20章 Time类与Date类

作者 bluetea
网站:https://github.com/bluetea

20.1 Time类和Date类

通过Time.new 或者Time.now来获得当前时间的对象

```
[9] pry(main)> a = Time.new
=> 2015-01-13 21:27:58 +0800
[10] pry(main)> a
=> 2015-01-13 21:27:58 +0800
[11] pry(main)> a
=> 2015-01-13 21:27:58 +0800
[12] pry(main)> Time.now
=> 2015-01-13 21:28:10 +0800
[13] pry(main)> Time.now
=> 2015-01-13 21:28:11 +0800
```

当你获取一个Time.new对象后a的值是不会再变的

```
[14] pry(main)> t = Time.now
=> 2015-01-13 21:31:22 +0800
[15] pry(main)> t.year
=> 2015
[16] pry(main)> t.month
=> 1
[17] pry(main)> t.day
=> 13
```

```
=> 22
[21] pry(main)> t = Time.now
=> 2015-01-13 21:32:21 +0800
[22] pry(main)> t.hour
=> 21
[23] pry(main)> t.min
=> 32
[24] pry(main)> t.sec
=> 21
```

t.to_i 从1970年1月1日到当前时间的秒数
t.wday 一周中的第几天（0表示星期日）

```
[26] pry(main)> t.wday
=> 2
```

t.mday 一个月中的第几天
t.yday 一年中的第几天
t.zone 时区

```
[27] pry(main)> t.zone
=> "CST"
```

Time.mktime方法可以根据制定时间获取Time对象

```
[28] pry(main)> t = Time.mktime(2016, 2, 30)
=> 2016-03-01 00:00:00 +0800
[29] pry(main)> t.year
=> 2016
[30] pry(main)> t.month
=> 3
[31] pry(main)> t.day
=> 1
```

20.3 时间的计算
Time对象之间可以互相比较和计算的

```
[32] pry(main)> t1 = Time.now
=> 2015-01-13 21:38:21 +0800
[33] pry(main)> sleep 10
=> 10
[34] pry(main)> t2 = Time.now
=> 2015-01-13 21:38:50 +0800
[35] pry(main)> t1 > t2
=> false
[36] pry(main)> t1 < t2
=> true
[37] pry(main)> t2 - t1
=> 28.186516
```

```
[46] pry(main)> t
=> 2015-01-13 22:21:23 +0800
[47] pry(main)> t2 = t + 60 * 60 +24
=> 2015-01-13 23:21:47 +0800
```

t.strftime(format) 更改时间输出格式

```
[49] pry(main)> t = Time.now
 => 2015-01-13 22:23:52 +0800
 [50] pry(main)> t.to_s
 => "2015-01-13 22:23:52 +0800"
 [51] pry(main)> t.strftime("%Y-%m-%d ")
 => "2015-01-13 "
 [52] pry(main)> t.strftime("%Y-%m-%d %H:%M:%S")
 => "2015-01-13 22:23:52"
 [53] pry(main)> t.strftime(" %H:%M:%S %Y-%m-%d")
 => " 22:23:52 2015-01-13"
```

t.rfc2822 通过Time#rfc2822方法可以生成email头部文件信息中的Date:字段格式字符串，在互联网的相关文档RFC2822文档有关于email形式定义， 需要ruquire time库

```
1  require "time"
2  t = Time.now
3  p t.rfc2822
```

```
ruby
bogon:ruby wangmjc$ ruby encode.rb
"Tue, 13 Jan 2015 22:30:11 +0800"
bogon:ruby wangmjc$
```

t.iso8601 通过Time#iso 8601生成符合ISO8601国际标准时间格式的字符串

```
[59] pry(main)> require "time"
=> true
[60] pry(main)> t.iso8601
=> "2015-01-13T22:31:40+08:00"
[61] pry(main)> t.rfc2822
=> "Tue, 13 Jan 2015 22:31:40 +0800"
[62] pry(main)>
```

20.5 本地时间

世界各地都有时差，大家的计算机中也设置有时区， 一般计算机中的时间都是根据时区来设置的，Time#localtime 将UTC变更为本地时间

```
[64] pry(main)> t.utc
=> 2015-01-13 14:31:40 UTC
[65] pry(main)> t.localtime
=> 2015-01-13 22:31:40 +0800
```

