如果你从肯定开始，必将以问题结束，如果你从问题开始，必将以肯定结束。

所谓了解，就是知道对方心灵最深的地方的痛处，痛在哪里。

1900年至2100年公历、农历互转Js代码 (http://blog.jjonline.cn/userInterFace/173.html)

晶晶 (http://blog.jjonline.cn/author/1) 3年前 (2014-07-21) 28664次浏览

前端 (http://blog.jjonline.cn/sort/userInterFace)

调用代码示例Demo

阳历：2017年5月26日（双子座）

农历：2017年五月初一，丁酉年乙巳月癸丑日（鸡年）

上述显示的demo简单代码：

```
1.
$(function () {
    var lunar = calendar.solar2lunar();
    $('.solarlunar').html('<strong>调用代码示例Demo</strong><br />阳历：'+lunar.cYear + '年' +lunar.cMonth + '月' + lunar.cDay +'日（'+lunar.astro+'）<br />农历：'+lunar.lYear + '年' +lunar.lMonthCn+lunar.lDayCn+'，'+lunar.gzYear+'年'+lunar.gzMonth+'月'+lunar.gzDay+'日（'+lunar.Animal+'年）');
});
```

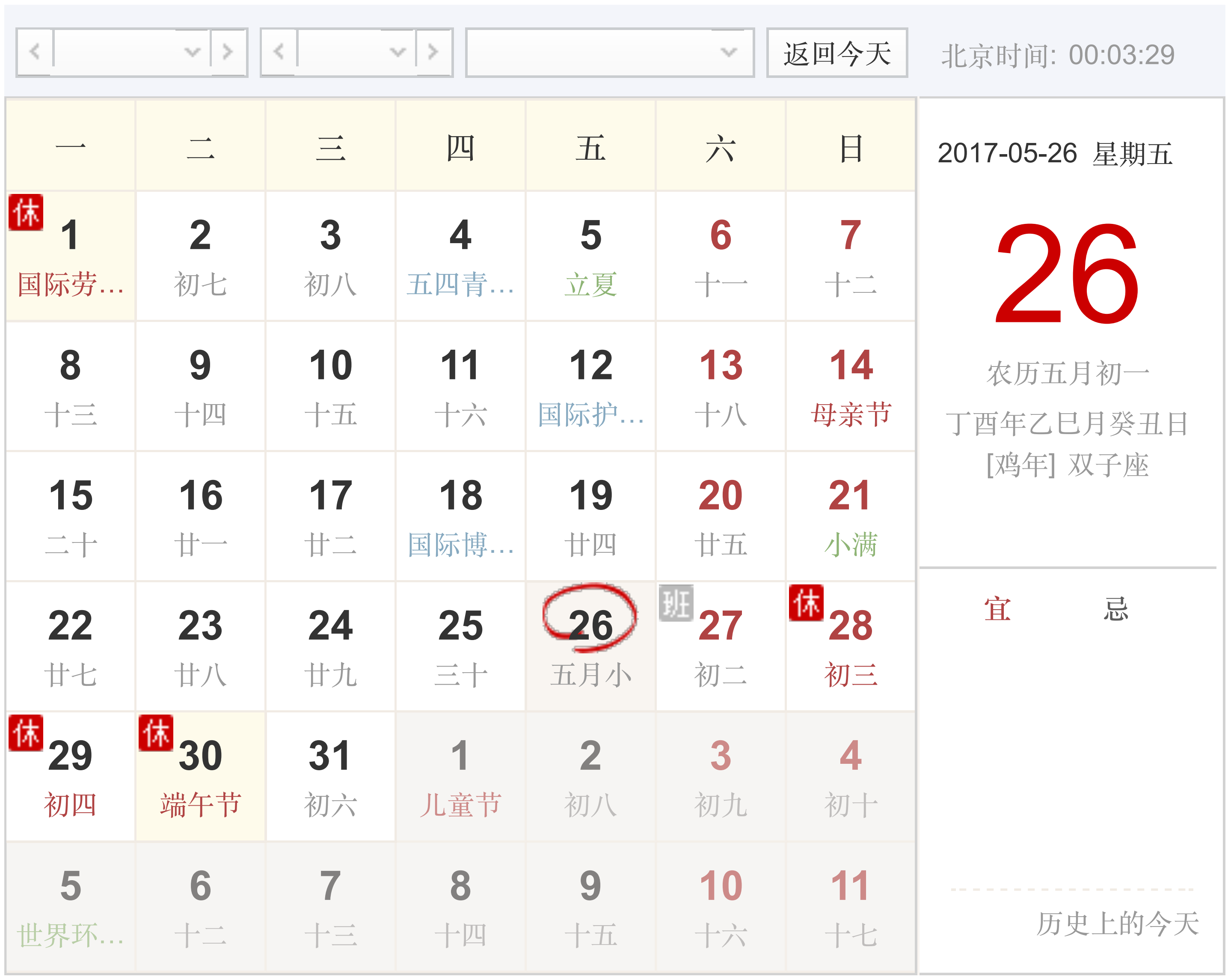
=======

2016-8-15更新：修正天干地支算法，修正农历润大小二进制表数据，添加星座字段astro

2016-9-25更新：修正农历闰月参数判断错误的bug

=======

最近在研究农历、公历的互转算法，墨迹了两三天发现我国农历（太阴历、月亮历）并没有世界通用的公历（太阳历）那么有规律性；我国农历的一些原数据并没有一个固定的算法可以通过程序产生，只能由天文台测定后提供，所以一般的所谓“万年历”都是采用“查表法”获取农历数据而产生的；这也就限定了“万年历”的区间范围。如：下方是360提供的一个万年历，这个万年历就只能查询1901年至2100年的公历、农历以及二十四节气等数据；更古老的或者说更遥远的时间就无法提供。


