

# Linux, L<sup>A</sup>T<sub>E</sub>X, Python

Felix, Felix, Daniel

Fachschaftsrat Mathematik

April 11, 2017

# Die Shell

- Die Kommandozeile erlaubt die direkte Kommunikation mit dem Rechner.

# Die Shell

- Die Kommandozeile erlaubt die direkte Kommunikation mit dem Rechner.
- Die Verwendung ist optional, aber sehr empfehlenswert.

# Die Shell

- Die Kommandozeile erlaubt die direkte Kommunikation mit dem Rechner.
- Die Verwendung ist optional, aber sehr empfehlenswert.
- Eine Shell ist auch eine vollwertige eigene Programmiersprache.

# Grundbefehle

- ls – Listet den Inhalt eines Verzeichnisses auf

# Grundbefehle

- ls – Listet den Inhalt eines Verzeichnisses auf
- cd – wechselt aktuelles Verzeichnis

# Grundbefehle

- ls – Listet den Inhalt eines Verzeichnisses auf
- cd – wechselt aktuelles Verzeichnis
- cat – gibt den Inhalt einer Datei auf der Konsole aus

# Grundbefehle

- ls – Listet den Inhalt eines Verzeichnisses auf
- cd – wechselt aktuelles Verzeichnis
- cat – gibt den Inhalt einer Datei auf der Konsole aus
- cp – Kopiert Datei



# Grundbefehle

- ls – Listet den Inhalt eines Verzeichnisses auf
- cd – wechselt aktuelles Verzeichnis
- cat – gibt den Inhalt einer Datei auf der Konsole aus
- cp – Kopiert Datei
- mv – verschiebt Datei

# Grundbefehle

- ls – Listet den Inhalt eines Verzeichnisses auf
- cd – wechselt aktuelles Verzeichnis
- cat – gibt den Inhalt einer Datei auf der Konsole aus
- cp – Kopiert Datei
- mv – verschiebt Datei
- rm – Löscht eine Datei

# Grundbefehle

- ls – Listet den Inhalt eines Verzeichnisses auf
- cd – wechselt aktuelles Verzeichnis
- cat – gibt den Inhalt einer Datei auf der Konsole aus
- cp – Kopiert Datei
- mv – verschiebt Datei
- rm – Löscht eine Datei
- rmdir – lösche (leeres) Verzeichnis

# Grundbefehle

- ls – Listet den Inhalt eines Verzeichnisses auf
- cd – wechselt aktuelles Verzeichnis
- cat – gibt den Inhalt einer Datei auf der Konsole aus
- cp – Kopiert Datei
- mv – verschiebt Datei
- rm – Löscht eine Datei
- rmdir – lösche (leeres) Verzeichnis
- mkdir – erstelle leeres Verzeichnis

# Grundbefehle

- ls – Listet den Inhalt eines Verzeichnisses auf
- cd – wechselt aktuelles Verzeichnis
- cat – gibt den Inhalt einer Datei auf der Konsole aus
- cp – Kopiert Datei
- mv – verschiebt Datei
- rm – Löscht eine Datei
- rmdir – lösche (leeres) Verzeichnis
- mkdir – erstelle leeres Verzeichnis
- echo – gibt Argumente an die Standardausgabe

# Grundbefehle

- ls – Listet den Inhalt eines Verzeichnisses auf
- cd – wechselt aktuelles Verzeichnis
- cat – gibt den Inhalt einer Datei auf der Konsole aus
- cp – Kopiert Datei
- mv – verschiebt Datei
- rm – Löscht eine Datei
- rmdir – lösche (leeres) Verzeichnis
- mkdir – erstelle leeres Verzeichnis
- echo – gibt Argumente an die Standardausgabe
- grep – sucht nach einer Zeichenkette

# Grundbefehle

- ls – Listet den Inhalt eines Verzeichnisses auf
- cd – wechselt aktuelles Verzeichnis
- cat – gibt den Inhalt einer Datei auf der Konsole aus
- cp – Kopiert Datei
- mv – verschiebt Datei
- rm – Löscht eine Datei
- rmdir – lösche (leeres) Verzeichnis
- mkdir – erstelle leeres Verzeichnis
- echo – gibt Argumente an die Standardausgabe
- grep – sucht nach einer Zeichenkette
- man - Zeigt Dokumentation von Kommandos an

# Tab-Completion

Das drücken der TAB Taste erspart euch viel Ärger und Arbeit, denn es vervollständigt Kommandos

Beispiel:

- TAB nach ls: Vervollständigung von Dateien



# Tab-Completion

Das drücken der TAB Taste erspart euch viel Ärger und Arbeit, denn es vervollständigt Kommandos

Beispiel:

- TAB nach ls: Vervollständigung von Dateien
- TAB nach cd: Vervollständigung von Verzeichnissen

