf) entfernten Satz wieder ein
Plus, Pos1	Schneidet den Datensatz bis zur Pos 1 (Zeilenbeginn) aus
Plus, Plus, Pos1	Fügt den mit (Plus, Pos1) entfernten Text wieder ein
Plus, Ende	Schneidet den Datensatz bis zum Zeilenende aus
Plus, Plus, Ende	Fügt den mit (Plus, Ende) entfernten Text wieder ein, mehrfach möglich
Plus, Page Down	Fügt eine Leerzeile <i>über</i> der aktuellen Zeile ein
Plus, Plus Page Down	<i>Wiederholt</i> die komplette Zeile einmal darunter
Plus, Page up	<i>Löscht</i> die komplette aktuelle Zeile, nicht aber den Datensatz!!!
Plus, Plus, Page up	Fügt den mit (Plus, Page up) gelöschten Text ein
Plus, Minus	Schneidet aus bis zum nächsten Blank inclusive
Plus, Plus, Minus	Fügt den mit (Plus, Minus) gelöschten Text wieder ein

Erstellen und Einfügen von Textbausteinen (Plustaste)

```
TUSTEP Sitzung HHEIN
Datei HHEIN*HALLO PCKPH715
8 |   entsteht
9 |   jöentstehlajklökdfsjdlkafentstehjtlsdkfjlasdfjöasldfja
  | jalsdkfjalskdfjalösd f entsteht  sdfgsdfg sd g d f a s d f
10 |   entsteht
11 |   entsteht Tustep-Test-Textentsteht entsteht
12 | Das ist mein Tustep-Test-Text, der sogenannte TTT. Um eine zweite
13 | Zeile zu erreichen, entsteht kjkljhjkh kljhkljhkj TTT.
  | schreibeNonsenslkhkljhkj lkhklhkljhklh lkh jkhklh klhklj schreibe
  | ich jetzt entstehtnoch etwas Nonsens hinzu, damit der sogenannte
  | schreibeTTT. Tustep-Test-Textentsteht schreibe
  | schreibe
14 | auch die dritte Zeile entsteht. Denn ientstehcht brauche mehrere
  | Datensätze ;- ) ; - ) ; - ) ; - ) ; - ) ; - )
15 | der sogenannte TTT. Tustep-Test-Textentsteht schreibe der
  | sogenannte TTT. ; - ) schreibe ; - )
  | Tustep-Test-Textentstehchtaslödfkjaösldfjaösldfjalöskdfjöalskd
  | jföalskjflöaskjf a alösd f jaslödf jaösldfasdfasdfasdf asdf asdfadf
  | asfdDas schreibeist mein Tustep-Test-Text, der sogenannte TTT. Um eine
  | zweite schreibe alksdfjaskdf lslsdfjasl lasdkfasldf TTT. schreibe
  | schreibe
16 | Zeile zu erreichen, ich jetzt noch etwas hinzu, damit
*=16 Gib Anweisung >
13:16 ENTER INSERT 14:21
```

Einfügen von Textbausteinen (Mauszwischenablage)

```
TUSTEP Sitzung HHEIN
Datei HHEIN*HALLO PCKPH715
8 | entsteht
9 | jöntstehltajklökdfsjdlkafentstehjtlsdkfjlasdfjöasldfja
| jalsdkfjalskdfjalösdf entsteht sdfgsdfg sd g dfasd df
10 | entsteht
11 | entsteht Tustep-Test-Textentsteht entsteht
12 | Das ist mein Tustep-Test-Text, der sogenannte TTT. Um eine zweite
13 | Zeile zu erreichen,entsteht kjkljhjkh kljhkljhkj TTT.
| schreibeNonsenslkhljkhkj lkjhklhlkjhkjhklh lkjh jkhlkh klhlkj schreibe
| ich jetzt entstehtnoch etwas Nonsens hinzu, damit der sogenannte
| schreibeTTT. Tustep-Test-Textentsteht schreibe
| schreibe
14 | auch die dritte Zeile entsteht. Denn ientstehcht brauche mehrere
| Datensätze ;-)) ;-)) ;-)) ;-)) ;-)) ;-))
15 | der sogenannte TTT. Tustep-Test-Textentsteht schreibe der
| sogenannte TTT. ;-)) schreibe ;-))
| Tustep-Test-Textentstehchtaslödfkjaösldfjaösldfjalöskdfjöalskd'
| jföalskjflöaskjf a alösdfjaslödf jaösldfasdfasdfasdf asdf asdfadf
| asfdDas schreibeist mein Tustep-Test-Text, der sogenannte TTT. Um eine
| zweite schreibe alksdfjaskdf lsldsfjasl lasdkfasldf TTT. schreibe
| schreibe
16 | Zeile zu erreichen, ich jetzt noch etwas hinzu, damit
```

