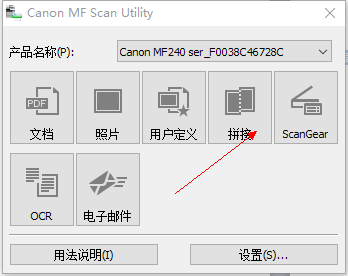
***灵魂画师***

我是black，我一直都在默默无闻着，我很内向暴躁，希望可以努力做一个受欢迎的人，那么我想从现在一步一步的记录下我的脚印。

交流

如何发表自己的意见，为了锻炼自己，我决定开始接触聊天工具，例如微博，奥不，没兴趣。

打印机如何复印身份证复印件：

1. 安装该打印机的驱动：
2. 选择拼接：
3. 把身份证都统一放在右侧（左侧）打印，最后调整大小即可。

UI

对于UI设计这一块，我是不擅长，但是看多了网上的素材，还是有自己的艺术美感的。嘻嘻，一切都在成长，我当然也不会退步。常用的UI软件有PS、Core、AI。但真正的艺术还是设计。设计才是灵魂，设计思路才是无法被取缔的，当然专业软件则更加如虎添翼，也起着不可忽视的作用。

就像我在网上看到的一个视频，一个美国大叔在超市干了三十多年的录入数据的工作,那手速肉眼已经无法判断了，厉害啊。

什么事都能熟能生巧，我告诉你骑自相车的空气阻力、摩擦力、重心。你一开始也肯定会摔倒。这是无法避免的，但就像现在网上的一些教学视频，一开始将了一大堆类似黑科技的东西，最后只用一个简单demo了事的虎头蛇尾。可这是无法避免的，这是经验错误，懂得越多反而越难解释简单的东西。就像你问一个拳击手如何打倒对手，他可能只会说，要有很好的躲避攻击意识，还要有很好的耐力。你听的头头是道，可是实际上对新手是没有这类固有思维的，也就是说并无法理解你说的躲避是右肩退30cm还是大腿收紧。

