要去掉某个css样式就直接设置为null就行了；；

Eg: background:null;

任意居中写法：width：50%；margin-left：-70px.

文字阴影：text-shadow;

盒子阴影：box-shadow:右 下 半径 颜色

-webkit-box-reflect:direction offset

direction:设置倒影生成的方向，above(上)、below(下)、left(左)、right(右)

offset:设置倒影和原图之间的距离，可为像素或者百分比，可正可负；

-webkit-box-reflect:below 5px –webkit-linear-gradient(top,rgba(0,0,0,0) 50%,rga(0,0,0,0.5) 100%)

内减盒子高宽：box-sizing:border-box;

背景图片随窗口变化：background-size:cover;

P标签本身是有padding间距的；

margin-top会使上一层继承这个属性（兼容性问题，应该用padding-top解决）；

在div里面添加contentEditable=’true’可以使div变成编辑框，outline：none使边框消失；

body字体默认为16px；

竖直方向内容超过盒子高度增加滚动条：overflow-y:scroll；

::-webkit-scrollbar 滚动条整体部分

::-webkit-scrollbar-thumb 滚动条里面的小方块，能向上向下移动（或往左往右移动，取决于是垂直滚动条还是水平滚动条）

::-webkit-scrollbar-button: 滚动条的轨道的两端按钮，允许通过点击微调小方块的位置

样式设置eg:

#gundong::-webkit-scrollbar{width:5xp; background:#ddd}

#dundong::-webkit-scrollbar-thumb{background:#666}

遍历数组： for(var n in nr)

oncontextmenu=”return false” 禁止鼠标右键

ondragstart=”return false” 禁止鼠标拖拽

onselectstart=”return false” 禁止鼠标选中文字

onmousewheel=”return false” 禁止鼠标滚动

onselect=”document.selection.empty()” 禁止复制文字

e.stopPropagation(); jQ中阻止冒泡事件

e.preventDefault(); js中阻止冒泡事件

<embed wmode=”transparent”><embed>因为视频的层级最高，用wmode=“transparent”属性可以使层级降低。

<select disabled=”true”>禁止选项，false不禁止。

<input placeholder=”默认文字”>

replace(/呀/,”和”);后者替换前者字符，只替换一个；

replace(/呀/g,”和”)；后者替换前者所有字符。

Substring() 用法:<字符串对象>.subtring(始，终)；返回源字符串的子字符串，该字符串是原字符串从始到终位置的前一段位置的一段，始-终=返回字符串的长度。如果没有指定终或指定得超过字符串长度，则子字符串从始位置一直取到原字符串尾。如果所指定的位置不能返回字符串，则返回空字符串。

要将两个单选按钮联动起来就要加相同的 name名：

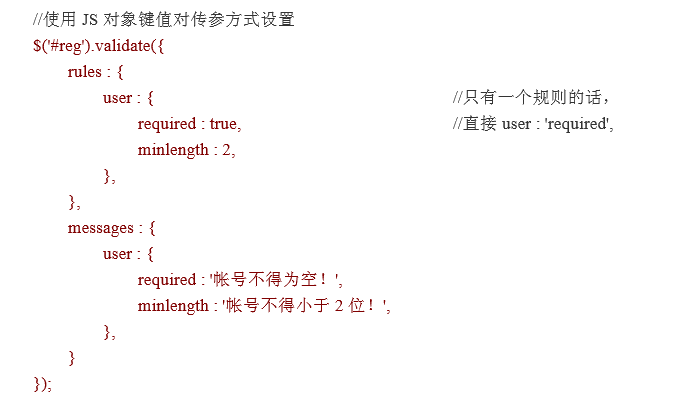
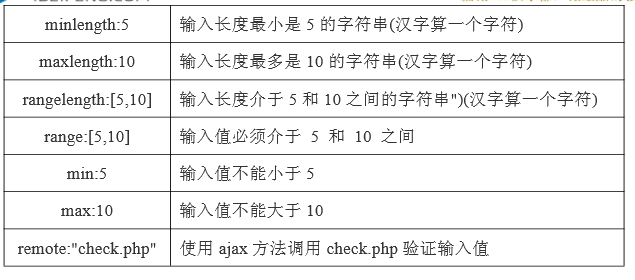
<input type="radio" name="aa">男

<input type="radio" name="aa">女

**表单验证：引入 jquery.**validate.js

插件下载地址：<http://bassistance.de/jquery-plugins/jquery-plugin-validation>

个性化表单设计思路就是自己设计页面然后将所选择的值传给<input>里面的value;



