

表单专题

- 1、传统的表单元素都有哪些？
 - form
 - input: text、radio、checkbox、button、submit、reset、file、password、hidden...
 - select: 下拉框,option是下拉框中的选项
 - textarea: 文本域
 - label: 标记标签,有点类似于span标签
- 2、HTML5中新增加的表单元素
 - input: search、url、email、tel、number、range、color、date、time...
 - progress: 进度条
 - datalist: 二级下拉框

新增加的表单元素在IE6~8下不兼容，而且没办法处理兼容(HTML5.MIN.JS是不能处理表单这个兼容的)

new : HTML5 中的新属性。

属性	值	描述
accept	mime_type	规定通过文件上传来提交的文件的类型。
align	left right top middle bottom	不赞成使用。规定图像输入的对齐方式。
alt	text	定义图像输入的替代文本。
autocomplete	on off	规定是否使用输入字段的自动完成功能。
autofocus	autofocus	规定输入字段在页面加载时是否获得焦点。 (不适用于 type="hidden")
checked	checked	规定此 input 元素首次加载时应当被选中。
disabled	disabled	当 input 元素加载时禁用此元素。
form	formname	规定输入字段所属的一个或多个表单。

formaction	URL	覆盖表单的 action 属性。 (适用于 type="submit" 和 type="image")
formenctype	见注释	覆盖表单的 enctype 属性。 (适用于 type="submit" 和 type="image")
formmethod	get post	覆盖表单的 method 属性。 (适用于 type="submit" 和 type="image")
formnovalidate	formnovalidate	覆盖表单的 novalidate 属性。 如果使用该属性，则提交表单时不进行验证。
formtarget	_blank _self _parent _top framename	覆盖表单的 target 属性。 (适用于 type="submit" 和 type="image")
height	pixels %	定义 input 字段的高度。(适用于 type="image")
list	datalist-id	引用包含输入字段的预定义选项的 datalist 。

max	number date	规定输入字段的最大值。 请与 "min" 属性配合使用，来创建合法值的范围。
maxlength	number	规定输入字段中的字符的最大长度。
min	number date	规定输入字段的最小值。 请与 "max" 属性配合使用，来创建合法值的范围。
multiple	multiple	如果使用该属性，则允许一个以上的值。
name	field_name	定义 input 元素的名称。
pattern	regex_pattern	规定输入字段的值的模式或格式。 例如 pattern="[0-9]" 表示输入值必须是 0 与 9 之间的数字。
placeholder	text	规定帮助用户填写输入字段的提示。
readonly	readonly	规定输入字段为只读。
required	required	指示输入字段的值是必需的。
size	number_of_char	定义输入字段的宽度。
src	URL	定义以提交按钮形式显示的图像的 URL。
step	number	规定输入字段的合法数字间隔。

type	button checkbox file hidden image password radio reset submit text	规定 input 元素的类型。
value	value	规定 input 元素的值。
width	pixels %	定义 input 字段的宽度。（适用于 type="image"）

resize : none 禁止文本域的手动缩放

```
<input type="range" max="65" min="18" value="25" id="rangeInp"/>
<input type="number" value="25" style="width: 50px; text-align: center;" disabled id="ageInp"/>
<script>
    rangeInp.oninput = function () {
        ageInp.value = this.value;
    };
</script>
```

oninput：相当于PC端的keydown、keyup事件，但是在移动端，由于键盘都是虚拟的，我们无法像PC端一样监听到键盘的触发(也就是不能使用keydown、keyup)，所以我们统一使用input事件来代替之前的事件 =>“当表单元素操作过程中触发这个事件”

新的表单元素的增加带来了什么作用？

1. 方便我们的开发，新的元素提供很多强大的功能，例如：日历。。
2. 在移动端，我们使用input新的类型，当用户输入的时候，手机会根据类型调取出最符合用户输入的虚拟键盘，例如：我们使用的时候number类型，那么调取出来的就是数字键盘....
3. 某些类型自带了表单验证

```
<input type="email" placeholder="请输入邮箱!"/>
```

CSS:

```
#userEmail:valid{}
#userEmail:invalid{}
```

JS:

`this.checkValidity()` 内置验证：不输入内容是TRUE，输入格式错误为FALSE，输入正确为TRUE

4. 提供了一个强大的提示属性：placeholder

```
//解决TRIM方法的兼容性,实现去除字符串的首尾空格
String.prototype.myTrim = function () {
    return this.replace(/(^s+)|(\s+$)/g, '');
};
```

解决placeholder的兼容性

思想：如果是标准浏览器（非IE）用placeholder的方式，反之（IE）采用盒子定位的方式解决

```
<div class="inpBox">
  <input type="text" id="userName" placeholder="请设置用户名"/>
  <span id="tipName">请设置用户名</span>
</div>
<script>
  if (window.navigator.userAgent.indexOf('MSIE') >= 0) {
    // -> IE
    var tipName = document.getElementById('tipName'),
        userName = document.getElementById('userName');
    tipName.style.display = 'block';
    userName.placeholder = null;

    tipName.onclick = function () {
      userName.focus();
    };
    userName.onkeyup = userName.onkeydown = function () {
      var val = this.value;
      tipName.style.display = val.length === 0 ? 'block' : 'none';
    }
  }
</script>
```