

指数

此处出现的所有页码均指本书的印刷版本，也可用作指向本书电子版中相应位置的超文本链接。

！（感叹号）， [8](#)
“（双引号）， [216](#)
#! r6rs, [456](#)
#’（语法）， [300](#)
#，（unsyntax）， [305](#)
#，@（unsyntax-splicing）， [305](#)
#;（基准注释）， [455](#)
#\， [211](#)
#’（quasisyntax）， [305](#)
#|...|#（块注释）， [455](#)
#b（二进制）， [169](#)
#d（十进制）， [169](#)
#f， [7](#)， [36](#)， [143](#)
#o（八进制）， [169](#)
#t， [7](#)， [36](#)， [143](#)
#x（十六进制）， [169](#)
&断言， [366](#)

&condition, [362](#)
&error, [367](#)
&i/o, [371](#)
&i/o-decoding, [375](#)
&i/o-encoding, [376](#)
&i/o-file-already-exists, [374](#)
&i/o-file-does-not-exist, [374](#)
&i/o-file-is-read-only, [374](#)
&i/o-file-protection, [373](#)
&i/o-filename, [373](#)
&i/o-invalid-position, [372](#)
&i/o-port, [375](#)
&i/o-read, [372](#)
&i/o-write, [372](#)
&implement-restriction, [369](#)
&irritants, [368](#)
&lexical, [370](#)
&message, [368](#)
&no-infinities, [376](#)
&no-nans, [377](#)
&non-continuable, [369](#)
&serious, [366](#)
&syntax, [370](#)
&undefined, [371](#)
&违规, [366](#)

&warning, [367](#)
&who, [369](#)
' (quote), [17](#), [22](#), [59](#), [141](#)
(), [7](#), [19](#)
*, [16](#), [172](#)
+, [16](#), [171](#)
, (未引用), [142](#)
, @ (未引用拼接), [142](#)
-, [16](#), [172](#)
->, [8](#)
. (dot), [19](#), [460](#)
... (省略号), [61](#), [294](#), [297](#)
/, [16](#), [172](#)
; (注释), [7](#), [455](#)
<, [170](#)
<=, [170](#)
=, [170](#)
=>, [111](#), [112](#)
>, [170](#)
>=, [170](#)
? (问号), [8](#), [37](#)
_ (下划线), [61](#), [296](#), [315](#)
_ (下划线), [294](#), [297](#)
' (准引号), [142](#)
abs, [34](#), [178](#), [183](#)

抽象对象, [53](#), [408](#)
acos, [185](#)
实际参数, [27](#), [92](#)
Algol 60, [6](#)
and, [37](#), [62](#), [110](#)
角度, [183](#)
追加, [46](#), [160](#)
适用, [107](#)
任意精度, [167](#)
asin, [185](#)
断言, [359](#)
断言违规, [358](#)
断言违规? , [366](#)
赋值, [102](#)
赋值, [47](#), [102](#)
assoc, [165](#)
协会列表, [165](#), [166](#), [243](#), [404](#)
assp, [166](#)
assq, [165](#)
assv, [165](#)
atan, [185](#)
原子? , [41](#)
辅助关键字, [61](#), [294](#)
基本情况, [41](#)
be-like-begin, [313](#)

begin, [51](#), [60](#), [101](#), [108](#)
二进制端口, [257](#)
二叉树, [155](#)
二进制端口? , [270](#)
绑定, [4](#)
位和, [186](#)
位按算术移位, [190](#)
位按算术移位- 左移, [189](#)
位按位算术 - 移位 - 右, [189](#)
位 - 按位计数, [187](#)
位宽位字段、[189](#)
位按位集? 、[188](#)
位按位复制位、[188](#)
位按位复制位字段、[189](#)
位按位第一位集、[187](#)
位按 if、[186](#)
位按 ior、[186](#)
位长度、[187](#)
位非位、[186](#)
位反向位场、[191](#)
位旋转位场、[190](#)
位向异或、[186](#)
块缓冲, [258](#)
块注释 (#|...|#) , [455](#)
块结构, [4](#)

布尔语法, [457](#)

布尔值, [7](#)

布尔=? , [243](#)

布尔? , [150](#)

绑定标识符=? , [302](#)

个括号 ([]) , [7](#)、[155](#)

个中断、[308](#)

个缓冲区模式、[258](#)

个缓冲区模式、[261](#)

个缓冲区模式? 、[262](#)

个字节向量语法、[461](#)

个字节向量>列表、[238](#)

个字节向量>字符串、[286](#)

字节向量->u8-list、[232](#)

字节向量->uint-list、[238](#)