实现页面跳转：window.location=”info.html”;

**.load() 方法讲解**

**.load三个参数**

参数一： url 必选 (url,参数类型字符串）

参数二：data 可选，发送key:value 数据，参数类型为object

参数三：callback 可选 回调函数，参数类型为函数 Function

案例一：本地.html 文件

安例二：服务器文件 .php 文件 (获取方式 get 和 post)

以什么方式发送数据就以什么方式获取数据；

$(“#main”).load(“info.php?name=qx”); get（默认）方式

后台处理

<?php

if($\_GET[‘name’==”qx”])

echo ‘呼啦啦’;

?>

$(“#main”).load(“info.html”,{name:”qx”}); post方式

<?php

if($\_POST‘name’==”qx”])

echo ‘呼啦啦’;

else echo ‘好牛逼’;

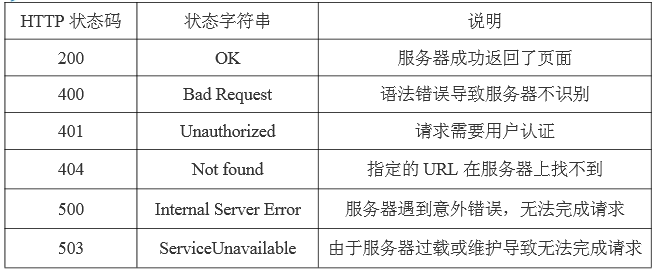
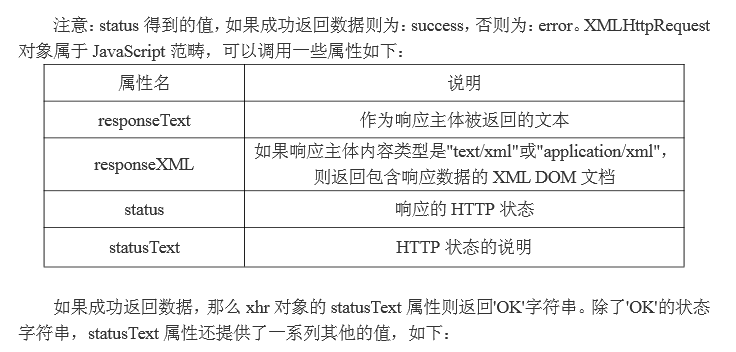
?>

回调函数有三个值：eg:

$(“#main”).load(“info.html”,{name:”qx”},finction(response,status,xhr){});

response就是我们返回的数据;status是我们返回的状态，只有两种值success或者error;

xhr是一个对象,xhr.responseText其实就是第一个参数response,xhr.status返回响应的HTTP状态码。



**//实时监听视频或者音频的播放时间：**

**$(“#myvideo“).get(0).addEventListener(‘timeupdate’,function(){**

**});**

**//实时监听输入框内容**

//IE

if(isIE){

$(“#sno”).get(0).attachEvent(“onpropertychange”,function(e){

alert(e.srcElement.value);

});

}

Else{

$(“#sno”).get(0).addEventListener(‘input’,function(e){

alert(e.target.value);

},false);

}

//全屏显示

//webkit

element.webkitRequestFullscreen();

element.webkitCancelFullscreen();

//FireFox

element.mozRequestFullscreen();

element.mozCancelFullscreen();

//W3C

element.requestFullscreen();

element.exitFullscreen();

function fullscreen(){

if(element.requestFullscreen)

element.requestFullscreen;

else if(element.mozRequestFullscreen)

element.mozRequestFullscreen;

else if(element.msRequestFullscreen)

element.msRequestFullscreen;

else if(element.oRequestFullscreen)

element.oRequestFullscreen;

else if(element.webkitRequestFullscreen)

element.webkitRequestFullscreen;

}

将数字型数转化为字符型：

var a=10;

b=a.toString();

alert(b.length);

将字符型数转化为数字型：

var a=”11210”;

parseInt(a);

**input文本框输入禁止在某些情况下是很有必要的，**



readonly表示此域的值不可修改，仅可与 type="text" 配合使用，可复制，可选择,可以接收焦点，后台会接收到传值.



