

-- NB: These are partial lists
for mnemonic purposes only:
consult manuals for more! --

Select (+ ⌘-OPT for non-contig)

Word

2-Click

Sentence

3-Click

(to end or start of sentence: SH-3-click)

Line

OPT-3-Click

Para

4-Click

Doc

5-Click / ⌘-A

Rect

opt-drag

Move (+ SH for Extend)

by char

rt/left arrow

by word

OPT-rt/left

by line

up/dn arrow
by screen

⌘-up/dn

by doc

OPT-up/dn

by this para

⌘-rt/left

by next para

⌘-ent/ret
(SH + Jump menu extends)

Delete (left)

by char

del

by word

OPT-del
(OPT + Jump menu deletes)

[NB: to recover last deleted text,

choose Copy with nothing selected;

this puts it on the clipboard]

Format

(Unnamed Rulers)

all similar paras

⌘-change

(to name all similar unnamed rulers,

type name in popup menu, ⌘-<ret>)

all similar, selected

⌘-change

(changes all similar to first)

all dependent paras

OPT-change

change identically

move indent

change relatively

SH-move indent

(Tabs)

lead char, this tab

dbl-click leaderbar

lead char, later tabs
dbl-click tab icon

(Layout)

in Layout, OPT-margin does not

move text

in Layout, SH-drag does not

move other page

(Rulers and Styles)

Insert Ruler makes an unnamed ruler

appear

directly changing a paragraph's format

causes an Insert Ruler below; pasting

does not

choosing a ruler Name with insertion pt

in paragraph causes an Insert Ruler

below; doing it with a ruler selected

does not

applying a Style with a named ruler

selected affects all material governed

by that ruler

applying a Style with text selected

affects only selected

text if no ruler is named; it affects

whole paragraph if ruler is named

— and furthermore, subsequent

para gets an Insert Ruler, and, if

named, the Inserted ruler is named!

to import a ruler, paste it with both

docs open

to import a style (and associated ruler),

apply it with both docs open

cannot paste a ruler if ruler of same

name exists; so, to make styles

conform, change ruler name first,

then apply with ruler selected

Windows

tile 2

SH-click in zoom

tile all

OPT-click in zoom

stack

OPT-SH-click in zoom

zoom

dbl-click title bar

keep inactive

hold ⌘

close other

⌘-OPT-click this close

synch scroll
SH scroll

rev scroll

OPT scroll

unsplit

dbl-click split bar

Files

open 2nd window

SH-dbl-click catalog

open by name

select, Open Selection

Characters

soft hyphen

⌘-hyphen

soft return

SH-enter

Menus

OPT makes command effect all docs

OPT-macro/style edits it

OPT-click clip title for clip menu

⌘-click wind title for windows menu

OPT-click wind title for macros menu

SH-cut/copy appends

SH-paste swaps

modified menus:

File: OPT, ⌘, ⌘-OPT-SH

Edit: SH, OPT, SH-OPT

Tools: all combs of SH, OPT, ⌘

(affect find/replace)

Format: SH

Size: SH

Graphics: SH, OPT

Locate

named ruler

SH-select in menu

hdr, ftr, ftnt
SH-close window

cursor

enter

Footnotes

change symbol
SH-click

change placing
SH-click sep line

Graphics

select all touched OPT-drag

restore char graphic SH-dbl-click

Find

enter ⌘ or ⌘

use SH

clear text attributes clear both & close

general help

find works in loops; in each loop

a selection is made, and if it fails

to match a new try is made starting

outside the selection

find expressions are evaluated from

right to left, meaning that a selection

is made that satisfies each component

before the next component is looked at

the metacharacters

wild

.

any exc return

:[space] any ctrl exc return

:!

any ctrl exc tab, ret; + hi ascii

:@

any ctrl, + hi ascii, + SP; not ret

:.

any at all

:A

alpha + diacritic + under

:a

alpha + diacritic; not under

:Å

not alpha nor diacritic nor under

:å

not alpha nor diacritic

:l
locase alpha + diacritic

:u
upcase alpha + diacritic

:d
digit

:n
alpha + diacritic + digit

:N
alpha + diacritic + digit + under

:ð
diacritic + under

:î
not diacritic nor under

:o
octal digit

:x
hex digit

:w
not punctuation

:b
space or tab

≈
whole word (+ space)

[...] any in set of chars and/or ranges

[^...] any not in set

:[...] set can include return

times

* zero or more

+ one or more

- zero or one

.* :+ :-

shortest match

position

:s doc start

:e doc end

^ para start

\$ para end

:^(
line start

\<
word start

\>
word end

:<x
after x

:>x
before x

[what's looked at is the rtmost

part of the insertion point]

clipboard

\CC current clip

\C0
clip 0, etc.