FIND << COPY COPY COPY >> DELETE << MOVE MOVE MOVE >> IGNORE

Einfügen von Textbausteinen (Mauszwischenablage): **Rechte** Maustaste

## Tastaturmakro mit der Haupttastatur

1. Definition eines Tastaturmakros in der Anweisungszeile:  
`Y,CA=SET_INS,|TEXTBAUSTEIN|`  
Erläuterung: „a“ ist der Buchstabe, der mit der STRG und der ALT-Taste gemeinsam gedrückt werden soll. Wenn das der Fall ist: erst auf Einfügemodus umschalten, dann den Text der zwischen | steht, schreiben
2. Aufrufen des Tastaturmakros: STRG+ALT+a  
(gleichzeitig)

## Tastaturmakro mit dem Nummernblock

1. Definition eines Tastaturmakros in der Anweisungszeile:  
`Y,N_2=SET_INS,|TEXTBAUSTEIN|`
2. Aufrufen des Tastaturmakros: Drücken der Zahl 2 im Nummernblock, NUM muss leuchten!

Einfügen von Textbausteinen (Tastaturmakros)

# Der TUSTEP-EDITOR

## Die wichtigsten Anweisungen

### ➤ **Texteingabe:**

ee: Eingeben am Ende der Datei

ez, \*: Einfügen der Windowszwischenablage (etwas mit STRG + C gemerktes) an der „Sternchenposition“, falls mit STRG+V nicht möglich

e, \*: Einfügen von max. 9 Zeilen

### ➤ **Umstellung von Datensätzen:**

u, pos1, pos2: Stellt den Satz pos1 hinter pos2 um

u, (pos1, pos2), pos3: Stellt den Bereich pos1 bis pos2 hinter pos3 um

### ➤ **Umnummerieren:**

u, , 1.1: Nummeriert so um, dass der erste Satz 1.1 hat

u, , 1: Nummeriert so um, dass der erste Satz 1 hat

u!: Nummeriert so um, dass alle Zwischendatensätze einsortiert werden, also keine „/“ mehr

# Der TUSTEP-EDITOR

## Die wichtigsten Anweisungen

### ➤ **Text kopieren:**

`k,ber,pos1`: Kopieren eines Bereiches hinter `pos1`

`k,datei,ber,pos1`: Kopieren eines Bereiches aus der Datei `<datei>` hinter `pos1`

### ➤ **Löschen von Datensätzen:**

`l!,ber`: Löschen eines Bereichs (Das Ausrufezeichen verhindert unabsichtliches Löschen!)