返回今天　北京时间: 00:03:29

一	二	三	四	五	六	日
休 1 国际劳…	2 初七	3 初八	4 五四青…	5 立夏	6 十一	7 十二
8 十三	9 十四	10 十五	11 十六	12 国际护…	13 十八	14 母亲节
15 二十	16 廿一	17 廿二	18 国际博…	19 廿四	20 廿五	21 小满
22 廿七	23 廿八	24 廿九	25 三十	26 五月小	班 27 初二	休 28 初三
休 29 初四	休 30 端午节	31 初六	1 儿童节	2 初八	3 初九	4 初十
5 世界环…	6 十二	7 十三	8 十四	9 十五	10 十六	11 十七

2017-05-26 星期五

26

农历五月初一

丁酉年乙巳月癸丑日

[鸡年] 双子座

宜　忌

历史上的今天

在网络中找了找相关的素材资料，要么老旧、要么有稍许错误，又把我搞烦了，遂自己写了一个可查询、互转1900年至2100年区间农历与公历的javascript库；本库所有农历数据来源于香港天文台（地址：http://data.weather.gov.hk/gts/time/conversion1_text_c.htm (http://data.weather.gov.hk/gts/time/conversion1_text_c.htm)）；本来想找紫金山天文台的数据的，结果硬是没找着有200年区间的农历数据的网页。

源数据经过抓取以及匹配获得Js里查表法所需要用的“表”，剩下的就是一些常规的js算法代码了，没什么技术含量；就不多啰嗦了。

js文件地址：http://www.jjonline.cn/Public/Js/calendar.js (http://www.jjonline.cn/Public/Js/calendar.js)