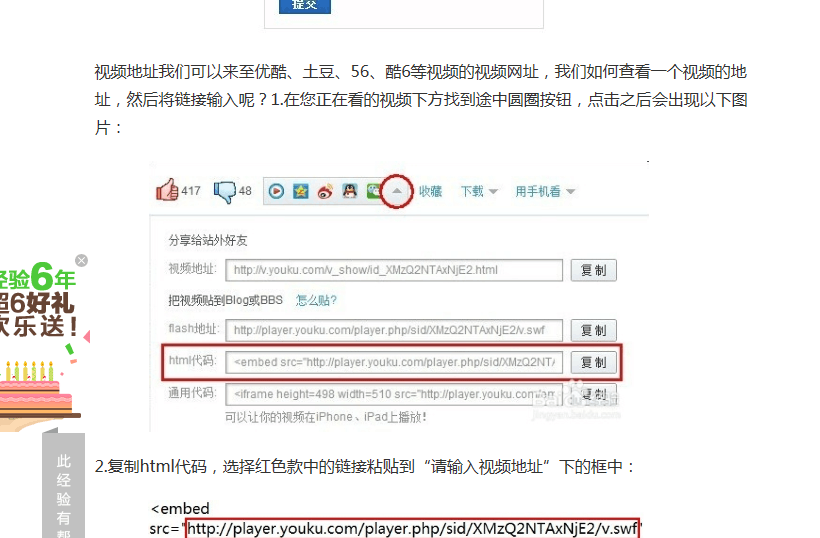
disabled表示禁用input元素，不可编辑，不可复制，不可选择，不能接收焦点,后台也不会接收到传值



最后介绍一个常用的标签，浏览器通常会记录input输入框的记录，所以你在输入的时候，经常会下拉很多内容，如下图：如果你想去掉的话，最好加上autocomplete="off"，使用方法如下：autocomplete="off"



获取网页的视频：



保持animation动画最后动态 animation-fill-mode:forwords;

包吃标签内部标签<xmp></xmp>;

function browserRedirect() {

var sUserAgent = navigator.userAgent.toLowerCase();

var bIsIpad = sUserAgent.match(/ipad/i) == "ipad";

var bIsIphoneOs = sUserAgent.match(/iphone os/i) == "iphone os";

var bIsMidp = sUserAgent.match(/midp/i) == "midp";

var bIsUc7 = sUserAgent.match(/rv:1.2.3.4/i) == "rv:1.2.3.4";

var bIsUc = sUserAgent.match(/ucweb/i) == "ucweb";

var bIsAndroid = sUserAgent.match(/android/i) == "android";

var bIsCE = sUserAgent.match(/windows ce/i) == "windows ce";

var bIsWM = sUserAgent.match(/windows mobile/i) == "windows mobile";

document.writeln("您的浏览设备为：");

if (bIsIpad || bIsIphoneOs || bIsMidp || bIsUc7 || bIsUc || bIsAndroid || bIsCE || bIsWM) {

document.writeln("phone");