### ➤ **Beenden des Editors:**

F3

ESC

B

# Weitere nützliche Anweisungen und Funktionen im Editor

- D, <name>: wechselt in die Datei <name>, auch ohne Komma möglich
- D? : öffnet ein Popup mit den zuletzt im Editor bearbeiteten Dateien, über Pfeiltasten kann eine gewählt werden. In diese wird per Eingabetaste gewechselt (= F6, falls keine Änderungen in der TUSTEP.INI vorgenommen werden, seit Version 2008)
- G+: History-Funktion rückwärts (= F9)
- G-: History-Funktion vorwärts (= F10)
- GG: listet die letzten 80 Befehle im Popup
- GGA: listet die letzten Austauschweisungen, GGZ die Zeigeanweisungen usw.
- M,1: Teilt den Bildschirm in zwei Teile, die jeweils eigene Editorfenster darstellen, und wechselt in den oberen Teil (= F11)
- M,2: Teilt den Bildschirm in zwei Teile, die jeweils eigene Editorfenster darstellen, und wechselt in den unteren Teil (= F12)
- D: informiert im Popup über die Datei an sich (Typ, Größe, Änderungsdatum usw.)
- F: informiert im Popup über die Belegung der Funktionstasten
- Y: informiert im Popup über die definierten Makros

Arbeiten im Editor:  
Zeigen  
Austauschen  
Patternmatching

# Arbeiten im Editor: Zeigen (1)

„Separator“/Trennzeichen:

Anweisungszeile: ZN , , , | Suchtext |

- Trennt die Suchzeichenfolge von der Anweisung
- Trennt einzelne Suchzeichenfolgen voneinander
- Beliebig wählbar

Begrenzung des zu durchsuchenden Bereichs innerhalb der Datei

Z.B. von Datensatz 3.15-4.27

zn , ( 3 . 15 , 4 . 27 ) , , | Suchtext |

Begrenzung des zu durchsuchenden Bereichs innerhalb eines Datensatzes

Z.B. von Position 4-28:

zn , , 4 - 28 , | Suchtext |

1.1 |Plötzlich drängte Johannes aus dem Strom der Promenierenden  
|hinaus und ging auf ein Tischchen zu, an dem noch niemand saß.  
|Giovanni hatte das  
1.2 |Gefühl, die Stühle dieses Cafés seien für ihn zu klein, aber  
|Iwan saß schon. Er hätte auch nie einen Platz in der ersten  
|Reihe genommen. So  
1.3 |dicht an den in beiden Richtungen Vorbeiströmenden sah man doch  
|nichts. Er hätte sich möglichst nah an die Hauswand gesetzt.  
|Hans saß auch schon.  
1.4 |Zu Giovannis Füßen. Hans sah aber noch zu Jean herauf, als  
|wolle er sagen, er betrachte sein Sitzen, so lange Joanes sich  
|noch nicht gesetzt habe, als  
1.5 |vorläufig. Johann bestellte schon den Kaffee, legte ein Bein  
|über das andere und schaute dem trägen Durcheinander auf der  
|Uferpromenade mit  
1.6 |einem Ausdruck des Vergnügens zu, der ausschließlich für Jean  
|bestimmt war. Ioanes verlegte seinen Blick auch wieder auf die  
|Leute, die zu dicht an  
1.7 |ihm vorbeipromenierten. John sah wenig. Von dem wenigen aber  
|zuviel. Er verspürte eine Art hoffnungslosen Hungers nach  
|diesen hell- und  
1.8 |leichtbekleideten Braungebrannten. Die sahen hier schöner aus  
|als daheim in Stuttgart. Von sich selbst hatte Juan dieses  
|Gefühl nicht. Er kam sich in  
1.9 |hellen Hosen komisch vor. Wenn Giovanni keine Jacke anhatte,  
|sah man von ihm wahrscheinlich nichts als seinen Bauch. Nach  
|acht Tagen würde ihm das egal  
1.10 |sein. Am dritten Tag noch nicht. So wenig wie die gräßlich  
|gerötete Haut. Nach acht Tagen würden Jan und Hanns auch braun  
|sein. Bei Sabine

