**一. 简介**

**1. 简介**

目前的版本是:4.7

**2. 注意事项**

* My97DatePicker目录是一个整体,不可破坏里面的目录结构,也不可对里面的文件改名,可以改目录名
* My97DatePicker是必须文件,不可删除
* 各目录及文件的用途:  
  WdatePicker.js配置文件,在调用的地方仅需使用该文件,可多个共存,以xx\_WdatePicker.js方式命名  
  config.js语言和皮肤配置文件,无需引入  
  calendar.js日期库主文件,无需引入  
  My97DatePicker.htm临时页面文件,不可删除  
  目录lang存放语言文件,你可以根据需要清理或添加语言文件  
  目录skin存放皮肤的相关文件,你可以根据需要清理或添加皮肤文件包
* 当WdatePicker.js里的属性:$wdate=true时,在input里加上class="Wdate"就会在选择框右边出现日期图标,如果您不喜欢这个样式,可以把class="Wdate"去掉,另外也可以通过修改skin目录下的WdatePicker.css文件来修改样式

**3. 支持的浏览器**

IE 6.0+ , Firefox 2.0+ , Chrome, Opera 9.5+ , Safari 3.0+

注意:IE 8.0是完美支持的,如果你在IE8上使用遇到问题,请与我取得联系,务必附上能再现你的问题的**纯HTML**代码包

© 2010 [My97](mailto:smallcarrot@163.com) All Rights Reserved.

**二. 功能及示例**

**1. 常规功能**

1. **支持多种调用模式**

除了支持常规在input单击或获得焦点调用外,还支持使用其他的元素如:<img><div>等触发WdatePicker函数来调用弹出日期框

**示例1-1-1 常规调用**

<input id="d11" type="text" onClick="WdatePicker()"/>

**示例1-1-2 图标触发**

<input id="d12" type="text"/>  
<img onclick="WdatePicker({el:'d12'})" src="../skin/datePicker.gif" width="16" height="22" align="absmiddle">  
注意:只需要传入控件的id即可



1. **下拉,输入,导航选择日期**

年月时分秒输入框都具备以下三种特性  
1. 通过导航图标选择  
  
  
2. 直接使用键盘输入数字  
  
  
3. 直接从弹出的下拉框中选择  
  
  
另:年份输入框有智能提示功能,当用户连续点击同一个导航按钮5次时,会自动弹出年份下拉框



1. **支持周显示**

可以通过配置isShowWeek属性决定是否限制周,并且在返回日期的时候还可以通过自带的自定义事件和API函数返回选择的周

**示例1-2-1 周显示简单应用**

<input id="d121" type="text" onfocus="WdatePicker({isShowWeek:true})"/>  
  
注意:周算法参考的是ISO8601定义的方法,如果您对此有疑问,请详见:<http://en.wikipedia.org/wiki/ISO_week_date>  
  
周算法选择(4.7新增)  
相关属性:whichDayIsfirstWeek  
周算法不同的地方有一些差异  
常见算法有三种  
1. ISO8601:规定第一个星期四为第一周,默认值: 4  
2. MSExcel:1月1日所在的周: 可以填写: 7  
3. 自己根据需要自定义,每年的第一个星期X作为第一周,可以填写: X (X可以是1-7之间任何一个数字)

**示例1-2-2 利用onpicked事件把周赋值给另外的文本框**

您选择了第 (W格式)周, 另外您可以使用WW格式: 周  
<input type="text" class="Wdate" id="d122" onFocus="WdatePicker({isShowWeek:true,onpicked:function() {$dp.$('d122\_1').value=$dp.cal.getP('W','W');$dp.$('d122\_2').value=$dp.cal.getP('W','WW');}})"/>  
  
onpicked 用法详见[自定义事件](file:///E:/Down/My97DatePicker日期控件4.7文档/2.5.htm#m251)  
$dp.cal.getP 用法详见[内置函数和属性](file:///E:/Down/My97DatePicker日期控件4.7文档/4.htm#m5)

1. **只读开关,高亮周末功能**

设置readOnly属性 true 或 false 可指定日期框是否只读  
设置highLineWeekDay属性 ture 或 false 可指定是否高亮周末

1. **操作按钮自定义**

清空按钮和今天按钮,可以根据需要进行自定义,它们分别对应 isShowClear 和 isShowToday 默认值都是true

**示例1-5 禁用清空功能**

最好把readOnly置为true,否则即使隐藏了清空按钮,用户依然可以在输入框里把值delete掉  
  
<input class="Wdate" type="text" id="d15" onFocus="WdatePicker({isShowClear:false,readOnly:true})"/>

1. **自动选择显示位置**

当控件处在页面边界时,它会自动选择显示的位置,所以没有必要担心弹出框会被页面边界遮住的问题了.

1. **自定义弹出位置**

当控件处在页面边界时,它会自动选择显示的位置.此外你还可以使用position参数对弹出位置做调整.

**示例1-6 通过position属性,自定义弹出位置**

使用positon属性指定,弹出日期的坐标为{left:100,top:50}  
  
<input class="Wdate" type="text" id="d16" onfocus="WdatePicker({position:{left:100,top:50}})"/>  
  
position属性的详细用法详见[属性表](file:///E:/Down/My97DatePicker日期控件4.7文档/3.htm#m31)

1. **自定义星期的第一天(4.6新增)**

各个国家的习惯不同,有些喜欢以星期日作为第一天,有些以星期一作为第一天.  
相关属性:firstDayOfWeek: 可设置 0 - 6 的任意一个数字,0:星期日 1:星期一 以此类推

**示例1-7 以星期一作为第一天**

<input class="Wdate" type="text" id="d17" onfocus="WdatePicker({firstDayOfWeek:1})"/>

**2. 特色功能**

1. **平面显示**

日期控件支持平面显示功能,只要设置一下eCont属性就可以把它当作日历来使用了,无需触发条件,直接显示在页面上

**示例2-1 平面显示演示**

<div id="div1"></div>  
<script>  
WdatePicker({eCont:'div1',onpicked:function(dp){alert('你选择的日期是:'+dp.cal.getDateStr())}})  
</script>  
  
$dp.cal.getDateStr 用法详见[内置函数和属性](file:///E:/Down/My97DatePicker日期控件4.7文档/4.htm#m5)

1. **支持多种容器**

除了可以将值返回给input以外,还可以通过配置el属性将值返回给其他的元素(如:textarea,div,span)等,带有innerHTML属性的HTML元素

**示例2-2 将日期返回到<span>中**

2008-01-01   
  
代码:  
<span id="demospan">2008-01-01</span>  
<img onClick="WdatePicker({el:'demospan'})" src="My97DatePicker/skin/datePicker.gif" width="16" height="22" align="absmiddle" style="cursor:pointer" />



