

html5 中表单新增属性

新增属性	定义	适用标签
autocomplete	规定表单是否应该启用自动完成，自动完成允许浏览器预测对字段的输入，当用户在输入字段时，浏览器基于之前输入的值，会在下面提示字段中写过的内容	<form> 标签 <input> 标签的 text、search、url、tel、email、password、datepickers、range 以及 color
autofocus	规定当页面加载时域自动的获得焦点	<input> 标签的所有类型
multiple	规定输入域中可选择多个文件或者输入多个值	<input> 标签的 email 和 file 类型
placeholder	提供一个提示，描述输入域中所期待的值	<input> 标签的 text、search、url、tel、email、password 类型
required	规定必须在提交之前填写输入域，输入域不能为空	<input> 标签的 text、search、url、tel、email、password、datepickers、number、checkbox、radio 以及 file

示例：

(1) onchange

✚ 代码如下：

```
<!DOCTYPE html>

<html>

  <head lang="en">

    <meta charset="UTF-8">

    <title>onchange</title>

  </head>

  <body>

    <select id="sel">

      <option value="1">1</option>

      <option value="2">2</option>

      <option value="3">3</option>

      <option value="4">4</option>

      <option value="5">5</option>

    </select>

    <script type="text/javascript">

      var sel=document.getElementById("sel");

      sel.onchange=function(){

        console.log(this.value)

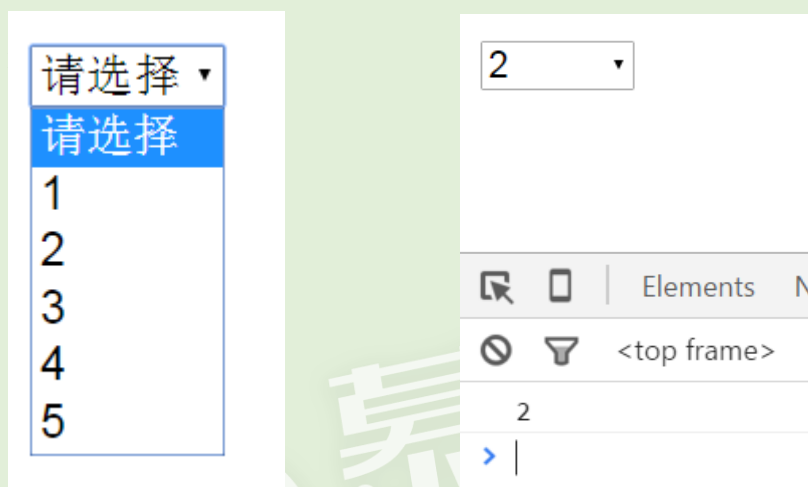
      }

    </script>

  </body>

</html>
```

效果图如下：



分析：当下拉列表的选项发生变化时，触发 `onchange` 事件，控制台打印下拉列表的 `value` 值

(2) `onsubmit`

代码如下：

```
<!DOCTYPE html>

<html>

  <head lang="en">

    <meta charset="UTF-8">

    <title>onsubmit</title>

    <style type="text/css">

      #btn{width:100px;margin-left:65px;}

    </style>

  </head>

  <body>

    <form id="form">
```

```
用户名：<input type="text"> <br/> <br/>

      &nbsp;密码：<input type="text"> <br/> <br/>

      <input type="submit" value="按钮" id="btn">

</form>

<script type="text/javascript">

    var btn=document.getElementById("btn");

    var form=document.getElementById("form");

    form.onsubmit=function(){

        alert("提交成功");

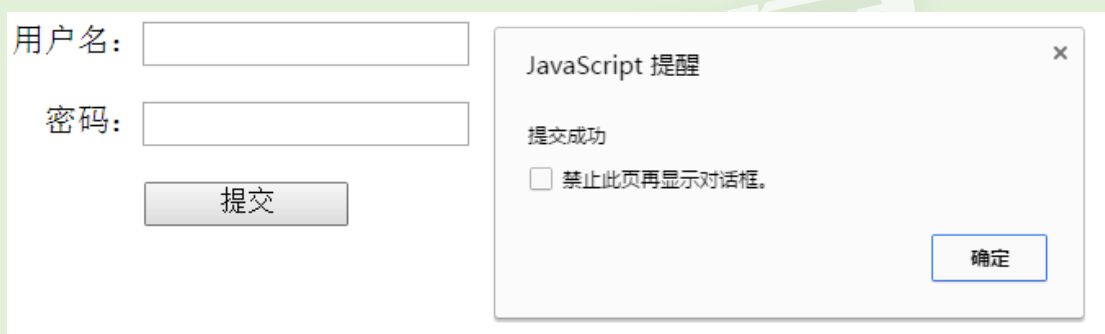
    }

</script>

</body>

</html>
```