1.1 | Plötzlich drängte Johannes aus dem Strom der Promenierenden  
| hinaus und ging auf ein Tischchen zu, an dem noch niemand saß.  
| Giovanni hatte das  
1.5 | vorläufig. Johann bestellte schon den Kaffee, legte ein Bein  
| über das andere und schaute dem trägen Durcheinander auf der  
| Uferpromenade mit  
1.7 | ihm vorbeipromenierten. John sah wenig. Von dem wenigen aber  
| zuviel. Er verspürte eine Art hoffnungslosen Hungers nach  
| diesen hell- und  
1.17 | fand sie interessant. War es auch. Aber nicht auszuhalten.  
| Hanns hatte sich vorgenommen, Johns Tagebücher zu lesen. Hannes  
| hatte alle fünf Bände  
1.18 | dabei. Wehe dir, Jean, wenn er nur vier Bände schafft. Hans  
| wußte überhaupt nicht, was Johannes in seinen Tagebüchern  
| notiert hatte.  
1.19 | Unvorstellbar, daß John etwas Privates notiert haben konnte.  
| Juan sehnte sich danach, Kierkegaard näherzukommen. Vielleicht  
| sehnte er sich  
1.25 | \$ Ans Wasser wollte er Johns Bücher nicht mitnehmen. Das hatte  
| er als Fünfzehnjähriger getan. Zarathustra hatte er auf dem  
| Bauch liegend  
1.27 | \$ Sabines Vergnügen an den Vorbeiströmenden hatte inzwischen  
| ein Lächeln erzeugt, das sich nicht mehr änderte. Johannes  
| genierte sich für  
1.31 | Umwelt über ihn und Sabine dachte, es sollte falsch sein.  
| Sobald es John gelang, Fehlschlüsse zu befördern, fühlte er  
| sich wohl. Inkognito war  
1.59 | Lächeln, das Frau Zürn, sobald er und Sabine auftauchten,  
| produzierte, nichts mit ihr zu tun hatte. Um so besser. Ihr  
| Mann Johann hatte in elf Jahren

\*\*\*\*\*

Dateiende

10 Zchflg. gefunden

10 Sätze gefunden

63 durchsucht

\*=2.3 Gib Anweisung &gt;

\*\*;01

ENTER

INSERT

16:27

## Arbeiten im Editor: Zeigen (2)

zn: Zeige nur = Die Datensätze werden angezeigt, die die gesuchte Zeichenfolge enthalten, und zwar nur die Datensätze. Mehrere Treffer werden gleichzeitig angezeigt. (Vergleichbar mit KWOC)

zu: Zeige Umgebung = Die Datensätze, die die gesuchte Zeichenfolge enthalten, werden einzeln nacheinander angezeigt. Jeweils ein Treffer in der Mitte des Bildschirms mit Umgebung. (Vergleichbar mit KWIC)

zb: Zeige bis = Die Datensätze, die die gesuchte Zeichenfolge enthalten, werden einzeln nacheinander angezeigt. Jeweils ein Treffer am Ende des Bildschirms und alle Datensätze davor, die noch auf den Bildschirm passen.

za: Zeige am Anfang = Die Datensätze, die die gesuchte Zeichenfolge enthalten, werden einzeln nacheinander angezeigt. Jeweils ein Treffer am Anfang des Bildschirms und alle Datensätze dahinter, die noch auf den Bildschirm passen.

**Als Zeigeanweisung immer 3 Kommata, Separatoren und Suchzeichenfolge!!!**

## Arbeiten im Editor: Zeigen (3)

Ausnahmezeichenfolgen: Zeige mir das eine nicht, aber das andere:

```
zn , , , | | Ausnahme | | Suchtext |
```

Mit doppelten Separatoren wird die Ausschlusszeichenfolge eingerahmt.

```
zn , , , | | Johann | | Johannes |
```

## Arbeiten im Editor: Austauschen

Anweisungszeile:

```
a , , , | Suchtext | Austauschtext |
```

```
a , , , | Johannes | johnny |
```

**Wichtig: Die angegebene Austauschzeichenfolge wird 1:1 übernommen, also auch Groß-/Kleinschreibung!!!**

## Arbeiten im Editor: Patternmatching (1)

- Was ist Patternmatching?

