1.1 Outline

- Exkurs: wieso braucht es das ganze Tooling?
  - Wissenschaft als Praxis
- Grundverständis für Mensch-Maschine-Interaktion
- Installation
  - abschliessen/Fehler beheben
  - Zeitplan für Installation schwierig abzuschätzen
  - Feierabend wenn System läuft

1.2 Recap last Lecture

- Bedeutung ist kontextabhängig
- Unvergleichbarkeit diskreter Symbole
- Ziel: Abstraktion + Kontextualität kombinieren
  - Quali + Quanti
  - Generalisierung + Rekontextualisierung
- Fragen
  - wichtigste Dateiformate?
    - txt, csv, tsv (xml)
  - Sinn von Texteditor?
    - keine Formatierung

2.2 The Zen of Organizing
2.3 Research means Organizing

- Wissenschaft auch praktische Seite, nicht nur theoretische
  - chaotisch statt strukturiert (aufgeräumte Papers kommen erst ganz am Ende)
  - Organisation von Komplexität als Problem
- Reproduzierbarkeit ist Arbeitsgrundlage und höchster Wert (mehr noch als Wahrheit)
  - Wiederverwendung von Code/Datensatz + Literatur/Theorien
- Wieso computational approach?
  - Reproduzierbarkeit und Kritisierbarkeit garantiert
  - Automatisierung von repetitiven Aufgaben -> spart Zeit
- Code strukturiert/dokumentiert Arbeitsablauf/Entscheidungen
  - gegen Vergessenheit
  - schützt nicht vor Fehler, aber sichert inkrementellen Fortschritt
- wichtig für grössere Projekte + Abschlussarbeiten
- kurzfristig langsamer, langfristig effizienter
  - keine mühsame Fehlersuche, Wiederholung

image-src

2.4 Organizing Literature

- Organisation betrifft auch Literatur
- Fragen, wer Literaturverwaltungssystem nutzt
  - was? wieso nützlich?
- Zotero
  - open-source, viele Features, konstante Weiterentwicklung
- Nutzen
  - verschiedene Zitationsstile
  - einmal indexiert, immer gleich
  - Recherche: Metadaten automatisch setzen
  - Bibliographie automatisch zusammenstellen

3.4 Computers
3.5 Two Trends in Computing

- Computer haben sich massiv verändert -> am meisten an Oberfläche
- Spannungsfeld zweier Trends
  - Einfachheit –> Eingeschränktheit; nur machen, was vorgesehen ist
  - Flexibilität –> technisches Vorwissen
- Apple: GUI/Bedienung Angleichung an physische Welt
- Engineering: schwieriger Einstieg/Lernen, dafür sehr viel effektiver/vielseitiger
  - ist alternativlos für Standardisierung unstandardisierter Daten
  - keine Zauberei, aber sehr nützlich
- von Wissen/Tools der Engineers profitieren
  - nicht selber Tools bauen
  - geniale Arbeitsabläufe + Modularität

3.6 Operating Systems (OS)

- OS
- Windows = Dominator, Alltag
- Linux = stable, secure, free, innovative. Became more user-friendly, sometimes still issue
- Mac = restricted to Mac HW, Unix-Derivat, vereint beide Welten
- Betriebssysteme wie Fahrräder oder Autos
- Systeme gleichen sich immer mehr an
  - Installation von Linux in Windows

3.7 User Interfaces

- historisch nur CLI zur Computersteuerung

- GUI von Apple entwickelt 1984, in 90er Standard
- CLI ist mehr als Sentimentalität
  - mächtiger dank Automatisierbarkeit
  - schneller
  - auf Server einzig mögliche Interaktion

3.8 "An awesome Programmer saves the World"****

- CLI auch in Popkultur, allerdings falsch dargestellt

3.9 Human-Machine Interaction

- Automatisierung über GUI nicht möglich
  - statt "hier" klicken, einfach Skript
- viele verschiedene Programmiersprachen
- Syntax sehr restriktiv
  - Computer sind ziemlich doof, aber sehr gehorsam.
  - Machen genau, was man ihnen sagt, nichts mehr, nichts weniger. Nie.

3.10 Programming

- CLI primär für Dateimanipulation/Verarbeitung
- Python
  - einfach: reduziert auf das elementare
  - Python is a general-purpose language whereas R is a statistical programming language.
- Software heisst neuerdings Algorithmus
  - falsch: "Algo als Böses/Mystisches"
  - korrekt: schrittweise Umwandlung von Input zu Output
  - software = program + data (complement HW)

3.11 Package Manager

- SW baut auf weiterer SW auf
  - keine vollständigen Programme
  - bei Installation hunderte von Kompatibilitäts-Checks
- zentrale Verwaltung installierter SW
  - update aller Programme mit einem Befehl
- 2 Manager: systemweit, Python
- app stores keine Innovation
- gehört auch zur Arbeitsorganisation

3.12 Open-Source is a Mindset

- Zusammen erreicht man mehr -> Abhängigkeiten

- Abhängigkeiten funktionieren am besten, wenn offen
  - Rückmeldungen -> gemeinsame Verbesserung
  - schnelle Weiterentwicklung
- CS offenste Disziplin
  - Private + Firmen
  - oft unentgeltlich

3.13 Resources everyone is using

- Kollaboration nicht nur für SW, auch für Fragenbeantwortung
- allermeiste Fragen/Probleme nicht neu, schon beantwortet
  - bash commands auf stackoverflow
  - installationsprobleme
- Github
  - source code + anleitung für Millionen von Programmen (klein und gross)
  - repository

3.14 Learning by doing, doing by Googleing. :woman_cartwheeling::man_cartwheeling:

- Technical problems are normal + solutions around the corner
  - Fehlermeldung lesen + googlen
- Without the internet, you are a nobody
- Installation is sometimes harder and much more poorly documented than mere usage

4.14 Set up your System
4.15 Backup :japanese_ogre:
4.16 Setting up your Development Environment

- verschiedene Installationsmöglichkeiten
  - Idee: plattformübergreifend, relativ einfach, uneingeschränkt
- Windows Leute installieren Ubuntu für Bash
- Installations Guide folgen
  - Verstehen aktuell egal
  - Fragen/Verbesserungsvorschläge willkommen
  - Ablauf verfolgen & auf Fehler achten

5.16 Relax. It takes a while.
5.17 VS Code Editor

- in-class demonstrieren
- Visual Studio: Code, Erklärung, Output an selbem Ort

5.18 First Steps in Python

- Python individuell vertiefen

5.19 Readings
6.19 Questions?
6.20 References