但它可以完善你的思维，它已经在你的脑中建立了一个框架，分为三部分：躲避、攻击、耐力。完善这个框架就看自己了。

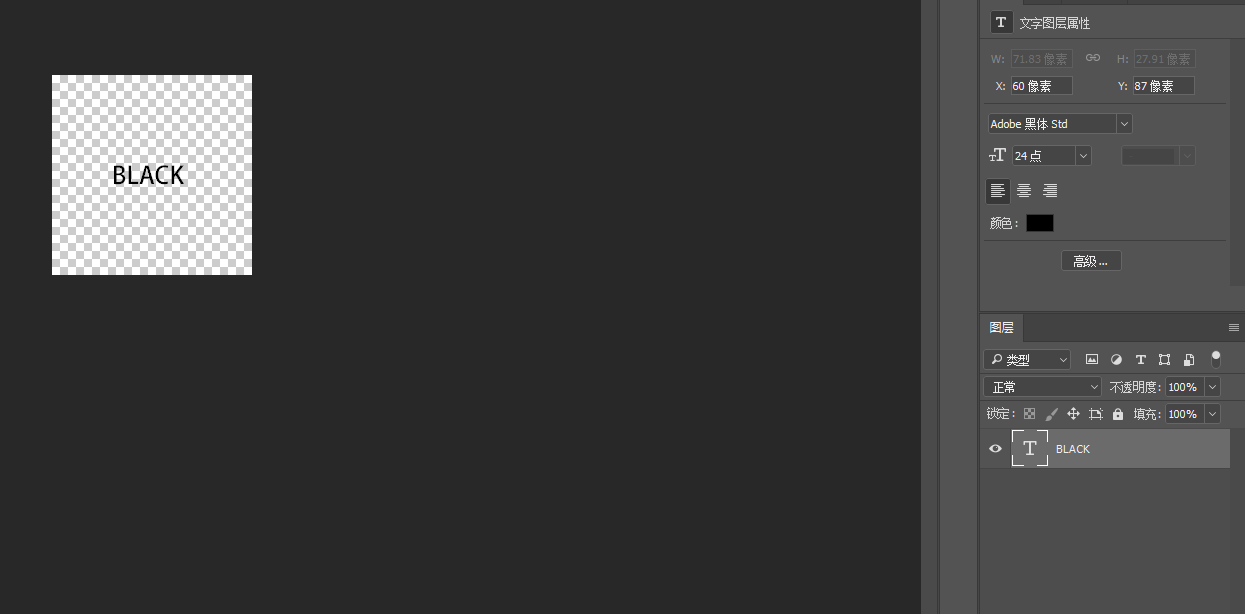
PS添加水印：

1.新建一个200像素的正方形来放置我们需要添加的水印：



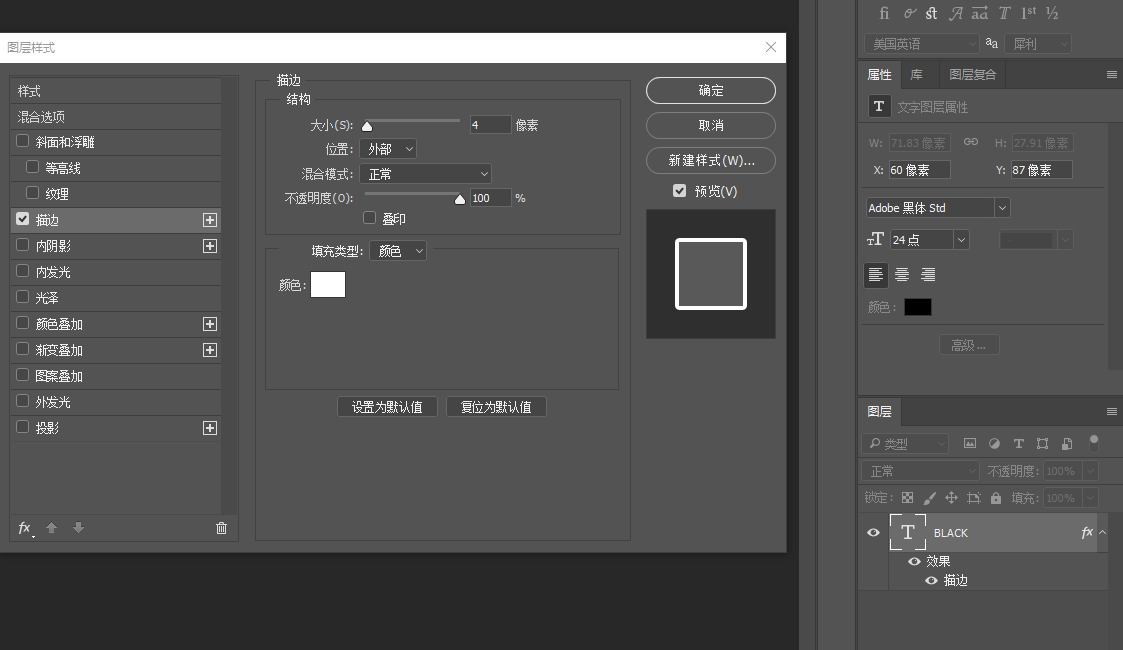
2.先点击T文本工具写入自己想要写入的字体，并可以使用ctrl+t来实现大小旋转的变换。