**Pattern Matching** (engl. für **Musterabgleich**) ist ein mathematischer Suchalgorithmus. Das Verfahren ermittelt in endlicher Zeit, ob sich ein gegebenes [Muster](#) (Pattern) in einem (begrenzten) Suchbereich wiederfindet. Im Gegensatz zur [Mustererkennung](#) (dem Finden von Mustern in Signalen) wird der Pattern vorher angegeben.

In einigen [funktionalen](#) oder [logischen](#) Programmiersprachen wird es genutzt, um Daten anhand ihrer Struktur zu verarbeiten.

Es wird auch verwendet, um Text zu bearbeiten. In Programmiersprachen wie [Perl](#) oder [awk](#) und auch in den meisten [Texteditoren](#) existieren Werkzeuge um einen Text nach einem Muster zu durchsuchen. Die Muster bestehen aus [regulären Ausdrücken](#).

Quelle: Wikipedia

In der [Informatik](#) ist ein **Regulärer Ausdruck** (Abk. **RegExp** oder **Regex**, engl. *regular expression*) eine [Zeichenkette](#), die der Beschreibung von [Mengen](#) beziehungsweise Untermengen von Zeichenketten mit Hilfe bestimmter [syntaktischer](#) Regeln dient. Reguläre Ausdrücke finden vor allem in der [Softwareentwicklung](#) Verwendung; für fast alle [Programmiersprachen](#) existieren Implementierungen.

Reguläre [Ausdrücke](#) stellen erstens eine Art *Filterkriterium* für Texte dar, indem der jeweilige reguläre Ausdruck in Form eines Musters mit dem Text abgeglichen wird. So ist es beispielsweise möglich, alle Wörter, die mit *S* beginnen und mit *D* enden, zu *matchen* (von englisch „to match“ – „auf etwas passen“, „finden“), ohne die zwischenliegenden Buchstaben explizit vorgeben zu müssen.

Ein weiteres Beispiel für den Einsatz als [Filter](#) ist die Möglichkeit, komplizierte Textersetzungen durchzuführen, indem man die zu suchenden Zeichenketten durch reguläre Ausdrücke beschreibt.

Zweitens lassen sich aus regulären Ausdrücken, als eine Art *Schablone*, auch Mengen von Wörtern erzeugen, ohne jedes Wort einzeln angeben zu müssen. So lässt sich beispielsweise ein Ausdruck angeben, der alle denkbaren Zeichenkombinationen (*Wörter*) erzeugt, die mit *S* beginnen und mit *D* enden.

Quelle: Wikipedia

## Arbeiten im Editor: Patternmatching (2)

Ausdrücke über die Qualität eines Zeichens:

- Beliebiger Großbuchstabe
- Beliebiger Kleinbuchstabe
- Beliebige Ziffer
- Beliebiger Buchstabe
- Beliebiges Zeichen
- ...

Ausdrücke über die Quantität eines Zeichens:

- Beliebig viele
- Mindestens
- Genau
- ...

Bei Zahlen:

- Größer als
- Kleiner als
- Zwischen zwei Werten

## Arbeiten im Editor: Patternmatching (3)

Beschreibung	WORD	TUSTEP (neues Pattern Matching)
Beliebiges Zeichen	^?	?
Beliebige Ziffer	^#	{\0}
Beliebiger Buchstabe	^\$	{&a}
Beliebiger Großbuchstabe	„Auf Groß-/Kleinschreibung achten“ anklicken	{\A}
Beliebiger Kleinbuchstabe	„Auf Groß-/Kleinschreibung achten“ anklicken	{\a}
Bestimmter Großbuchstabe z.B. A oder M	„Auf Groß-/Kleinschreibung achten“ anklicken	\A oder \M
Bestimmter Kleinbuchstabe z.B. a oder m	„Auf Groß-/Kleinschreibung achten“ anklicken	\a oder \m

