

支持史前及公元任意年份的datetime — 从史前到宇宙终结的万年历算法

在 2021年7月8日 上张贴 由 qyfcool 已关闭评论

众所周知，程序中的datetime通常会有一个时间基准点，通常是1991年1月1日，并以距此时间点的秒数来表示时间点，而这秒数的范围通常会受到其类型（uint32）的限制。

由于我需要做一个历史时间轴软件，需要史前及亿万年的时间表达，因此需要自己写一个时间表征算法。即以公元元年为起点（0秒），以负的秒数表达史前时间，并以此为基准推算任意时间对应的年月日及星期。

这个项目在此：<https://github.com/SleepySoft/History>

在开始之前，让我们分析一下时间的表达形式：秒，分，时，日，月，周，年

其中秒为根本。而分，时，日，周的进制都是恒定的，给一个秒数，可以用简单的除法算出分，时，日，周。而月分大小月以及闰月，年分平年闰年，也就是说它们的进制并不确定，这是最为麻烦的地方。

而其中麻烦的源头，则是闰年带来的周期性变化。而反过来说，一旦确定闰年与否，那么该年的天数，以及每个月的天数就能确定了。月分尽管天数有所不同，但通过查表的方式可以很好地解决。

如此说来，我们需要一个函数，用来判断某年是否闰年，基于这个结果，我们才能往下走。相信这个算法大家都已经熟得不能再熟了：

```
(year % 4 == 0 and year % 100 != 0) or (year % 400 == 0)
```

基于闰年的判断，我们可以得到以下函数（TICK指秒数，下同）：

```
def year_ticks(leap_year: bool) -> TICK
def year_days(leap_year: bool) -> int
def month_ticks(month: int, leap_year: bool) -> TICK
def month_days(month: int, leap_year: bool) -> int
```

现在我们已经能知道一年的平闰性及该年的秒数，那么我任意给出一个秒数，你能算出它对应多少年吗？没错，这是一大难点，因为你无法直接除以一个确切的数。如果你想用迭代的方法来说，不好意思，对于 10^{60} 年，那么计算的速度会让人无法接受。我采用的计算方法是：

1. 用总天数计算有几个400年
2. 用上一个算式的余数计算有几个100年
3. 用上一个算式的余数计算有几个4年
4. 如果最终余下的正好为4年的天数-1，那么实际剩余年份应该是3年；否则除以365即得到剩余的年份
5. 将上述的商分别乘以对应的除数并加上最终剩余的年份即可

其中各个周期的天数计算方法如下：

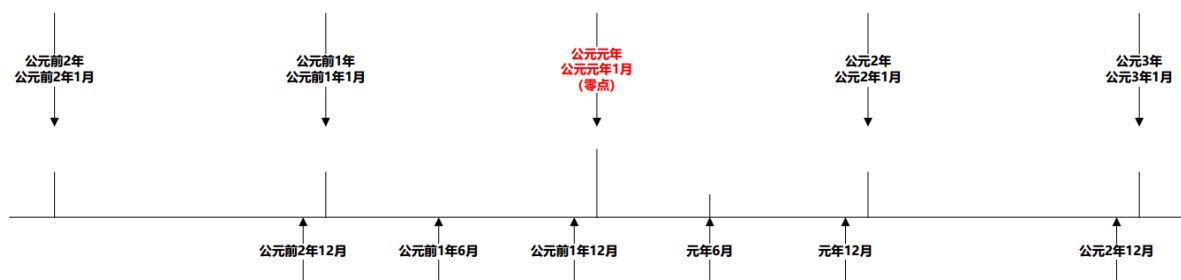
```
DAYS_PER_4_YEARS = 365 * 4 + 4 // 4 - 4 // 100 + 4 // 400
DAYS_PER_100_YEARS = 365 * 100 + 100 // 4 - 100 // 100 + 100 // 400
DAYS_PER_400_YEARS = 365 * 400 + 400 // 4 - 400 // 100 + 400 // 400
```

后面的事情就简单了，除去年份后，还剩余多少天，通过查表（注意平闰年）即可得到是几月几日。然而其中还有个问题，这是我在查找“公元元年是星期几”这个问题查到的：

<https://www.zhihu.com/question/27580677/answer/270147876>

不过在这里我就不做这个调整了，我的程序只需要表征时间而不需要算星期，所以问题并不大。

这么一来公元纪年的问题基本都解决了，剩下的就是公元前的问题。如果以负数表征公元前时间的话，我们发现它和公元后的时间并非镜像的关系。确切来说，年与公元排布为镜像，而月日则并非如此。



对于这样的映射，作为程序员，我们通常采用取补的方式。举个例子：-1日（公元前1日）是几月几号？

没错应该是公元前1年12月31号而非1月1号。程序中是怎么算的呢，很简单，用当年的总天数减掉当前天数，再按公元纪年后的方法计算就可以了。

然而在计算公元前年份的时候要注意边界的处理，比如公元前365天是公元前几年？公元前1年？不，是公元前2年，即公元前2年12月31日。导致这个问题的原因在于，0秒被分配给了公元纪年，所

以公元前的时间需要偏移一秒 -- 这仅对年份能够被“整除”的情况。

具体实现的代码在这个文件里：

https://github.com/SleepySoft/History/blob/master/Utility/history_time.py

配合上文的解释，代码应该很好理解，就不赘述了。

最后希望大家喜欢我的这个项目。谢谢。

标签：[datetime](#)、[万年历](#)、[时间轴编辑](#)