**起始日期功能**  **注意:日期格式必须与 realDateFmt 和 realTimeFmt 一致而不是与 dateFmt 一致**有时在项目中需要选择生日之类的日期,而默认点开始日期都是当前日期,导致年份选择非常麻烦,你可以通过起始日期功能加上配置alwaysUseStartDate属性轻松解决此类问题

**示例2-3-1 起始日期简单应用**

默认的起始日期为 1980-05-01  
当日期框为空值时,将使用 1980-05-01 做为起始日期  
  
  
<input type="text" id="d221" onFocus="WdatePicker({startDate:'1980-05-01'})"/>

**示例2-3-2 alwaysUseStartDate属性应用**

默认的起始日期为 1980-05-01  
当日期框无论是何值,始终使用 1980-05-01 做为起始日期  
  
  
<input type="text" id="d222" onFocus="WdatePicker({startDate:'1980-05-01',alwaysUseStartDate:true})"/>

**示例2-3-3 使用内置参数**

除了使用静态的日期值以外,还可以使用动态参数(如:%y,%M分别表示当前年和月)  
  
下例演示,年月日使用当年当月的1日,时分秒使用00:00:00作为起始时间  
  
  
<input type="text" id="d233" onFocus="WdatePicker({startDate:'%y-%M-01 00:00:00',dateFmt:'yyyy-MM-dd HH:mm:ss',alwaysUseStartDate:true})"/>

1. **自定义格式**

yMdHmswW分别代表年月日时分秒星期周,你可以任意组合这些元素来自定义你个性化的日期格式.

1. **日期格式表**

|  |  |  |
| --- | --- | --- |
| **格式** | **说明** | |
| y | 将年份表示为最多两位数字。如果年份多于两位数，则结果中仅显示两位低位数。 | |
| yy | 同上，如果小于两位数，前面补零。 | |
| yyy | 将年份表示为三位数字。如果少于三位数，前面补零。 | |
| yyyy | 将年份表示为四位数字。如果少于四位数，前面补零。 | |
| M | 将月份表示为从 1 至 12 的数字 | |
| MM | 同上，如果小于两位数，前面补零。 | |
| MMM | 返回月份的缩写 一月 至 十二月 (英文状态下 Jan to Dec) 。 | |
| MMMM | 返回月份的全称 一月 至 十二月 (英文状态下 January to December) 。 | |
| d | 将月中日期表示为从 1 至 31 的数字。 | |
| dd | 同上，如果小于两位数，前面补零。 | |
| H | 将小时表示为从 0 至 23 的数字。 | |
| HH | 同上，如果小于两位数，前面补零。 | |
| m | 将分钟表示为从 0 至 59 的数字。 | |
| mm | 同上，如果小于两位数，前面补零。 | |
| s | 将秒表示为从 0 至 59 的数字。 | |
| ss | 同上，如果小于两位数，前面补零。 | |
| w | 返回星期对应的数字 0 (星期天) - 6 (星期六) 。 | |
| D | 返回星期的缩写 一 至 六 (英文状态下 Sun to Sat) 。 | |
| DD | 返回星期的全称 星期一 至 星期六 (英文状态下 Sunday to Saturday) 。 | |
| W | 返回周对应的数字 (1 - 53) 。 | |
| WW | 同上，如果小于两位数，前面补零 (01 - 53) 。 | |
| **示例 格式字符串** | | **值** |
| yyyy-MM-dd HH:mm:ss | | 2008-03-12 19:20:00 |
| yy年M月 | | 08年3月 |
| yyyyMMdd | | 20080312 |
| 今天是:yyyy年M年d HH时mm分 | | 今天是:2008年3月12日 19时20分 |
| H:m:s | | 19:20:0 |
| y年 | | 8年 |
| MMMM d, yyyy | | 三月 12, 2008 |

**示例 2-4-1: 年月日时分秒**

<input type="text" id="d241" onfocus="WdatePicker({dateFmt:'yyyy年MM月dd日 HH时mm分ss秒'})" class="Wdate" style="width:300px"/>  
  
注意:点两次才能选择日期的原因,详见 [autoPickDate](file:///E:/Down/My97DatePicker日期控件4.7文档/3.htm#autopickdate) 属性

**示例 2-4-2 时分秒**

<input type="text" id="d242" onfocus="WdatePicker({skin:'whyGreen',dateFmt:'H:mm:ss'})" class="Wdate"/>  
  
注意:这里提前使用了皮肤(skin)属性,所以你会看到一个不同的皮肤,皮肤属性详见[自定义和动态切换皮肤](file:///E:/Down/My97DatePicker日期控件4.7文档/2.3.htm#m232)

**示例 2-4-3 年月**

<input type="text" id="d243" onfocus="WdatePicker({skin:'whyGreen',dateFmt:'yyyy年MM月'})" class="Wdate"/>

**示例 2-4-4 取得系统可识别的日期值(重要)**

类似于 1999年7月5日 这样的日期是不能够被系统识别的,他必须转换为能够识别的类型如 1999-07-05  
  
 真实的日期值是:   
<input id="d244" type="text" class="Wdate" onfocus="WdatePicker({dateFmt:'yyyy年M月d日',vel:'d244\_2'})"/>  
<input id="d244\_2" type="text" />  
  
注意:在实际应用中,一般会把vel指定为一个hidden控件,这里是为了把真实值展示出来,所以使用文本框  
关键属性: vel 指定一个控件或控件的ID,必须具有value属性(如input),用于存储真实值(也就是realDateFmt和realTimeFmt格式化后的值)

**示例 2-4-5 星期, 月 日, 年(4.6新增)**

<input type="text" id="d245" onfocus="WdatePicker({dateFmt:'DD, MMMM d, yyyy'})" class="Wdate"/>

1. **双月日历功能(4.6新增)**

可以同时弹出两个月的日历

**示例2-5 双月日历功能**

<input class="Wdate" type="text" onfocus="WdatePicker({doubleCalendar:true,dateFmt:'yyyy-MM-dd'})"/>  
  
注意:双月日历一般只用于包含年月日三个元素的场景,另外设置该属性时,autoPickDate自动设置为true

1. **自动纠错功能**

纠错处理可设置为3种模式:提示(默认) 自动纠错 标记,当日期框中的值不符合格式时,系统会尝试自动修复,如果修复失败会根据您设置的纠错处理模式进行处理,错误判断功能非常智能它可以保证用户输入的值是一个合法的值

**示例2-6-1 不合法的日期演示**

请在下面的日期框中填入一个不合法的日期(如:1997-02-29),再尝试离开焦点  
使用默认容错模式 提示模式 errDealMode = 0 在输入错误日期时,会先提示  
  
  
注意:1997年不是闰年哦

**示例2-6-2 超出日期限制范围的日期也被认为是一个不合法的日期**

最大日期是2000-01-10 ,如果在下框中填入的日期 大于 2000-01-10(如2000-01-12)也会被认为是不合法的日期  
自动纠错模式 errDealMode = 1 在输入错误日期时,自动恢复前一次正确的值