# History

- ↑-Taste ruft das zuletzt eingegeben Kommando auf

# History

- ↑-Taste ruft das zuletzt eingegeben Kommando auf
- Funktioniert auch Rekursiv

# History

- $\uparrow$ -Taste ruft das zuletzt eingegeben Kommando auf
- Funktioniert auch Rekursiv
- $\downarrow$ -Taste funktioniert dementsprechend in der anderen Richtung

# History

- ↑-Taste ruft das zuletzt eingegeben Kommando auf
- Funktioniert auch Rekursiv
- ↓-Taste funktioniert dementsprechend in der anderen Richtung
- Ctrl-r startet eine Suche nach Befehlen

# History

- $\uparrow$ -Taste ruft das zuletzt eingegeben Kommando auf
- Funktioniert auch Rekursiv
- $\downarrow$ -Taste funktioniert dementsprechend in der anderen Richtung
- Crtl-r startet eine Suche nach Befehlen
- Eine Begonnene Eingabe kann mit Alt-p vervollständigt werden.

# History

- ↑-Taste ruft das zuletzt eingegeben Kommando auf
- Funktioniert auch Rekursiv
- ↓-Taste funktioniert dementsprechend in der anderen Richtung
- Ctrl-r startet eine Suche nach Befehlen
- Eine Begonnene Eingabe kann mit Alt-p vervollständigt werden.
- Und das war nur die Spitze des Eisbergs

# Patterns

Manchmal will man nicht Alles ausschreiben, dafür gibt es Patterns.

Beispiel:



# Patterns

Manchmal will man nicht Alles ausschreiben, dafür gibt es Patterns.

Beispiel:

- `ls -l *` – Listet erweiterte Informationen zu aller Dateien im aktuellen Verzeichnis auf

# Patterns

Manchmal will man nicht Alles ausschreiben, dafür gibt es Patterns.

Beispiel:

- `ls -l *` – Listet erweiterte Informationen zu aller Dateien im aktuellen Verzeichnis auf

# Patterns

Manchmal will man nicht Alles ausschreiben, dafür gibt es Patterns.

Beispiel:

- `ls -l *` – Listet erweiterte Informationen zu aller Dateien im aktuellen Verzeichnis auf
- `ls -l *.jpg` – Listet erweiterte Informationen zu allen Dateien mit der Endung `.jpg` auf

# Patterns

Manchmal will man nicht Alles ausschreiben, dafür gibt es Patterns.

Beispiel:

- `ls -l *` – Listet erweiterte Informationen zu aller Dateien im aktuellen Verzeichnis auf
- `ls -l *.jpg` – Listet erweiterte Informationen zu allen Dateien mit der Endung `.jpg` auf

# Patterns

Manchmal will man nicht Alles ausschreiben, dafür gibt es Patterns.

Beispiel:

- `ls -l *` – Listet erweiterte Informationen zu aller Dateien im aktuellen Verzeichnis auf
- `ls -l *.jpg` – Listet erweiterte Informationen zu allen Dateien mit der Endung `.jpg` auf
- `ls *.???` – Listet alle Dateien mit einer dreibuchstabigen Endung auf

# Patterns

Manchmal will man nicht Alles ausschreiben, dafür gibt es Patterns.

Beispiel:

- `ls -l *` – Listet erweiterte Informationen zu aller Dateien im aktuellen Verzeichnis auf
- `ls -l *.jpg` – Listet erweiterte Informationen zu allen Dateien mit der Endung `.jpg` auf
- `ls *.???` – Listet alle Dateien mit einer dreibuchstabigen Endung auf

# Patterns

Manchmal will man nicht Alles ausschreiben, dafür gibt es Patterns.

Beispiel:

- `ls -l *` – Listet erweiterte Informationen zu aller Dateien im aktuellen Verzeichnis auf
- `ls -l *.jpg` – Listet erweiterte Informationen zu allen Dateien mit der Endung `.jpg` auf
- `ls *.???` – Listet alle Dateien mit einer dreibuchstabigen Endung auf
- `ls [a-z]*` – Listet alle Dateien auf, welche mit einem Kleinbuchstaben beginnen

# Patterns

Manchmal will man nicht Alles ausschreiben, dafür gibt es Patterns.

Beispiel:

- `ls -l *` – Listet erweiterte Informationen zu aller Dateien im aktuellen Verzeichnis auf
- `ls -l *.jpg` – Listet erweiterte Informationen zu allen Dateien mit der Endung `.jpg` auf
- `ls *.???` – Listet alle Dateien mit einer dreibuchstabigen Endung auf
- `ls [a-z]*` – Listet alle Dateien auf, welche mit einem Kleinbuchstaben beginnen



# Patterns

Manchmal will man nicht Alles ausschreiben, dafür gibt es Patterns.