然后双击背景，把背景变成图层再删除掉（因为我们需要水印覆盖在图片上，不需要白色背景）。



（想象一下，整个宣传图片要被分割成这样200像素大小的方块填充文字，请根据自己的美感调整字体大小角度让图片更美观）

1. 双击文字图层空白处，会出现一个图层样式，我们选择描边：



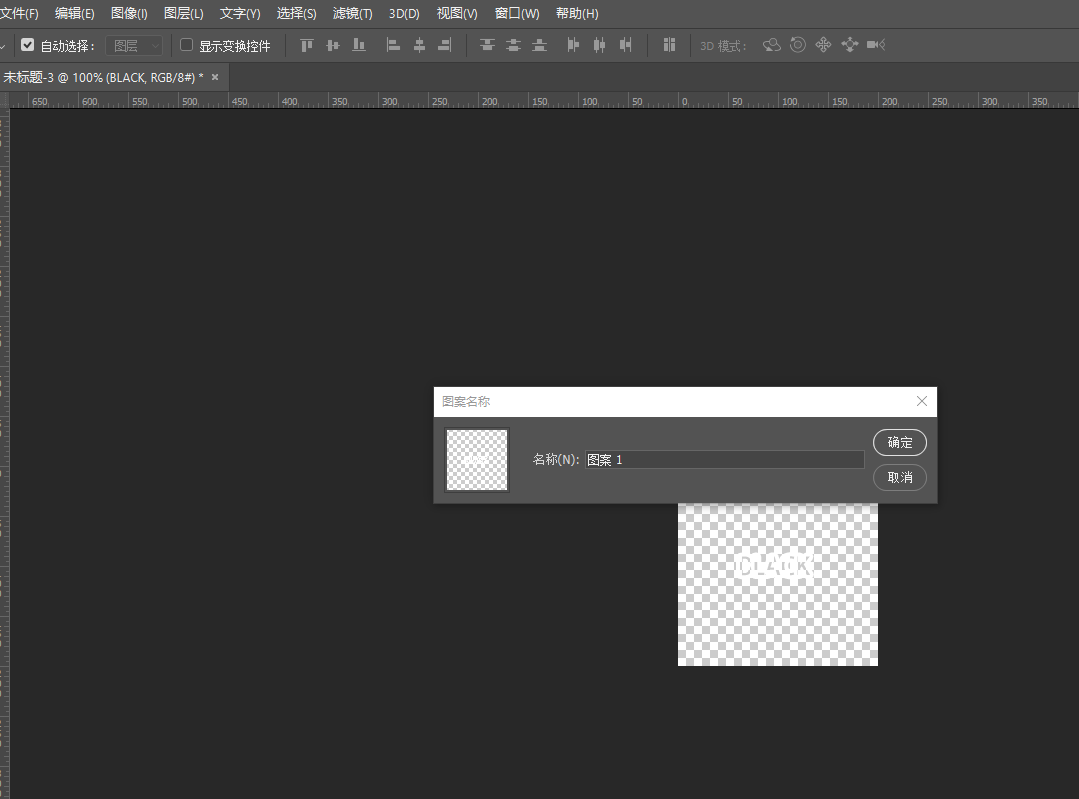
一般选择大小4像素，位置外部，其它的都不需要动，颜色你可以更换。

点击确认，在将文字图层填充调为0，实现水印的效果：



可能有些太显眼了，我们不妨可以把透明度调为60%。

1. 编辑→定义图案→命名→确认：

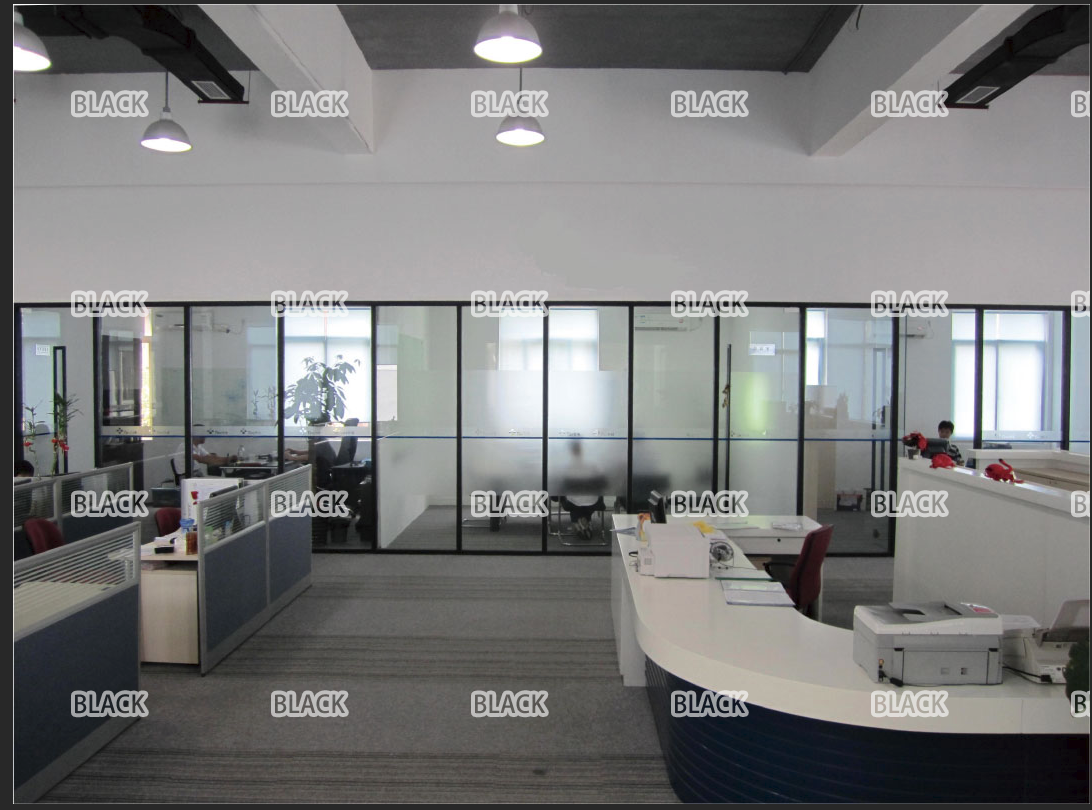


1. 调用它，我们在需要水印的大图片上

编辑→填充(一般你新建的水印在最后一个，选择它，点击确定)：



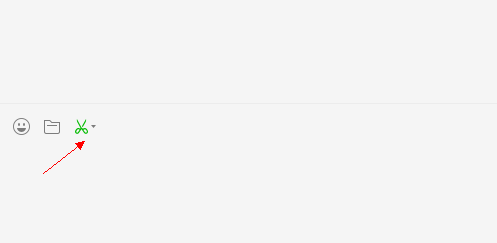
1. 完工：



7.感觉效果不理想，你可以重复3-4-5。

如何截屏：

一般人使用微信或者qq来截屏就可以了，微信截屏步骤：



当然没那么简单，怎么截全屏呢，怎么截那些鼠标移上去才显示的内容呢？

，这时我们打开微信的设置，发现热键：



也就是说我们按alt+A就可以完成截屏了。

qq也是同理，在设置里qq的截屏时shift+alt+A，这可能会造成冲突，一般想使用的话还是在qq设置里把截屏热键改为alt+A好。

前端

浏览器常见错误：

3XX 重定向

4XX 客户端错误

5XX 服务器错误

浏览器内核：

一、Trident内核代表产品Internet Explorer，又称其为IE内核。Trident（又称为MSHTML），是微软开发的一种排版引擎。使用Trident渲染引擎的浏览器包括：IE、傲游、世界之窗浏览器、Avant、腾讯TT、Netscape 8、NetCaptor、Sleipnir、GOSURF、GreenBrowser和KKman等。

二、Gecko内核代表作品Mozilla FirefoxGecko是一套开放源代码的、以C++编写的网页排版引擎。Gecko是最流行的排版引擎之一，仅次于Trident。使用它的最著名浏览器有Firefox、Netscape6至9。

三、WebKit内核代表作品Safari、Chromewebkit 是一个开源项目，包含了来自KDE项目和苹果公司的一些组件，主要用于Mac OS系统，它的特点在于源码结构清晰、渲染速度极快。缺点是对网页代码的兼容性不高，导致一些编写不标准的网页无法正常显示。主要代表作品有Safari和Google的浏览器Chrome。

