

<form action="/server.html" method="get">()

Name 是提交内容的名字

Value 是提交内容的具体

Name=value

● 单纯文本

<input type="text" name="text1" value="">这个是单纯的输入框

<input type="password" name="password" value="">密码输入变***的输入框

用户名:
密码:

● 单选框

Name 名字相同是是一组

性别: 男<input type="radio" name="gender" value="男" checked="默认选中">

女<input type="radio" name="gender" value="女">

性别: 男: ☒ 女: ☐

● 多选框

爱好:

羽毛球 <input type="checkbox" name="hobby" value="yumao">

兵乓球 <input type="checkbox" name="hobby" value="pingpang">

篮球 <input type="checkbox" name="hobby" value="basketball">

足球 <input type="checkbox" name="hobby" value="football"

checked="checked">

爱好: 羽毛球 ☒ 兵乓球 ☐ 篮球 ☒ 足球 ☒

● 按钮

<input type="submit" value="submit">提交键

<input type="reset" value="reset">, 点击全部回归到默认值 重置键

<input type="button" value="button">按钮键

● 下拉列表

地址: 省

<select name="province" id="">

<option value="1">广东省</option>

<option value="2">福建省</option>

<option value="3" selected (默认)>浙江省</option>

</select>

范冰冰
女明星
范冰冰
范冰冰
男明星

这个在 option 前面加一句<optgroup label="女明星">即可

● 文本域

自我介绍

```
<textarea name="info"></textarea>
```

Label

`<label for="id">点击</label>`，给一个标签放上和文本框一样的 id，就可以在点击标签的时候，光标也可以直接到文本框选择框中

● Fieldset 分组

在表单中可以使用 `fieldset` 来为表单进行分组

可以将表单项中的同一组放到 `fieldset` 中

用户爱好

性别 ☐ 男 ☒ 女

爱好 ☐ 足球 ☐ 篮球 ☒ 羽毛球 ☒ 乒乓球

```
<fieldset>
```

```
<legend>放组名：用户爱好</legend>
```

```
内容
```

```
</fieldset>
```

框架集:用 `frameset` 来创建框架集，注意 `frameset` 不能和 `body` 出现同一个页面

`<frame src="要引入界面"/>`整个界面都是引入的

新增加表单验证

● placeholder 隐形提示

```
<input type="text" placeholder="社会我飞哥 人丑路子短" />
```


作为提示，一点就消失要修改提示的样式只能在 style 中修改

```
input[type="text"]::-webkit-input-placeholder{  
    color: pink;  
}
```


● autofocus, 自动聚焦

```
<input type="text" placeholder="社会我飞哥路子短" autofocus />
```


能自动获取焦点

● required, 是否是必填的

```
<input type="text" placeholder="社会我飞哥" autofocus />  
<input type="text" placeholder="社会我飞哥" autofocus required  
>  
<input type="submit"/>
```


! 请填写此字段。

如果加了这个属性就是必填的项

● pattern 放正则表达式

```
<input type="text" pattern="\d{1,5}" />  
<input type="submit"/>
```

● formaction 在submit里定义提交地址

```
<form action="http://www.atguigu.com">//这里填表单整体的  
    <input type="submit"/>  
    <input type="submit" formaction="http://www.baidu.com" />  
//这个是填写这个提交要去的  
</form>
```

● datalist列表标签:

```
<input type="text" placeholder="世界上最好的人"
list="list"></input>
```

<datalist id="list">//这个id必须和list的id一样

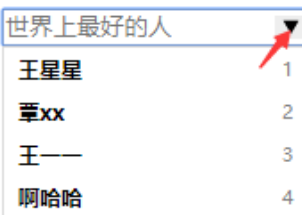
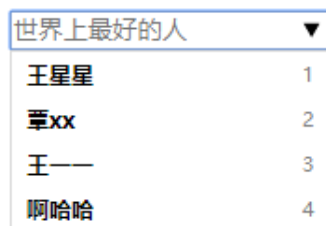
```
<option value="王星星">1</option>
```

```
<option value="覃xx">2</option>
```

```
<option value="王一一">3</option>
```

```
<option value="啊哈哈">4</option>
```