## Arbeiten im Editor: Patternmatching (4)

Beschreibung	TUSTEP
Beliebig viele	{00}
Kann vorkommen, muss aber nicht (1 oder 0)	{0}
Mindestens, z.B. 3, möglichst viele	{3--0}
Mindestens, z.B. 3, möglichst wenige	{3-0}
Genau n, z.B. 4	{4}
Notwendige Escape-Folgen für bestimmte Einzelzeichen	\?, \*, \{, \}, \[, \], \\ \\
Anzeige des neuen Pattern Matching im Editor	STRG+K+Leertaste

## Arbeiten im Editor: Patternmatching (5)

Beschreibung	TUSTEP
n-tes Element von links	{+n=}
n-tes Element von rechts	{-n=}
Umschalten in der Suchzeichenfolge auf einen neuen Elementbereich	{ }
Rechter Rand	{}
Linker Rand	{}

## Arbeiten im Editor: Patternmatching (6), Ersetzungszeichenfolgen

Beschreibung	TUSTEP
Verweise	Wie in Suchzeichenfolgen, z.B. $\{+1=\}$ oder $\{-3=\}$
Verweis auf gesamte Suchzeichenfolge/ Kernzeichenfolge	$\{=0=\}$
Element/ Suchzeichenfolge in Groß-/Kleinbuchstaben austauschen	$\{....+\}$ , $\{....-\}$ , Bsp.: $\{=0+\}$ (gesamter Suchstring in Großbuchstaben)
m-tes bis n-tes Element von rechts/links gezählt	$\{+m-n=\}$ , $\{-m-n=\}$ , Bsp.: $\{+3-5=\}$ , $\{-5-9=\}$

## Arbeiten im Editor: Patternmatching (7)

Kombination von Qualitäten:

Zeige nur die Großbuchstaben:

```
zn, , , | {\A} |
```

Zeige nur die Kleinbuchstaben:

```
zn, , , | {\a} |
```

Zeige nur Großbuchstaben, denen eine Ziffer folgt:

```
zn, , , | {\A} {\0} |
```

Zeige nur die Zeichenfolge Großbuchstabe, 2 Kleinbuchstaben, 1 Ziffer:

```
zn, , , | {\A} {2} {\a} {\0} |
```

## Arbeiten im Editor: Patternmatching (8)

Kombination von Qualitäten und Quantitäten:

Zeige mindestens 4, aber möglichst viele, Großbuchstaben:

```
zn , , , | {4--0} {\A} |
```

Zeige nur Großbuchstaben, denen 4 Ziffern folgen:

```
zn , , , | {\A} {4} {\0} |
```

Zeige nur die Zeichenfolge Großbuchstabe, fakultativer Punkt, Ziffer:

```
zn , , , | {\A} {0} . {\0} |
```

## Arbeiten im Editor: Patternmatching (9)

Verweise:

Zeige mir Datensätze mit Doppelbuchstaben:

zn , , , | {&a} {+1=} |

zn , , , | {&a} {-2=} |

Ebenso im Austausch:

Tausche alle Wörter mit Doppelbuchstaben so aus, dass die Doppelbuchstaben durch \* umrahmt sind:

a , , , | {&a} {+1=} | \\* {=0=} \\* |

# Individualisierungen im Editor

- Farbeinstellungen für die Arbeitsbereiche: STRG + F
- Größen, Längen des Bildschirms: STRG + L
- Cursor und Signalton: STRG + G
- Colorierungen

# TUSTEP Helpdesk

- Handbuch als PDF im Installationsverzeichnis + per `#*ZEBE` (Kommandoebene)
- Handbuch aufrufen mit `#hilfe` (Kommandoebene) oder `STRG + O` (Editor)
- Durchsuchbares Handbuch: `#suche`
- TUSTEP-Wiki (<http://tustep.wikispaces.com/>)