4

1. Presto内核代表作品OperaPresto是由Opera Software开发的浏览器排版引擎，供Opera 7.0及以上使用。它取代了旧版Opera 4至6版本使用的Elektra排版引擎，包括加入动态功能，例如网页或其部分可随着DOM及Script语法的事件而重新排版。

实现点击产生位移的效果：

.button:active { -webkit-transform: translateY(1px); -moz-transform: translateY(1px); }

搜索框隐藏样式：

appearance:none;

-moz-appearance:none; /\* Firefox \*/

-webkit-appearance:none; /\* Safari 和 Chrome \*/

页面如何播放视频：

一句话：

<video width="640" height="480" controls="controls" src="">ben</video>

css单位;

% 百分比

in 英寸

cm 厘米

mm 毫米

em

1em 等于当前的字体尺寸。

2em 等于当前字体尺寸的两倍。

例如，如果某元素以 12pt 显示，那么 2em 是24pt。

在 CSS 中，em 是非常有用的单位，因为它可以自动适应用户所使用的字体。

ex 一个 ex 是一个字体的 x-height。 (x-height 通常是字体尺寸的一半。)

pt 磅 (1 pt 等于 1/72 英寸)

pc 12 点活字 (1 pc 等于 12 点)

px 像素 (计算机屏幕上的一个点)

vw：（取值1-100），理解全屏幕宽为100vw，自适应屏幕。

Vh：（取值1-100），理解全屏幕高为100vh，自适应屏幕。

BOM是浏览器对象模型，用来获取或设置浏览器的属性，行为。例如：新建窗口、获取屏幕分辨率、浏览器版本号等。

DOM是文档对象模型，用来获取或设置文档中标签的属性，例如获取或者设置input表单的value值。

Js实现表单验证：

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>用户注册</title>

<style type="text/css">

input:focus,textarea:focus{

border:1px solid #f00;

background:#fcc;

}

textarea{

width:230px;

height:50px;

}

body

{

font-size:15px;

/\* 字体的样式 \*/

font-family:Microsoft YaHei;

}

select option{

font-size:10px;

font-family:Microsoft YaHei;

}

</style>

<script type="text/javascript">

//及时验证用户名

function checkuse(){

//在每个函数中定义check变量是为了在表单提交后，能够逐个验证每个函数是否通过，很好很好。（以下同理）

var check;

var username = document.getElementById("username").value;

if (username.length > 18 || username.length < 6) {

alert("用户名输入不合法，请重新输入！");

//此处甚妙，既然你在此处输入错误，那么按理说当然要在此处继续输入了。（在此处继续获取焦点！）

document.getElementById("username").focus();

check = false;

} else {

document.getElementById("checktext1").innerHTML = "\* 用户名由6-18位字符组成 √";

check = true;

}

return check;

}

//利用正则表达式判断密码符合否

function checkpwd() {

var check;

var reg = /[^A-Za-z0-9\_]+/;

var regs = /^[a-zA-Z0-9\_\u4e00-\u9fa5] + $ /;

var password = document.getElementById("password").value;

if (password.length < 6 || password.length > 18 || regs.test(password)) {

alert("密码输入不合法，请重新输入！");

document.getElementById("password").focus();

check = false;

} else {

document.getElementById("checktext2").innerHTML = "\* 密码由6-18位字符组成，且必须包含字母、数字和标点符号 √";

check = true;

}

return check;

}

//验证密码是否不一致！

function checkpwdc() {

var check;

var password = document.getElementById("password").value;

var pwdc = document.getElementById("pwdc").value;

if (password != pwdc) {

alert("两次输入密码不一致，请重新输入！");

document.getElementById("pwdc").focus();

check = false;

} else {

document.getElementById("checktext3").innerHTML = "\* 请再次输入你的密码 √";

check = true;

}

return check;

}

//提交时验证用户类别

function checkut(){

var check;

if(document.getElementById("selUser").selectedIndex == 0)

{

alert("请选择用户类型！");

document.getElementById("selUser").focus();

check = false;

