Rückblick und Vertiefung Commandline

Navigation

Wir haben die Commandline oder den Terminal bereits gestern kennengelernt. Damit installieren wir neue Programme, wir halten alten aktuell, und wir können damit unsere Eigen-Kreationen, unsere eigene Programme auslösen. Das Terminal können wir auch dazu nutzen, sehr komplexe Aufgaben zu lösen.

Deshalb wollen wir uns dieses Werkzeug näher anschauen, bevor wir uns in die Python-Pogrammiersprache stürzen.

Üben wir zunächst etwas mit dem Terminal.

- Übung 1

Say

Aber die Commandline ist natürlich nicht nur dazu da, Dokumente herumzuschieben, oder neue zu erschaffen. Die Commanline kann viel mehr. Lassen wir den Computer damit reden.

- `say hello`
- `say hallo, ich bins`
- `man say` damit rufen wir das ganze Manual auf. Schauen wir uns an, was wir alles damit machen könnne. Mit `q` verlassen wir den Modus wieder.
- `say -v ?` oder `say voice ?` damit können wir alle gespeicherte Sprachen abrufen.
- Mit dem Befehl `say -f "namedesfiles"` können wir uns ganze Dokumente vorlesen lassen.

Wirkt der Computer nicht bereits menschlicher, weil er sprechen kann? Aber natürlich spricht er nur dann, wenn wir es ihm sagen. Noch beeindruckender ist, wenn der Computer scheinbar unaufgefordert zu uns spricht.

Wir können den Computer sagen, dass er gewisse Tätigkeiten zu genau bestimmten Zeiten ausführt. Mit derselben Funktionialität funktionieren Dutzende Dienste, und Apps. E-Mail-Dienste zum Beispiele. Sie sind je nach Einstellung einfach alle 15 Minuten oder jede Minute so eingestellt.

Crontabs

Damit wir dem Computer entsprechend einstellen können, müssen wir zuerst den entsprechenden Editor auswählen. Auch das können wir mit dem Terminal erledigen. Das ist das Eingabesystem für unser Gerät. Der gängigste Editor für Unix-Geräte (Apple und Linux) ist der VIM Editor. Dieser ist allerdings ziemlich kompliziert. Wir arbeiten besser mit dem Nano Editor.

- Dafür gebt ihr ein: `export EDITOR=nano`
- Nun öffnen wir den Crontab: ```crontab -e```
- Und wir geben die fünf * Zeichen ein, und dann `say "hello you"`
- Der Computer wird nun jede Minute "hello you" sagen.
- Das lässt sich natürlich verfeinern. Wie, erschliesst sich auf der Site Crontab.guru.
- Übung 2

So, nun wir können also bereits Aufgaben automatisieren. Das ist schon ziemlich viel.

wc

Gehen wir zurück auf die Commandline. Und schauen wir an, wie wir damit riesige Datensätze befragen und behandelnd können.

Wir haben gesten alle SNF-Projekte heruntergeladen. Für die, die nicht mehr wissen, wo diese Daten sind. Geht auf die SNF-Site: http://p3.snf.ch und klicken oben auf Daten und Dokumentation. Dann runter scrollen und P3_GrantExport.csv herunter laden. Ihr solltet am Ende ein File mit circa 38 MB auf eurem Gerät haben. Zieht es auf euren Desktop. Und versucht es mit Excel oder Google Spreadsheets aufzumachen. Nein, macht das nicht! Das Programm kann damit nicht umgehen. Das heisst aber nicht, dass es für euer Gerät ein Problem ist. Die Commandline kann damit locker umgehen.

- Zählen wir, wie viele Wörter das File hat: `wc P3_GrantExport.csv` .
- Das Ergebnis zählt eigentlich nicht nur die Wörter, sondern die Linien, die Wörter und die Zeichen.
- Zeigen wir nur die Zeilen an: `wc -l P3P3_GrantExport.csv` . Das gleich können wir nun auch für die Wörter: `wc -w P3P3_GrantExport.csv` oder die Zeichen `wc -m P3P3_GrantExport.csv` tun. Wer mehr für `wc` wissen will, googelt `wc unix` . So viel mehr kann wc aber nicht wirklich, aber es ist sehr nützlich, um sich über eine grössere Datensammlunge eine Übersicht zu verschaffen.

grep und Piping

Grep steht für "Globally search a Regular Expression and Print". Also: Ein bestimmtes Textmuster suchen und das Resultat dann ausdrucken. Macht das Sinn?

Versuchen wir es.

- Geben wir folgendes ein `grep "Geschichte" P3_GrantExport.csv`
- Wir bekommen ein Fenster voller Text in der Commandline. Diesen ganze Inhalt können wir abspeicher, folgendermassen `grep "Geschichte" P3_GrantExport.csv > geschichte.csv`
- Auf eurem Desktop sollte nun ein File mit dem Namen "geschichte.csv" erscheinen. Versucht das nun mit Excel oder Google Spreadsheets zu öffnen. Das macht schon sehr viel mehr Sinn.
- Wir können Befehle auch kombinieren. Also **wc** und **grep**. Mit dem Piping-Zeichen. Auf dem Mac findet ihr das Zeichen mit "alt" und 7.
- Dann geben wir ein ``grep "Geschichte" P3_GrantExport.csv | wc -l```
- Mit dieser Suche ist der Buchstabe Wichtig

Üben wir ein bisschen:

- Übung 3

Mit diesen Befehlen kann man theoretisch riesiege Dateien befragen. Und es gibt viele Forscher, die damit sehr umfangreiche Analysen machen. Wenn ihr mehr wissen wollt, schaut euch dieses Buch an.

Als nächstes wollen wir uns weitere Zusatz-Packages anschauen, um Daten kennenzulernen.