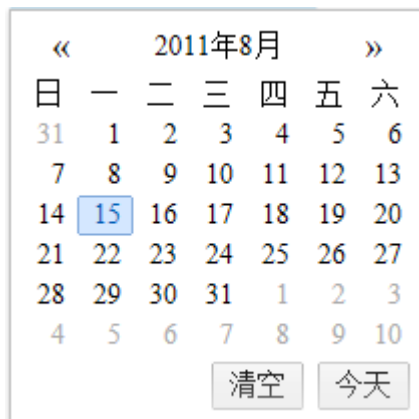


时间选择控件



datepicker 是使用原生 javascript 写的一个时间选择控件，主体思想来源于 win7 日历。

一、iDate

首先我们来说 iDate.parse 和 iDate.format，这两个属于内部私有静态方法，格式化的时间解析为 Date、Date 的格式化都靠它们了，注意：外部不能使用。

原型

parse

`<Date> iDate.parse(<string> format, <string> value)`

```
iDate.parse('yyyy-MM-dd HH:mm:ss', '2011-08-19 18:00:00');  
// 结果 Date {Fri Aug 19 2011 18:00:00 GMT+0800}
```

format

`<string> iDate.format(<string> format, <Date> date)`

```
iDate.format('yyyy-MM-dd HH:mm:ss', new Date());  
// 结果 2011-08-19 18:00:00
```

格式化可用占位符

占位符	说明
Y	4 位数字完整表示的年份 例如：1999 或 2003
yyyy	4 位数字完整表示的年份 例如：1999 或 2003
yy	2 位数字表示的年份 例如：99 或 03

M	数字表示的月份，没有前导零 1 到 12
MM	数字表示的月份，有前导零 01 到 12
MMM	三个字母缩写表示的月份 Jan 到 Dec
MMMM	月份，完整的文本格式，例如 January 或者 March January 到 December
d	月份中的第几天，没有前导零 1 到 31
dd	月份中的第几天，有前导零的 2 位数字 01 到 31
tt	上午和下午值 am 或 pm
H	小时，24 小时格式，没有前导零 0 到 23
HH	小时，24 小时格式，有前导零 00 到 23
h	小时，12 小时格式，没有前导零 1 到 12
hh	小时，12 小时格式，有前导零 01 到 12
m	分钟，没有前导零 0 到 59
mm	分钟，有前导零 00 到 59
s	秒数，没有前导零 0 到 59
ss	秒数，有前导零 00 到 59
w	星期中的第几天，数字表示 0（表示星期天）到 6（表示星期六）
N	1（表示星期一）到 7（表示星期天）
D	星期中的第几天，文本表示，3 个字母 Mon 到 Sun
DD	星期几，完整的文本格式 Sunday 到 Saturday
O	与格林威治时间相差的小时数 例如：+0200

二、DatePanel

我们可以通过 `(new DatePanel(options)).create()`，创建一个日期面板。

原型

`new DatePanel(options)`

参数设置

OPTIONS:

- `minDate` 未实现
- `maxDate` 未实现
- `switchOnClick`
 - 默认 `true`，点击灰色（日期不属于当月）日期，是否立即切换到所属月
- `format`
 - 默认 `yyyy-MM-dd`，`iDate` 认识的格式化字符串，
 - 格式化字符串中包含 `H`, `h`, `m`, `s` 将会显示时间面板
- `value`
 - 默认 `null`，初始化时间，该值是能被解析的字符串值，满足以下条件将能被解析：
 - 格式为 `options.format`
 - 能被 `Date.parse` 解析
 - 格式符合以下 9 种格式：
 - `Y-M-d H:m:s` `Y/M/d H:m:s` `Y 年 M 月 d 日 H:m:s`

d MMM Y H:m:s Y-M-d Y/M/d
M/d/Y Y 年 M 月 d 日 d MMM Y

startDay

默认 0，一周的第一天 0:周日 到 6: 周六

place

默认 null，日期面板放置区<HTMLElement>，null 时追加到 document.body

buttons

默认 null，添加的按钮, 形如:

```
{  
  "确定":function(){  
    alert(this.format())  
  },  
  "今天":function(){  
    alert(this.formatNow())  
  }  
}
```