代码如下：

```
/**
* @1900-2100区间内的公历、农历互转
* @charset  UTF-8
* @Author  Jea杨(JJonline@JJonline.Cn)
* @Time    2014-7-21
* @Time    2016-8-13 Fixed 2033hex、Attribution Annals
* @Version 1.0.1
* @公历转农历：calendar.solar2lunar(1987,11,01); //[you can ignore params of prefix 0]
* @农历转公历：calendar.lunar2solar(1987,09,10); //[you can ignore params of prefix 0]
*/
var calendar = {

    /**
      * 农历1900-2100的润大小信息表
      * @Array Of Property
      * @return Hex
      */
    lunarInfo:[0x04bd8,0x04ae0,0x0a570,0x054d5,0x0d260,0x0d950,0x16554,0x056a0,0x09ad0,0x055d2,//1900-1909
            0x04ae0,0x0a5b6,0x0a4d0,0x0d250,0x1d255,0x0b540,0x0d6a0,0x0ada2,0x095b0,0x14977,//1910-1919
            0x04970,0x0a4b0,0x0b4b5,0x06a50,0x06d40,0x1ab54,0x02b60,0x09570,0x052f2,0x04970,//1920-1929
            0x06566,0x0d4a0,0x0ea50,0x06e95,0x05ad0,0x02b60,0x186e3,0x092e0,0x1c8d7,0x0c950,//1930-1939
            0x0d4a0,0x1d8a6,0x0b550,0x056a0,0x1a5b4,0x025d0,0x092d0,0x0d2b2,0x0a950,0x0b557,//1940-1949
            0x06ca0,0x0b550,0x15355,0x04da0,0x0a5b0,0x14573,0x052b0,0x0a9a8,0x0e950,0x06aa0,//1950-1959
            0x0aea6,0x0ab50,0x04b60,0x0aae4,0x0a570,0x05260,0x0f263,0x0d950,0x05b57,0x056a0,//1960-1969
            0x096d0,0x04dd5,0x04ad0,0x0a4d0,0x0d4d4,0x0d250,0x0d558,0x0b540,0x0b6a0,0x195a6,//1970-1979
```

```
        0x095b0,0x049b0,0x0a974,0x0a4b0,0x0b27a,0x06a50,0x06d40,0x0af46,0x0ab60,0x09570,//1980-1989
        0x04af5,0x04970,0x064b0,0x074a3,0x0ea50,0x06b58,0x055c0,0x0ab60,0x096d5,0x092e0,//1990-1999
        0x0c960,0x0d954,0x0d4a0,0x0da50,0x07552,0x056a0,0x0abb7,0x025d0,0x092d0,0x0cab5,//2000-2009
        0x0a950,0x0b4a0,0x0baa4,0x0ad50,0x055d9,0x04ba0,0x0a5b0,0x15176,0x052b0,0x0a930,//2010-2019
        0x07954,0x06aa0,0x0ad50,0x05b52,0x04b60,0x0a6e6,0x0a4e0,0x0d260,0x0ea65,0x0d530,//2020-2029
        0x05aa0,0x076a3,0x096d0,0x04afb,0x04ad0,0x0a4d0,0x1d0b6,0x0d250,0x0d520,0x0dd45,//2030-2039
        0x0b5a0,0x056d0,0x055b2,0x049b0,0x0a577,0x0a4b0,0x0aa50,0x1b255,0x06d20,0x0ada0,//2040-2049
        /**Add By JJonline@JJonline.Cn**/
        0x14b63,0x09370,0x049f8,0x04970,0x064b0,0x168a6,0x0ea50, 0x06b20,0x1a6c4,0x0aae0,//2050-2059
        0x0a2e0,0x0d2e3,0x0c960,0x0d557,0x0d4a0,0x0da50,0x05d55,0x056a0,0x0a6d0,0x055d4,//2060-2069
        0x052d0,0x0a9b8,0x0a950,0x0b4a0,0x0b6a6,0x0ad50,0x055a0,0x0aba4,0x0a5b0,0x052b0,//2070-2079
        0x0b273,0x06930,0x07337,0x06aa0,0x0ad50,0x14b55,0x04b60,0x0a570,0x054e4,0x0d160,//2080-2089
        0x0e968,0x0d520,0x0daa0,0x16aa6,0x056d0,0x04ae0,0x0a9d4,0x0a2d0,0x0d150,0x0f252,//2090-2099
        0x0d520],//2100


    /**
      * 公历每个月份的天数普通表
      * @Array Of Property
      * @return Number
      */
    solarMonth:[31,28,31,30,31,30,31,31,30,31,30,31],


    /**
      * 天干地支之天干速查表
      * @Array Of Property trans["甲","乙","丙","丁","戊","己","庚","辛","壬","癸"]
```

```
     * @return Cn string
     */
    Gan:["\u7532","\u4e59","\u4e19","\u4e01","\u620a","\u5df1","\u5e9a","\u8f9b","\u58ec","\u7678"],


    /**
     * 天干地支之地支速查表
     * @Array Of Property
     * @trans["子","丑","寅","卯","辰","巳","午","未","申","酉","戌","亥"]
     * @return Cn string
     */
    Zhi:["\u5b50","\u4e11","\u5bc5","\u536f","\u8fb0","\u5df3","\u5348","\u672a","\u7533","\u9149","\u620c","\u4ea5"],


    /**
     * 天干地支之地支速查表<=>生肖
     * @Array Of Property
     * @trans["鼠","牛","虎","兔","龙","蛇","马","羊","猴","鸡","狗","猪"]
     * @return Cn string
     */
    Animals:["\u9f20","\u725b","\u864e","\u5154","\u9f99","\u86c7","\u9a6c","\u7f8a","\u7334","\u9e21","\u72d7","\u732a"],


    /**
     * 24节气速查表
     * @Array Of Property
     * @trans["小寒","大寒","立春","雨水","惊蛰","春分","清明","谷雨","立夏","小满","芒种","夏至","小暑","大暑","立秋","处暑","白露","秋分","寒露","霜降","立冬","小雪","大雪","冬至"]
     * @return Cn string
     */
    solarTerm:["\u5c0f\u5bd2","\u5927\u5bd2","\u7acb\u6625","\u96e8\u6c34","\u60ca\u86f0","\u6625\u5206","\u6e05\u660e","\u8c37\u96e8","\u7acb\u590f","\u5c0f\u6ee1","\u8292\u79cd","\u590f\u81f3","\u5c0f\u6691","\u5927\u6691","\u7acb\u79cb","\u5904\u6691","\u767d\u9732","\u79cb\u5206","\u5bd2\u9732","\u971c\u964d","\u7acb\u51ac","\u5c0f\u96ea","\u5927\u96ea","\u51ac\u81f3"],


    /**
     * 1900-2100各年的24节气日期速查表
     * @Array Of Property
     * @return 0x string For splice
     */
    sTermInfo:[
'9778397bd097c36b0b6fc9274c91aa','97b6b97bd19801ec9210c965cc920e','97bcf97c3598082c95f8c965cc920f',
'97bd0b06bdb0722c965ce1cfcc920f','b027097bd097c36b0b6fc9274c91aa','97b6b97bd19801ec9210c965cc920e',