Beispiel:

- `ls -l *` – Listet erweiterte Informationen zu aller Dateien im aktuellen Verzeichnis auf
- `ls -l *.jpg` – Listet erweiterte Informationen zu allen Dateien mit der Endung `.jpg` auf
- `ls *.???` – Listet alle Dateien mit einer dreibuchstabigen Endung auf
- `ls [a-z]*` – Listet alle Dateien auf, welche mit einem Kleinbuchstaben beginnen
- `ls eins-{x,y,z}` – Listet Dateien `eins-x`, `eins-y`, `eins-z` auf

# IO redirection

Es gibt verschiedene Kommunikationskanäle:

- Die Standardeingabe stdin (Nummer 0)

# IO redirection

Es gibt verschiedene Kommunikationskanäle:

- Die Standardeingabe `stdin` (Nummer 0)
- Die Standardausgabe `stdout` (Nummer 1)

# IO redirection

Es gibt verschiedene Kommunikationskanäle:

- Die Standardeingabe `stdin` (Nummer 0)
- Die Standardausgabe `stdout` (Nummer 1)
- Die Standard-Fehlerausgabe `stderr` (Nummer 2)

# IO redirection

Es gibt verschiedene Kommunikationskanäle:

- Die Standardeingabe `stdin` (Nummer 0)
- Die Standardausgabe `stdout` (Nummer 1)
- Die Standard-Fehlerausgabe `stderr` (Nummer 2)

Umleitungen:

- `stdout` und `stderr` werden üblicherweise auf das Terminal geschrieben.

# IO redirection

Es gibt verschiedene Kommunikationskanäle:

- Die Standardeingabe `stdin` (Nummer 0)
- Die Standardausgabe `stdout` (Nummer 1)
- Die Standard-Fehlerausgabe `stderr` (Nummer 2)

Umleitungen:

- `stdout` und `stderr` werden üblicherweise auf das Terminal geschrieben.
- Ausgaben können mit `<` und `>` umgeleitet werden.

# IO redirection

Es gibt verschiedene Kommunikationskanäle:

- Die Standardeingabe stdin (Nummer 0)
- Die Standardausgabe stdout (Nummer 1)
- Die Standard-Fehlerausgabe stderr (Nummer 2)

Umleitungen:

- stdout und stderr werden üblicherweise auf das Terminal geschrieben.
- Ausgaben können mit `<` und `>` umgeleitet werden.

Beispiele:

- `ls > datei`

# IO redirection

Es gibt verschiedene Kommunikationskanäle:

- Die Standardeingabe stdin (Nummer 0)
- Die Standardausgabe stdout (Nummer 1)
- Die Standard-Fehlerausgabe stderr (Nummer 2)

Umleitungen:

- stdout und stderr werden üblicherweise auf das Terminal geschrieben.
- Ausgaben können mit `<` und `>` umgeleitet werden.

Beispiele:

- `ls > datei` – schreibe stdout von ls in datei



# IO redirection

Es gibt verschiedene Kommunikationskanäle:

- Die Standardeingabe stdin (Nummer 0)
- Die Standardausgabe stdout (Nummer 1)
- Die Standard-Fehlerausgabe stderr (Nummer 2)

Umleitungen:

- stdout und stderr werden üblicherweise auf das Terminal geschrieben.
- Ausgaben können mit `<` und `>` umgeleitet werden.

Beispiele:

- `ls > datei` – schreibe stdout von ls in datei
- `ls 2> datei` – schreibe stderr von ls in datei

# IO redirection

Es gibt verschiedene Kommunikationskanäle:

- Die Standardeingabe stdin (Nummer 0)
- Die Standardausgabe stdout (Nummer 1)
- Die Standard-Fehlerausgabe stderr (Nummer 2)

Umleitungen:

- stdout und stderr werden üblicherweise auf das Terminal geschrieben.
- Ausgaben können mit `<` und `>` umgeleitet werden.

Beispiele:

- `ls > datei` – schreibe stdout von ls in datei
- `ls 2> datei` – schreibe stderr von ls in datei

# IO redirection

Es gibt verschiedene Kommunikationskanäle:

- Die Standardeingabe stdin (Nummer 0)
- Die Standardausgabe stdout (Nummer 1)
- Die Standard-Fehlerausgabe stderr (Nummer 2)

Umleitungen:

- stdout und stderr werden üblicherweise auf das Terminal geschrieben.
- Ausgaben können mit `<` und `>` umgeleitet werden.

Beispiele:

- `ls > datei` – schreibe stdout von ls in datei
- `ls 2> datei` – schreibe stderr von ls in datei
- `ls &> datei` – schreibe stdout und stderr ls in datei

# Pipes

Ausgaben und Eingaben von Kommandos können durch sogenannte Pipes geschrieben mit einem `—` verbunden werden.

Beispiel:

- `ls -l | grep log` – Gibt alle Dateien aus die log enthalten