}else{

document.getElementById("checktext4").innerHTML = "\* 请选择用户类型 √";

check = true;

}

return check;

}

//提交时验证用户性别

function checkGender(){

var check;

var gender = "";

//获取所有名称为sex的标签

var sex = document.getElementsByName("sex");

//遍历这些名称为sex的标签

for(var i=0;i<sex.length;++i){

//如果某个sex被选中，则记录

if(sex[i].checked)

gender = sex[i].value;

}

if(gender == "")

{

alert("请选择性别！");

check = false;

}else{

document.getElementById("checktext5").innerHTML = "\* 请选择你的性别 √";

check = true;

}

return check;

}

//及时验证出生日期

function checkDate(){

var check;

if(document.getElementById("txtDate").value ==""){

alert("请填写出生日期！");

document.getElementById("txtDate").focus();

check = false;

}else{

document.getElementById("checktext6").innerHTML = "\* 请选择你的出生日期 √";

check = true;

}

return check;

}

//及时验证兴趣爱好

function checkHobby(){

var check;

var hobby = 0;

//objNum为所有名称为hobby的input标签

var objNum = document.getElementsByName("hobby");

//遍历所有hobby标签

for(var i=0;i<objNum.length;++i){

//判断某个hobby标签是否被选中

if(objNum[i].checked==true)

hobby++;

}

//如果有选中的hobby标签

if(hobby >=1){

document.getElementById("checktext7").innerHTML = "\* 请选择你的兴趣爱好 √";

check = true;

}else{

alert("请填写爱好！");

check = false;

}

return check;

}

//正则表达式验证电子邮件（及时）

function checkemail(){

var check;

//电子邮件的正则表达式

var e1 = document.getElementById("email").value.indexOf("@",0);

var e2 = document.getElementById("email").value.indexOf(".",0);

if(email == "" || (e1==-1 || e2==-1) || e2<e1 )

{

alert("E\_mail输入错误!");

document.getElementById("email").focus();

check = false;

} else {

document.getElementById("checktext8").innerHTML = "\* 请填写常用的EMAIL，将用于密码找回 √";

check = true;

}

return check;

}

//及时验证自我介绍

function checkintro(){

var check;

var intro = document.getElementById("introduction").value;

if (intro.length > 100) {

alert("字数超限！");

check = false;

} else {

document.getElementById("checktext9").innerHTML = "\* 限100字内 √";

document.getElementById("checktext9").focus();

check = true;

}

return check;

}

//提交表单时所有都验证一遍(若任何一个验证不通过，则返回为false，阻止表单提交)

function check() {

var check = checkuse() && checkpwd() && checkpwdc() && checkut() && checkGender() && checkDate() && checkHobby()

&& checkemail() &&checkintro();

return check;

}

</script>

</head>

<body >

<!-- <form action ="跳转页面" method ="get"|"post" name ="表单名称" target ="打开方式" enctype="multipart/form-data" > -->

<!-- onsubmit()函数在返回值为true时提交表单。 -->

<form action="#" method="get" onsubmit="return check()" >

<fieldset>

<legend>

表单及时验证小例子

</legend>

<table align="left" style="background-image: url('img/4.jpg');" >

<tr>

<td>用户名:</td>

<td><input type="text" name="username" id="username" onchange=" checkuse()" /></td>

<td id="checktext1">\* 用户名由6-18位字符组成</td>

</tr>

<!-- onblur 事件处理程序:当元素或窗口失去焦点时触发该事件 -->

<!-- onchange事件处理程序:当表单元素获取焦点，并且内容发生改变时，触发该事件 -->

<!-- 以下同理 -->

<tr>

<td>密码:</td>

<td><input type="password" name="password" id="password" onchange="checkpwd()" /></td>