🔗 效果图如下：



🔗 分析：onsubmit 事件绑定在<form>标签上，当点击“提交”时，触发 onsubmit 事件，执行事件的执行代码

(3) keyCode

🔗 代码如下：

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head lang="en">
```

```
    <meta charset="UTF-8">
```

```
    <title>onsubmit</title>
```

```
  </head>
```

```
  <body>
```

```
    输入字符： <input type="text" id="text">
```

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

```
      var text=document.getElementById("text");
```

```
      text.onkeydown=function(){
```

```
        console.log(event.keyCode);
```

```
      }
```

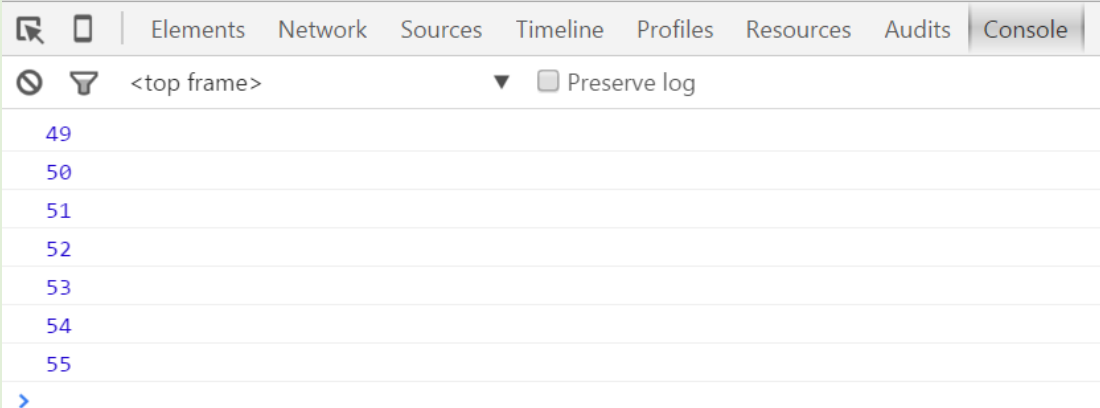
```
    </script>
```

```
  </body>
```

```
</html>
```

效果图如下：

输入字符：



🚦 分析：当按下键盘上的 1 时，控制台输出 1 的代码值：49；输出 2 时，控制台输出：50，等等。

event.keyCode 返回的是键盘按键的字符代码，配合键盘事件使用

(4) 键盘事件绑定

◆ JS 代码中给 html 对象添加

```
<!DOCTYPE html>

<html>

  <head lang="en">

    <meta charset="UTF-8">

    <title>DOM 0 级事件</title>

  </head>

  <body>

    用户名:<input type="text">

    <script type="text/javascript">

      document.onkeyup=function(){

        alert("123");

      }

    </script>

  </body>

</html>
```

因为这个事件是给 document 上添加，所以会导致按键被松开时，别的元素也会触发这个事件

◆ 直接在 HTML 元素标签内添加事件

```
<!DOCTYPE html>

<html>

  <head lang="en">

    <meta charset="UTF-8">

    <title>DOM 0 级事件</title>

  </head>

  <body>

    用户名:<input type="text" onkeyup="fn()">

    <script type="text/javascript">

      function fn(){

        alert("123");

      }

    </script>

  </body>

</html>
```