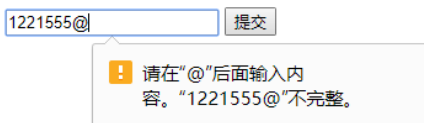
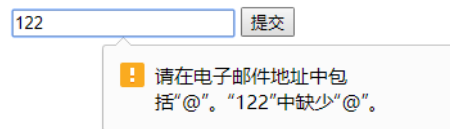
```
</datalist>
```



● email, 是否是邮箱格式

```
<input type="email" />
```

```
<input type="submit" value="提交"/>
```

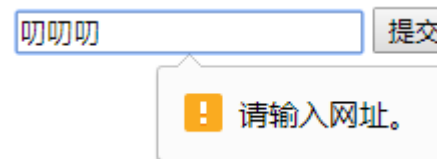


@前后都有数值

● url 验证是否是网址格式

```
<input type="url" />
```

```
<input type="submit" value="提交"/>
```



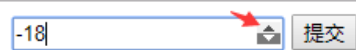
- range 进度条格式

`<input type="range" step="2" min="0" max="10" value="2"/>`
每点一步走 2，最短在 0，最长在 10，



- number 只能填与数字有关的

`<input type="number"/>`
`<input type="submit"/>`



只能输入数字



可以输入与任何数字有关的，e 是 log 的底

- color，出现色板

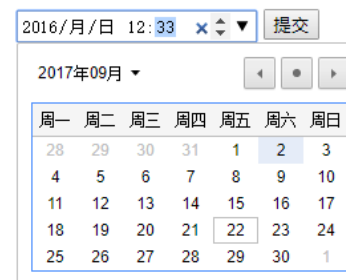
`<input type="color"/>`
`<input type="submit"/>`



- datetime 移动端显示，显示完整的时间

- datetime-local 显示完整日期，不含时区

`<input type="datetime-local"/>`
`<input type="submit"/>`



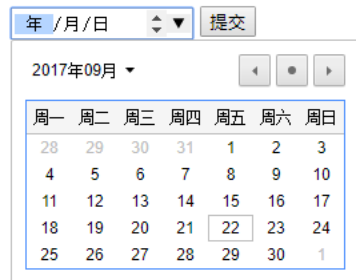
- time显示时间，不含时区

```
<input type="time"/>  
<input type="submit"/>
```



--:59 x [up/down arrows] 提交

- date 显示日期

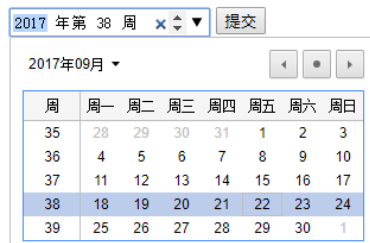


年 / 月 / 日 [up/down arrows] 提交

2017年09月 [left/right/center arrows]

周一	周二	周三	周四	周五	周六	周日
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1

- week 显示周数



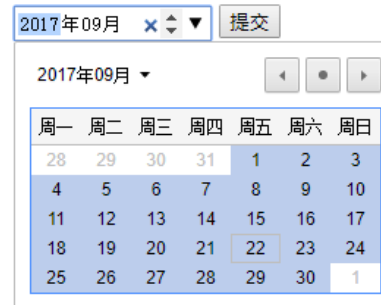
2017 年第 38 周 x [up/down arrows] 提交

2017年09月 [left/right/center arrows]

周	周一	周二	周三	周四	周五	周六	周日
35	28	29	30	31	1	2	3
36	4	5	6	7	8	9	10
37	11	12	13	14	15	16	17
38	18	19	20	21	22	23	24
39	25	26	27	28	29	30	1

- month 选择月份

```
<input type="month"/>  
<input type="submit"/>
```



2017年09月 x [up/down arrows] 提交

2017年09月 [left/right/center arrows]

周一	周二	周三	周四	周五	周六	周日
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1

表单验证 (js 里面的)

```
var textNode = document.querySelector("input[type='text']");
//验证没有通过的时候 才会触发 (八种验证)
textNode.addEventListener("invalid",function(ev){
    console.log(this.validity);
    //输入值为空时返回true
    console.log(this.validity.valueMissing);
})
```