<td id="checktext2">\* 密码由6-18位字符组成，且必须包含字母、数字和标点符号</td>

</tr>

<tr>

<td>确认密码：</td>

<td><input type="password" name="pwdc" id="pwdc" onchange="checkpwdc()" /></td>

<td id="checktext3">\* 请再次输入你的密码</td>

</tr>

<tr>

<td>用户类型：</td>

<td>

<select id="selUser" onblur="checkut()">

<option name="selUser" value="0">请选择</option>

<option name="selUser" value="1">管理员</option>

<option name="selUser" value="2">普通用户</option>

</select>

</td>

<td id="checktext4">\* 请选择用户类型</td>

</tr>

<tr>

<td>性别：</td>

<td>

<input type="radio" value="1" name="sex" onchange="checkGender()"/>男

<input type="radio" value="2" name="sex" onchange="checkGender()"/>女

</td>

<td id="checktext5">\* 请选择你的性别</td>

</tr>

<tr>

<td>出生日期：</td>

<td><input type="date" name="date" id="txtDate" onblur="checkDate()"/></td>

<td id="checktext6">\* 请选择你的出生日期</td>

</tr>

<tr>

<td>兴趣爱好：</td>

<td>

<input type="checkbox" name="hobby" value="reading" onchange="checkHobby()">阅读

<input type="checkbox" name="hobby" value="music" onchange="checkHobby()">音乐

<input type="checkbox" name="hobby" value="sports" onchange="checkHobby()">运动

</td>

<td id="checktext7">\* 请选择你的兴趣爱好</td>

</tr>

<tr>

<td>电子邮件：</td>

<td><input type="text" name="email" id="email" onchange="checkemail()"/></td>

<td id="checktext8">\* 请填写常用的EMAIL，将用于密码找回</td>

</tr>

<tr>

<td>自我介绍：</td>

<td><textarea cols="30" rows="3" name="introduction" id="introduction" onchange="checkintro()">这是自我介绍...</textarea></td>

<td id="checktext9">\* 限100字内</td>

</tr>

<tr>

<td colspan="2" align="center">

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

<input type="reset" name="reset" value="重置" />

</td>

</tr>

</table>

</fieldset>

</form>

</body>

</html>

Js打印页面属性：

console.log("网页可见区域宽:"+document.body.clientWidth);

console.log("网页可见区域高:"+document.body.clientHeight);

console.log("网页可见区域宽(包括边线的宽):"+document.body.offsetWidth);

console.log("网页可见区域高(包括边线的宽):"+document.body.offsetHeight);

console.log("网页正文全文宽:"+document.body.scrollWidth);

console.log("网页正文全文高:"+document.body.scrollHeight);

console.log("网页被卷去的高:"+document.body.scrollTop);

console.log("网页被卷去的左:"+document.body.scrollLeft);

console.log("网页正文部分上:"+window.screenTop);

console.log("网页正文部分左:"+window.screenLeft);

console.log("屏幕分辨率的高:"+window.screen.height);

console.log("屏幕分辨率的宽:"+window.screen.width);

console.log("屏幕可用工作区高度:"+window.screen.availHeight);

console.log("屏幕可用工作区宽度:"+window.screen.availWidth);

Bootstrap有很好的借鉴意义。



后端

= == ===：

= 是赋值：就是说给一个变量赋值

== 是轻量级的比较运算，只看值不看类型

=== 是重量级的比较运算，既看值，也看类型，要绝对相等才会为true

Mysql使用命令：

mysql -uroot -p

root

1.首先显示数据库 show databases；//建立数据库 create database (databases\_name);

2.进入数据库use (databases\_name)；

3.显示数据库里的所有表show tables；

4.显示数据表内容select \* from (tables\_name)；

5.建立表结构;

create table MyClass(

id int(4) not null primary key auto\_increment,

name char(20) not null,

sex int(4) not null default '0',

degree double(16,2));

1. 删除表结构：

drop table MyClass;

\*查看表结构：desc (tables\_name)；

1. 查询数据内容：

select \* from MyClass;

1. 修改其中一条数据：

update MyClass set name='Mary' where id=1;

1. 删除其中一条数据：

delete from myclass where id=1;

入侵