Date! □□□□

□□

1. □□

`date!`
□□□□□□□□□□□□□□□□□□□□□□□date□□□□□□□□□□□□□time□□□□□□□□□□□□□□□date□□□time□24□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Red□date□ ISO 8601 □□□□□□□□□□□□□□□□□□□□□□□□□□

Note:

- date! は グレゴリオ暦 （1582年から採用）に基づいていますが、これ以前の日付についても同様に date! の日付規則が適用されるものとします。

- 内部表現における年の有効範囲は、理論的には西暦 1/Jan/-16384 から 31/Dec/16383 までですが、現在の実装では西暦 1/Jan/-9999 から 31/Dec/9999 までに制限されています。

2. 生成

date! 値はソースコードをロードもしくはパースするときに生成されるほか、 make を使った構築やほかの型からの to 変換でも生成されます。

2.1. リテラル構文

日付リテラルにはいくつかの書式がありますが、これはRebol言語の日付リテラルとISO日付書式の両方をサポートするためです。

```
<date>/<time>
<date>/<time><zone>
<date>T<time>
<date>T<time>Z
<date>T<time><zone>

<date> の書式は以下:
    <yyyy><sep><mmm><sep><dd>
    <dd><sep><mmm><sep><yyyy>
    <dd><sep><mmm><sep><yy>

Tプレフィックスが付いた <date> の書式は以下:
    <yyyy><mm><dd>
    <yyyy>-W<ww>
    <yyyy>-W<ww>-<d>
    <yyyy>-<ddd>

<mmm> の書式は以下 のどれか:
    <m>
    <mon>
    <month>

<time> の書式は以下:
    <hour>:<min>:<sec>
    <hour>:<min>:<sec><zone>
    <hhmmss>
    <hhmmss><zone>
    <hhmmss>.<dec>
    <hhmmss>.<dec><zone>
    <hhmm>
    <hhmm><zone>

<zone> の書式は以下のどちらか:
    <sign><hour>
```

```
    <sign><hour>:<min15>
    <sign><hhmm>

<sec> □□□□□□□□□□
    <ss>
    <ss>.<dec>

<sep>     : `-` □□□ `/`
<yyyy>    : □□□□3□□□□4□□□□□4□□ISO□□□□□□□□
<yy>      : 2000□□□□□□□□□□2□□□□□
<m>       : □□□□□1□□□□2□□□□□
<mm>      : □□□□□2□□□□□
<mon>   : □□□□□□3□□
<month>  : □□□□
<d>       : □□□□□□□□□1□□7□
<dd>      : □□□□□□□□□1□□□□2□□□□□
<ddd>     : □□□□□□□□□3□□□□
<ww>      : □□□□□□□□2□□□□□
<time>    : time!□□
<hour>    : □□□□□1□□□□2□□□□
<min>     : □□□□1□□□□2□□□□□
<ss>      : □□□□1□□□□2□□□□□
<dec>     : □□□□□□□□□
<sign>    : `+` □□□ `-` □□□□□□□□□□□□
<min15>   : <min>□□□□□□□15□□□□□□□□□□□
<hhmm>    : □□□□□□□□□4□□□□□□□□□□□□□
<hhmmss> : □□□□□□□□□6□□□□□□□□□□□□□
```

date□□□

```
<dd>-<mon>-<yyyy>
<dd>-<mon>-<yyyy>/<hour>:<min>:<sec>
<dd>-<mon>-<yyyy>/<hour>:<min>:<sec><sign><hour>:<min15>
```

time□zone□□□-□□□□□□□□/□□□□□□□□□□□□□□□□□□□□□□□□□

Notes:

- □□
- □□2□□□□□□yy□□□□□□□□□□□□□□□□□50□□□□□□□□□2000□□□□□□□□□□□50□□□□□□□1900□□□□□□□□□□□□□□□

□□□□date□□□□□□□□□□□□□□□□□

```
1999-10-5
1999/10/5
5-10-1999
5/10/1999
5-October-1999
1999-9-11
11-9-1999
5/sep/2012
5-SEPTEMBER-2012

02/03/04
02/03/71

5/9/2012/6:0
5/9/2012/6:00
5/9/2012/6:00+8
5/9/2012/6:0+0430
4/Apr/2000/6:00+8:00
1999-10-2/2:00-4:30
1/1/1990/12:20:25-6

2017-07-07T08:22:23+00:00
2017-07-07T08:22:23Z
20170707T082223Z
20170707T0822Z
20170707T082223+0530

2017-W01
2017-W23-5
2017-W23-5T10:50Z
2017-001
2017-153T10:50:00-4:00
```