个字节向量-copy、[229](#)

个字节向量-copy! 、[230](#)

个字节向量-填充! 、[229](#) 个

bytevector-ieee-double-native-ref, [239](#)

bytevector-ieee-double-native-set! , [239](#)

bytevector-ieee-double-ref, [240](#)

bytevector-ieee-double-set! , [240](#)

bytevector-ieee-single-native-ref, [239](#)

bytevector-ieee-single-native-set! , [239](#)

bytevector-ieee-single-ref, [240](#)

bytevector-ieee-single-set! , [240](#)
bytevector-length, [229](#)
bytevector-s16-native-ref, [232](#)
bytevector-s16-native-set! , [233](#)
bytevector-s16-ref, [235](#)
bytevector-s16-set! , [236](#)
bytevector-s32-native-ref, [232](#)
bytevector-s32-native-set! , [233](#)
bytevector-s32-ref, [235](#)
bytevector-s32-set! , [236](#)
bytevector-s64-native-ref, [232](#)
bytevector-s64-native-set! , [233](#)
bytevector-s64-ref, [235](#)
bytevector-s64-set! , [236](#)
bytevector-s8-ref, [231](#)
bytevector-s8-set! , [231](#)
bytevector-sint-ref, [237](#)
bytevector-sint-set! , [238](#)
bytevector-u16-native-ref, [232](#)
bytevector-u16-native-set! , [233](#)
bytevector-u16-ref, [235](#)
bytevector-u16-set! , [236](#)
bytevector-u32-native-ref, [232](#)
bytevector-u32-native-set! , [233](#)
bytevector-u32-ref, [235](#)

bytevector-u32-set! , [236](#)
bytevector-u64-native-ref, [232](#)
bytevector-u64-native-set! , [233](#)
bytevector-u64-ref, [235](#)
bytevector-u64-set! , [236](#)
bytevector-u8-ref, [230](#)
bytevector-u8-set! , [231](#)
bytevector-uint-ref, [237](#)
bytevector-uint-set! , [238](#)
bytevector=? , [229](#)
bytevector? , [155](#)
C, [393](#)
caaaar, [157](#)
caaadr, [157](#)
caaar, [157](#)
caadar, [157](#)
caaddr, [157](#)
caadr, [157](#)
caar, [157](#) caar, [157](#)
caar, cadr, ..., cddddr, [34](#)
cadaar, [157](#)
cadadr, [157](#)
cadar, [157](#)
caddar, [157](#)
caddr, [157](#)

caddr, [157](#) caddr

cadr, [31](#), [32](#), [34](#), [157](#)

按名称调用, [408](#)

按值调用, [407](#)

使用字节向量输出端口调用, [266](#)

当前继续调用, [123](#), [426](#)

使用输入文件调用, [281](#)

使用输出文件调用, [282](#)

使用端口调用, [272](#)

使用字符串输出端口调用, [267](#)

使用值调用, [130](#), [131](#)

调用/cc, [74](#), [122](#), [123](#), [126](#), [133](#),
[425](#), [426](#)

汽车, [18](#), [155](#), [156](#)

案例, [55](#), [113](#), [306](#)

案例-lambda, [94](#)

cdaaar, [157](#)

cdaadr, [157](#)

cdaar, [157](#)

cdadar, [157](#)

cdaddr, [157](#)

cdadr, [157](#)

cdar, [157](#)

cddaar, [157](#)

cddadr, [157](#)

cddar, [157](#)
cdddar, [157](#)
cddddr, [157](#)
cdddr, [157](#)
cddr, [31](#), [34](#), [157](#)
cdr, [18](#), [38](#), [155](#), [156](#)
天花板, [177](#)
char->integer, [215](#)
char-alphabetic? , [213](#)
char-ci<=? , [212](#)
char-ci<? , [212](#)
char-ci=? , [212](#)
char-ci>=? , [212](#)
char-ci>? , [212](#)
char-downcase, [214](#)
char-foldcase, [215](#)
char-general-category、 [214](#)
字符小写? 、 [213](#)
字符数字? 、 [213](#)
字符标题大小写、 [213](#)
字符标题大小写、 [214](#)
字符大写、 [214](#)
字符大写? , [213](#)
char-whitespace? , [213](#)
char<=? , [212](#)

