

132 lines (91 sloc) 6.63 KB

# Setup

## 01 Command Line Intro

### Start

Bevor wir loslegen, müssen wir die Commandline kennenlernen. Sie ist so etwas wie unsere Hauptsteuerungszentrum und unsere neue Maus. Von hier aus starten wir Programme, auch unsere eigenen, wir halten unsere Software uptodate und installieren neue Software.

In dieser Erklärung braucht ihr [] nicht anzuschreiben. Also `cd [dirname]` bedeutet, dass ihr einfach `cd dirname` schreibt.

### Wir sind faul

- ↑ und ↓ scrollen uns zu alten Eingaben. Sehr nützlich.
- Mit `Tab` können Sachen autocompleted werden. Probiert es aus.
- Auf einem Mac könnte ihr auch mit Drag und Drop arbeiten, um Files und Folders hin und her zu bewegen.

### Navigation

- `pwd` : prints working directory — zeigt uns an, wo wir uns gerade befinden.
- `cd [dirname]` : changes directory, bewegt uns ins jeweilige Directory.
- `cd ..` : Bewegt uns eine Stufe rauf.
- `ls` : zeigt alle subdirectories an. `ls -lah` zeigt uns alles an.
- `locate [filename]` : versucht Files zu finden. Wenn es etwas nicht findet, bedeutet es nicht, dass es auf dem Gerät nicht ist.
- `find` : Dauert länger als `locate`, findet aber alles.
- `mv [source] [destination]` bewegt eine Datei von A nach B
- `cp [source] [destination]` kopiert eine Datei an einen anderen Ort, nun habt ihr also zwei identische Kopien derselben Datei.
- `rm [file]` deletes zerstört eine Datei, ihr findet sie auch nicht mehr im Papierkorb
- `mkdir [directoryname]` Baut ein neues Verzeichnis
- `rmdir [directoryname]` löscht ein Verzeichnis mit dem kompletten Inhalt
- `tar cvf [filename] [filename] [filename]...` komprimiert eine Datei `.tar.gz`
- `tar xvf [filename]` extrahiert eine Datei `.tar.gz` file

### Individuelle Files

- `cat [filename]` : displays the contents of a file

- `wc [filename]` : displays the word count of a file
- `wc -l [filename]` : displays the line count of a file
- `head -n 10 [filename]` : displays the first 10 lines of a file
- `tail -n 20 [filename]` : displays the last 20 lines of a file
- `more [filename]` : displays the contents of a file one screen at a time (spacebar to continue)
- `grep [text] [filename]` : show all of the lines in filename that contain text.
- `sort` : sorts the lines of a file
- `uniq` : removes duplicate adjacent lines of a file
- Und **wichtig**: Mit diesem wunderbaren Zeichen kann man Befehle kombinieren: `|`

## Lustiges

- `banner`
- `cowsay`
- `say`

## Abspeichern und Kombinieren

- `grep xyz 'file' > output.txt`
- `grep -E 'MyVariable = False|MyVariable = True' FormA.frm`

## Escaping commands

Has something gone wrong? Taking too long? Hold down Control and hit C, a.k.a. Ctrl+C.

## Mehr zu den einzelnen Befehlen lernen

- `man` pages are manuals that you can use from the command line.
- `man grep` would give you the entry for `grep`, and then you'd use the `d` and `u` to navigate up and down. You should probably just `googlegrep man page` or `grep examples`, though.

## 02 Installs

Hier installieren wir Zusatzprogramme und bereiten den Computer darauf vor, dass er in Zukunft einfacher Zusatzpakete installieren kann. Den Terminal öffnen und mit folgenden Befehlen die Software installieren. Manchmal dauern die Installs etwas lange. Geduld. Es funktioniert alles richtig. Und all das müssen wir nur einmal machen. Nicht mehr, aber auch nicht weniger.

### Weitere Install-Programme, jetzt auch für Linux

- **Pip3** ist das gängigste Install-Modul, das wir verwenden werden. Es lässt sich folgendermassen installieren: `pip3 install -U pip` zur Sicherheit installieren wird noch `pip install -U pip`. Wenn das nicht funktioniert hat versuche es mit `sudo pip install -U pip`.

### Install-Programme und Python3

**Mac OS X** (auf Linux sollte Python 3 schon installiert sein)

- Diese Install-Programme sind auf manchen Macs bereits vorinstalliert. Auf anderen nicht. Zur Sicherheit einfach diesen Befehl eingeben: `xcode-select --install`
- Mit **Homebrew** ist es einfacher, Pakete zu installieren. Der Befehl zum Initialsetup ist etwas kompliziert. Müssen wir aber nur einmal eingeben. `ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"`
- Und, damit alles etwas runde läuft, diesen Befehl:
- `echo 'export PATH=/usr/local/bin:$PATH' >> ~/.bash_profile`

- `source ~/.bash_profile`
- Und jetzt können wir mit Befehl Python 3 installieren: `brew install python3`

## Virtual Environments

- Wenn Du joggen gehst, ziehst Du Dich anders an, als wenn Du schwimmen gehst. Dasselbe gilt für Python. Um die unterschiedlichen Installationen von einander zu trennen, installieren wir sogenannte virtual environments. So: `pip3 install virtualenv` wenn es eine Fehlermeldung gegeben hat: `sudo pip3 install virtualenv`
- Wir müssen dem Computer zudem sagen, dass er virtual environments in Python3 verwendet, und nicht Python2. Das machen wir diesem extrem kompliziert anzusehenden Befehl `echo 'export VIRTUALENV_PYTHON=`which python3`' >> ~/.bash_profile` und `source ~/.bash_profile`.
- Jetzt noch ein paar Zusätze: `pip3 install virtualenvwrapper`, und dann wieder eine ganze Reihe komplizierter Befehle:
- `echo 'export VIRTUALENVWRAPPER_PYTHON=$VIRTUALENV_PYTHON' >> ~/.bash_profile`
- `echo 'export WORKON_HOME=$HOME/.virtualenvs' >> ~/.bash_profile`
- `export VIRTUALENVWRAPPER_SH_PATH=`which virtualenvwrapper.sh``
- `echo "source $VIRTUALENVWRAPPER_SH_PATH" >> ~/.bash_profile`
- `source ~/.bash_profile`
- So, und jetzt können wir virtuelle Umgebungen, virtual environments, kreieren. Versuche wir es mit diesem Befehl `mkvirtualenv erstesVE`
- Und dann `workon erstesVE`. Nun sollte in der Kommandozeile eine Anzeige auftauchen, dass Du Dich in einem virtual environment aufhältst, mit dem Namen VE. Von jetzt an brauchst Du auch nicht mehr `pip3` zu schreiben, da `pip` weiss, was er zu tun hat.

## Erste Zusatzpakete

- Installieren wir also BeautifulSoup, eine Python-Library, die wir später brauchen werden, um Websites zu lesen. Also: `pip install beautifulsoup4`.
- Und dann installieren wir noch Jupyter Notebook: `pip install jupyter notebook`.

## Github

- Anleitung zur [Github-Installation](#), den Rest werden wir morgen anschauen.
- `git clone https://github.com/MAZ-CAS-DDJ/kurs_18_19.git`

## Letzter Install (vorerst)

- Jetzt, ganz am Ende, installieren wir noch eine letzte Software: Den Texteditor [Atom](#). Das tun wir ganz normal. So, wie ihr es gewohnt seid, Software zu installieren.
- Wenn noch Zeit bleibt, kurzes Intro zu MD