表单验证反馈(帮给表单的节点)

validity 对象, 通过下面的 valid 可以查看验证是否通过, 如果八种验证都通过返回 true, 一种验证失败返回 false

node.addEventListener("invalid",fn1,false);

valueMissing : 输入值为空时返回 true

typeMismatch : 控件值与预期类型不匹配返回 true

patternMismatch: 输入值不满足 pattern 正则返回 true

tooLong : 超过 maxLength 最大限制返回 true

rangeUnderflow: 验证的 range 最小值返回 true

rangeOverflow : 验证的 range 最大值返回 true

stepMismatch : 验证 range 的当前值 是否符合 min、max 及 step

的规则返回 true

customError 不符合自定义验证返回 true

setCustomValidity

formnovalidate 属性给提交 html 按钮添加, 禁止验证

自定义标签

```
var textNode = document.querySelector("input[type='text']");
var submitNode = document.querySelector("input[type='submit']");
submitNode.onclick=function(){
    if(textNode.value == "xfz"){//自定义标签要求
        this.setCustomValidity("不要输入敏感词");
    }else{
        //自定义便签的提示
        this.setCustomValidity("");
    }
}
submitNode.addEventListener("invalid",function(ev){
    console.log(this.validity);//不通过时显示提示
})//验证吧不通过时验证
```

前端验证

1. 首先导入验证库和 jQuery 的库, js 函数库中
2. 写 html 的样式
3. 写 js 的, 注意规则和提示之间都是用 “,”

```
$(function () {  
    $("#myForm").validate({  
        rules: {  
            username: {  
                required: true,  
                minlength: 2,  
                maxlength: 10  
            }  
        },  
        messages: {  
            username: {  
                required: '用户名不能为空',  
                minlength: '用户名长度最少为两位',  
                maxlength: '用户名长度最多为十位'  
            }  
        }  
    })  
})
```

验证函数

form 表单

1. 每个的规则

2. 每个规则的提示

这个是表单里面 name = "username"

// 不能为空 最大最小长度的设置

required: true, 是否为空

minlength: 2, 最短长度的值

maxlength: 10 最长长度

激活 Wir
转到“设置”

常用 rule:

required: true, 判断是否为空,

remote: "check.php" 使用 ajax 方法调用 check.php 验证输入值,

email: true, 判断是不是合法的 email 格式,

url: true, 判断是不是合法的网址,

date: true, 判断是不是时间的格式,

dateISO: true

必须输入正确格式的日期 (ISO), 例如: 2009-06-23, 1998/01/22。只验证格式, 不验证有效性,

number: true, 判断是不是数字的格式,

digits: true, 必须输入整数

equalTo: "写 id", 看看这个值与这个 id 是否相等,

maxlength: 10, -----> 最长只能写 10 个字段,

minlength: 2, -----> 最短要写 2 个字段,

rangelength: [5,10] 输入长度在 5 和 10 之间的字符串 (汉字算一个字符)

range: [5,10] 输入值必须介于 5 和 10 之间。

max: 5, 输入值不能大于 5

min: 2, 输入值不能小于 5

step: \$.validator.format("Please enter a multiple of {0}.")


```
$('#myForm').validate({
  debug: true,
  //写这个不跳转，用于测试和编写来测试，不用点击，写完就知道对不对
  提交按不了，如果一直都没有提示，就是验证通过。
```

```
  rules: {
    username: {
      required: true,
      minlength: 2,
    },
    pwd1: {
      required: true,
      minlength: 6
      maxlength: 8
    },
    pwd2: {
      equalTo: '#password1'//写 id
    }
  }
  messages: {
    username: {
      required: '用户名是必须的',
      minlength: '用户名至少为 6 位'
    },
    pwd1: {
      required: '密码是必须的',
```

```
      minlength: '密码至少为 6 位',
      maxlength: '密码最多 8 位'
    },
    pwd2: {
      equalTo: '必须与密码相同'
    }
  }
})
```