

Paredit Cheat Sheet

Basic Insertion Commands

```
paredit-open-round (
(a b | c d)
(a b ( | ) c d)
(foo "bar | baz" quux)
(foo "bar ( | baz" quux)

paredit-close-round )
(a b | c )
(a b c |)

; Hello, | world!
; Hello, | world!

paredit-close-round-and-newline M-)
(defun f (x | ))
(defun f (x)
)

; (Foo. |
; (Foo.) |

paredit-open-square [
(a b | c d)
(a b [ | ] c d)

(foo "bar | baz" quux)
(foo "bar [baz" quux)

paredit-close-square ]
(define-key keymap [frob | ] 'frobnicate)
(define-key keymap [frob | ] 'frobnicate)

; [Bar. |
; [Bar.] |

paredit-doublequote "
(frob grovel | full lexical)
(frob grovel " | " full lexical)

(foo "bar | baz" quux)
(foo "bar \" | baz" quux)

paredit-meta-doublequote M-"
(foo "bar | baz" quux)
(foo "bar baz"
 | quux)

(foo | (bar #\x "baz \\ quux") zot)
(foo " | (bar #\\x \"baz \\\\ quux\")" zot)

paredit-backslash \
(string # | )
Escaping character... (x)
(string #\ | x)

"foo | bar"
Escaping character... (")
"foo\ | bar"
```

```
paredit-comment-dwim M-;
(foo | bar) ; baz
(foo bar) ; | baz

(frob grovel |)
(frob grovel) ; |

| (defun hello-world ...)
;;; |
(defun hello-world ...)

paredit-newline C-j
(let ((n (frobboz))) | (display (+ n 1)
port))
(let ((n (frobboz)))
(display (+ n 1)
port))
```

Deleting & Killing

```
paredit-forward-delete
C-d, <delete>, <deletechar>

(quu | x "zot")
(quu | "zot")

(quux | "zot")
(quux " | zot")
(quux " | ot")

(foo ( | ) bar)
(foo | bar)

| (foo bar)
(| foo bar)

paredit-backward-delete DEL
("zot" q | uux)
("zot" | uux)

("zot" | quux)
("zot" | quux)
("zo" | quux)

(foo ( | ) bar)
(foo | bar)

(foo bar) |
(foo bar | )

paredit-kill C-k
(foo bar) | ; Useless comment!
(foo bar) |

(| foo bar) ; Useful comment!
(| ) ; Useful comment!

| (foo bar) ; Useless line!
|

(foo " | bar baz"
quux)
(foo " | "
quux)
```

```
paredit-forward-kill-word M-d
| (foo bar) ; baz
(| bar) ; baz
(| ) ; baz
(| ) ; |

;; | Frobnicate
(defun frobnicate ...)
;; |
(defun frobnicate ...)
;;
(| frobnicate ...)

paredit-backward-kill-word M-DEL
(foo bar) ; baz
(quux) |
(foo bar) ; baz
(| )
(foo bar) ; |
(| )
(| ) ;
(| ) ;
(| )
```

Movement & Navigation

```
paredit-forward C-M-f
(foo | (bar baz) quux)
(foo (bar baz) | quux)

(foo (bar) | )
(foo (bar)) |

paredit-backward C-M-b
(foo (bar baz) | quux)
(foo | (bar baz) quux)

(| (foo) bar)
| ((foo) bar)
```

Depth-Changing Commands

```
paredit-wrap-round M-(
(foo | bar baz)
(foo ( | bar) baz)

paredit-splice-sexp M-s
(foo (bar | baz) quux)
(foo bar | baz quux)

paredit-splice-sexp-killing-backward
M-<up>, ESC <up>
(foo (let ((x 5)) | (sqrt n)) bar)
(foo (sqrt n) bar)

paredit-splice-sexp-killing-forward
M-<down>, ESC <down>
```

```
(a (b c | d e) f)
(a b c | f)

paredit-raise-sexp M-r
(dynamic-wind in (lambda () | body) out)
(dynamic-wind in | body out)
| body
```

Barfage & Slurpage

```
paredit-forward-slurp-sexp
C-), C-<right>
(foo (bar | baz) quux zot)
(foo (bar | baz quux) zot)

(a b ((c | d)) e f)
(a b ((c | d) e) f)

paredit-forward-barf-sexp
C-\}, C-<left>
(foo (bar | baz quux) zot)
(foo (bar | baz) quux zot)

paredit-backward-slurp-sexp
C-(, C-M-<left>, ESC C-<left>
(foo bar (baz | quux) zot)
(foo (bar baz | quux) zot)

(a b ((c | d)) e f)
(a (b (c | d) e) f)

paredit-backward-barf-sexp
C-{, C-M-<right>, ESC C-<right>
(foo (bar baz | quux) zot)
(foo bar (baz | quux) zot)
```

Miscellaneous Commands

```
paredit-split-sexp M-S
(hello | world)
(hello) | (world)

"Hello, | world!"
"Hello, " | "world!"

paredit-join-sexps M-J
(hello) | (world)
(hello | world)

"Hello, " | "world!"
"Hello, | world!"

hello-
| world
hello- | world

paredit-recentre-on-sexp C-c C-M-l No
examples
paredit-reindent-defun M-q No examples
```