**示例2-6-3 使用无效天和无效日期功能限制的日期也被认为是一个不合法的日期**

如:  
2008-02-20 无效日期限制  
2008-02-02 2008-02-09 2008-02-16 2008-02-23 无效天限制  
都是无效日期  
您可以尝试在下框中输入这些日期,并离开焦点  
  
标记模式 errDealMode = 2 在输入错误日期时,不做提示和更改,只是做一个标记,但此时日期框不会马上隐藏  
  
  
注意:标记类:WdateFmtErr是在skin目录下WdatePicker.css中定义的

1. **跨无限级框架显示**

无论你把日期控件放在哪里,你都不需要担心会被外层的iframe所遮挡进而影响客户体验,因为My97日期控件是可以跨无限级框架显示的

**示例2-7 跨无限级框架演示**

可无限跨越框架iframe,无论怎么嵌套框架都不必担心了,即使有滚动条也不怕  
注意:Javascript是无法跨越frameset的,但是My97日期控件可无限跨越框架iframe,任何的frameset都可以使用iframe代替

1. **民国年日历和其他特殊日历**

当年份格式设置为yyy格式时,利用年份差量属性yearOffset(默认值1911民国元年),可实现民国年日历和其他特殊日历

**示例2-8 民国年演示**

<input type="text" id="d28" onClick="WdatePicker({dateFmt:'yyy/MM/dd'})"/>  
  
注意:年份格式设置成yyy时,真正的日期将会减去一个差量yearOffset(默认值为:1911),如果是民国年使用默认值即可无需另外配置,如果是其他的差量,可以通过参数的形式配置

1. **编辑功能**

当日期框里面有值时,修改完某个属性后,只要点击这个按钮就可以实现时间和日期的编辑

**示例2-9 日期和时间的编辑演示**

您可以尝试对下面框中的月份改为1,然后点击更新,你会发现日期由 2000-02-29 01:00:00 变为 2000-01-29 01:00:00

1. **为编程带来方便**

如果el的值是this,可省略,即所有的el:this都可以不写  
日期框设置为disabled时,禁止更改日期(不弹出选择框)  
如果没有定义onpicked事件,自动触发文本框的onchange事件  
如果没有定义oncleared事件,清空时,自动触发onchange事件

1. **其他属性**

设置readOnly属性,可指定日期框是否只读  
设置highLineWeekDay属性,可指定是否高亮周末  
设置isShowOthers属性,可指定是否显示其他月的日期  
加上class="Wdate"就会在选择框右边出现日期图标

**3. 多语言和自定义皮肤**

1. **多语言支持**

通过lang属性,可以为每个日期控件单独配置语言,当然也可以通过WdatePicker.js配置全局的语言  
语言列表和语言安装说明详见[语言配置](file:///E:/Down/My97DatePicker日期控件4.7文档/3.htm#m32)

**示例3-1 多语言示例**

繁体中文:   
<input id="d311" class="Wdate" type="text" onFocus="WdatePicker({lang:'zh-tw'})"/>

英文:   
<input id="d312" class="Wdate" type="text" onFocus="WdatePicker({lang:'en'})"/>

简体中文:   
<input id="d313" class="Wdate" type="text" onFocus="WdatePicker({lang:'zh-cn'})"/>  
  
注意:默认情况lang='auto',即根据浏览器的语言自动选择语言.  
  
技巧:很多项目中,可能有固定的语言选项,希望可以在日期选择的属性设置中强制某种语言,即:可以通过后台代码控制语言的选择,其实这个实现起来是很容易的,My97Datepicker是支持多个配置文件共存的,你可以配置类似cn\_WdatePicker.js,en\_WdatePicker.js(注意命名规则,必须以WdatePicker.js结尾)等等,在这些WdatePicker.js里面设置不同的默认语言,皮肤,格式等其他可以设置的参数,然后在页面里根据你的系统变量引入不同的xx\_WdatePicker.js即可

1. **自定义和动态切换皮肤** **[立刻转到皮肤中心](file:///E:/skin.htm)**

通过skin属性,可以为每个日期控件单独配置皮肤,当然也可以通过WdatePicker.js配置全局的皮肤  
皮肤列表和皮肤安装说明详见[皮肤配置](file:///E:/Down/My97DatePicker日期控件4.7文档/3.htm#m33)

**示例3-2 皮肤演示**

**默认皮肤default**: skin:'default'  
  
<input id="d321" class="Wdate" type="text" onfocus="WdatePicker()"/>  
  
注意:在WdatePicker里配置了skin='default',所以此处可省略,同理,如果你把WdatePicker里的skin配置成'whyGreen'那么在不指定皮肤的情况下都使用'whyGreen'皮肤了

**whyGreen皮肤**: skin:'whyGreen'   
<input id="d322" class="Wdate" type="text" onfocus="WdatePicker({skin:'whyGreen'})"/>  
  
注意:更多皮肤,请到[皮肤中心](file:///E:/skin.htm)下载

**4. 日期范围限制**

**静态限制** **注意:日期格式必须与 realDateFmt 和 realTimeFmt 一致而不是与 dateFmt 一致**你可以给通过配置minDate(最小日期),maxDate(最大日期)为静态日期值,来限定日期的范围

**示例4-1-1 限制日期的范围是 2006-09-10到2008-12-20**

<input id="d411" class="Wdate" type="text" onfocus="WdatePicker({skin:'whyGreen',minDate:'2006-09-10',maxDate:'2008-12-20'})"/>

**示例4-1-2 限制日期的范围是 2008-3-8 11:30:00 到 2008-3-10 20:59:30**

<input type="text" class="Wdate" id="d412" onfocus="WdatePicker({skin:'whyGreen',dateFmt:'yyyy-MM-dd HH:mm:ss',minDate:'2008-03-08 11:30:00',maxDate:'2008-03-10 20:59:30'})" value="2008-03-09 11:00:00"/>

**示例4-1-3 限制日期的范围是 2008年2月 到 2008年10月**

<input type="text" class="Wdate" id="d413" onfocus="WdatePicker({dateFmt:'yyyy年M月',minDate:'2008-2',maxDate:'2008-10'})"/>

**示例4-1-4 限制日期的范围是 8:00:00 到 11:30:00**

<input type="text" class="Wdate" id="d414" onfocus="WdatePicker({dateFmt:'H:mm:ss',minDate:'8:00:00',maxDate:'11:30:00'})"/>

**动态限制** **注意:日期格式必须与 realDateFmt 和 realTimeFmt 一致而不是与 dateFmt 一致**你可以通过系统给出的动态变量,如%y(当前年),%M(当前月)等来限度日期范围,你还可以通过{}进行表达式运算,如:{%d+1}:表示明天  
  
