Paredit Cheatsheet

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!
```

M-) paredit-close-round-and-newline

```
(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)
```

M-" paredit-meta-doublequote

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

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

\ paredit-backslash

```
(string #| )
; Escaping character...  (x)
(string #\x| )
```

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

M-; paredit-comment-dwim

```
(foo | bar)    ; baz
(foo bar)      ; | baz
```

```
(frob grovel)|
(frob grovel) ; |
```

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

C-j paredit-newline

```
(let ((n (frobbotz))) | (display (+
n 1)
port))
(let ((n (frobbotz)))
| (display (+ n 1)
port))
```

Deleting and Killing

C-d paredit-forward-delete

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

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

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

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

DEL paredit-backward-delete

```
("zot" q| uux)
("zot" | uux)
```

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

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

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

C-k paredit-kill

```
(foo bar)| ; Useless comment!
(foo bar)|
```

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

```
| (foo bar) ; Useless line!
|
```

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

M-d paredit-forward-kill-word

```
| (foo bar) ; baz
(| bar) ; baz
(| ) ; baz
() ;|
```

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

M-DEL paredit-backward-kill-word

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

Movement and Navigation

C-M-f paredit-forward

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

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

C-M-b paredit-backward

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

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

C-M-u paredit-backward-up
C-M-d paredit-forward-down
C-M-p paredit-backward-down
C-M-n paredit-forward-up

Depth Changing

M-(paredit-wrap-round

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

M-s paredit-splice-sexp

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

M-↑ paredit-splice-sexp-killing-backward

```
(foo (let ((x 5)) | (sqrt n)) bar)
(foo (sqrt n) bar)
```

M-↓ paredit-splice-sexp-killing-forward

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

M-r paredit-raise-sexp

```
(dynamic-wind in (lambda () | body)
out)
(dynamic-wind in | body out)
| body
```

Barfage and Slurpage

C-→ paredit-forward-slurp-sexp

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

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

C-← paredit-forward-barf-sexp

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

C-M-← paredit-backward-slurp-sexp

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

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

C-M-→ paredit-backward-barf-sexp

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

Miscellaneous

M-S paredit-split-sexp

```
(hello| world)
(hello)| (world)
```

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

M-J paredit-join-sexps

```
(hello)| (world)
(hello| world)
```

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

```
hello-
| world
hello-| world
```

C-c C-M-l paredit-recentre-on-sexp
M-q paredit-reindent-defun