实例化对象公开方法

create

`<void> create()`

创建视图，new DatePanel(options)不会出现日期面板，只有当调用 create 函数时，才会渲染加入到文档中。

setMonth

`<void> setMonth(<string> date, <bool> quick)`

月视图，切换到月视图 date;

date 格式为 2011-7 或 2011-7-19

表示切换到 2011 年 8 月份，这里的月份为 zero based 即 0:一月 至 11:十二月；
如果有号数则将该天 focus，2011-7-19 表示切换到 8 月并选中 19 号。

quick 为 true 时表示不使用动画，快速切换，默认为 false

setTime

`<void> setTime(<string> value)`

value 格式同 options.value，此函数会触发面板切换

getDate

`<Date> getDate(<void>)`

获取当前时间面板所显示的日期、时间。

format

`<string> format(<string> format, <Date> date)`

格式化时间，参数均可选。

formatNow

`<string> formatNow(<string> format)`

格式化当前（本地计算机）时间，参数可选。

toString

`<string> toString(<void>)`

格式化当前时间面板所显示的时间。

bind

`<DatePanel> bind(<string>event, <function> fn)`

绑定事件，可用事件如下：

DECADE_CHANGED

参数：

start	int	年代开始年 如 1900
end	int	年代结束年 如 1909
year	int	获得焦点的年 如 1901

YEAR_CHANGED

参数：

year	int	切换到的新年
------	-----	--------

MONTH_CHANGED

参数：

year	int	切换到的新年
month	int	切换到的新月，zero based

VIEW_CHANGED

参数：

mode	int	视图模式 1: 月视图 2: 年视图 3:年代视图
args	array	参数

DATE_FOCUSED

参数：

year	int
month	int
date	int

ITEM_CLICKED

参数:

value string
element DOMElement

DATE_CLICKED

参数:

value string
element DOMElement

YEAR_PREPARED

参数:

element DOMElement
year int

MONTH_PREPARED

参数:

element DOMElement
year int
month int

三、DatePicker

DatePicker 是建立在 DatePanel 基础上的二层封装。设计目的通过一个按钮快速选择一个日期。比如按钮点击时 `onclick="DatePicker(this, {})"`

DatePicker 具有二线性，函数与类

- 作为函数调用时如 `DatePicker(this, {})`，这种机制适合作为 `onclick`，`onfocus` 属性值，可以避免重复创建实例对象，节省内存。
- 作为类调用时如 `new DatePicker(elem, {})`，这样适合二次开发。

原型

`DatePicker(button, options)`
`new DatePicker(button, options)`

具体参数设置

button: DOMElement

作为按钮，或者说触发显示时间面板的元素，推荐用 `input`，`img`，`span`，`em`，`a`，`b`，`i`，`q`，`div`

OPTIONS:

startDay 同 DatePanel

format 默认 `yyyy-MM-dd`，同 DatePanel

`hideOnClick` 默认 true，点击选中完成即隐藏面板

`value` function/DOMElement

- 传 null 值，默认设定为与 button 同值
- 传 function 应该具有集 set、get 一体的逻辑，如：

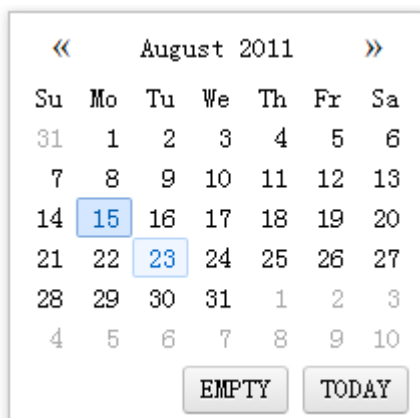
```
function(v){
    if (v === undefined) {
        return some;
    } else {
        some = v
    }
}
```
- 传 DOMElement，元素推荐 input、textarea、span、div、a，通过存取 value/innerHTML 属性，读取储存时间值

四、Locale

本地语言化的支持，默认使用的简体中文，比如

支持 E 文：

```
DatePicker.setLocale({
    WEEKNAME:['Su','Mo','Tu','We','Th','Fr','Sa'],
    MONTHNAME:[
        'Jan','Feb','Mar','Apr','May','Jun',
        'Jul','Aug','Sep','Oct','Nov','Dec'],
    FORMAT:'MMMM yyyy',
    TIME:'TIME',
    OK:'OK',
    TODAY:'TODAY',
    NOW:'NOW',
    EMPTY:'EMPTY'
});
```



支持繁体中文：

```
DatePicker.setLocale({
    WEEKNAME:['日','一','二','三','四','五','六'],
    MONTHNAME:[
        '一月','二月','三月','四月','五月','六月',
        '七月','八月','九月','十月','十一月','十二月'],
    FORMAT:'yyyy年M月',
    TIME:'時間',
    OK:'確定',
    TODAY:'今天',
    NOW:'現在',
    EMPTY:'清空'
});
```



五、jQuery.fn.DatePicker

如果你喜欢 jQuery 式调用，请在加载 datepicker.js 之前加载 jQuery，使用形如：

```
$('#button').DatePicker({
    format:'yyyy-m-d'
});
```