动态变量表

|  |  |
| --- | --- |
| **格式** | **说明** |
| %y | 当前年 |
| %M | 当前月 |
| %d | 当前日 |
| %ld | 本月最后一天 |
| %H | 当前时 |
| %m | 当前分 |
| %s | 当前秒 |
| {} | 运算表达式,如:{%d+1}:表示明天 |
| #F{} | {}之间是函数可写自定义JS代码 |

**示例4-2-1 只能选择今天以前的日期(包括今天)**

<input id="d421" class="Wdate" type="text" onfocus="WdatePicker({skin:'whyGreen',maxDate:'%y-%M-%d'})"/>

**示例4-2-2 使用了运算表达式 只能选择今天以后的日期(不包括今天)**

<input id="d422" class="Wdate" type="text" onfocus="WdatePicker({minDate:'%y-%M-{%d+1}'})"/>

**示例4-2-3 只能选择本月的日期1号至本月最后一天**

<input id="d423" class="Wdate" type="text" onfocus="WdatePicker({minDate:'%y-%M-01',maxDate:'%y-%M-%ld'})"/>

**示例4-2-4 只能选择今天7:00:00至明天21:00:00的日期**

<input id="d424" class="Wdate" type="text" onfocus="WdatePicker({dateFmt:'yyyy-M-d H:mm:ss',minDate:'%y-%M-%d 7:00:00',maxDate:'%y-%M-{%d+1} 21:00:00'})"/>

**示例4-2-5 使用了运算表达式 只能选择 20小时前 至 30小时后 的日期**

<input id="d425" class="Wdate" type="text" onClick="WdatePicker({dateFmt:'yyyy-MM-dd HH:mm',minDate:'%y-%M-%d {%H-20}:%m:%s',maxDate:'%y-%M-%d {%H+30}:%m:%s'})"/>

**脚本自定义限制** **注意:日期格式必须与 realDateFmt 和 realTimeFmt 一致而不是与 dateFmt 一致**系统提供了$dp.$D和$dp.$DV这两个API来辅助你进行日期运算,此外你还可以通过在 #F{} 中填入你自定义的脚本,做任何你想做的日期限制

**示例4-3-1 前面的日期不能大于后面的日期且两个日期都不能大于 2020-10-01**

合同有效期从 到   
<input id="d4311" class="Wdate" type="text" onFocus="WdatePicker({maxDate:'#F{$dp.$D(\'d4312\')||\'2020-10-01\'}'})"/>  
<input id="d4312" class="Wdate" type="text" onFocus="WdatePicker({minDate:'#F{$dp.$D(\'d4311\')}',maxDate:'2020-10-01'})"/>  
  
注意:  
两个日期的日期格式必须相同  
  
$dp.$ 相当于 document.getElementById 函数.  
那么为什么里面的 ' 使用 \' 呢? 那是因为 " 和 ' 都被外围的函数使用了,故使用转义符 \ ,否则会提示JS语法错误.  
所以您在其他地方使用时注意把 \' 改成 " 或者 ' 来使用.  
  
#F{$dp.$D(\'d4312\')||\'2020-10-01\'} 表示当 d4312 为空时, 采用 2020-10-01 的值作为最大值

**示例4-3-2 前面的日期+3天 不能大于 后面的日期**

日期从 到   
<input type="text" class="Wdate" id="d4321" onFocus="WdatePicker({maxDate:'#F{$dp.$D(\'d4322\',{d:-3});}'})"/>  
<input type="text" class="Wdate" id="d4322" onFocus="WdatePicker({minDate:'#F{$dp.$D(\'d4321\',{d:3});}'})"/>  
  
使用 $dp.$D 函数 可以将日期框中的值,加上定义的日期差量:  
两个参数: id={字符类型}需要处理的文本框的id值 , obj={对象类型}日期差量  
日期差量用法:  
属性y,M,d,H,m,s分别代表年月日时分秒  
如  
为空时,表示直接取值,不做差量(示例4-3-1中的参数就是空的)  
{M:5,d:7} 表示 五个月零7天  
{y:1,d:-3} 表示 1年少3天  
{d:1,H:1} 表示一天多1小时

**示例4-3-3 前面的日期+3月零2天 不能大于 后面的日期 且 前面日期都不能大于 2020-4-3减去3月零2天 后面日期 不能大于 2020-4-3**

住店日期从 到   
<input type="text" class="Wdate" id="d4331" onFocus="WdatePicker({maxDate:'#F{$dp.$D(\'d4332\',{M:-3,d:-2})||$dp.$DV(\'2020-4-3\',{M:-3,d:-2})}'})"/>  
<input type="text" class="Wdate" id="d4332" onFocus="WdatePicker({minDate:'#F{$dp.$D(\'d4331\',{M:3,d:2});}',maxDate:'2020-4-3'})"/>  
  
注意:#F{$dp.$D(\'d4332\',{M:-3,d:-2}) || $dp.$DV(\'2020-4-3\',{M:-3,d:-2})}  
表示当 d4332 为空时, 采用 $dp.$DV(\'2020-4-3\',{M:-3,d:-2})} 的值作为最大值

使用 $dp.$DV 函数 可以将显式传入的值,加上定义的日期差量:  
两个参数: value={字符类型}需要处理的值 , obj={对象类型}日期差量  
用法同上面的 $dp.$D 类似,如 $dp.$DV(\'2020-4-3\',{M:-3,d:-2}) 表示 2020-4-3减去3月零2天

**示例4-3-4 发挥你的JS才能,定义任何你想要的日期限制**

自动转到随机生成的一天,当然,此示例没有实际的用途,只是为演示目的  
   
<script>  
//返回一个随机的日期  
function randomDate(){  
var Y = 2000 + Math.round(Math.random() \* 10);  
var M = 1 + Math.round(Math.random() \* 11);  
var D = 1 + Math.round(Math.random() \* 27);  
return Y+'-'+M+'-'+D;  
}  
</script>  
<input type="text" class="Wdate" id="d434" onFocus="var date=randomDate();WdatePicker({minDate:date,maxDate:date})"/>

1. **无效天**

可以使用此功能禁用周日至周六所对应的日期,相关属性:disabledDays (0至6 分别代表 周日至周六)

**示例4-4-1 禁用 周六 所对应的日期**

<input id="d441" type="text" class="Wdate" onFocus="WdatePicker({disabledDays:[6]})"/>

**示例4-4-2 通过position属性,自定义弹出位置**

<input id="d442" type="text" class="Wdate" onFocus="WdatePicker({disabledDays:[0,6]})"/>

**无效日期** **注意:日期格式必须与 realDateFmt 和 realTimeFmt 一致而不是与 dateFmt 一致**可以使用此功能禁用,所指定的一个或多个日期,只要你熟悉正则表达式,你可以尽情发挥  
  
用法(正则匹配):  
如果你熟悉正则表达式,会很容易理解下面的匹配用法  
如果不熟悉,可以参考下面的常用示例  
['2008-02-01','2008-02-29'] 表示禁用 2008-02-01 和 2008-02-29  
['2008-..-01','2008-02-29'] 表示禁用 2008-所有月份-01 和 2008-02-29  
['200[0-8]]-02-01','2008-02-29'] 表示禁用 [2000至2008]-02-01 和 2008-02-29  
['^2006'] 表示禁用 2006年的所有日期

此外,您还可以使用 %y %M %d %H %m %s 等变量, 用法同动态日期限制 注意:%ld不能使用  
['....-..-01','%y-%M-%d'] 表示禁用 所有年份和所有月份的第一天和今天  
['%y-%M-{%d-1}','%y-%M-{%d+1}'] 表示禁用 昨天和明天

当然,除了可以限制日期以外,您还可以限制时间  
['....-..-.. 10\:00\:00'] 表示禁用 每天10点 (注意 : 需要 使用 \: )

不再多举例了,尽情发挥你的正则才能吧!