char<? , [212](#)
char=? , [212](#)
char>=? , [212](#)
char>? , [212](#)
char? , [154](#)
字符语法, [457](#)
字符, [211](#)
Chez 方案, [ix](#), [42](#)
子类型, [325](#)
循环列表, [156](#)
关闭输入端口, [285](#)
关闭输出端口, [285](#)
关闭端口, [270](#)
编解码器, [257](#)
命令行, [350](#)
注释, [7](#), [455](#)
通用 Lisp, [6](#)
编译器, [4](#)
完整, 参见引擎
复数, [167](#), [412](#)
complex? , [151](#), [167](#)
compose, [34](#)
compound condition, [362](#)
cond, [39](#), [44](#), [111](#), [304](#)
condition, [362](#)

condition object, [361](#)
条件类型, [361](#)
条件访问器, [365](#)
条件刺激物, [368](#)
条件消息, [368](#)
条件谓词, [365](#)
条件谁, [369](#)
条件?, [362](#)
条件, [109](#)
条件, [357](#)
缺点, [19](#), [156](#)
缺点单元格, [155](#)
缺点*, [158](#)
缺点, [19](#)
常量, [141](#)
常量, [21](#), [141](#)
延续传递风格, [78](#), [418](#)
延续, [5](#), [73](#), [124](#), [421](#)
控制结构, [107](#)
核心语法形式, [4](#), [22](#), [59](#), [404](#)
cos, [185](#)
CPS、[78](#)
个电流异常处理程序、[357](#)
个电流错误端口、[263](#)
个电流输入端口、[263](#)

个电流输出端口、[263](#)
个循环列表、[56](#)
d（双精度）、[169](#) 个
数据、[141](#)
基准注释（#;），[455](#)
基准语法，[455](#)，[456](#)
基准>语法，[308](#)，[317](#)，[320](#)
默认协议，[327](#)
定义，[30](#)，[81](#)，[100](#)
定义条件类型、[364](#)
定义枚举、[250](#)
定义可积、[315](#)
定义对象、[408](#)
定义记录类型、[323](#)、[328](#)
定义结构、[318](#)
定义语法、[61](#)，[291](#)，[292](#)，[389](#)
定义句法扩展，[60](#)
defun 语法，[33](#)，[60](#)
延迟，[128](#)
延迟评估，[408](#)
删除文件，[286](#)
delq!，[54](#)
分母，[181](#)
描述段，[132](#)
显示，[285](#)，[397](#)

div, [175](#)
div-and-mod, [175](#)
div0, [176](#)
div0-and-mod0, [176](#)
除数, [115](#), [116](#)
do, [115](#), [312](#)
点 (.), [19](#), [460](#)
点对, [20](#), [155](#)
双精度, [27](#), [33](#)
双引号, [216](#)
双任意, [30](#)
双缺点, [27](#), [33](#)
倍加, [33](#)
双递归, [70](#)
dxdy, [131](#)
动态分配, [3](#)
动态风, [124](#)
省略号 (...), [61](#), [294](#)
其他, [111](#), [112](#), [113](#)
空列表, [7](#), [19](#)
字节序, [228](#)
引擎, [421](#)
枚举集>列表, [252](#)
个枚举集补码, [254](#)
个枚举集构造函数, [251](#)

个枚举集差异, [253](#)
个枚举集索引器, [254](#) 个
枚举集枚举集交集, [253](#)
枚举集成员? , [253](#)
枚举集投影, [254](#)
枚举集子集? , [252](#)
枚举集并集, [253](#)
枚举集- 宇宙, [252](#)
枚举集=? , [252](#)
环境, [137](#)
环境, [404](#)
eof 对象, [257](#)
eof-object, [273](#)
eof-object? , [257](#), [273](#)
eol 样式、[257](#)
eol 样式、[259](#)
eq? 、[143](#)
等算、[245](#)
等于? 、[148](#)
等价谓词、[143](#)
eqv? 、[38](#), [146](#)
错误, [358](#)
错误处理模式, [258](#)
错误处理模式, [260](#)
错误? , [367](#)

评估, [136](#)
甚至? , [47](#), [66](#), [81](#), [174](#)
精确, [180](#)
精确>精度, [181](#)
精确整数 sqrt, [184](#)
精确? , [167](#), [170](#)
精确度, [167](#), [180](#)
精确度保留, [167](#)
除导入集, [346](#)
例外, [9](#), [357](#)
感叹号 (!) , [8](#)
存在, [119](#)
出口, [350](#)
exp, [184](#)
扩展, [59](#)
过期, 看引擎
导出, [345](#)
导出级别, [345](#)
表达式, [7](#)
expt, [179](#)
扩展示例, [381](#)
f (单) , [169](#)
因子, [71](#), [72](#), [73](#)
阶乘, [68](#), [75](#), [116](#)
假, [7](#), [36](#)