'97bcf97c359801ec95f8c965cc920f','97bd0b06bdb0722c965ce1cfcc920f','b027097bd097c36b0b6fc9274c91aa',
'97b6b97bd19801ec9210c965cc920e','97bcf97c359801ec95f8c965cc920f','97bd0b06bdb0722c965ce1cfcc920f',
'b027097bd097c36b0b6fc9274c91aa','9778397bd19801ec9210c965cc920e','97b6b97bd19801ec95f8c965cc920f',
'97bd09801d98082c95f8e1cfcc920f','97bd097bd097c36b0b6fc9210c8dc2','9778397bd197c36c9210c9274c91aa',
'97b6b97bd19801ec95f8c965cc920e','97bd09801d98082c95f8e1cfcc920f','97bd097bd097c36b0b6fc9210c8dc2',
'9778397bd097c36c9210c9274c91aa','97b6b97bd19801ec95f8c965cc920e','97bcf97c3598082c95f8e1cfcc920f',
'97bd097bd097c36b0b6fc9210c8dc2','9778397bd097c36c9210c9274c91aa','97b6b97bd19801ec9210c965cc920e',
'97bcf97c3598082c95f8c965cc920f','97bd097bd097c35b0b6fc920fb0722','9778397bd097c36b0b6fc9274c91aa',
'97b6b97bd19801ec9210c965cc920e','97bcf97c3598082c95f8c965cc920f','97bd097bd097c35b0b6fc920fb0722',
'9778397bd097c36b0b6fc9274c91aa','97b6b97bd19801ec9210c965cc920e','97bcf97c359801ec95f8c965cc920f',
'97bd097bd097c35b0b6fc920fb0722','9778397bd097c36b0b6fc9274c91aa','97b6b97bd19801ec9210c965cc920e',
'97bcf97c359801ec95f8c965cc920f','97bd097bd097c35b0b6fc920fb0722','9778397bd097c36b0b6fc9274c91aa',
'97b6b97bd19801ec9210c965cc920e','97bcf97c359801ec95f8c965cc920f','97bd097bd07f595b0b6fc920fb0722',
'9778397bd097c36b0b6fc9210c8dc2','9778397bd19801ec9210c9274c920e','97b6b97bd19801ec95f8c965cc920f',
'97bd07f5307f595b0b0bc920fb0722','7f0e397bd097c36b0b6fc9210c8dc2','9778397bd097c36c9210c9274c920e',
'97b6b97bd19801ec95f8c965cc920f','97bd07f5307f595b0b0bc920fb0722','7f0e397bd097c36b0b6fc9210c8dc2',
'9778397bd097c36c9210c9274c91aa','97b6b97bd19801ec9210c965cc920e','97bd07f1487f595b0b0bc920fb0722',
'7f0e397bd097c36b0b6fc9210c8dc2','9778397bd097c36b0b6fc9274c91aa','97b6b97bd19801ec9210c965cc920e',
'97bcf7f1487f595b0b0bb0b6fb0722','7f0e397bd097c35b0b6fc920fb0722','9778397bd097c36b0b6fc9274c91aa',
'97b6b97bd19801ec9210c965cc920e','97bcf7f1487f595b0b0bb0b6fb0722','7f0e397bd097c35b0b6fc920fb0722',
'9778397bd097c36b0b6fc9274c91aa','97b6b97bd19801ec9210c965cc920e','97bcf7f1487f531b0b0bb0b6fb0722',
'7f0e397bd097c35b0b6fc920fb0722','9778397bd097c36b0b6fc9274c91aa','97b6b97bd19801ec9210c965cc920e',
'97bcf7f1487f531b0b0bb0b6fb0722','7f0e397bd07f595b0b6fc920fb0722','9778397bd097c36b0b6fc9274c91aa',
'97b6b97bd19801ec9210c9274c920e','97bcf7f0e47f531b0b0bb0b6fb0722','7f0e397bd07f595b0b0bc920fb0722',
'9778397bd097c36b0b6fc9210c91aa','97b6b97bd197c36c9210c9274c920e','97bcf7f0e47f531b0b0bb0b6fb0722',
'7f0e397bd07f595b0b0bc920fb0722','9778397bd097c36b0b6fc9210c8dc2','9778397bd097c36c9210c9274c920e',
'97b6b7f0e47f531b0723b0b6fb0722','7f0e37f5307f595b0b0bc920fb0722','7f0e397bd097c36b0b6fc9210c8dc2',
'9778397bd097c36b0b70c9274c91aa','97b6b7f0e47f531b0723b0b6fb0721','7f0e37f1487f595b0b0bb0b6fb0722',
'7f0e397bd097c35b0b6fc9210c8dc2','9778397bd097c36b0b6fc9274c91aa','97b6b7f0e47f531b0723b0b6fb0721',
'7f0e27f1487f595b0b0bb0b6fb0722','7f0e397bd097c35b0b6fc920fb0722','9778397bd097c36b0b6fc9274c91aa',
'97b6b7f0e47f531b0723b0b6fb0721','7f0e27f1487f531b0b0bb0b6fb0722','7f0e397bd097c35b0b6fc920fb0722',
'9778397bd097c36b0b6fc9274c91aa','97b6b7f0e47f531b0723b0b6fb0721','7f0e27f1487f531b0b0bb0b6fb0722',
'7f0e397bd097c35b0b6fc920fb0722','9778397bd097c36b0b6fc9274c91aa','97b6b7f0e47f531b0723b0b6fb0721',
'7f0e27f1487f531b0b0bb0b6fb0722','7f0e397bd07f595b0b0bc920fb0722','9778397bd097
```

```
c36b0b6fc9274c91aa',
'97b6b7f0e47f531b0723b0787b0721','7f0e27f0e47f531b0b0bb0b6fb0722','7f0e397bd07f595b0b0bc920fb0722',
'9778397bd097c36b0b6fc9210c91aa','97b6b7f0e47f149b0723b0787b0721','7f0e27f0e47f531b0723b0b6fb0722',
'7f0e397bd07f595b0b0bc920fb0722','9778397bd097c36b0b6fc9210c8dc2','977837f0e37f149b0723b0787b0721',
'7f07e7f0e47f531b0723b0b6fb0722','7f0e37f5307f595b0b0bc920fb0722','7f0e397bd097c35b0b6fc9210c8dc2',
'977837f0e37f14998082b0787b0721','7f07e7f0e47f531b0723b0b6fb0721','7f0e37f1487f595b0b0bb0b6fb0722',
'7f0e397bd097c35b0b6fc9210c8dc2','977837f0e37f14998082b0787b06bd','7f07e7f0e47f531b0723b0b6fb0721',
'7f0e27f1487f531b0b0bb0b6fb0722','7f0e397bd097c35b0b6fc920fb0722','977837f0e37f14998082b0787b06bd',
'7f07e7f0e47f531b0723b0b6fb0721','7f0e27f1487f531b0b0bb0b6fb0722','7f0e397bd097c35b0b6fc920fb0722',
'977837f0e37f14998082b0787b06bd','7f07e7f0e47f531b0723b0b6fb0721','7f0e27f1487f531b0b0bb0b6fb0722',
'7f0e397bd07f595b0b0bc920fb0722','977837f0e37f14998082b0787b06bd','7f07e7f0e47f531b0723b0b6fb0721',
'7f0e27f1487f531b0b0bb0b6fb0722','7f0e397bd07f595b0b0bc920fb0722','977837f0e37f14998082b0787b06bd',