**示例4-5-1 禁用 每个月份的 5日 15日 25日**

<input id="d451" type="text" class="Wdate" onFocus="WdatePicker({disabledDates:['5$']})"/>  
  
注意 :'5$' 表示以 5 结尾 注意 $ 的用法

**示例4-5-2 禁用 所有早于2000-01-01的日期**

<input id="d452" type="text" class="Wdate" onFocus="WdatePicker({disabledDates:['^19']})"/>  
  
注意:'^19' 表示以 19 开头 注意 ^ 的用法  
当然,可以使用minDate实现类似的功能 这里主要是 在演示 ^ 的用法

**示例4-5-3 配合min/maxDate使用,可以把可选择的日期分隔成多段**

本示例本月可用日期分隔成五段 分别是: 1-3 8-10 16-24 26,27 29-月末  
  
<input id="d453" type="text" class="Wdate" onFocus="WdatePicker({minDate:'%y-%M-01',maxDate:'%y-%M-%ld',disabledDates:['0[4-7]$','1[1-5]$','2[58]$']})"/>

**示例4-5-4 min/maxDate disabledDays disabledDates 配合使用 即使在要求非常苛刻的情况下也能满足需求**

<input id="d454" type="text" class="Wdate" onFocus="WdatePicker({minDate:'%y-%M-01',maxDate:'%y-%M-%ld',disabledDates:['0[4-7]$','1[1-5]$','2[58]$'],disabledDays:[1,3,6]})"/>

**示例4-5-5 禁用前一个小时和后一个小时内所有时间 使用 %y %M %d %H %m %s 等变量**

鼠标点击 小时输入框时,你会发现当然时间对应的前一个小时和后一个小时是灰色的  
  
<input id="d2a25" type="text" class="Wdate" onFocus="WdatePicker({dateFmt:'yyyy-MM-dd HH:mm:ss',disabledDates:['%y-%M-%d {%H-1}\:..\:..','%y-%M-%d {%H+1}\:..\:..']})"/>  
  
注意:%y %M %d等详见[动态变量表](#m242)

**示例4-5-6 #F{}也是可以使用的**

本示例利用自定义函数 随机禁用0-23中的任何一个小时  
打开小时选择框,你会发现有一个小时被禁用的,而且每次禁用的小时都不同  
   
<script>  
function randomH(){  
//产生一个随机的数字 0-23  
var H = Math.round(Math.random() \* 23);  
if(H<10) H='0'+H;  
//返回 '^' + 数字  
return '^'+H;  
}  
</script>  
<input type="text" class="Wdate" id="d456" onFocus="WdatePicker({dateFmt:'HH:mm:ss',disabledDates:['#F{randomH()}']})"/>

1. **有效天与有效日期**

使用无效天和无效日期可以很方便的禁用不可用的日期,但是在只需要启用少部分日期的情况下,有效天和有效日期的功能就非常适合了.  
关键属性: opposite 默认为false, 为true时,无效天和无效日期变成有效天和有效日期

**示例4-6 只启用 每个月份的 5日 15日 25日**

<input id="d46" type="text" class="Wdate" onFocus="WdatePicker({opposite:true,disabledDates:['5$']})"/>  
  
注意 :'5$' 表示以 5 结尾 注意 $ 的用法

1. **特殊天和特殊日期**

特殊天和特殊日期的用法跟完全无效天和无效日期完全相同,但是opposite属性对其无效  
  
关键属性:  
specialDays(0至6 分别代表 周日至周六) 用法同无效天  
specialDates 用法同无效日期,但是对时分秒无效

**示例4-7-1 高亮每周 周一 周五**

<input id="d471" type="text" class="Wdate" onFocus="WdatePicker({specialDays:[1,5]})"/>

**示例4-7-2 高亮每月 1号 15号**

<input id="d472" type="text" class="Wdate" onFocus="WdatePicker({specialDates:['....-..-01','....-..-15']})"/>

**5. 自定义事件**

1. **自定义事件**

如果你需要做一些附加的操作,你也不必担心,日期控件自带的自定义事件可以满足你的需求.此外,你还可以在自定义事件中调用提供的API库来做更多的运算和扩展,绝对可以通过很少的代码满足你及其个性化的需求.  
  
注意下面几个重要的指针,将对你的编程带来很多便利  
this: 指向文本框  
dp: 指向$dp  
dp.cal: 指向日期控件对象  
注意:函数原型必须使用类似 function(dp){}的模式,这样子,在函数内部才可以使用dp

1. **onpicking 和 onpicked 事件**

**示例5-2-1 onpicking事件演示**

<input type="text" id="5421" onFocus="WdatePicker({onpicking:function(dp){if(!confirm('日期框原来的值为: '+dp.cal.getDateStr()+', 要用新选择的值:' + dp.cal.getNewDateStr() + '覆盖吗?')) return true;}})" class="Wdate"/>  
  
注意:你注意到dp.cal.getDateStr和dp.cal.getNewDateStr的用法了嘛? 详见[内置函数和属性](file:///E:/Down/My97DatePicker日期控件4.7文档/4.htm?#m5)

**示例5-2-2 使用onpicked实现日期选择联动**

选择第一个日期的时候,第二个日期选择框自动弹出  
日期从: 至   
注意:下面第一个控件代码的写法  
<input id="d5221" class="Wdate" type="text" onFocus="var d5222=$dp.$('d5222');WdatePicker({onpicked:function(){d5222.focus();},maxDate:'#F{$dp.$D(\'d5222\')}'})"/>  
至  
<input id="d5222" class="Wdate" type="text" onFocus="WdatePicker({minDate:'#F{$dp.$D(\'d5221\')}'})"/>  
  
注意:$dp.$是一个[内置函数](file:///E:/Down/My97DatePicker日期控件4.7文档/4.htm?#m5),相当于document.getElementById