快速傅里叶变换 (FFT) , [412](#)
挡泥板, [299](#), [301](#)
斐波那契, [69](#), [102](#), [116](#), [422](#)
斐波那契数列, [69](#), [102](#)
字段, [331](#)
文件, [257](#)
文件存在? , [286](#)
文件选项, [261](#)
过滤器, [164](#)
查找, [165](#)
有限? , [174](#)
一类数据值, [3](#)
一类过程, [5](#)
修正数, [192](#)
fixnum->flonum, [211](#)
fixnum-width, [193](#)
fixnum? , [193](#)
f1*, [207](#)
f1+, [206](#)
f1-, [206](#)
f1/, [207](#)
f1<=? , [203](#)
f1<? , [203](#)
f1=? , [203](#)
f1>=? , [203](#)

fl>? , [203](#)
flabs, [209](#)
flacos, [210](#)
flasin, [210](#)
flatan, [210](#)
flceiling, [208](#)
flcos, [210](#)
fldenominator, [209](#)
fldiv, [207](#)
fldiv-and-mod, [207](#)
fldiv0, [208](#)
fldiv0-and-mod0, [208](#)
fleven? , [205](#)
fexp, [209](#)
flexpt, [210](#)
flfinite? , [205](#)
flfloor, [208](#)
flinfinite? , [205](#)
flinteger? , [204](#)
flip-flop, [102](#)
fllog, [209](#)
flmax, [205](#)
flmin, [205](#)
flmod, [207](#)
flmod0, [208](#)

flnan? , [205](#)
flnegative? , [204](#)
flnumerator, [209](#)
floating point, [167](#)
flod? , [205](#)
flonum, [202](#)
flonum? , [203](#)
floor, [177](#)
flpositive? , [204](#)
flround, [208](#)
flsin, [210](#)
flsqrt, [210](#)
flt看, [210](#)
fltruncate, [208](#)
流体装订, [125](#)
齐平输出端口, [280](#)
flzero? , [204](#)
折叠左, [120](#)
折叠右, [121](#)
折叠, [120](#), [121](#)
for-all, [119](#)
for-each, [118](#)
force, [128](#)
formal parameters, [26](#), [29](#), [92](#)
格式化输出, [401](#)

`fprintf`, [401](#)
`free variable`, [28](#)
自由标识符=? , [302](#)
频率, [393](#)
`fx*`, [195](#)
`fx*/携带`, [197](#)
`fx+`, [195](#)
`fx+/carry`, [197](#)
`fx-`, [195](#)
`fx-/carry`, [197](#)
`fx<=?` , [193](#)
`fx<?` , [193](#)
`fx=?` , [193](#)
`fx>=?` , [193](#)
`fx>?` , [193](#)
`fxand`, [197](#)
`fxarithmetic-shift`, [201](#)
`fxarithmetic-shift-left`, [201](#)
`fxarithmetic-shift-right`, [201](#)
`fxbit-count`, [198](#)
`fxbit-field`, [200](#)
`fxbit-set?` , [199](#)
`fxcopy-bit`, [200](#)
`fxcopy-bit-field`, [200](#)
`fxdiv`, [196](#)

fxdiv-and-mod, [196](#)
fxdiv0, [196](#)
fxdiv0-and-mod0, [196](#)
fxeven? , [194](#)
fxfirst-bit-set, [199](#)
fxif, [198](#)
fxior, [197](#)
fxlength, [198](#)
fxmax, [195](#)
fxmin, [195](#)
fxmod, [196](#)
fxmod0, [196](#)
fxnegative? , [194](#)
fxnot, [197](#)
fxodd? , [194](#)
fxpositive? , [194](#)
fxreverse-bit-field, [202](#)
fxrotate-bit-field, [201](#)
fxxor, [197](#)
fxzero? , [194](#)
垃圾回收器, [3](#)
gcd, [179](#)
生成临时, [310](#)
生成, [324](#)
get-bytevector-all, [275](#)

get-bytevector-n, [274](#)
get-bytevector-n! , [274](#)
get-bytevector-some, [275](#)
get-char, [275](#)
get-datum, [278](#)
get-line, [277](#)
get-string-all, [277](#)
get-string-n, [276](#)
get-string-n! , [276](#)
get-u8, [274](#)
getq, [54](#)
goodbye, [41](#)
最大 - fixnum , [193](#)
警卫 , [361](#)
野兔和, [56](#) , [66](#)
哈希可清除! , [249](#)
哈希表包含? , [246](#)
哈希表副本 , [248](#)
hashtable-delete! , [248](#)
hashtable-entries, [250](#)
hashtable-equivalence-function, [245](#)
hashtable-hash-function, [245](#)
hashtable-keys, [249](#)
hashtable-mutable? , [245](#)
hashtable-ref, [246](#)

