Musa Al-hassy https://github.com/alhassy/CheatSheet February 17, 2019

Reference Sheet for “What I’m Currently Learning”

Hello, World!

Pretty PDF Enter `M-x compile` to produce a nice looking PDF of your reference sheet.
- ◇ I’ve bound this command to `C-c C-m` in my Emacs setup ;-)

Section Headers A usual Org header, say `* my section`, results in the boxed headers used in this cheat sheet.

Parallel Environments The sequence `<p TAB` produces a ‘parallel’ environment for producing text side-by-side. The column break is automatic, but as this is sugar for a `minipage` containing a `multicolum` we can force a column separation with `\columnbreak`: This command, in Org, necessities newlines between the items being separated.

To learn more, manipulating this source is the way to go!

Also, opening this file produces a `README.md` ;-)

CheatSheet Examples

Reference sheets created from this project include,

CatsCheatSheet Listing of common theorems in elementary category theory.
LatticesCheatSheet Reference sheet for definitions and results in Lattice Theory.
CoqCheatSheet Reference sheet for the Coq language.

The steps to utilising this git project for your own cheat sheet may be:

1. Go to the repo you want to make a cheat sheet.
2. Add this project as a submodule then copy its core to where you’re working:
```
git submodule add https://github.com/alhassy/CheatSheet.git
; cp CheatSheet/CheatSheet.org .
; cp CheatSheet/README.org .
```
3. Open `CheatSheet.org` and locate `#+INCLUDE: CheatSheetSetup.org` then rewrite `CheatSheetSetup.org` → `CheatSheet/CheatSheetSetup.org`.
4. Within the `README.org`, if you’re using it, alter the regions marked `!!CHANGE ME!!`.

I don’t think this is difficult to automate, so I will likely get to doing it.

Basic Equational Support

Basic name-formula equational support. `\eqn{name}{formula}` yields a displayed equation with `formula` left aligned and `name` right aligned:

$$formula \qquad \text{NAME}$$

Moreover, we can refer to such a formula by invoking `\ref{name}` –e.g., NAME. However, if `name` involves unicode symbols, then this may cause problems.

Org-mode Basics

Read Org-mode for beginners for a refresher!
- ◇ For more see The Compact Org-mode Guide.

Reloading To reload a file with updated org settings, press `C-c C-c` on a settings line –i.e., one beginning with a `#+`, to reset the temporary file cache.

Inclusion During export, you can include the content of another file.
- ◇ Syntax: `#+INCLUDE: "fileName" [markup [language]]`
  - ○ `markup ::= src | example`
  - ○ `language ::= C | haskell | emacs-lisp |`
  - ○ If the markup is not given, the text will be assumed to be in Org mode format and will be processed normally; c.f., Setup files.
- ◇ To visit the file, `C-c '` while the cursor is on the line with the file name.
- ◇ Include only portions of a file by appending with `:lines "x-y"` where `x` is the first line and `y` is the second-to-last line. Also `"-y"` for upto but not including line `y`, and `"x-"` for taking line `x` until the end of the file.

Emacs

C-x r k M-x kill-rectangle
C-l C-l move buffer top to be current cursor location.
Delete a region of text, e.g., white space `C-SPC` at the beginning of the first line then `C-x r k` (rectangular kill) at the end of the last line of the (indentation) region you want to remove.

Git

Revert a file to a particular commit `git checkout 0cdf -- myfiles`
- ◇ Where `0cdf` is your commit identifier, which is usually much longer.

`git whatchanged` Like `git log` but informs exactly which files were altered.

Grep

Find all files containing specific text ‘r’ecursively look for the ‘w’hole given pattern: `grep -rw '/path/to/somewhere/' -e 'pattern'`

Better `ack 'text-to-find-here' locationToBeginLooking`
- ◇ ack is like grep, but for source code.
- ◇ It looks prettier and more informative.

Linux

- ◇ The way to “double-click” on a file from the command line is `xdg-open`.

CheatSheet Helper Elisp

The following utilities are loaded when this file is opened. After the first time the file `CheetSheet.el` is created and this section may be deleted. When you delete this section, ensure the `load` in the footer below loads `CheatSheet/CheatSheet.el`.

1. Make some changes, look at them with `f7`.
2. Commit each change with `f8`.
3. Push your changes with `f9`.

```
(defun my-org-latex-export-to-pdf ()
  "Produce a PDF from the CheatSheet then show it via the evince PDF viewer."

  (interactive)
  (org-latex-export-to-pdf)
  (eshell-command
     (concat "evince "
             (file-name-sans-extension buffer-file-name) ".pdf &"))
)

;; Preview and commit

(local-set-key (kbd "<f7>") 'my-org-latex-export-to-pdf)

(local-set-key (kbd "<f8>") '(lambda () (interactive)
  (shell-command
     (format "git commit CheatSheet.org CheatSheet.pdf -m \"CheatSheet: %s\""
     (read-string "Commit Message for CheatSheet: ")))
))

;; Stuff that should be loaded whenever CheatSheet.org is opened.

(visual-line-mode t)

(require 'ox-extra)
(ox-extras-activate '(ignore-headlines))

;; for the <X-TAB short-cuts
(make-variable-buffer-local 'org-structure-template-alist)

(setq PARALLEL (concat "# \n#+begin_parallel latex \n?\n#+end_parallel"))
(add-to-list 'org-structure-template-alist `("p" ,PARALLEL))
```

Example Use <p: Loops implement finite quantifications

A finite quantification can be defined axiomatically by the empty-range rule and split-off term rules. Together these form a recursive definition which can be phrased as a loop.

```
// For -⊕- : T → T → T,
// fold(A,a,b) ≈ (⊕ x : a..b-1 • A[x])
/*@ axiomatic Fold {
  @
  @ logic T
  @   fold{L}(T *A, integer a, integer b)
  @   reads a,b,A, A[..] ;
  @
  @ axiom foldEmptyRange{L} :
  @   \forall T *A, integer a, b; a >= b
  @   ==>  fold(A,a,b) == identity(⊕);
  @
  @ axiom foldSplitOffTerm{L} :
  @   \forall T *A, integer a, b; a <= b
  @   ==>      fold(A, a, b+1)
  @        == fold(A, a, b  ) ⊕ A[b];
  @ }
  @*/
```

```
/*@ requires \valid(A+(0..N-1));
  @ assigns \nothing;
  @ ensures \result == fold(A,0,N);
  @*/
T fold(int N, T* A) {

    T total = identity(⊕);

    /*@ loop invariant 0 <= n <= N;
      @ loop invariant total == fold(A,0,n);
      @ loop assigns n, total;
      @ loop variant N-n;
    */
    for(int n = 0; n != N; n++)
        total = total ⊕ A[n];
    return total;
}
```

This pseudo-code is reified by giving concrete values for (*T*, ⊕, `identity`) such as `(int, +, 0)` or `(bool, ||, false)`. Any monoid will do.