# Zusatzpackages Wget und CSVkit

Es gibt sehr viele Zusatzprogramme für das Terminal, alle mit sehr unterschiedlichen Funktionialtäten. Es lohnt sich in den Dokumentationen herumzustöbern, um die unterschiedlichen Programme kennenzulernen.

Wir wollen hier zwei kennenlernen: `wget` und `csvkit` und am Ende wollen wir alle Befehle mit einem Crontab kombinieren, um Files dann automatisch abzuspeichern.

## wget

Was ist wget? Um die Funktionen kennenzulernen googelt ihr am besten immer nach "wget documentation". Ihr stösst dann zum Beispiel auf darauf. Die kurze Definition von Wget ist "GNU Wget is a free utility for non-interactive download of files from the Web".

- Zuerst müsst ihr es installieren. Das geschieht mit `pip install wget` .
- Mit `wget --version` erfährt ihr, welche Version ihr nun installiert habt, und, wer eigentlich hinter Wget steckt.
- Mit `wget --help` könnte ihr alle Funktionen aufrufen. Das wird euch zunächst nicht viel sagen. Aber lasst euch nicht abschrecken. Ihr müsst nicht alles auf einmal wissen. Und vieles braucht ihr gar nicht zu wissen. Wenn ihr euch entschliesst, dass Wget für euch nichts taugt, müsst ihr euch gar nie damit befassen.
- Ich verwende wget beispielsweise, um regelmässig Snapshots von Websites oder von Facebook-Gruppen zu nehmen. z.B. `wget https://www.balthasar-glaettli.ch/`
- Viele Dienste erkennen Wget als Roboter, deshalb muss man manchmal mitgeben, dass es sich hier um einen Browser handelt, z.B.: `wget --user-agent=Firefox https://www.facebook.com/glaettli.ch`
- Für eine ganze Website, ist folgendes nötig. `wget --recursive --no-clobber --page-requisites --html-extension --convert-links --restrict-file-names=windows --no-parent https://www.balthasar-glaettli.ch`

Jetzt wollen wir das üben:

- Übung 4
- Wget ist ein sehr effizienter Scraper. Es lohnt sich damit zu arbeiten.
- Ihr habt vielleicht bemerkt, dass ihr beim bauen des Scrapers neue Dateien immer wieder über die alten abgespeichert habt. Damit ihr neue Files abspeichert, könnt ihr mit der eingebauten Uhr arbeiten. Ihr speichert Dateien also immer mit der aktuellen Uhrzeit und dem aktuellen Datum ab.
- `wget -O glaettlibeispiel.htm https://www.balthasar-glaettli.ch/` . Der Filename kommt zuerst, davor -O, und am Ende dann die URL.
- Mit dem folgenden Befehl ruft ihr in der Commandline das aktuelle Datum auf: `date +"%Y%m%d_%H%M%S"` .
- Nun muss alles verbunden werden: `wget -O `date +"%Y%m%d_%H%M%S".htm` https://www.balthasar-glaettli.ch/` .
- Ihr müsst jetzt nicht verzweifeln, weil ihr die ganze Syntax so unglaublich kompliziert findet. Im Verlaufe der Woche werdet ihr sehen, wie viel sich wiederholt.

## CSVkit

Als letzte Commandline-Zusatzsoftware wollen wir uns CSVKit anschauen. Es hilft uns dabei, grosse Datensammlungen zu verstehen. Ihr findet hier eine gute Dokumentation.

- Zuerst müssen wir das Programm installieren: `pip install csvkit` , das muss in einem Virtualenv passieren, das mit Python3 funktioniert. Das haben wir gestern installiert.
- Falls `pip install csvkit` nicht funktioniert, arbeiten wir mit `sudo pip install csvkit` .
- Arbeiten wit mit "P3_GrantExport.csv", die Zusammenstellung der Nationalfond-Projekte. Schauen wir uns das File mit `csvlook P3_GrantExport.csv` an. Das wird eine Weile dauern, denn das Programm wird alle Zellen im Terminal anzeigen wollen. Das ist also nicht wirklich hilfreich.
- Und das Ergebnis ist nicht wirklich übersichtlicht. Schauen wir mit `csvcut -n P3_GrantExport.csv` an, wie die Spalten heissen. Nicht sehr übersichtlich. Der Grund dafür ist, dass die Spalten nicht durch Kommas getrennt sind, sondern durch ";". Formatieren wir deshalb das ganze um. Das tun wir mit `csvformat -d ";" P3_GrantExport.csv > data.csv` .
- Nun arbeiten wir mit data.csv weiter. Geben wir `csvcut -n data.csv` ein. Und nun haben wir alle Spalten übersichtlich zusammen.
- Wählen wir die Spalten aus, die uns interessieren: `csvcut -c 5,8,9,10,17 data.csv > data_selected.csv`
- Und nun kommen wir zum magischen Befehl csvstat. Dieser Befehl macht eine kleine statistische Zusammenfassung des Inhalts der Datensammlung. `csvstat data_selected.csv` . Bei grösseren Datensammlungen braucht es ein paar Augenblick Geduld.
- Die ersten Ergebnisse sind eindrücklich. Aber es werden auch gleichzeitig die Grenzen von CSVStat ersichtlich. Die Zahlen zum Beispiel liest der Computer nicht als Zahlen, sondern als Text.

Und was wir jetzt lernen werden, ist selber Programme zu schreiben, die Daten reinigen und in eine Form bringen, damit wir sie befragen können. Wir können immer wieder auf csvstat zurückgreifen, wenn es schnell gehen soll. Aber in der Regel werdet ihr lieber selber einen Code schreiben, um Daten zu analysieren und zu manipulieren.

Als erstes werde ich euch in die Umgebung einführen, in der ihr Programmieren lernen werden. Wir werden Python lernen.

**Warum Python?** Es gibt dafür zwei einfache Antworten: Es ist erstens die von der Syntax einfachste Programmiersprache. Sie ist der menschlichen Sprache am nächsten. Und die Sprache ist mittlerweile die populärste Sprache im Internet überhaupt. Eine Analyse wie es in den letzten Jahren dazu gekommen, könnt ihr hier lesen. Die Grafiken zeigen, wie Python populäre Sprachen wir Javascript oder Java abgelöst haben. Auch R hat Python längst abgehängt. Ihr werdet oft auf R stossen. Die Sprache ist bei Datenjournalisten und vor allem in der Wissenschaft sehr populär, vorallem wenn es um Statistik geht. Python kann auch Statistik und sie ist ausserdem sehr viel flexibler einsetzbar, etwa wenn es darum geht, Sites zu scrapen oder mit Textdateien umzugehen.