**示例5-2-3 将选择的值拆分到文本框**

年 月 日 时 分 秒   
<input type="text" id="d523\_y" size="5"/> 年  
<input type="text" id="d523\_M" size="3"/> 月  
<input type="text" id="d523\_d" size="3"/> 日  
<input type="text" id="d523\_HH" size="3"/> 时  
<input type="text" id="d523\_mm" size="3"/> 分  
<input type="text" id="d523\_ss" size="3"/> 秒  
<img onclick="WdatePicker({el:'d523',dateFmt:'yyyy-MM-dd HH:mm:ss',onpicked:pickedFunc})" src="My97DatePicker/skin/datePicker.gif" width="16" height="22" align="absmiddle" style="cursor:pointer"/>  
<script>  
function pickedFunc(){  
$dp.$('d523\_y').value=$dp.cal.getP('y');  
$dp.$('d523\_M').value=$dp.cal.getP('M');  
$dp.$('d523\_d').value=$dp.cal.getP('d');  
$dp.$('d523\_HH').value=$dp.cal.getP('H');  
$dp.$('d523\_mm').value=$dp.cal.getP('m');  
$dp.$('d523\_ss').value=$dp.cal.getP('s');  
}  
</script>  
  
注意:el:'d523'中,如果你不需要d523这个框,你可以把他改成hidden,但是el属性必须指定  
$dp.$和$dp.cal.getP都是[内置函数](file:///E:/Down/My97DatePicker日期控件4.7文档/4.htm?#m5)



1. **onclearing 和 oncleared 事件**

**示例5-3-1 使用onclearing事件取消清空操作**

<input type="text" class="Wdate" id="d531" onFocus="WdatePicker({onclearing:function(){if(!confirm('日期框的值为:'+this.value+', 确实要清空吗?'))return true;}})"/>  
  
注意:当onclearing函数返回true时,系统的清空事件将被取消,  
函数体里面没有引用$dp,所以函数原型里面可以省略参数dp

**示例5-3-2 使用cal对象取得当前日期所选择的月份(使用了 dp.cal)**

<input type="text" class="Wdate" id="d532" onFocus="WdatePicker({oncleared:function(dp){alert('当前日期所选择的月份为:'+dp.cal.date.M);}})"/>

**示例5-3-3 综合使用两个事件**

<script>  
function d533\_focus(element){  
var clearingFunc = function(){ if(!confirm('日期框的值为:'+this.value+', 确实要清空吗?')) return true; }  
var clearedFunc = function(){ alert('日期框已被清空'); }  
WdatePicker({el:element,onclearing:clearingFunc,oncleared:clearedFunc})  
}  
</script>  
<input type="text" class="Wdate" id="d533" onFocus="d533\_focus(this)"/>

1. **年月日时分秒的 changing和changed**

年月日时分秒都有对应的changing和changed事件,分别是:  
ychanging ychanged  
Mchanging Mchanged  
dchanging dchanged  
Hchanging Hchanged  
mchanging mchanged  
schanging schanged

**示例5-4-1 年月日改变时弹出信息**

<input type="text" class="Wdate" onFocus="WdatePicker({dchanging:cDayFunc, Mchanging:cMonthFunc, ychanging:cYearFunc, dchanged:cDayFunc, Mchanged:cMonthFunc, ychanged:cYearFunc})"/>  
<script>  
function cDayFunc(){  
cFunc('d');  
}  
function cMonthFunc(){  
cFunc('M');  
}  
function cYearFunc(){  
cFunc('y');  
}  
function cFunc(who){  
var str,p,c = $dp.cal;  
if(who=='y'){  
str='年份';  
p='y';  
}  
else if(who=='M'){  
str='月份';  
p='M';  
}  
else if(who=='d'){  
str='日期';  
p='d';  
}  
alert(str+'发生改变了!\n$dp.cal.date.'+p+'='+c.date[p]+'\n$dp.cal.newdate.'+p+'='+c.newdate[p]);  
}  
</script>  
  
这个例子用到了 $dp.cal.date 和 $dp.cal.newdate 属性,你能从这里发现他们的不同之处吗?  
下面是有关这两个属性的描述详见[内置函数和属性](file:///E:/Down/My97DatePicker日期控件4.7文档/4.htm#m5)

**6. 快速选择功能**

此功能允许指定5个最常用的日期,可以方便用户选择,如果不指定,系统将自动生成  
相关属性:  
qsEnabled 是否启用快速选择功能, 注意:如果日期格式里不包含 d(天) 这个元素时,快速选择将一直显示,不收此属性控制  
quickSel 快速选择数据,可以传入5个快速选择日期,日期格式同min/maxDate  
  
注意:  
日期格式必须与 realDateFmt realTimeFmt 相匹配  
除了使用静态的日期值以外,还可以使用动态参数(如:%y,%M分别表示当前年和月)

**示例6-1 传入2个静态日期**

<input class="Wdate" type="text" onfocus="WdatePicker({dateFmt:'yyyy年MM月dd日',qsEnabled:true,quickSel:['2000-1-10','2000-2-20']})"/>  
  
注意:当传入的数据不足5个时,系统将自动补全

**示例6-2 传入2个动态日期,1个静态日期**

<input type="text" class="Wdate" onFocus="WdatePicker({dateFmt:'yyyy年MM月dd日',qsEnabled:true,quickSel:['2000-10-01','%y-%M-01','%y-%M-%ld']})"/>  
  
