jqueryValidation: jquery-UI 小组组长;

https://jqueryvalidation.org/;

从页面性能的角度来说:

最好是把js的引入放在结束的body标签上面;

基本功能：

validate:插件的核心方法;

rules:对那些元素规定规则：

required:必须填写,值为true,false

digits:正整数;

range:取值范围;[18,100]

min:最小值;取值数字

max:最大值,取值数字

messages:自定义提示信息;

debug:true,不会向远程发送请求,方便本地调试,节约资源;

minlength:最小长度;取值数字

maxlength:最大长度;取值数字

rangelength:[2,10]:长度2-10之间;

email:邮箱格式,取值true、false

url: 校验网址,取值true、false,网址格式:必须http://

dateISO：YYYY-MM-DD校验比较严格

date:校验规则比较宽泛;

number:数字,取值true,false

equalTo:'#pass':值必须和#pass的值相等;

remote:'url',把数据发送到远程进行校验;默认GET

进阶功能：

valid():此方法返回true或者false;代表表单校验是否通过;

rules():获取表单元素的校验规则;比如$('#username').rules();获取到的是验

证规则;

rules('add',rules):

向表单元素增加校验规则;比如：

$('#user').rules('add',{minlength:10,maxlength:20});第一个参数必须是add,第二个参数是规

则配置项;

rules('remove',rules):删除表单元素校验规则;

删除表单元素的校验规则,比如:

$('#user').rules('remove','minlength maxlength ···');第一个参数必须是remove,第二个参数

是删除的规则,多个规则之间以空格隔开;

ignore:'#user':忽略#user的校验;

submitHandler:验证通过后执行的函数,验证通过后会触发;

invalidHandler:function(event,validator){ }//验证不通过执行的函数;

validator.numberOfInvalids():错误数量;

groups:{//规定把哪些元素的错误信息一起显示;

login:'user pass'

},

errorPlacement:function(error,element){//规定把错误信息统一显示在什么位置;

error.insertBefore('#info');//错误信息统一显示在#info的前面;

}

onsubmit:是否在提交时验证,取值true，false;

onfocusout:是否在获取焦点是验证

onkeyup:是否在敲击键盘是验证

onclick:是否在鼠标点击时验证,一般用于checkbox或者radio

focusInvalid：提交表单后,未通过验证的表单是否获得焦点;取值true，false

focusCleanup:当未通过验证的元素获得焦点的时候,是否移出错误信息;取值true，false

errorClass:指定错误提示的css类名,可以自定义错误提示的样式;

errorClass:'wrong':定义错误信息class名字为wrong

validClass:指定验证通过的css类名

validClass:'right':定义正确信息class名字为right;

errorElement:使用什么标签标记错误;

errorElement:'div'

wrapper:使用什么标签把上边的errorElement包起来

errorLabelContainer:把错误信息统一放在一个容器里面;

highlight：可以给未通过的元素加效果;

\*\*\*\*unhiglight:取出未通过验证的元素的效果,一般和highlight同时使用;

$.validator.addMethod(name,method,message);

name:自定义规则的方法名称

method:function(value,element){}

//方法逻辑;

//value:元素的value值;

//element:当前的元素

message:提示信息;