hashtable-set! , [246](#)
hashtable-size, [248](#)
hashtable-update! , [247](#)
hashtable? , [155](#)
hashtables, [243](#)
i/o-decoding-error? , [375](#)
i/o-encoding-error-char, [376](#)
i/o-encoding-error? , [376](#)
i/o-error-filename, [373](#)
i/o-error-port, [375](#)
i/o-error-position, [372](#)
i/o-error? , [371](#)
i/o-file-already-exists-error? , [374](#)
i/o-file-does-not-exist-error? , [374](#)
i/o-file-is-read-only-error? , [374](#)
i/o-file-protection-error? , [373](#)
i/o-filename-error? , [373](#)
i/o-invalid-position-error? , [372](#)
i/o-port-error? , [375](#)
i/o-read-error? , [372](#)
i/o-write-error? , [372](#)
标识符语法, [291](#), [297](#), [307](#), [316](#), [317](#)
标识符? , [301](#)
标识符, [6](#)
if, [35](#), [36](#), [39](#), [51](#), [59](#), [109](#)

imag-part, [182](#)
出口不变性, [349](#)
不可变, [331](#)
实施限制 - 违反? , [369](#)
隐式开始, [109](#)
导入, [345](#)
导入级别, [345](#)
导入规范, [345](#), [346](#)
不正确的列表, [19](#), [155](#)
包含, [309](#)
间接导出, [349](#)
不精确, [180](#)
不精确→精确, [181](#)
不精确? , [167](#), [170](#)
无限? , [174](#)
继承, [412](#)
记录继承, [325](#)
输入端口、[257](#)
个输入端口、[270](#)
个整数>字符、[215](#)
个整数除法、[79](#)
个整数值、[153](#)
个整数? 、[151](#)、[167](#)
整数, [167](#)
个可积过程, [315](#)

个内部定义, [81](#), [92](#)

个内部状态, [49](#)

个解释, [404](#)

个解释器, [4](#), [404](#)

行内空白, [455](#)

个刺激条件? , [368](#)

迭代, [5](#), [45](#), [68](#), [114](#), [115](#), [117](#),
[118](#), [120](#), [121](#), [122](#)

关键字, [4](#), [61](#), [291](#)

1 (长整型), [169](#)

lambda, [26](#), [29](#), [59](#), [92](#), [93](#)

lambda*, [94](#)

latin-1, [257](#)

latin-1-编解码器, [259](#)

lazy, [51](#)

lazy evaluation, [51](#), [127](#)

lcm, [179](#)

最小固定数, [193](#)

长度, [42](#), [159](#)

let, [23](#), [28](#), [65](#), [95](#), [114](#)

let*, [64](#), [96](#)

let*值、[99](#)、[134](#)

let 绑定变量、[23](#)

let 语法、[291](#)、[293](#)、[314](#)

let 值、[99](#)、[134](#)、[310](#)

letrec, [65](#), [81](#), [97](#), [310](#)
letrec*, [98](#)
letrec-syntax, [291](#), [293](#), [314](#)
词汇范围, [4](#), [5](#), [25](#), [63](#)
词法违规?、[370](#)
个库、[343](#)
个库正文、[348](#)
个库版本、[344](#)
个库版本引用、[347](#)
个轻量级线程、[421](#)
行缓冲、[258](#)
行结尾、[455](#)
Lisp, [ix](#), [6](#)
lisp-cdr, [38](#)
list, [20](#), [31](#), [32](#), [158](#)
list 常量, [7](#)
list syntax, [460](#)
list->string, [223](#)
list->vector, [226](#)
list-copy, [43](#)
list-ref, [159](#)
list-sort, [167](#), [387](#)
list-tail, [160](#)
list? , [56](#), [66](#), [67](#)、[81](#), [158](#)
列表, [17](#), [18](#), [155](#)

