# $.fn.Calendar

## 常量

|  |  |  |
| --- | --- | --- |
| **名称** | **类型** | **说明** |
| WorkDayType | Object | 工作日类型  HOLIDAY: 休息  WORKDAY: 上班  OFFWORK: 请假  TAKEOFF: 调休  OVERTIME: 加班 |
| CalendarType | Object | 历法类型  SOLAR: 阳历  LUNAR: 农历 |

## 属性

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| calendar | CalendarModel | 当前月份日期数据模型对象 |
| preCalendar | CalendarModel | 上个月份日期数据模型对象 |
| nextCalendar | CalendarModel | 下个月份日期数据模型对象 |
| view | View | 日历面板 |
| setting | Object | 万年历设置 |

## 方法

|  |  |  |
| --- | --- | --- |
| 名称 | 参数 | 说明 |
| init(obj, cSetting) | obj: 需要加载万年历的html元素jquery对象  cSetting: 万年历的配置 | 返回Calendar对象 |
| drawCld(SY, SM) | SY: 公历年份4位数值  SM: 公历月份[0-11] | 在页面中渲染SY,SM的日历数据 |
| setWorkDay($day, dayType) | $day: 日期单元格td的jquery对象或公历年月日字符串(yyyyMMdd)  dayType: WorkDayType枚举类型，需要设置的工作日类型（上班，休息，请假，调休，加班...） | 设置工作日的标记 |
|  |  |  |

# Calendar

## 常量

|  |  |  |
| --- | --- | --- |
| **名称** | **类型** | **说明** |
| WorkDayType | Object | 工作日类型  HOLIDAY: 休息  WORKDAY: 上班  OFFWORK: 请假  TAKEOFF: 调休  OVERTIME: 加班 |
| CalendarType | Object | 历法类型  SOLAR: 阳历  LUNAR: 农历 |

## 属性

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| setting | Object | 万年历设置 |
| workDays | Array | 服务器设置的工作日数据  [{ymd:'yyyyMMdd',isWorkDay:'1是上班日0是休息日'}] |

## 方法

|  |  |  |
| --- | --- | --- |
| 名称 | 参数 | 说明 |
| drawCalendar(sY, sM) | sY: 公历4位年份  sM: 公历月份[0-11] | 渲染加载指定年，月的日历数据到视图 |
| goYear(pY) | pY: 公历4位年份 | 跳转到指定年份的日历视图，月份默认当前视图中的月份 |
| goMonth(pM) | pM: 公历月份[1-12] | 跳转到指定月份的日历视图，年份默认当前视图中的月份 |
| goToday() |  | 切换到当前年月日历视图（公历） |
| setWorkDay($day, dayType) | $day: 需要设置的日期单元格jquery对象  dayType: WorkDayType枚举 | 设置工作日（上班，休息，请假，调休，加班...） |
| getGridSDay(ymd|sy ,sm,sd) | ymd: 8位或10位年月日字符串（20181204,2018-12-04,2018/12/04）  或  sy: 4位年份  sm: 月份[1-12]  sd: 日 | 获取具体公历日期的单元格jquery对象（当前日历视图范围内，否则返回null） |
| getGridLDay(ly,lm,ld) | ly: 4位年份  lm: 月份[1-12]  ld: 日 | 获取具体农历日期的单元格jquery对象（当前日历视图范围内，否则返回null） |
| getDayBySDay(sm,sd) | sm: 月份[1-12]  sd: 日 | 返回DayElement对象，参数sm，sd均是在当前视图内，否则返回null |

# cSetting配置

setting = {

calendarId: ""

,url: null

,isSuccessive: false

, festival: {

lunarFestival = [] // 农历节日

,solarFestival = [] // 公历节日

,weekFestival = [] // 周节日

,isMerge

}

,callback: {

onClick: function

}

}

calendarId: 实例化日历控件的元素id

url: 请求服务器工作日的地址，返回json数据格式{data: [{ymd:'yyyyMMdd',isWorkDay:'1是上班日0是休息日'}]}

isSuccessive: 是否连续（默认false），表示是否显示上个月的月末日期和下一个月的月初日期

festival:

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 说明 | 示例 |
| lunarFestival | Array | 农历节日 | ["20180101\*元旦",...]  20180101：表示节日的日期  \*：这一天放假，空格表示不放假  元旦：节日的名称，不可为空 |
| solarFestival | Array | 公历节日 | 同上 |
| weekFestival | Array | 周节日 | ["0520\*母亲节",...]  0520：表示5月的第二个周日  前两位：表示月份  第三位：  1,2,3,4 表示第1,2,3,4 个星期几  5,6,7,8 表示到数第1,2,3,4 个星期几  第四位：表示周几[0-6表示周日-周六]  \*：表示放假，空格表示不放假  母亲节：表示节日名称 |
| isMerge | boolean | 是否与控件自带节日数据合并，默认false | true: 表示设定的festival将与控件自带的节日数据合并，同一日期将覆盖控件  false: 表示festival完全覆盖控件自带的节日数据 |

# 事件

|  |  |  |
| --- | --- | --- |
| 名称 | 参数 | 说明 |
| onClick: function(obj,dayCld,cldModel) | obj: 点击的单元格jquery对象  dayCld: DayElement对象  cldModel: CalendarModel对象 | 日历单元格的点击事件 |
| dateChange:function(year,month) | year: 切换后的年份  month: 切换后的月份 | 日历视图发生切换时的事件 |

# 类型

## DayElement

具体某一天的日历属性

### 属性

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| sYear | number | 公历年 |
| sMonth | number | 公历月[1-12] |
| sDay | number | 公历日 |
| week | number | 星期 |
| lYear | number | 农历年 |
| lMonth | number | 农历月 |
| lDay | number | 农历日 |
| isLeap | boolean | 是否闰 |
| cYear | string | 年柱 |
| cMonth | string | 月柱 |
| cDay | string | 日柱 |
| solarFestival | string | 公历节日名称 |
| lunarFestival | string | 农历节日名称 |
| solarTerms | string | 节气名称 |

### 方法

|  |  |  |
| --- | --- | --- |
| 名称 | 参数 | 说明 |
| setWorkDay(dayType) | WorkDayType枚举 | 设置上班，休息，请假，调休，加班...等标记 |
| getGridDay() |  | 返回当前DayElement对应日期的单元格jquery对象 |

## CalendarModel

返回具体某一月份的日历数据模型（公历，农历，节假日，节气，年柱，月柱，日柱信息）

构造CalendarModel(year, month)

说明：year:年份，4位数字，month:月份数字[0-11]

属性：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| length | number | 当月总天数 |
| firstWeek | number | 当月1日是周几[0-6]表示周日到周六 |
| DayElement对象 | 数组 | 当月所有日期的DayElement对象数组 |

# 使用示例

<script type=*"text/javascript"*>

**var** calendar;

$(**function**(){

**var** \_setting = {

calendarId: ""

,isSuccessive: **true**

,festival:{

lunarFestival: [] // 农历节日

,solarFestival: [] // 公历节日

,weekFestival: [] // 周节日

}

,callback: {

onClick: dayOnclick

}

};

calendar = $.fn.Calendar.init($("#cldPanle"), \_setting);

//calendar.drawCalendar(2017, 9);

//calendar.goYear(2016);

calendar.goMonth(10);

});

**function** dayOnclick(obj,dayCld,cldModel){

console.log(obj);

console.log(dayCld);

console.log(cldModel);

//calendar.setWorkDay(obj, calendar.WorkDayType.WORKDAY);

/\*

calendar.setWorkDay('20181220',calendar.WorkDayType.WORKDAY);

var gd = calendar.getGridSDay(2018,12,17);

console.log(gd);

\*/

}

</script>

<table id=*"cldPanle"* class=*"table table-striped biao"*>

</table>