2.2. 🔲🔲🔲🔲🔲

```
make date! [<day> <month> <year>]
make date! [<year> <month> <day>]
make date! [<day> <month> <year> <time>]
make date! [<day> <month> <year> <time> <zone>]
make date! [<day> <month> <year> <hour> <minute> <second>]
make date! [<day> <month> <year> <hour> <minute> <second> <zone>]

<year>   : integer!□□
<month>  : integer!□□
<day>    : integer!□□
<time>   : time!□□
<zone>   : integer!□time!□□□pair!□□
<hour>   : integer!□□
<minute> : integer!□□
<second> : integer!□□
```

Notes:

- □□ make □□□□ to □□□□□□□□□□□□□□□□□□□□□□□□

- year □ day □□100□□□3□□□□□□□□□□□□□□□□□□□□□ □□ 1□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□3□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□3□□□□□□□□□□□□□□□□□□

例

```
make date! [1978 2 3]
== 3-Feb-1978

make date! [1978 2 3 5:0:0 8]
== 3-Feb-1978/5:00:00+08:00

make date! [1978 2 3 5:0:0]
== 3-Feb-1978/5:00:00

make date! [1978 2 3 5 20 30]
== 3-Feb-1978/5:20:30

make date! [1978 2 3 5 20 30 -4]
== 3-Feb-1978/5:20:30-4:00


make date! [100 12 31]
== 31-Dec-0100

; 32 isn't a valid day
make date! [100 12 32]
*** Script Error: cannot MAKE/TO date! from: [100 12 32]
*** Where: make

; First field is < 100, so not considered as a year
make date! [99 12 31]
*** Script Error: cannot MAKE/TO date! from: [99 12 31]
*** Where: make
```

3. □□□□□□

□□□□□□□□□□□□□□□□□□ date! □□□□□□□□□□□□□□□□□□□

3.1. /date

语法

```
<date>/date
<date>/date: <date2>

<date>   : date!□□□□□□□word□□□□□□□□
<date2> : date!□□
```

描述

□□□□□□□□□□□□□□□time□zone□□□□□□□□□□□□□□□□□□□□□□□□□□ date! □□□□□□□□□□□□

```
d:  now
== 10-Jul-2017/22:46:22-06:00
d/date
== 10-Jul-2017

d/date: 15/09/2017
== 15-Sep-2017/22:46:22-06:00
```

3.2. /year

```
<date>/year
<date>/year: <year>

<date> : date!▯▯▯▯▯▯word▯▯▯▯▯▯▯
<year> : integer!▯
```

▯▯

```
d:  now
== 10-Jul-2017/22:46:22-06:00
d/year: 16383
== 16383
d
== 10-Jul-16383/22:46:22-06:00
d/year: 16384
== 16384
d
== 10/Jul/-16384/22:46:22-06:00      ; 16384▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
d/year: 32767
== 32767
d
== 10/Jul/-1/22:46:22-06:00
d/year: 32768
== 32768
d
== 10-Jul-0000/22:46:22-06:00
```

3.3. /month

```
<date>/month
<date>/month: <month>

<date>  : date!形式（またはword形式）の値
<month> : integer!値
```

説明

月を取得または設定します。月を設定すると、範囲を超えた場合は年が繰り上がり（または繰り下がり）ます。

例

```
d: now
== 10-Jul-2017/22:48:31-06:00
d/month: 12
== 12
d
== 10-Dec-2017/22:48:31-06:00
d/month: 13
== 13
d
== 10-Jan-2018/22:48:31-06:00    ; 月が範囲を超えたので、年が繰り上がった
d/month
== 1                             ; 月の部分だけが返される
```

3.4. /day

構文

```
<date>/day
<date>/day: <day>

<date> : date!形式（またはword形式）の値
<day>  : integer!値
```

説明

日を取得または設定します。日を設定すると、範囲を超えた場合は月が繰り上がり（または繰り下がり）ます。

```
 d: 1-jan-2017
== 1-Jan-2017
d/day: 32
== 32
d
== 1-Feb-2017
d/day: 0          ; 0は前の月の最後の日を表すので、一日戻ったことになる
== 0
d
== 31-Jan-2017
```

3.5. /time

構文

```
<date>/time
<date>/time: <time>

<date> : date!。上位互換性をもつwordも同様に使える。
<time> : time!。またはnone!。
```