文本, [294](#)
加载, [13](#)
局部变量绑定, [95](#)
日志, [184](#)
前瞻字符, [275](#)
lookahead-u8, [274](#)
loop, [308](#)
looping, [5](#)
macros, [291](#)
magnitude, [178](#), [183](#)
make-assertion-violation, [366](#)
make-bytevector, [228](#)
make-counter, [50](#), [54](#)
make-custom-binary-input-port, [267](#)
make-custom-binary-input/output-port, [267](#)
make-custom-binary-output-port, [267](#)
make-custom-textual-input-port, [268](#)
make-custom-textual-input/output-port, [268](#)
make-custom-textual-output-port, [268](#)
make-enumeration, [251](#)
make-eq-hashtable, [243](#)
make-eqv-hashtable, [244](#)
make-error, [367](#)
make-hashtable, [244](#)
make-i/o-decoding-error, [375](#)

make-i/o-encoding-error, [376](#)
make-i/o-error, [371](#)
make-i/o-file-already-exists-error, [374](#)
make-i/o-file-does-not-exist-error, [374](#)
make-i/o-file-is-read-only-error, [374](#)
make-i/o-file-protection-error, [373](#)
使 i/o 文件名错误、[373](#)
使 i/o 无效位置错误、[372](#)
使 i/o 端口错误、[375](#)
使 i/o 读取错误、[372](#)
使 i/o 写入错误、[372](#) 使 i/o 写入错误
使-实现-限制-违规, [369](#)
使刺激条件, [368](#)
使词法违规, [370](#)
使列表, [46](#), [94](#)
使消息条件, [368](#)
不无穷大违规, [376](#)
不做南违规, [377](#)
不可延续违规, [369](#)
制造极地, [183](#)
做出承诺, [129](#)
make-queue, [54](#)
make-record-constructor-descriptor, [332](#)
make-record-type-descriptor, [323](#), [331](#)
make-recangular, [182](#)

make-serious -condition, [366](#)
make-stack, [52](#), [55](#)
make-string, [218](#)
make-syntax-violation, [370](#)
make-transcoder, [259](#)
make-undefined-violation, [371](#)
make-variable-transformer, [291](#), [298](#),
[306](#)
make-vector, [224](#)
make-violation, [366](#)
make-warning, [367](#)
make-who-condition, [369](#)
map, [45](#), [47](#), [117](#), [392](#)
map1, [46](#)
mapping, [45](#), [117](#), [118](#), [121](#), [122](#)
矩阵乘法, 最大 [381](#)
, [178](#)
成员, [161](#)
memp, [163](#)
memq, [161](#)
memv, [43](#), [161](#)
合并, [387](#)
message-condition? , [368](#)
消息, [52](#), [408](#)
元循环解释器, [404](#)

方法, [317](#)
分钟, [178](#)
mod, [175](#)
mod0, [176](#)
模, [175](#)
mul, [382](#)
多个值, [9](#)
多处理, [421](#)
可变, [331](#)
个相互递归过程, [66](#), [97](#)
命名 let, [67](#), [71](#), [114](#)
命名约定, [8](#)
nan? , [174](#)
native-endianness, [228](#)
native-eol-style, [260](#)
native-transcoder, [259](#)
negative? , [173](#)
嵌套引擎, [429](#)
个嵌套 let 表达式, [96](#)
个换行符, [285](#)
个无穷大违规? , [376](#)
个无 nans-violation? , [377](#)
个不可连续违规? , [369](#)
非确定性计算, [421](#), [424](#)
非生成性, [331](#)

非生成性, [324](#)
非局部退出, [123](#), [124](#)
非, [36](#), [110](#)
空环境, [137](#)
null? , [37](#), [151](#)
数字语法, [459](#)
数字>字符串, [191](#)
数字? , [38](#), [151](#)
数字, [16](#), [167](#)
分子, [181](#)
对象标识, [144](#)
对象>字符串, [267](#)
面向对象编程, [317](#), [408](#)
对象, [3](#)
随机出现, [28](#), [30](#)
八位字节, [257](#)
奇数? , [47](#), [66](#)、[81](#)、[174](#)
仅导入集, [346](#)
不透明, [331](#)
不透明记录类型, [330](#)、[336](#)
开放字节向量输入端口、[264](#)
个开放字节向量输出端口、[265](#)
个打开文件输入端口, [262](#)
个打开文件输入/输出端口, [263](#)
个打开文件输出端口, [262](#)

个打开输入文件, [280](#)
个打开输出文件, [281](#) 个
open-string-input-port, [265](#)
open-string-output-port, [266](#)
操作系统, [423](#), [429](#)
对对象的操作, [141](#)
运算符优先级, [16](#)
个可选参数, [93](#)
或者, [36](#), [63](#), [110](#)
评估顺序, [22](#), [107](#)
输出端口, [257](#)
输出端口缓冲模式, [273](#)
输出端口?, [270](#)
对?, [38](#), [151](#)
对、[19](#)、[155](#)
父类型、[331](#)
父类型、[325](#)
父 rtd、[331](#)
分区、[164](#)
模式变量、[294](#)
模式变量、[61](#), [299](#)
模式, [294](#)
窥视字符, [284](#)
小切兹方案, [ix](#)
指针, [4](#)