20.6 从字符串中获取时间

Time.parse(str)

```
[67] pry(main)> t = Time.parse("Tue, 13 Jan 2015 22:31:40 +0800")
=> 2015-01-13 22:31:40 +0800
[68] pry(main)> t = Time.parse("2015-01-13T22:31:40+08:00")
=> 2015-01-13 22:31:40 +0800
```

)

20.7 日期的获取

Date类用于处理不包含时间的日期，使用Date.today方法可以得到表示当前日期的的Date对象， 需要require "date"

```
2.2.0 :001 > require "date"
 => true
2.2.0 :002 > d = Date.new
 => #<Date: -4712-01-01 ((0j,0s,0n),+0s,2299161j)>
2.2.0 :003 > d = Date.today
 => #<Date: 2015-01-13 ((2457036j,0s,0n),+0s,2299161j)>
2.2.0 :004 > d.year
 => 2015
2.2.0 :005 > d.month
 => 1
2.2.0 :006 > d.day
 => 13
2.2.0 :007 > d.wday
 => 2
2.2.0 :008 > d.mday
 => 13
2.2.0 :009 > d.yday
 => 13
```

用Date.new方法生成指定对象

```
[3] pry(main)> d = Date.new(2013,3,2)
=> #<Date: 2013-03-02 ((2456354j,0s,0n),+0s,2299161j)>
[4] pry(main)> puts d
2013-03-02
=> nil
```

Date类还有一个特点，就是 可以对月末的日期做-1处理（-2表示月末的前一天）

```
[5] pry(main)> d = Date.new(2013,3,2)
=> #<Date: 2013-03-02 ((2456354j,0s,0n),+0s,2299161j)>
[6] pry(main)> d = Date.new(2013,3,-1)
=> #<Date: 2013-03-31 ((2456383j,0s,0n),+0s,2299161j)>
[7] pry(main)> d = Date.new(2013,3,-2)
=> #<Date: 2013-03-30 ((2456382j,0s,0n),+0s,2299161j)>
[8] pry(main)> d = Date.new(2013,-1,-2)
=> #<Date: 2013-12-30 ((2456657j,0s,0n),+0s,2299161j)>
```

20.8日期的运算

```
[13] pry(main)> d1 = Date.new(2013,1,1)
=> #<Date: 2013-01-01 ((2456294j,0s,0n),+0s,2299161j)>
[14] pry(main)> d2 = Date.new(2013,1,4)
=> #<Date: 2013-01-04 ((2456297j,0s,0n),+0s,2299161j)>
[15] pry(main)> d2-d1
=> (3/1)
[16] pry(main)> d1 +1
=> #<Date: 2013-01-02 ((2456295j,0s,0n),+0s,2299161j)>
[17] pry(main)> d1 +100
=> #<Date: 2013-04-11 ((2456394j,0s,0n),+0s,2299161j)>
[18] pry(main)> d1 -1
=> #<Date: 2012-12-31 ((2456293j,0s,0n),+0s,2299161j)>
[19] pry(main)> d1 -100
=> #<Date: 2012-09-23 ((2456194j,0s,0n),+0s,2299161j)>
```

使用>>运算符，可以获得一个月后相同日期的date对象 << 相反

```
[21] pry(main)> d1
=> #<Date: 2013-01-01 ((2456294j,0s,0n),+0s,2299161j)>
[22] pry(main)> d1 >> 2
=> #<Date: 2013-03-01 ((2456353j,0s,0n),+0s,2299161j)>
[23] pry(main)> d1 << 1
=> #<Date: 2012-12-01 ((2456263j,0s,0n),+0s,2299161j)>
```

Date对象也可以使用strftime方法，时间部分会变为0

```
[25] pry(main)> d1
=> #<Date: 2013-01-01 ((2456294j,0s,0n),+0s,2299161j)>
[26] pry(main)> d1.strftime("%Y-%m-%d %H:%M:%S")
=> "2013-01-01 00:00:00"
```

从字符串中获取日期 Date.parse(str)