注意:当传入的数据不足5个时,系统将自动补全

**三. 配置说明**

**1. 属性配置**

1. **属性表**

|  |  |  |  |
| --- | --- | --- | --- |
| **属性** | **类型** | **默认值** | **说明** |
| 静态属性:只能在WdatePicker.js中配置,一般情况下,不建议您修改静态属性的值 | | | | |
| $wdate | bool | true | 是否自动引入Wdate类 设置为true时,可直接在引入WdatePicker.js的页面里使用 class="Wdate" Wdate可在skin目录下的WdatePicker.css文件中定义 建议使用默认值 |
| $dpPath | string | '' | 是否显示指定程序包的绝对位置,一般情况下为空即可(程序自动创建),该属性是为防止极其少数的情况下程序创建出错而设置的 设置方法: 如果,程序包所在http中的地址为 http://localhost/proName/My97DatePicker/ 则 $dpPath = '/proName/My97DatePicker/'; 建议使用默认值 |
| $crossFrame | bool | true | 是否跨框架,一般设置为true即可,遇到跨域错误时可以将此功能关闭 |
| 可配置属性:可以在WdatePicker方法是配置 | | | | |
| el | Element 或 String | null | 指定一个控件或控件的ID,必须具有value或innerHTML属性(如input,textarea,span,div,p等标签都可以),用户存储日期显示值(也就是dateFmt格式化后的值) |
| vel | Element 或 String | null | 指定一个控件或控件的ID,必须具有value属性(如input),用于存储真实值(也就是realDateFmt和realTimeFmt格式化后的值) |
| doubleCalendar | bool | false | 是否是双月模式,如果该属性为true,则弹出同时显示2个月的日期框 |
| enableKeyboard | bool | true | 键盘控制开关 |
| enableInputMask | bool | true | 文本框输入启用掩码开关 |
| autoUpdateOnChanged | bool | null | 在修改年月日时分秒等元素时,自动更新到el,默认是关闭的(即:需要点击确定或点击日期才更新) 为false时 不自动更新 为true时 自动更新 为null时(默认值) 如果有日元素且不隐藏确定按钮时 为false,其他情况为true |
| whichDayIsfirstWeek | int(1-7之间任何一个数字) | 4 | 周算法不同的地方有一些差异 常见算法有三种 1. ISO8601:规定第一个星期四为第一周,默认值: 4 2. MSExcel:1月1日所在的周: 可以填写: 7 3. 自己根据需要自定义,每年的第一个星期X作为第一周,可以填写: X (X可以是1-7之间任何一个数字)  相关链接:<http://en.wikipedia.org/wiki/ISO_week_date> |
| position | object | {} | 日期选择框显示位置 注意:坐标默认单位是px,是相对当前框架坐标(不受滚动条),left属性只接受数字,top属性除接受数字外还可以接受 'above' 上方显示, 'under' 下方显示, 'auto' 系统根据页面大小自动选择(默认) 如: {left:100,top:50}表示固定坐标[100,50] {top:50}表示横坐标自动生成,纵坐标指定为 50 {left:100}表示纵坐标自动生成,横坐标指定为 100 {top:'above'}表示上方显示 {top:'under'}表示下方显示 请参考示例 |
| lang | string | 'auto' | 当值为'auto'时 自动根据客户端浏览器的语言自动选择语言 当值为 其他 时 从langList中选择对应的语言 你可以参考[语言配置](#m32) |
| skin | string | 'default' | 皮肤名称 默认自带 default和whyGreen两个皮肤 另外如果你的css够强的话,可以自己做皮肤 你可以参考[皮肤配置](#m33) |
| dateFmt | string | 'yyyy-MM-dd' | 日期显示格式 你可以参考[自定义格式](file:///E:/Down/My97DatePicker日期控件4.7文档/2.2.htm#m224) |
| realDateFmt | string | 'yyyy-MM-dd' | 计算机可识别的,真正的日期格式 无效日期设置(disabledDates),最大日期(maxDate),最小日期(minDate)以及快速日期都必须与它们相匹配 建议使用默认值 |
| realTimeFmt | string | 'HH:mm:ss' |
| realFullFmt | string | '%Date %Time' |
| minDate | string | '1900-01-01 00:00:00' | 最小日期(注意要与上面的real日期相匹配) |
| maxDate | string | '2099-12-31 23:59:59' | 最大日期(注意要与上面的real日期相匹配) |
| startDate | string | '' | 起始日期,既点击日期框时显示的起始日期 为空时,使用今天作为起始日期(默认值) 否则使用传入的日期作为起始日期(注意要与上面的real日期相匹配) 你可以参考[起始日期示例](file:///E:/Down/My97DatePicker日期控件4.7文档/2.2.htm#m223) |
| firstDayOfWeek | int | 0 | 周的第一天 0表示星期日 1表示星期一 |
| isShowWeek | bool | false | 是否显示周 你可以参考[周显示示例](file:///E:/Down/My97DatePicker日期控件4.7文档/2.1.htm#m213) |
| highLineWeekDay | bool | true | 是否高亮显示 周六 周日 |
| isShowClear | bool | true | 是否显示清空按钮 |
| isShowToday | bool | true | 是否显示今天按钮 |
| isShowOthers | bool | true | 为true时,第一行空白处显示上月的日期，末行空白处显示下月的日期,否则不显示 |
| readOnly | bool | false | 是否只读 |
| errDealMode | int | 0 | 纠错模式设置 可设置3中模式 0 - 提示 1 - 自动纠错 2 - 标记 |
| autoPickDate | bool | null | 为false时 点日期的时候不自动输入,而是要通过确定才能输入 为true时 即点击日期即可返回日期值 为null时(推荐使用) 如果有时间置为false 否则置为true |
| qsEnabled | bool | true | 是否启用快速选择功能 |
| autoShowQS | bool | false | 是否默认显示快速选择 |
| quickSel | Array | null | 快速选择数据,可以传入5个快速选择日期 注意:日期格式必须与 realDateFmt realTimeFmt realFullFmt 相匹配 你可以参考[快速选择示例](file:///E:/Down/My97DatePicker日期控件4.7文档/2.6.htm#m26) |
| disabledDays | Array | null | 可以使用此功能禁用周日至周六所对应的日期 0至6 分别代表 周日至周六 你可以参考[无效天示例](file:///E:/Down/My97DatePicker日期控件4.7文档/2.4.htm#m244) |
| disabledDates | Array | null | 可以使用此功能禁用所指定的一个或多个日期 你可以参考[无效日期示例](file:///E:/Down/My97DatePicker日期控件4.7文档/2.4.htm#m245) |
| opposite | bool | false | 默认为false, 为true时,无效天和无效日期变成有效天和有效日期 你可以参考[有效天与有效日期示例](file:///E:/Down/My97DatePicker日期控件4.7文档/2.4.htm#m246) |
| specialDates | Array | null | 特殊日期,对指定的日期进行高亮显示 你可以参考[特殊天与特殊日期示例](file:///E:/Down/My97DatePicker日期控件4.7文档/2.4.htm#m247) |
| specialDays | Array | null | 特殊天,使用此功能禁用周日至周六所对应的日期进行高亮显示 0至6 分别代表 周日至周六 你可以参考[特殊天与特殊日期示例](file:///E:/Down/My97DatePicker日期控件4.7文档/2.4.htm#m247) |
| onpicking | function | null | 此四个参数为事件参数 你可以参考[自定义事件示例](file:///E:/Down/My97DatePicker日期控件4.7文档/2.5.htm#m25) |
| onpicked | function | null |
| onclearing | function | null |
| oncleared | function | null |
| ychanging ychanged Mchanging Mchanged dchanging dchanged Hchanging Hchanged mchanging mchanged schanging schanged | function | null | (4.6Beta3新增)  y M d H m s 分别表示年月日时分秒 changing 事件发生在属性改变之前 changed 事件发生在属性改变之后  你可以参考[示例5-4-1](file:///E:/Down/My97DatePicker日期控件4.7文档/2.5.htm?#m254) |

1. **配置全局默认值**

通过配置WdatePicker.js的属性可以避免每次调用都传入配置值,为变成带来很多方便.  
在默认情况下My97为每个属性都配置了默认值,这些默认值都可以在WdatePicker.js中修改的  
你可以根据你个人的喜好更改这些值  
  
比如你比较不喜欢默认的皮肤default 而更喜欢 whyGreen 这个皮肤,  
你可以直接在WdatePicker.js把skin值改为 whyGreen  
这样,你就不必每次调用控件的时候都传入 skin:'whyGreen' 了  
你学会了吗?