por （并行或）， [424](#)
端口， [257](#)
端口-eof？， [278](#)
port-has-port-position？， [271](#)
port-has-set-port-position！？， [272](#)
port-position， [271](#)
port-transcoder， [271](#)
port？， [270](#)
positive？， [173](#)
谓词， [8](#)， [37](#)， [143](#)
前缀导入集， [346](#)
前缀表示法， [15](#)， [16](#)
基元过程， [4](#)
printf， [401](#)
过程应用程序， [16](#)， [17](#)， [21](#)， [27](#)， [107](#)
程序定义， [5](#)， [31](#)， [100](#)
程序？， [155](#)
程序， [26](#)， [91](#)， [92](#)
产品， [74](#)， [80](#)
正确列表， [19](#)， [56](#)， [155](#)
协议， [331](#)
记录协议， [326](#)， [332](#)
推特字节向量， [279](#)
推杆字符， [279](#)
推閤数据， [279](#)， [397](#)

put-string, [279](#)
put-u8, [278](#)
putq! , [54](#)
二次公式, [48](#)
quasiquote (') , [142](#)
quasisyntax (#') , [305](#)
问号 (?) , [8](#), [37](#)
队列, [53](#)
报价 (') , [17](#), [22](#), [59](#), [141](#)
商, [175](#)
加价, [357](#)
个可加持数, [357](#)
个有理数, [167](#)
个有理值? , [153](#)
个有理数? , [151](#), [167](#)
合理化, [181](#)
rcd, [332](#)
read, [284](#)
read-char, [284](#)
real number, [167](#)
real->flonum, [211](#)
real-part, [182](#)
real-value? , [153](#)
real? , [151](#), [167](#)
rec, [311](#)

reciprocal, [15](#), [37](#), [39](#), [80](#)
记录生成性, [324](#)
记录继承, [325](#)
记录 uid, [325](#)
记录访问器、[334](#)
记录构造函数、[333](#)
记录构造函数描述符、[332](#)
记录构造函数描述符、[333](#)
记录字段可变? 、[338](#)
记录变量、[334](#)
记录谓词、[333](#)
记录 rtd、[338](#)
记录类型描述符、[331](#)
记录类型描述符、[333](#)
记录类型描述符? 、[332](#)
记录类型字段名称、[337](#)
记录类型生成名称、[337](#)
记录类型名称、[336](#)
记录类型不透明、[337](#)
记录类型父级、[336](#)
记录类型密封? , [337](#)
记录类型 uid, [336](#)
记录? , [338](#)
记录, [323](#)
递归, [5](#), [41](#), [65](#), [114](#)

递归步骤, [41](#)
递归过程, [41](#)
其余部分, [175](#)
删除, [163](#)
remv, [163](#)
remq, [163](#)
remv, [44](#), [163](#)
重命名导入集, [346](#)
重试, [75](#), [80](#)
反向, [161](#)
修订报告, [ix](#), [3](#)
回合, [178](#)
循环赛, [423](#)
rtd, [331](#)
秒（短）, [169](#)
方案标准, [ix](#)
方案报告环境, [137](#)
范围, [25](#)
密封, [331](#)
密封记录类型, [330](#)
段长度, [132](#)
段斜率, [132](#)
分号（ ; ）， [7](#), [455](#)
序列, [313](#)
序列, [108](#)

严重疾病? , [366](#)
set! , [47](#), [59](#), [102](#)
集车! , [157](#)
set-cdr! , [56](#), [157](#)
set-of, [389](#)
set-port-position! , [272](#)
sets, [389](#)
shadowing, [4](#), [25](#), [31](#)
shhh, [50](#)
更短, [41](#), [47](#)
更短? , [47](#)
副作用, [8](#), [108](#)
简单条件, [362](#)
简单条件, [363](#)
sin, [185](#)
sint-list->bytevector, [239](#)
sort, [387](#)
split, [133](#)
sqrt, [183](#)
square, [14](#)
stack objects, [52](#)
standard-error-port, [264](#)
标准输入端口、[264](#)
个标准输出端口、[264](#)
个流、[128](#)

