Korny's Emacs/clojure cheatsheet

Assumes: emacs 24 + nrepl + projectile + paredit
Note: excludes things I find easy to remember :)

Paredit:

```
Nav fwd:    C-M-f
    back:   C-M-b
    in:     C-M-<down>
    out:    C-M-<up>
 fn fwd:    C-M-a
 fn bak:    C-M-e
Select fn:  C-M-h
wrap:       M-(
splice:     M-s
grow right: C-<right>
grow left:  C-M-<left>
shrink rgt: C-<left>
shrink lft: C-M-<right>
split:      M-S-s
join:       M-S-j
reindent fn: M-q
```

Windows:

```
New win below: C-x 2
New win right: C-x 3
One win:       C-x 1
Kill win:      C-x 0
```

Help:

```
Search: C-h a
Mode: C-h m
Key: C-h k
```

Projectile:

```
C-c p then
F - find file
B - find buffer
G - grep
R - replace
O - occurs
D - dired
E - recent files
H - help
```

Basics I forget:

```
Undo: C-/
Cancel: C-g C-g C-g
Regexp search: C-S-s
      replace: C-M-%
  groups \( \) or \|
  replace \1 \2
```

Cut/Paste:

```
Mark:   C-space     Swap mark: C-x C-x
Cut:    C-w         Copy:      M-w
Paste:  C-y         Pop-Paste: M-y
```

Namespaces:

```
(ns foo.bar
  "docstring"
  (:require [clojure.tools.cli :as cli
             foo.baz :refer [spoo bwat]
             foo.bar :refer :all])
  (:use [foo.bar]) ; avoid!
  (:import [java.util Date HashMap]))
```

Clojure / Nrepl:

```
load buffer in repl: C-c C-k
switch repl to ns:   C-c M-n
jump to repl:        C-c C-z
scroll other win:  C-M-v and C-M-S-v

Jump to defn: M-.
Back after jump: M-,
Complete: M-/

eval: at point: C-c C-C
      last exp: C-x C-e
      in popup: C-c C-p
```

Destructuring:

```
[a b & m :as all]
{a :a b :b}
{:keys [a b] :or {:a "a"}}
```