<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



* **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">



* **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">



* **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>

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

* **文本域**



<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firldset中



<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 />



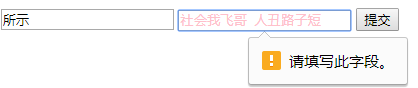
能自动获取焦点

* 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"/>



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>

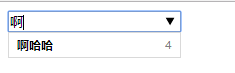
<option value="覃xx">2</option>

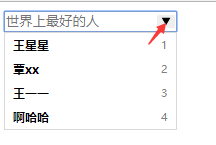
<option value="王一一">3</option>

<option value="啊哈哈">4</option>

</datalist>



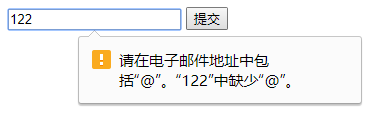


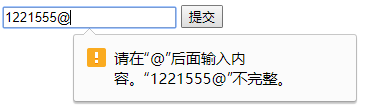


* email，是否是邮箱格式

<input type="email" />

<input type="submit" value="提交"/>



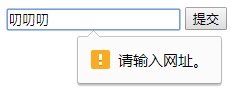


@前后都有数值

* 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

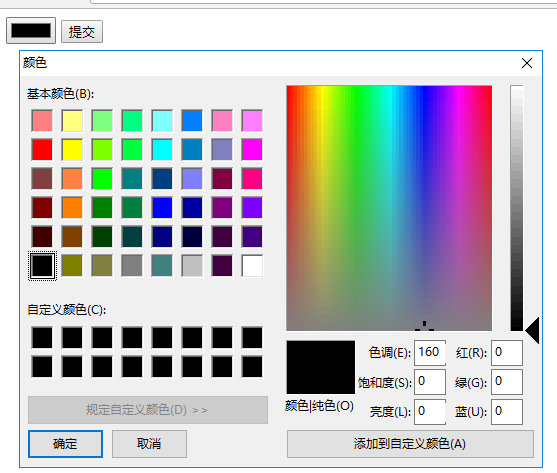


可以输入与任何数字有关的，e是log的底

* color，出现色板

<input type="color"/>

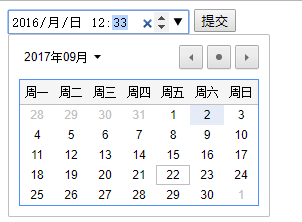
<input type="submit"/>



* 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"/>



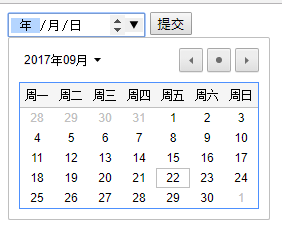
* time显示时间，不含时区

<input type="time"/>

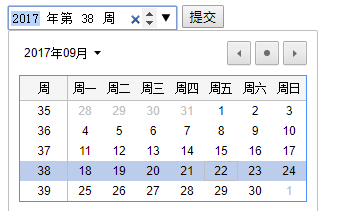
<input type="submit"/>



* date显示日期



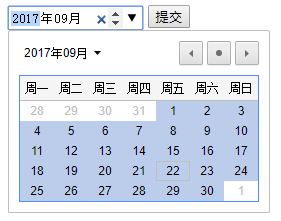
* week 显示周数



* month 选择月份

<input type="month"/>

<input type="submit"/>



**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']");

varsubmitNode = 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“，”**



**常用rulse：**

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: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'

}

}

})