'7f07e7f0e47f149b0723b0787b0721','7f0e27f0e47f531b0b0bb0b6fb0722','7f0e397bd07f595b0b0bc920fb0722',
'977837f0e37f14998082b0723b06bd','7f07e7f0e37f149b0723b0787b0721','7f0e27f0e47f531b0723b0b6fb0722',
'7f0e397bd07f595b0b0bc920fb0722','977837f0e37f14898082b0723b02d5','7ec967f0e37f14998082b0787b0721',
'7f07e7f0e47f531b0723b0b6fb0722','7f0e37f1487f595b0b0bb0b6fb0722','7f0e37f0e37f14898082b0723b02d5',
'7ec967f0e37f14998082b0787b0721','7f07e7f0e47f531b0723b0b6fb0722','7f0e37f1487f531b0b0bb0b6fb0722',
'7f0e37f0e37f14898082b0723b02d5','7ec967f0e37f14998082b0787b06bd','7f07e7f0e47f531b0723b0b6fb0721',
'7f0e37f1487f531b0b0bb0b6fb0722','7f0e37f0e37f14898082b072297c35','7ec967f0e37f14998082b0787b06bd',
'7f07e7f0e47f531b0723b0b6fb0721','7f0e27f1487f531b0b0bb0b6fb0722','7f0e37f0e37f14898082b072297c35',
'7ec967f0e37f14998082b0787b06bd','7f07e7f0e47f531b0723b0b6fb0721','7f0e27f1487f531b0b0bb0b6fb0722',
'7f0e37f0e366aa89801eb072297c35','7ec967f0e37f14998082b0787b06bd','7f07e7f0e47f149b0723b0787b0721',
'7f0e27f1487f531b0b0bb0b6fb0722','7f0e37f0e366aa89801eb072297c35','7ec967f0e37f14998082b0723b06bd',
'7f07e7f0e47f149b0723b0787b0721','7f0e27f0e47f531b0723b0b6fb0722','7f0e37f0e366aa89801eb072297c35',
'7ec967f0e37f14998082b0723b06bd','7f07e7f0e37f14998083b0787b0721','7f0e27f0e47f531b0723b0b6fb0722',
'7f0e37f0e366aa89801eb072297c35','7ec967f0e37f14898082b0723b02d5','7f07e7f0e37f14998082b0787b0721',
'7f07e7f0e47f531b0723b0b6fb0722','7f0e36665b66aa89801e9808297c35','665f67f0e37f14898082b0723b02d5',
'7ec967f0e37f14998082b0787b0721','7f07e7f0e47f531b0723b0b6fb0722','7f0e36665b66a449801e9808297c35',
'665f67f0e37f14898082b0723b02d5','7ec967f0e37f14998082b0787b06bd','7f07e7f0e47f531b0723b0b6fb0721',
'7f0e36665b66a449801e9808297c35','665f67f0e37f14898082b072297c35','7ec967f0e37f14998082b0787b06bd',
'7f07e7f0e47f531b0723b0b6fb0721','7f0e26665b66a449801e9808297c35','665f67f0e37f1489801eb072297c35',
'7ec967f0e37f14998082b0787b06bd','7f07e7f0e47f531b0723b0b6fb0721','7f0e27f1487f531b0b0bb0b6fb0722'],


    /**
      * 数字转中文速查表
      * @Array Of Property
      * @trans ['日','一','二','三','四','五','六','七','八','九','十']
      * @return Cn string
      */
    nStr1:["\u65e5","\u4e00","\u4e8c","\u4e09","\u56db","\u4e94","\u516d","\u4e03","\u516b","\u4e5d","\u5341"],


    /**
      * 日期转农历称呼速查表
      * @Array Of Property
      * @trans ['初','十','廿','卅']
      * @return Cn string
      */
    nStr2:["\u521d","\u5341","\u5eff","\u5345"],


    /**
      * 月份转农历称呼速查表
      * @Array Of Property
      * @trans ['正','一','二','三','四','五','六','七','八','九','十','冬','腊']
      * @return Cn string
      */
    nStr3:["\u6b63","\u4e8c","\u4e09","\u56db","\u4e94","\u516d","\u4e03","\u516b","\u4e5d","\u5341","\u51ac","\u814a"],


    /**
      * 返回农历y年一整年的总天数
      * @param lunar Year
      * @return Number
      * @eg:var count = calendar.lYearDays(1987) ;//count=387
      */
    lYearDays:function(y) {
        var i, sum = 348;
        for(i=0x8000; i>0x8; i>>=1) { sum += (calendar.lunarInfo[y-1900] & i)? 1: 0; }
        return(sum+calendar.leapDays(y));
    },


    /**
      * 返回农历y年闰月是哪个月；若y年没有闰月 则返回0
      * @param lunar Year
      * @return Number (0-12)
      * @eg:var leapMonth = calendar.leapMonth(1987) ;//leapMonth=6
      */
    leapMonth:function(y) { //闰字编码 \u95f0
        return(calendar.lunarInfo[y-1900] & 0xf);
    },


    /**
      * 返回农历y年闰月的天数 若该年没有闰月则返回0
      * @param lunar Year
      * @return Number (0、29、30)
      * @eg:var leapMonthDay = calendar.leapDays(1987) ;//leapMonthDay=29
      */
    leapDays:function(y) {
        if(calendar.leapMonth(y))  {
            return((calendar.lunarInfo[y-1900] & 0x10000)? 30: 29);
        }
        return(0);
    },


    /**
      * 返回农历y年m月（非闰月）的总天数，计算m为闰月时的天数请使用leapDays方法
      * @param lunar Year
      * @return Number (-1、29、30)
      * @eg:var MonthDay = calendar.monthDays(1987,9) ;//MonthDay=29
      */
    monthDays:function(y,m) {
        if(m>12 || m<1) {return -1}//月份参数从1至12，参数错误返回-1
        return( (calendar.lunarInfo[y-1900] & (0x10000>>m))? 30: 29 );
    },


    /**
      * 返回公历(!)y年m月的天数
      * @param solar Year
      * @return Number (-1、28、29、30、31)
      * @eg:var solarMonthDay = calendar.leapDays(1987) ;//solarMonthDay=30
      */
    solarDays:function(y,m) {
        if(m>12 || m<1) {return -1} //若参数错误 返回-1
        var ms = m-1;
        if(ms==1) { //2月份的闰平规律测算后确认返回28或29
            return(((y%4 == 0) && (y%100 != 0) || (y%400 == 0))? 29: 28);
        }else {
            return(calendar.solarMonth[ms]);