説明

日付の時刻部分をアクセスまたは設定します。設定する値は time! またはゼロが入ります。設定する値がtimeである場合、タイムゾーンは、変更されません。日付がタイムゾーンを保持していない場合はセットされます。設定する値が none! である場合は、日付の時刻部分が削除されます。時刻のない日付が戻ります。

timeが none! であるとき、日付が持つtimeとzoneは取り除かれ、0で返されます。以下の例で示します。

例

```
d: now
== 10-Jul-2017/23:18:54-06:00
d/time: 1:2:3
== 1:02:03
d
== 10-Jul-2017/1:02:03-06:00
d/time: none
== 10-Jul-2017
```

3.6. /hour

語法

```
<date>/hour
<date>/hour: <hour>

<date> : date!□□□□□□word□□□□□□
<hour> : integer!□
```

描述

这设置日期中的小时部分。合法的取值范围是0到23。注意更改后的值不会进行归一化处理，当超出范围时会调整日期。

例

```
d: now
== 10-Jul-2017/23:19:40-06:00
d/hour: 0
== 0
d
== 10-Jul-2017/0:19:40-06:00
d/hour: 24
== 24
d
== 11-Jul-2017/0:19:40-06:00
```

3.7. /minute

语法

```
<date>/minute
<date>/minute: <minute>

<date>   : date!□□□□□□word□□□□□□
<minute> : integer!□
```

描述

这设置日期中的分钟部分。合法的取值范围是0到59。注意更改后的值不会进行归一化处理，当超出范围时会调整日期。

の

```
== 10-Jul-2017/23:20:25-06:00
d/minute: 0
== 0
d
== 10-Jul-2017/23:00:25-06:00
d/minute: 60
== 60
d
== 11-Jul-2017/0:00:25-06:00
```

3.8. /second

構文

```
<date>/second
<date>/second: <second>

<date>   : date!（または、word、パス、）
<second> : integer!（または、float!）
```

説明

日付の秒フィールドにアクセスします。通常は0から59ですが、 integer! または float! を設定することで、任意の値を設定できます。範囲外の値は正規化されます。

例

```
d: now
== 10-Jul-2017/23:21:15-06:00
d/second: 0
== 0
d
== 10-Jul-2017/23:21:00-06:00
d/second: -1
== -1
d
== 10-Jul-2017/23:20:59-06:00
d/second: 60
== 60
d
== 10-Jul-2017/23:21:00-06:00
```

3.9. /zone

```
<date>/zone
<date>/zone: <zone>

<date> : date!□□□□□word□□□□□□
<zone> : time!□□□integer!□
```

□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ -16:00 □□ +15:00 □□□□ `time!` □□□□□□□□□□ `/zone` □□□□□□□□□□□□□□□□□□zone□□□□□□□□□□□□□□time□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□ `integer!` □□□□□□□□□□□□□□□□□□□□□□□□□□□□0□□□□□□□□□□□

□□□□□□□□□□□□15□□□□□□□□□□□□□□□□15□□□□□□□□□□□□□□□□□□

□

```
d: 1/3/2017/5:30:0
d/zone: 8
== 1-Mar-2017/5:30:00+08:00

d/zone: -4:00
== 1-Mar-2017/5:30:00-04:00
```

3.10. /timezone

□□

```
<date>/timezone
<date>/timezone: <zone>

<date>    : date!□□□□□word□□□□□□
<timezone> : integer!□time!□□□pair!□□
```

□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ -16:00 □□ +15:00 □□□□ `time!` □□□□□□□□□□□□□□□□ `/timezone` □□□□□□□□□time□zone□□□□□□□□□□□□□□□□□□□□□□□zone□□time□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

timezone□ `integer!` □□□□□□□□□□□□□□□□□□□□□□□□□□□□0□□□□□□□□□□□

□□□□□□□□□□□□15□□□□□□□□□□□□□□□□15□□□□□□□□□□□□□□□□□□

例

```
d: 1/3/2017/5:30:0
d/timezone: 8
== 1-Mar-2017/13:30:00+08:00

d/timezone: -4:00
== 1-Mar-2017/1:30:00-04:00
```

Note:

- `/timezone` 为0时，不显示时区，即默认为UTC时区的时间。

3.11. /yearday

格式

```
<date>/yearday
<date>/yearday: <day>

<date>    : date!类型的值，或word类型的变量。
<yearday> : integer!值
```