个字符串、[218](#)
个字符串语法、[458](#)
个字符串>字节向量、[287](#) 个
string->list, [222](#)
string->number, [191](#)
string->symbol, [242](#)
string->utf16, [287](#)
string->utf32, [287](#)
string->utf8, [287](#)
string-append, [219](#)
string-ci-hash, [245](#)
string-ci<=? , [217](#)
string-ci<? , [217](#)
string-ci=? , [217](#)
string-ci>=? , [217](#)
string-ci>? , [217](#)
string-copy, [219](#)
string-downcase, [221](#)
string-fill! , [220](#)
string-foldcase, [221](#)
string-for-each, [122](#)
string-hash, [245](#)
string-length, [218](#)
string-normalize-nfc, [222](#)
string-normalize-nfd, [222](#)

string-normalize-nfkc, [222](#)
string-normalize-nfkd, [222](#)
string-ref, [218](#)
string-set! , [219](#)
string-titlecase, [221](#)
string-upcase, [221](#)
string<=? , [216](#)
string<? , [216](#)
string=? , [216](#)
string>=? , [216](#)
string>? , [216](#)
string? , [38](#), [154](#)
字符串, [14](#), [216](#)
结构化形式, [6](#)
结构, [318](#)
子字符串, [95](#), [220](#)
和, [65](#)
符号语法, [458](#)
符号表, [241](#)
符号>字符串, [242](#)
符号哈希, [245](#)
符号=? , [242](#)
符号? , [38](#), [154](#)
符号, [18](#), [241](#)
句法扩展, [5](#), [22](#), [59](#), [60](#), [291](#)

句法形式, [18](#), [59](#), [291](#)
语法, [291](#)
语法 (#'), [300](#)
语法对象, [298](#)
语法违规, [9](#)
语法->datum, [308](#)
语法大小写, [291](#), [299](#)
语法规则, [291](#), [294](#), [300](#), [389](#)
语法违规, [359](#)
语法违规形式, [370](#)
语法-违规-子窗体, [370](#)
语法-违规?, [370](#)
尾调用, [5](#), [68](#)
尾递归, [5](#), [68](#)
tan, [185](#)
tconc, [53](#)
告诉, [50](#)
模板, [295](#)
文本端口, [257](#)
文本端口?, [270](#)
线程, [421](#)
thunk, [51](#), [124](#)
刻度, 请参阅引擎
定时抢占, [421](#)
计时器中断, [425](#)

个令牌, [455](#)

个顶级定义, [30](#) 个、[101](#)

个顶级程序、[343](#)

个跟踪、[42](#)

个跟踪、[42](#)

个转码端口、[271](#)

个转码器、[257](#)

转码器编解码器, [259](#)

转码器 eol 样式, [259](#)

转码器错误处理模式, [259](#)

转换器, [61](#)

树副本, [44](#)

真, [7](#), [36](#)

截断, [177](#)

类型谓词, [38](#)

u8-list->字节向量, [232](#)

uint-list->bytevector, [239](#)

undefined-violation? , [371](#)

下划线 (_), [61](#), [296](#), [315](#)

下划线 (_), [294](#)

统一, [417](#)

统一, [418](#)

除非, [64](#), [112](#)

取消引号 (,), [142](#)

未引用拼接 (, @), [142](#)

未指定, [9](#)
未语法 (#,), [305](#)
非语法拼接 (#, @), [305](#)
展开保护 (在 Lisp 中), [124](#)
utf-16, [257](#)
utf-16-编解码器, [259](#)
utf-8, [257](#)
utf-8-编解码器, [259](#)
utf16->字符串, [288](#)
utf32->string, [288](#)
utf8->string, [287](#)
值, [130](#), [131](#)
变量绑定, [23](#), [91](#)
变量引用, [91](#)
变量, [4](#), [18](#), [23](#), [30](#), [47](#)
向量, [224](#)
向量语法, [461](#)
个矢量>列表, [225](#)
个矢量填充!, 每个 [225](#)
个矢量, [122](#)
个矢量长度, [224](#)
个矢量映射, [121](#)
个矢量引用, [224](#)
vector-set!, [225](#)
vector-sort, [226](#)

vector-sort! , [226](#)
vector? , [154](#)
vectors, [55](#), [223](#), [383](#)
违规? , [366](#)
警告? , [367](#)
当, [64](#), [112](#)
空格, [455](#)
空白字符, [7](#)
谁条件? , [369](#)
绕线器, 请参阅 `dynamic-wind`
`with-exception-handler`, [360](#)
`with-input-from-file`, [283](#)
`with-output-to-file`, [283](#)
`with-syntax`, [304](#)
`write`, [284](#), [397](#)
写字符, [285](#)
`x++`, [316](#)
零? , [173](#)

R. Kent Dybvig / The Scheme Programming Language, Fourth Edition

Copyright © 2009 The MIT Press. 经许可以电子方式复制。

插图 © 2009 让-皮埃尔·赫伯特

ISBN 978-0-262-51298-5 / LOC

QA76. 73. S34D93

订购本书 / 关于这本书

<http://www.scheme.com>