NB: to use, may need single-

quotes to force metachar eval

other chars

|
Logical Or

&
found str

\(...\) parentheses

\1...\9 parenth expr

\E10...\E63 parenth expr

[count from left]

\0
null

\b
backspace

`\t`
tab

`\n`
linefeed

`\f`
forced pagebreak

`\r`
return

`\s`
space

`\v`
soft return

else, \ switches to literal

Macros

to copy, include ¶ before name

general

"..." strs must be literal

(for variable, put on clip

and use '\CC' etc.)

for " in str, use ""

use ' instead to include metas

use \return to insert fake

return into a quote

(e.g. to allow ruler in quote)

digit
change clipboard

:1 "text1" ["text2"]
makes dialog

box; text1 is prompt; text2 allows

user to enter text (goes to clip)

Key [text]

no quotes! no text = ret; use ASCII

Table to type arrows etc.

menu commands

enter name or title:name

if |-able, can take "on" or "off"

can precede with symbols Shift

Option Command from ASCII

Table (hex 19, 20, 17)

Copy["clipnum"]["text"]

Cut["clipnum"]

Paste["clipnum"]["text"]

(cannot omit clipnum, but can

leave empty, so you can insert

a literal string directly)

Mark Text... ["n1"]["n2"]

(n2 to mark start and end of

selection w/ diff names)

Jump
["pos1"["pos2"]]

(pos2 to select from one to

the other (but not including!!);

NB a pos can be a name, linenum,

or "END" (use "1" for start))

Ruler menu commands

Left, Center, Right, Full

L 1, L 1.5, L 2, L 3, L Other

¶ 1, ¶ 1.5, ¶ 2, ¶ 3, ¶ Other

(after Other, enter val and ret)

Find commands

Find Next ["exp" ["options"]]

Find/Replace ["exp1" "exp2" ["options"]]

where the options are:

*
use dialog's options

C
copy text and options into dialog

-g
normal search

e
power search

g
power search +

w
whole word

-w
not whole word

r
reverse (search backwards)

i

ignore case

W

wrap

-W

don't wrap

G

search all open

-G

search front doc only

o

ignore styles in Find

-o

use styles in Find

O

ignore styles in Replace

-O

use styles in Replace

a

replace all

l

replace all in selected paras

s

replace all in selection

n

with s, select last changed

l

index, not replace

t

no dialogs

v

dialogs

E

jump to end on failure

T

terminate macro on failure

M

search all files on search list

-M

search current doc only

X

with Find Next, puts pathnames

from search list onto clip

Programming Language

//

comment

label:

label

+

concatenate strs

-

remove first instance of str2

/

remove every instance of str2

variables

DOUBLE
printf() formatting str

LONG

ditto

[see 32-34 of manual]

DoubleLock
boolean; if true, forces

all constants to double-precision

clipboard
current clip

SelectWithRuler
boolean; if true,

selections include ruler just to left

ReportErrors
boolean; if false,

suppresses error reporting

commands

local x

make caller's x inaccessible

global x
make x live forever

(no need to repeat in other macros)

free x

kill x's var space

freeLocal, freeGlobal, freeAll

macroCopy, macroPaste between

clip and selection point

selectRuler(n)
selects ruler governing

char offset n

setSelect(n1,n2)

selects char range

setSelectMore(n1,n2)

builds non-

contig selections; first one is ignored,

so start with (0,0)

goto label

exit

from macro

if(E)

if E is true, non-zero, or non-

empty, then statements on same

line will execute; use semi-colon to

separate statements on same line;

operators include !, &&, ||, ==, <> ,

<=, >=, etc.

functions (e.g. cos(num))

sin, cos, tan, acos, asin, atan

sinh, cosh, tanh, exp

ceil
returns next int as float

floor
returns prev int as float

div, mod

fmod(f1,f2) rem f1/f2

modf

fractional part

random

integer

rrandom(f1,f2)

float betw args

seed

uses arg to seed later randoms

CharToNum, NumToChar

left(str,n)

right(str,n)

mid(str,startnum,n)

offset(soughtstr,targetstr)

upperRoman, lowerRoman
int to str

fabs (= abs val)

log, log1

functions (no arg)

docName, docPath

of front doc

dateCreated, lastDateSaved

timeModified

charNum, lineNum, PageNum

of

insertion point

pageLineNum

lineNum on this page

endCharNum

char offset to end

rulerName

governing insertion pt

rulerStart

char offset

nextRulerStart

selectStart, selectEnd

in chars

numFound

how many found in last find

arrays

array must already exist via method

arrays are 1-based

[n] addr absolutely

[[n]] addr relative, e.g. [[1]] = next

methods (syntax: var -> method(arg))

type

return string naming type of var

create(n)

n storage spaces at start

qcreate(n)

n storage spaces at end

push(E)

adds E to var front

qpush(E)

adds E to var end

NB: these keywords can be used as E

to obtain info about current non-contig

selection: strings; starts; ends;

startends;

ints, floats

treats str as number

pop

removes and refs first element

first

refs without removing

last

refs without removing

size

reports size

shuffle

randomizes

swap(n1,n2)

error
reports boolean whether last access

to var caused error