1. **配置单个控件**

在控件里面你可以使用 onfocus 或 onclick 事件来调用WdatePicker函数来触发日期控件  
WdatePicker({})其中{}中的内容都是只对当前实例有效,你可以任意配置属性表里有的所有属性  
你可以随意的组合这些属性,达到你的需求  
My97日期控件在这方面是做得非常灵活的.

1. **多套配置快速切换**

您可以设置多个WdatePicker.js文件,如 cn\_WdatePicker.js,en\_WdatePicker.js,simple\_WdatePicker.js等  
在不同的页面引入不同的 WdatePicker.js 达到配置快速切换的目的.  
注意:文件必须以 \_WdatePicker.js(大小写不限制) 为后缀,形如 <yourname>\_WdatePicker.js

**2. 语言配置**

1. **语言列表**

My97DatePicker目录下有个config.js,里面有段代码:  
var langList =  
[  
{name:'en', charset:'UTF-8'},  
{name:'zh-cn', charset:'gb2312'},  
{name:'zh-tw', charset:'GBK'}  
];  
  
这就是语言列表,每个项有name和charset两个属性.  
name 表示语言的名称(必须与浏览器的语言字符串命名相同),在配置的时候,lang属性只能是配置列表里面已有的项,否则将自动返回第一项  
charset 表示对应语言目录下的js文件所对应的编码格式

1. **语言安装说明**

分两步轻松实现:  
1 将语言文件拷贝到 lang 目录  
2 打开 config.js 配置语言列表

**3. 皮肤配置**

1. **皮肤列表**

My97DatePicker目录下有个config.js,里面有段代码:  
var skinList =  
[  
{name:'default', charset:'gb2312'},  
{name:'whyGreen', charset:'gb2312'},  
{name:'blue', charset:'gb2312'},  
{name:'simple', charset:'gb2312'}  
];  
  
这就是皮肤列表,每个项有name和charset两个属性.  
name 表示皮肤的名称,在配置的时候,skin属性只能是配置列表里面已有的项,否则将自动返回第一项  
charset 表示对应皮肤目录下的css文件:datepicker.css所对应的编码格式

1. **皮肤安装说明**

分两步轻松实现:  
1 将皮肤文件包拷贝到 skin 目录  
2 打开 config.js 配置皮肤列表  
  
注意:安装过多的皮肤会影响性能,一般只安装自己使用的皮肤,3个以下比较适宜

**四. 如何使用**

1. 在使用该日期控件的文件中加入JS库(仅这一个文件即可,其他文件会自动引入,请勿删除或改名), 代码如下 <script language="javascript" type="text/javascript" src="datepicker/WdatePicker.js"></script>  
注:src="datepicker/WdatePicker.js" 请根据你的实际情况改变路径  
  
2. 加上主调函数 WdatePicker  
关于 WdatePicker 的用法:  
  
如果您是新手,对js还不是很了解的话一定要多看看这份文档  
基本上每一个演示的示例下面都有相关代码,并且 关键属性用蓝字标出,关键值用红字标出 应该很容易看明白  
  
如果您有一定的水准  
希望能从头到尾把功能介绍好好看一遍,这样大部分功能你都会用了  
  
如果您是高手  
建议您通读配置说明和内置函数

**五. 内置函数和属性**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **函数名** | **返回值类型** | **作用域** | **参数** | **描述** |
| $dp.show | void | 全局 | 无 | 显示日期选择框 |
| $dp.hide | void | 全局 | 无 | 隐藏日期选择框 |
| $dp.$D | String | 全局 | id [string]:对象的ID arg [object]:日期差量,可以设置成 {y:[值],M:[值],d:[值],H:[值],m:[值],s:[值]} 属性 y,M,d,H,m,s 分别代表 年月日时分秒 {M:3,d:7} 表示 3个月零7天 {d:1,H:1} 表示1天多1小时 | 将id对应的日期框中的日期字符串,加上定义的日期差量,返回使用real格式化后的日期串 参考 示例 4-3-2 |
| $dp.$DV | String | 全局 | v [string]:日期字符串 arg [object]:同上例的arg | 将传入的日期字符串,加上定义的日期差量,返回使用real格式化后的日期串 参考 示例 4-3-3 |
| 以下函数只在事件自定义函数中有效 | | | | | |
| $dp.cal.getP | String | 事件function | p [string]:属性名称 yMdHmswWD分别代表年,月,日,时,分,秒,星期(0-6),周(1-52),星期(一-六) f [string]:format 格式字符串 设置方法参考 1.4 自定义格式 | 返回所指定属性被格式字符串格式化后的值[单属性],在changing,picking,clearing事件中返回选择前的值 参考 示例 1-2-2 |
| $dp.cal.getDateStr | String | 事件function | f [string]:格式字符串,为空时使用dateFmt | 返回所指定属性被格式字符串格式化后的值[整个值],在changing,picking,clearing事件中返回选择前的值 |
| $dp.cal.getNewP | String | 事件function | 用法同$dp.cal.getP | 返回所指定属性被格式字符串格式化后的值[单属性],在changing,picking,clearing事件中返回选择后的值 |
| $dp.cal.getNewDateStr | String | 事件function | 用法同$dp.cal.getDateStr | 返回所指定属性被格式字符串格式化后的值[整个值],在changing,picking,clearing事件中返回选择后的值 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **属性名** | **返回值类型** | **作用域** | **参数** | **描述** |
| $dp.cal.date | object | 事件function | $dp.cal.date.y:返回 年 $dp.cal.date.M:返回 月 $dp.cal.date.d:返回 日 $dp.cal.date.H:返回 时 $dp.cal.date.m:返回 分 $dp.cal.date.s:返回 秒 | 在changing,picking,clearing事件中返回选择前的日期对象 |
| $dp.cal.newdate | object | 事件function | 用法同$dp.cal.date | 在changing,picking,clearing事件中返回选择后的日期对象 |

**六. 疑难解答**

任何问题,请先参考 [技术支持](file:///E:/support.htm)  
如果找不到答案,您可以直接在blog留言,或者通过下面的联系方式与我联系

**七. 联系My97**

\* 如果您在使用过程中遇到问题,或者有更好的建议  
\* 欢迎您访问  
\* BLOG: <http://my97.cnblogs.com>  
\* MAIL: support$my97.net($换成@)  
\* 有问题在我blog留言或给我Email吧,最好先仔细看说明,很多问题都是因为没有仔细看说明导致的

© 2010 [My97](mailto:smallcarrot@163.com) All Rights Reserved. [LinezingStat](http://tongji.linezing.com/report.html?unit_id=324202) 浙ICP备09049265号