        }
    },

    /**
     * 农历年份转换为干支纪年
     * @param  lYear 农历年的年份数
     * @return Cn string
     */
    toGanZhiYear:function(lYear) {
        var ganKey = (lYear - 3) % 10;
        var zhiKey = (lYear - 3) % 12;
        if(ganKey == 0) ganKey = 10;//如果余数为0则为最后一个天干
        if(zhiKey == 0) zhiKey = 12;//如果余数为0则为最后一个地支
        return calendar.Gan[ganKey-1] + calendar.Zhi[zhiKey-1];

    },

    /**
     * 公历月、日判断所属星座
     * @param  cMonth [description]
     * @param  cDay [description]
     * @return Cn string
     */
    toAstro:function(cMonth,cDay) {
        var s   = "\u9b54\u7faf\u6c34\u74f6\u53cc\u9c7c\u767d\u7f8a\u91d1\u725b\u53cc\u5b50\u5de8\u87f9\u72ee\u5b50\u5904\u5973\u5929\u79e4\u5929\u874e\u5c04\u624b\u9b54\u7faf";
        var arr = [20,19,21,21,21,22,23,23,23,23,22,22];
        return s.substr(cMonth*2 - (cDay < arr[cMonth-1] ? 2 : 0),2) + "\u5ea7";//座
    },


    /**
      * 传入offset偏移量返回干支
      * @param offset 相对甲子的偏移量
      * @return Cn string
      */
    toGanZhi:function(offset) {
        return calendar.Gan[offset%10] + calendar.Zhi[offset%12];
    },


    /**
     * 传入公历(!)y年获得该年第n个节气的公历日期
     * @param y公历年(1900-2100)；n二十四节气中的第几个节气(1~24)；从n=1(小寒)算起
     * @return day Number
     * @eg:var _24 = calendar.getTerm(1987,3) ;//_24=4;意即1987年2月4日立春
     */
    getTerm:function(y,n) {
        if(y<1900 || y>2100) {return -1;}
        if(n<1 || n>24) {return -1;}
        var _table = calendar.sTermInfo[y-1900];
        var _info = [
            parseInt('0x'+_table.substr(0,5)).toString() ,
            parseInt('0x'+_table.substr(5,5)).toString(),
            parseInt('0x'+_table.substr(10,5)).toString(),
            parseInt('0x'+_table.substr(15,5)).toString(),
            parseInt('0x'+_table.substr(20,5)).toString(),
            parseInt('0x'+_table.substr(25,5)).toString()
        ];
        var _calday = [
            _info[0].substr(0,1),
            _info[0].substr(1,2),
            _info[0].substr(3,1),
            _info[0].substr(4,2),

            _info[1].substr(0,1),
            _info[1].substr(1,2),
            _info[1].substr(3,1),
            _info[1].substr(4,2),

            _info[2].substr(0,1),
            _info[2].substr(1,2),
            _info[2].substr(3,1),
            _info[2].substr(4,2),

            _info[3].substr(0,1),
            _info[3].substr(1,2),
            _info[3].substr(3,1),
            _info[3].substr(4,2),

            _info[4].substr(0,1),
            _info[4].substr(1,2),
            _info[4].substr(3,1),
            _info[4].substr(4,2),

            _info[5].substr(0,1),
            _info[5].substr(1,2),
            _info[5].substr(3,1),
            _info[5].substr(4,2),
        ];
        return parseInt(_calday[n-1]);
    },


    /**
      * 传入农历数字月份返回汉语通俗表示法
     * @param lunar month
     * @return Cn string
     * @eg:var cnMonth = calendar.toChinaMonth(12) ;//cnMonth='腊月'
     */
    toChinaMonth:function(m) { // 月 => \u6708
        if(m>12 || m<1) {return -1} //若参数错误 返回-1
        var s = calendar.nStr3[m-1];
        s+= "\u6708";//加上月字
        return s;
    },


    /**
     * 传入农历日期数字返回汉字表示法
     * @param lunar day
     * @return Cn string
     * @eg:var cnDay = calendar.toChinaDay(21) ;//cnMonth='廿一'
     */
    toChinaDay:function(d){ //日 => \u65e5
        var s;
        switch (d) {
            case 10:
            s = '\u521d\u5341'; break;
        case 20:
            s = '\u4e8c\u5341'; break;
            break;
        case 30:
            s = '\u4e09\u5341'; break;
            break;
        default :
            s = calendar.nStr2[Math.floor(d/10)];
            s += calendar.nStr1[d%10];
        }
        return(s);
    },


    /**
      * 年份转生肖[!仅能大致转换] => 精确划分生肖分界线是“立春”
      * @param y year
      * @return Cn string
      * @eg:var animal = calendar.getAnimal(1987) ;//animal='兔'
      */
    getAnimal: function(y) {
        return calendar.Animals[(y – 4) % 12]
    },


    /**
      * 传入公历年月日获得详细的公历、农历object信息 <=>JSON
      * @param y  solar year
      * @param m solar month
      * @param d  solar day
      * @return JSON object
      * @eg:console.log(calendar.solar2lunar(1987,11,01));
      */
    solar2lunar:function (y,m,d) { //参数区间1900.1.31~2100.12.31
        if(y<1900 || y>2100) {return –1;}//年份限定、上限
```

```
if(y==1900&&m==1&&d<31) {return -1;}//下限
if(!y) { //未传参  获得当天
    var objDate = new Date();
}else {
    var objDate = new Date(y,parseInt(m)-1,d)
}
var i, leap=0, temp=0;
//修正ymd参数