说明

返回一个表示距离该日期所在年的1月1日（第1天）有多少天的整数，或者给一个日期的这个域赋值，从而改变这个日期（年份不变），也就是将日期调整到距离该年的第一天有几天的位置。

Note:

- `/yearday` 域也可以使用别名 `/julian` 来进行访问。这是因为Rebol语言习惯使用这个别名来访问。

```
d: 1-jan-2017
== 1-Jan-2017
d/yearday
== 1
d: 31-dec-2017
== 31-Dec-2017
d/yearday
== 365
d: 31-dec-2020
== 31-Dec-2020
d/yearday
== 366                    ;  □□□□

d: 31-dec-2017
== 31-Dec-2017
d/yearday: 366
== 366
d
== 1-Jan-2018
```

3.12. /weekday

□□

```
<date>/weekday
<date>/weekday: <day>

<date>    : date!□□□□□□□word□□□□□□□
<weekday> : integer!□
```

□□

□□□□□□□□□□□□□□□□□□□□□1□□□□□□7□□□

例

```
d: now
== 10-Jul-2017/23:25:35-06:00
d/weekday
== 1
d/weekday: 2
== 2
d
== 11-Jul-2017/23:25:35-06:00
d/weekday: 7
== 7
d
== 16-Jul-2017/23:25:35-06:00
d/weekday: 8
== 8
d
== 17-Jul-2017/23:25:35-06:00
```

3.13. /week

语法

```
<date>/week
<date>/week: <day>

<date> : date!，日期或含有word！的时间。
<week> : integer!。
```

描述

按照自然周的规则进行计算，即：一年中含有1月1日的那个自然周，就是该年的第1周。一年中最多有53个自然周。可以读取和设置该值。设置该值的时候，将以最接近的方式进行计算（相隔最近的那个自然周对应的日期）。

Note:

- 设置该值的时候，会将当前日期设置为周日（即1而不是7，当一周的开始为周日的时候）。可以参考下面将会介绍的 `/isoweek` （如果你希望一周开始于周一）。

```
d: now
== 10-Jul-2017/23:28:07-06:00
d/week
== 28
d/week: 29
== 29
d
== 16-Jul-2017/23:28:07-06:00
d/week: 52
== 52
d
== 24-Dec-2017/23:28:07-06:00
d/week: 53
== 53
d
== 31-Dec-2017/23:28:07-06:00
d/week: 54
== 54
d
== 7-Jan-2018/23:28:07-06:00
```

3.14. /isoweek

構文

```
<date>/isoweek
<date>/isoweek: <day>

<date>    : date!型。日付のwordを含んでもよい
<isoweek> : integer!。
```

説明

[ISO   8601](#)
の週番号として日付を返します。返される値は各年で最初は1番であり、52番あるいは53番まで数えます。ほとんどの機能は週と共通していますが、各年の終わりを越えた週の扱いが異なります。曜日について上記を参照してください。

```
 d: now
 == 10-Jul-2017/23:29:13-06:00
 d/isoweek
 == 28
 d/isoweek: 29
 == 29
 d
 == 17-Jul-2017/23:29:13-06:00
 d/isoweek: 52
 == 52
 d
 == 25-Dec-2017/23:29:13-06:00
 d/isoweek: 53
 == 53
 d
 == 1-Jan-2018/23:29:13-06:00
```

3.15. 〇〇〇〇〇〇〇〇〇〇〇〇

〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇word〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇

```
<date>/<index>

<date>  : date!〇〇〇〇〇word〇〇〇〇〇〇
<index> : 〇〇〇〇〇〇〇〇〇〇〇〇integer!〇
```

〇〇

〇〇〇〇〇〇	〇〇〇〇〇〇
1	date
2	year
3	month
4	day
5	zone
6	time
7	hour
8	minute
9	second
10	weekday
11	yearday

□□□□□□□	□□□□□□□
12	timezone
13	week
14	isoweek

3.16. Pick□□□□□□□□□□□□□□□□□□□□□□□

□□□ `pick` □□□□□□□□

□□

```
pick <date> <field>

<date>  : date!□
<field> : integer!□
```

integer□□□

□

```
d: now
== 10-Jul-2017/23:35:01-06:00
names: system/catalog/accessors/date!
repeat i 14 [print [pad i 4 pad names/:i 10 pick d i]]
1     date       11-Jul-2017
2     year       2017
3     month      7
4     day        11
5     zone       8:00:00
6     time       21:43:52
7     hour       21
8     minute     43
9     second     52.0
10    weekday    2
11    yearday    192
12    timezone   8:00:00
13    week       28
14    isoweek    28
```