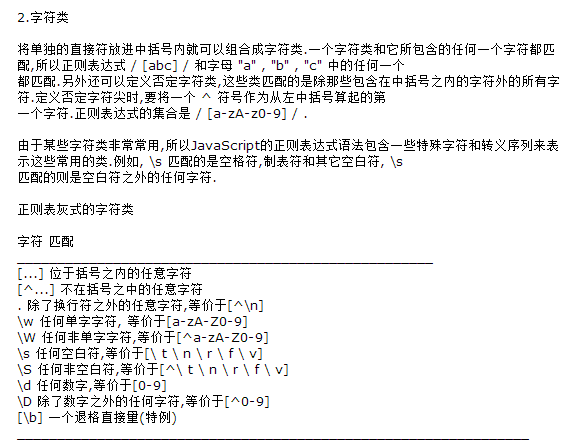
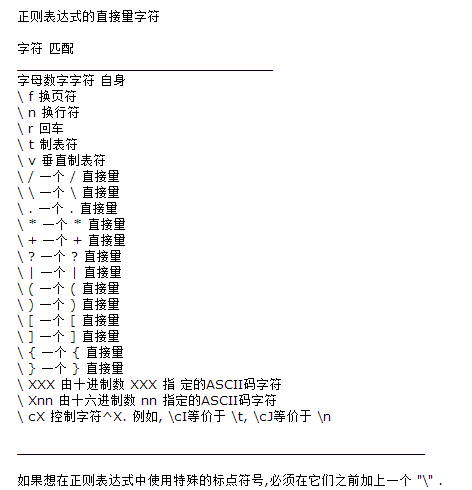
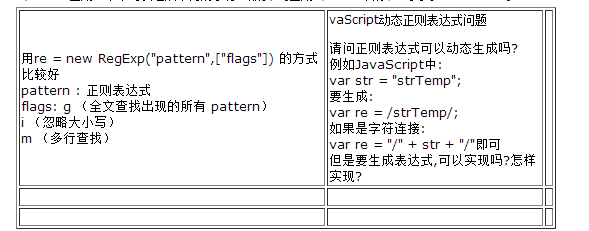
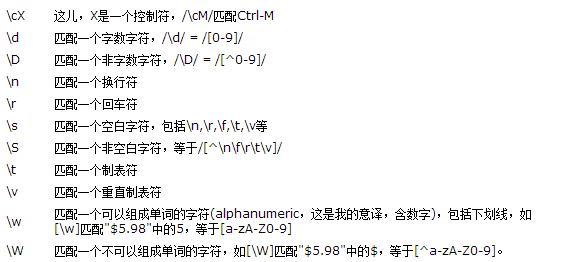
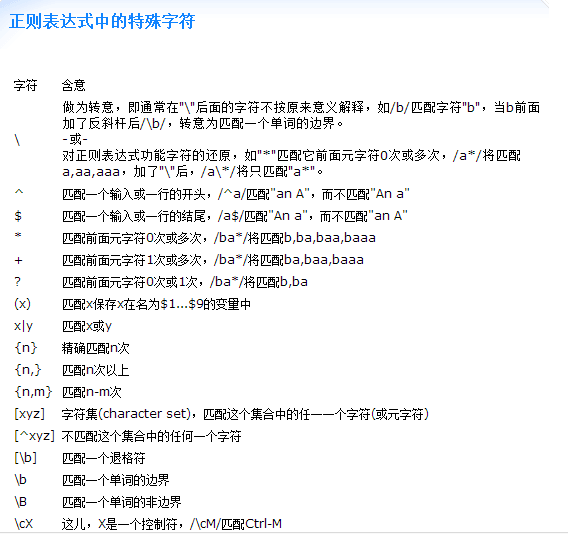
} else {

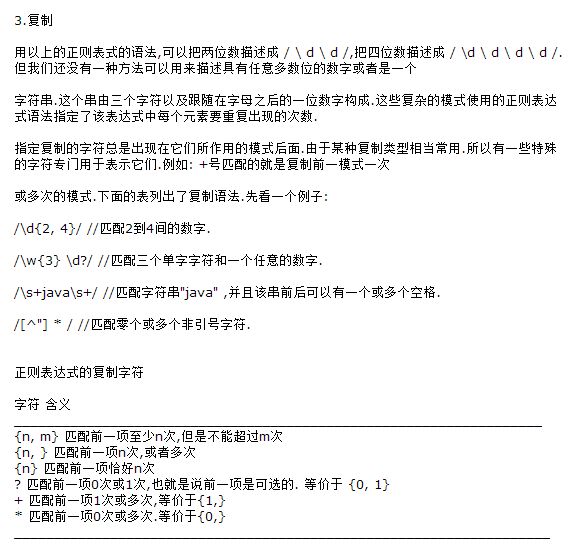
document.writeln("pc");

}

}







匹配类样式：

[class^="icon-"],

[class\*=" icon-"]{ font-family:'webfont';

font-style:normal; font-weight:normal;}

background-size:

length: 如100px 100px;

percent:如90% 90%;

cover:背景图片的较小边放大到目标大小结束;

contain:背景图片的较大边缩小到目标大小结束。

<img data-img-lg=”1.png”>

$(“img”).data(“img-lg”);

$element.data()是一个函数，专门用于取元素上自定义属性（data-xxx）；

函数的参数是我们要取得的属性名称（xxx）



Css3滚动条样式设置：

* ::-webkit-scrollbar 滚动条整体部分
* ::-webkit-scrollbar-thumb??滚动条里面的小方块，能向上向下移动（或往左往右移动，取决于是垂直滚动条还是水平滚动条）
* ::-webkit-scrollbar-track??滚动条的轨道（里面装有Thumb）
* ::-webkit-scrollbar-button 滚动条的轨道的两端按钮，允许通过点击微调小方块的位置。
* ::-webkit-scrollbar-track-piece 内层轨道，滚动条中间部分（除去）
* ::-webkit-scrollbar-corner 边角，即两个滚动条的交汇处
* ::-webkit-resizer 两个滚动条的交汇处上用于通过拖动调整元素大小的小控件

