

14. 日历和时间

学习要点：

1. 基本使用
2. 回调使用

本节课我们来开始了解 Layui 的内置模块：日历和时间。

一. 基本使用

1. laydate 模块，可以在选择表单时弹出日历和时间让其选择并输入到表单；

```
<input type="text" class="layui-input" id="date">
```

```
layui.use(['laydate'], () => {  
  const laydate = layui.laydate  
  
  //日历模块  
  laydate.render({  
    elem : '#date'      //绑定表单  
  })  
})
```

2. type 属性，可以设置模块的类型选择：year、month、date、time 和 datetime；

```
laydate.render({  
  elem : '#date',  
  type : 'year'      //默认 date  
})
```

3. range 属性，可开启范围选择，即左右两个面板；

```
laydate.render({  
  elem : '#date',  
  range : '~'  
})
```

4. format 属性，可以设置自定义格式，全部格式样式请对照手册；

```
laydate.render({  
  elem : '#date',  
  format : 'yyyy 年 MM 月 dd 日'  
})
```

5. value 属性，可以设置表单默认值，但必须遵循 format 格式；

```
laydate.render({  
  elem : '#date',
```

```
        value : '2030 年 01 月 01 日'  
    })
```

6. `isInitValue` 属性，配合 `value` 属性，是否运行初始值填充；

```
laydate.render({  
    elem : '#date',  
    isInitValue : true  
})
```

7. `min`、`max` 属性，设置最大值最小值；

```
laydate.render({  
    elem : '#date',  
    min : '2018-1-1',  
    max : '2022-1-1'  
})
```

8. `trigger` 属性，可以切换触发方法，默认 `focus`；

```
laydate.render({  
    elem : '#date',  
    trigger : 'dblclick'  
})
```

9. `show` 属性，是否初始就显示日历框，默认 `false`；

```
laydate.render({  
    elem : '#date',  
    show : true  
})
```

PS：剩下还有一些雷同，可对照手册自行验证：

- (1).`showButton`：是否显示底部按钮；
- (2).`btns`：三个按钮的设置；
- (3).`lang`：设置面板语言(`cn`,`en`)；
- (4).`theme`：设置主题(4 种方式)；
- (5).`calendar`：是否显示公历；
- (6).`mark`：标注重要日子；
- (7).`position`：定位方式，嵌入容器中；
- (8).`zIndex`：层叠顺序设置；

二. 基本使用

1. `ready` 方法，日历控件被初始打开的时候会触发回调；

```
laydate.render({  
    elem : '#date',  
    ready(date) {  
        console.log(date)  
    }  
})
```

```
    }  
  })
```

2. **change** 方法，日历控件被切换时的时候会触发回调；

```
laydate.render({  
  elem : '#date',  
  range : '~',  
  change(value, date, endDate) {  
    console.log(value)  
    console.log(date)  
    console.log(endDate)  
  }  
})
```

3. **done** 方法，日历控件选择结束的时候会触发回调；

```
laydate.render({  
  elem : '#date',  
  range : '~',  
  done(value, date, endDate) {  
    console.log(value)  
    console.log(date)  
    console.log(endDate)  
  }  
})
```