4. □□□□

4.1. UNIX□□

□□□□ UNIX□□ □□□□ to □□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
to-integer <date>
to-date <epoch>

<date>  : date!□
<epoch> : UNIX□□□□□integer!□
```

UNIX□□□□UTC□□□□□□□□□□□□□□UTC□□□□□□□□□UNIX□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
d: 8-Jul-2017/17:49:27+08:00
to-integer d
== 1499507367

to-integer 8-Jul-2017/9:49:27
== 1499507367

to-date to-integer d
== 8-Jul-2017/9:49:27
```

UNIX□□□□2038□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

4.2. □□□□□□□□□□□□□□□

格式

```
to date! <spec>

<spec> : □□□□□□□□□□□□□□□□□
```

□□□□□□□□□□□□□ make □□□□□□□□□□□□□2.2□ □□□□□□□□□ □□□□□□□□□□□□□□□□□□ date!
□□□ make □□□□□□□□□□□□□□

5. □□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□=, ==, <>, >, <, >=, <=, same? □□□□□min□max□sort □□□□□□□□□□□□□□□□□

```
3-Jul-2017/9:41:40+2:00 = 3-Jul-2017/5:41:40-2:00
== true

10/10/2017 < 1/1/2017
== false

max 10/10/2017 1/1/2017
== 10-Oct-2017

same? 1/1/1980 1-JAN-1980
== true

sort [1/1/2017 5/10/1999 3-Jul-2017/5:41:40-2:00 1/1/1950 1/1/1980/2:2:2]
== [1-Jan-1950 1-Jan-1980/2:02:02 5-Oct-1999 1-Jan-2017 3-Jul-2017/5:41:40-02:00]
```

6. 口口口口

口口口口口口口口口口口口口口口口口口口口口口口口

- 口口口口口口口口口口口口口口口口口口口口口口口口口口口口口口口口口口口口
- 口口口口口口口口口口口口口口口口口口口口口口口口口口口口口口口
- 口口口口口time口口口口口口口口口口date口口口口口口口口口口口口口口口口口口口口口口口口
- 2口口口口口口口口口口口口2口口口口口口口口口口口口口口口口口口口口口口口口口
- 2口口口口口口 difference 口口口口口口口口口口2口口口口口口口口口口口口 time! 口口口口口口口口口口

```
20-Feb-1980 + 50
== 10-Apr-1980

20-Feb-1980 + 3
== 23-Feb-1980

20-Feb-1980 - 25
== 26-Jan-1980

20-Feb-1980 + 100
== 30-May-1980

28-Feb-1980 + 20:30:45
== 28-Feb-1980/20:30:45

28-Feb-1980/8:30:00 + 20:30:45
== 29-Feb-1980/5:00:45

d: 20-Feb-1980
d/day: d/day + 50
== 10-Apr-1980

d: 20-Feb-1980
d/month: d/month + 5
== 20-Jul-1980

d: 28-Feb-1980/8:30:00
d/hour: d/hour + 48
== 1-Mar-1980/8:30:00

08/07/2017/10:45:00 - 20-Feb-1980/05:30:0
== 13653

difference 08/07/2017/10:45:00 20-Feb-1980/05:30:0
327677:15:00
```

7. ⬚⬚⬚⬚⬚⬚

now ⬚⬚⬚OS⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚ now
⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚

- /utc⬚ UTC⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚

- /precise⬚ ⬚⬚⬚⬚⬚⬚Windows⬚⬚60⬚⬚1⬚⬚⬚⬚⬚Unix⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚

例

```
now
== 8-Jul-2017/18:32:25+08:00

now/year
== 2017

now/hour
== 18

now/month
== 7

now/day
== 8

now/hour
== 18

now/zone
== 8:00:00

now/utc
== 8-Jul-2017/10:32:25
```

8. 生成随机值或序列生成器

8.1. Random

语法

```
random <date>

<date> : date!值
```

描述

该函数随机选取给定日期内的日期。如果日期有time和timezone的部分，则仅随机生成日期部分；否则会随机生成time和timezone，导致结果不同。

```
random 09/07/2017
== 18-May-1972

random 09/07/2017
== 13-Aug-0981

random 09/07/2017/12:00:00+8
== 28-Feb-0341/17:57:12+04:00

random 09/07/2017/12:00:00+8
== 13-Dec-1062/5:09:12-00:30
```