Scripting

- Scripting is a way of automating a sequence of commands to a computer system. (Shell commands, utility program commands...)
- Both text and graphical interfaces can be scripted.
- In its simplest form a script is a list of commands.
- To make it possible to repeat commands and choose different commands in different situations, many scripting languages include programming language features (data and control structures).
- The basic design of a scripting language can be command or programming oriented.
- Many recent scripting languages are more oriented towards rapidly constructing simple programs rather than executing commands.
  - simple programs - for many reasons scripting languages are not suited for larger programs.
Why (not) use scripting

- Good points
  - Automate command sequences.
  - Pre-process input to or post-process output from programs.
  - "Glue" different programs together.
  - Easy and fast to write and run scripts for simple functions.

- Bad points
  - Slow execution of the scripts themselves.
  - Unsuitable for large programs or functions which require complex data structures or algorithms.
  - Dubious portability, language definition connected to a particular implementation and version.

- Compare with a Swiss army knife or a multi tool:
  - Useful in a tight spot, better than nothing, but not really good at anything.

- Use care when determining whether to use a scripting language or a normal programming language.
“Features” of script languages

- (But certainly not every language has every feature!)
- Design motivated by the need to achieve something rather than sound design principles – no proper language definition.
- Very high-level in the problem domain (e.g. regular expressions).
- Poor and/or complicated data structures.
- Awkward syntax.
- Features which you have to try out to understand how they work.
- Interpreted languages – no compilation step but slow execution.
- Variables used by textually substituting their values into commands. (Error prone and a security risk!)
- Unsafe (no variable declarations, strong typing etc.)
Script languages

- **sh**
  - Unix shell. The classic! Ugly programming facilities built on top of the command function of the shell.

- **csh**
  - Unix shell with C-like programming facilities.

- **tcl/tk**
  - Programming language used to build GUIs. Extensible and embeddable.

- **perl**
  - A sh/sed/awk superset with many features, awkward syntax and semantics.

- **python**
  - High-level OO language. Good design. Inspired by functional and list processing languages.
Useful tools

- Typically these process take input from `stdin` and write results on `stdout`
- `awk` - pattern match based processing
- `sed` - "Stream editor"
- `fgrep, grep, egrep` - Search for regular expressions
- `wc` - count lines, words, characters
- `sort` - sort
- `uniq` - remove duplicate lines
- `cat` - copies/concatenates files/inputs
- `tee` - split a pipe
- Refer to the appropriate man pages for more info!