var y = objDate.getFullYear(),m = objDate.getMonth()+1,d = objDate.getDate();
var offset   = (Date.UTC(objDate.getFullYear(),objDate.getMonth(),objDate.getDate()) - Date.UTC(1900,0,31))/86400000;
for(i=1900; i<2101 && offset>0; i++) { temp=calendar.lYearDays(i); offset-=temp; }
if(offset<0) { offset+=temp; i--; }

//是否今天
var isTodayObj = new Date(),isToday=false;
if(isTodayObj.getFullYear()==y && isTodayObj.getMonth()+1==m && isTodayObj.getDate()==d) {
    isToday = true;
}
//星期几
var nWeek = objDate.getDay(),cWeek = calendar.nStr1[nWeek];
if(nWeek==0) {nWeek =7;}//数字表示周几顺应天朝周一开始的惯例
//农历年
var year = i;

var leap = calendar.leapMonth(i); //闰哪个月
var isLeap = false;

//效验闰月
for(i=1; i<13 && offset>0; i++) {
    //闰月
    if(leap>0 && i==(leap+1) && isLeap==false){
        --i;
        isLeap = true; temp = calendar.leapDays(year); //计算农历闰月天数
    }
    else{
        temp = calendar.monthDays(year, i);//计算农历普通月天数
    }
```

```
        //解除闰月
        if(isLeap==true && i==(leap+1)) { isLeap = false; }
        offset -= temp;
    }

    if(offset==0 && leap>0 && i==leap+1)
    if(isLeap){
        isLeap = false;
    }else{
        isLeap = true; --i;
    }
    if(offset<0){ offset += temp; --i; }
    //农历月
    var month   = i;
    //农历日
    var day     = offset + 1;

    //天干地支处理
    var sm      =   m-1;
    var gzY     =   calendar.toGanZhiYear(year);

    //月柱 1900年1月小寒以前为 丙子月(60进制12)
    var firstNode   = calendar.getTerm(year,(m*2-1));//返回当月「节」为几日开始
    var secondNode  = calendar.getTerm(year,(m*2));//返回当月「节」为几日开始

    //依据12节气修正干支月
    var gzM     =   calendar.toGanZhi((y-1900)*12+m+11);
    if(d>=firstNode) {
        gzM     =   calendar.toGanZhi((y-1900)*12+m+12);
    }

    //传入的日期的节气与否
    var isTerm = false;
    var Term    = null;
    if(firstNode==d) {
        isTerm  = true;
        Term    = calendar.solarTerm[m*2-2];
    }
    if(secondNode==d) {
            isTerm  = true;
            Term    = calendar.solarTerm[m*2-1];
        }
        //日柱 当月一日与 1900/1/1 相差天数
        var dayCyclical = Date.UTC(y,sm,1,0,0,0,0)/86400000+25567+10;
        var gzD = calendar.toGanZhi(dayCyclical+d-1);
        //该日期所属的星座
        var astro = calendar.toAstro(m,d);

        return {'lYear':year,'lMonth':month,'lDay':day,'Animal':calendar.getAnimal(year),'IMonthCn':(isLeap?"\u95f0":'')+calendar.toChinaMonth(month),'lDayCn':calendar.toChinaDay(day),'cYear':y,'cMonth':m,'cDay':d,'gzYear':gzY,'gzMonth':gzM,'gzDay':gzD,'isToday':isToday,'isLeap':isLeap,'nWeek':nWeek,'ncWeek':"\u661f\u671f"+cWeek,'isTerm':isTerm,'Term':Term,'astro':astro};
    },


    /**
     * 传入公历年月日以及传入的月份是否闰月获得详细的公历、农历object信息 <=>JSON
     * @param y  lunar year
     * @param m lunar month
     * @param d  lunar day
     * @param isLeapMonth  lunar month is leap or not.
     * @return JSON object
     * @eg:console.log(calendar.lunar2solar(1987,9,10));
     */
    lunar2solar:function(y,m,d,isLeapMonth) {   //参数区间1900.1.31~2100.12.1
        var isLeapMonth = !!isLeapMonth;
        var leapOffset  = 0;
        var leapMonth   = calendar.leapMonth(y);
        var leapDay     = calendar.leapDays(y);
        if(isLeapMonth&&(leapMonth!=m)) {return -1;}//传参要求计算该闰月公历 但该年得出的闰月与传参的月份并不同
        if(y==2100&&m==12&&d>1 || y==1900&&m==1&&d<31) {return -1;}//超出了最大极限值
        var day  = calendar.monthDays(y,m);
        var _day = day;
       //bugFix 2016-9-25
       //if month is leap, _day use leapDays method
```

```
        if(isLeapMonth) {
          _day = calendar.leapDays(y,m);
        }
        if(y < 1900 || y > 2100 || d > _day) {return -1;}//参数合法性效验

        //计算农历的时间差
        var offset = 0;
        for(var i=1900;i<y;i++) {
            offset+=calendar.lYearDays(i);
        }
        var leap = 0,isAdd= false;
        for(var i=1;i<m;i++) {
            leap = calendar.leapMonth(y);
            if(!isAdd) {//处理闰月
                if(leap<=i && leap>0) {
                    offset+=calendar.leapDays(y);isAdd = true;
                }
            }
            offset+=calendar.monthDays(y,i);
        }
        //转换闰月农历 需补充该年闰月的前一个月的时差
        if(isLeapMonth) {offset+=day;}
        //1900年农历正月一日的公历时间为1900年1月30日0时0分0秒(该时间也是本农历的最开始起始点)
        var stmap   =   Date.UTC(1900,1,30,0,0,0);
        var calObj  =   new Date((offset+d-31)*86400000+stmap);
        var cY      =   calObj.getUTCFullYear();
        var cM      =   calObj.getUTCMonth()+1;
        var cD      =   calObj.getUTCDate();

        return calendar.solar2lunar(cY,cM,cD);
    }
};
```