//js屏蔽浏览器自带的ctrl+s保存网页功能

$(document).keydown(function(e){

e.stopPropagation();

if(e.ctrlKey && e.keyCode==83){

window.event.returnValue = false; //屏蔽浏览器本身保存网页

console.log(1111);

}

})

Node express获取请求的ip地址 req.headers.host req.ip

FileReader()读取本地图片文件并显示

**var** reader = **new** FileReader();   
    //图片转为了base64编码  
    reader.onload = **function**(e)**{**

//或者e.target.result  
        result.innerHTML = '<img src="'+**this**.result+'" alt=""/>'   
    **}**

reader.readAsDataURL(file);

FileReader的方法和事件

|  |  |
| --- | --- |
| abort | 中断读取 |
| readAsText(file, [encoding]) | 将文件读取为文本 该方法有两个参数，其中第二个参数是文本的编码方式，默认值为 UTF-8。这个方法非常容易理解，将文件以文本方式读取，读取的结果即是这个文本文件中的内容。 |
| readAsBinaryString(file) | 将文件读取二进制码 通常我们将它传送到后端，后端可以通过这段字符串存储文件 |
| readAsDataURL(file) | 将文件读取为DataURL 将文件读取为一串Data URL字符串，将小文件以一种特殊格式的URL地址直接读入页面。小文件指图像与html等格式的文件。 |
| 事件 | |
| onabort | 数据读取中断时触发 |
| onerror | 数据读取出错时触发 |
| onloadstart | 数据读取开始时触发 |
| onload | 数据读取成功完成时触发 |
| onloadend | 数据读取完成时触发，无论成功失败 |

# web app <input type="file" > 只调用图库

…

<input type="file" accept="image/\*" capture="camera">

<input type="file" accept="video/\*" capture="camcorder">

<input type="file" accept="audio/\*" capture="microphone">

accept表示打开的系统文件目录

capture表示的是系统所捕获的默认设备，camera：照相机；camcorder：摄像机；microphone：录音；

其中还有一个属性multiple，支持多选，当支持多选时，multiple优先级高于capture，所以只用写成：<input type="file" accept="image/\*" multiple>就可以，

# [localtunnel内网服务器暴露至公网](http://www.cnblogs.com/yxiaoqian/p/6747897.html)

**1.安装**

**npm install -g localtunnel**

**2.运行**

**lt --port 8080**

**(your url is: http://xxxx.localtunnel.me ) 就实现了通过这个地址访问本地服务器8080端口的项目了～**

**另！以上是国外项目 速度慢，搭建自己的localtunnel服务端**

Meta标签中的format-detection属性及含义

format-detection翻译成中文的意思是“格式检测”，顾名思义，它是用来检测html里的一些格式的，那关于meta的format-detection属性主要是有以下几个设置：

meta name="format-detection" content="telephone=no"

meta name="format-detection" content="email=no"

meta name="format-detection" content="adress=no"

也可以连写：meta name="format-detection" content="telephone=no,email=no,adress=no"

下面具体说下每个设置的作用：

**一、telephone**

你明明写的一串数字没加链接样式，而iPhone会自动把你这个文字加链接样式、并且点击这个数字还会自动拨号！想去掉这个拨号链接该如何操作呢？这时我们的meta又该大显神通了，代码如下：

telephone=no就禁止了把数字转化为拨号链接！  
telephone=yes就开启了把数字转化为拨号链接，要开启转化功能，这个meta就不用写了,在默认是情况下就是开启！

**二、email**

告诉设备不识别邮箱，点击之后不自动发送

email=no禁止作为邮箱地址！  
email=yes就开启了把文字默认为邮箱地址，这个meta就不用写了,在默认是情况下就是开启！

**三、adress**

adress=no禁止跳转至地图！  
adress=yes就开启了点击地址直接跳转至地图的功能,在默认是情况下就是开启！

**使用nodejs将html5 canvas base64编码图片保存为文件**

//接收前台POST过来的base64

var imgData = req.body.imgData;

//过滤data:URL

var base64Data = imgData.replace(/^data:image\/\w+;base64,/, "");

var dataBuffer = new Buffer(base64Data, 'base64');

fs.writeFile("out.png", dataBuffer, function(err) {

if(err){

res.send(err);

}else{

res.send("保存成功！");

}

});