由于源数据较多，文件未压缩就达到了22kb，还凑合吧~

调用方法，详细本文开头的Demo示例已经很清楚了吧~还是cp下：

1.
```
/**公历年月日转农历数据 返回json**/
calendar.solar2lunar(1987,11,01);
/**农历年月日转公历年月日**/
calendar.lunar2solar(1987,9,10);
//调用以上方法后返回类似如下object（json）具体以上就不需要解释了吧！
//c开头的是公历各属性值 l开头的自然就是农历咯 gz开头的就是天干地支纪年的数据啦~
{
 Animal: "兔",
 IDayCn: "初十",
 IMonthCn: "九月",
 Term: null,
 astro: "天蝎座",
 cDay: 1,
 cMonth: 11,
 cYear: 1987,
 gzDay: "甲寅",
 gzMonth: "庚戌",
 gzYear: "丁卯",
 isLeap: false,
 isTerm: false,
 isToday: false,
 lDay: 10,
 lMonth: 9,
 lYear: 1987,
 nWeek: 7,
 ncWeek: "星期日"
}
//该代码还有其他可以调用的方法，请自己查看代码中的详细注释
```

标签： js (http://blog.jjonline.cn/tag/js) 农历 (http://blog.jjonline.cn/tag/%E5%86%9C%E5%8E%86) 公历 (http://blog.jjonline.cn/tag/%E5%85%AC%E5%8E%86) Js (http://blog.jjonline.cn/tag/Js)

上一篇 入职三年记---谨以此文纪念那即将逝去的青春 (http://blog.jjonline.cn/mine/174.html)

全国各省、市、县、镇、村的mysql数据库和JSON格式数据 (http://blog.jjonline.cn/phptech/172.html) 下一篇

相关推荐

- Jquery辅助图片裁剪插件jcrop简介，一款web图片裁剪插件 (http://blog.jjonline.cn/userInterFace/jquery_jcrop_img_corp.html) 2012-12-24
- jQuery Ajax 实例 全解析 转载来至博客园吊儿郎当 (http://blog.jjonline.cn/otherarticle/jquery_ajax.html) 2011-10-15
- JavaScript核心：Object属性和相关方法 (http://blog.jjonline.cn/userInterFace/224.html) 2017-04-11
- JavaScript核心：Function属性和相关方法 (http://blog.jjonline.cn/userInterFace/225.html) 2017-04-12
- JavaScript逻辑运算符或和且的返回值 (http://blog.jjonline.cn/userInterFace/226.html) 2017-04-21

网友评论 42

来盖楼吧~

Ctrl+Enter快速提交 提交评论

昵称	昵称（必填）
邮箱	邮箱（必填）
网址	网址（选填）

解决自己的一个小需求时用到了你的代码，感谢。项目地址：https://github.com/dusu/calendar4print #1

dusu 1周前 (05-16)

@dusu：👍👍👍

晶晶 (http://blog.jjonline.cn/) 6天前

写万年历时用了你的农历、干支、节气等算法。但该算法中月日干支以立春为界，年和生肖却不是，所以我简单修改了一下，统一以立春为界： #2

```
var lichunDay = calendar.getTerm(year,3);
if((m==1 && month==1) || (m==2 && month==1 && d<lichunDay)){
    year--;
}else if(m==2 && month==12 && d>=lichunDay){
    year++;
}
```
我的万年历地址是：https://wuxincai.com/fn/wnl/ 支持键盘和移动触屏操作。

无心菜 (https://wuxincai.com/) 2个月前 (03-16)

阳历：2017年1月5日（魔羯座）
农历：2016年腊月初八，丙申年庚子月壬辰日（猴年）
庚子月 应该是 辛丑月 #3

嘻嘻哈哈 5个月前 (01-05)

@嘻嘻哈哈：能否解决？？

家家 3个月前 (02-12)

月份干支返回值不对 #4

嘻嘻哈哈 5个月前 (01-05)

貌似😂月份干支显示的不正确 #5

嘻嘻哈哈 5个月前 (01-05)

收藏一年多了，没想到大神还在更新 #6

小明 5个月前 (01-03)

我是初学者，能否发个调用的html文件给我? #7

莫建广 6个月前 (12-07)

😃 #8

莫建广 6个月前 (12-07)

節氣生肖 #9

```
getAnimal2: function(y,m,d) {
      if(m == 1) {
        return calendar.Animals[(y - 5) % 12]
      } else if(m == 2) {
        var springDay = calendar.getTerm(y,3);
        if(d < springDay) {
          return calendar.Animals[(y - 5) % 12]
        } else {
          return calendar.Animals[(y - 4) % 12]
        }
      } else {
        return calendar.Animals[(y - 4) % 12]
    }
},
```
tirear 6个月前 (11-21)

非常感谢，发现你有准备24节气，但Demo中24节气没看见，调用的话calendar.getTerm(1987,3)，只得到4,1是小寒，4不应该是雨水吗，为什么说是立春，能依靠传入（1999,1,1）得知它是在什么节气之后吗? #10

赤羽飞鸿 7个月前 (11-07)

@赤羽飞鸿：理论基础：24节气很有规律，因为是将地球绕太阳公转划分24等分确定的时间点，而阳历又是地球绕太阳公转一周来划分年份的，所以阳历中每个月有两个节气。
calendar.getTerm方法第一个参数是阳历年份，第二参数是该阳历年24个节气中第几个，这里的24节气个数以小寒为开始，也就是第二个参数中1表示小寒，24表示冬至，由前面的理论基础可以得知getTerm方法的第二个参数与阳历月份也是有关联关系的，换种说法一年中第1个、第2个节气必定在这一年的1月，而第3个、第4个节气必定在这一年的2月，以此类推。
举两个例子：
例子1：想获得2016年立冬节气是11月<为什么立冬节气必定在11月呢？前面的理论基础决定的>的哪一天，传参方法为calendar.getTerm(2016,11*2-1)，返回结果7，即表示2016年11月7日为立冬（今天就是立冬，第二个参数刻意写成11*2-1，当中的11即为11月，直接写成calendar.getTerm(2016,21)可以理解成：2016年第21个节气<从小寒为1算起，第21个节气为立冬>的日期是哪一天，因为阳历每个月有2个节气，所以第21个节气所在的月份很容易推算出来，也就是11月）；
例子2：那么2016年11月的第二个节气也就是小雪又是11月的哪一天呢？（同样含义的问法：2016年第22个节气在11月的哪一天？）写法：calendar.getTerm(2016,11*2)，返回结果22，表示2016年11月22日为小雪，以此类推。😃

晶晶 (http://blog.jjonline.cn/) 7个月前 (11-07)

@晶晶：再次表示感谢。

赤羽飞鸿 7个月前 (11-07)

1 2 (http://blog.jjonline.cn/userInterFace/173.html/comment-page-2#comments)
3 (http://blog.jjonline.cn/userInterFace/173.html/comment-page-3#comments)