

***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 1](#_Toc516917639)

[打印机如何复印身份证复印件 1](#_Toc1671808306)

[电脑常识 2](#_Toc1983488030)

[电脑获得最高管理员权限 2](#_Toc1945319202)

[查看本机服务 2](#_Toc2033254777)

[命令提示符cd到桌面： 2](#_Toc1408535345)

[快捷键基本常识 3](#_Toc969761899)

[修改本地虚拟域名 3](#_Toc1288480808)

[UI 5](#_Toc1734954731)

[PS添加水印 6](#_Toc1788268987)

[如何截屏 10](#_Toc1755393257)

[前端 11](#_Toc201315137)

[浏览器常见错误 11](#_Toc1697741175)

[浏览器内核 12](#_Toc173932771)

[实现点击产生位移的效果 12](#_Toc2112691889)

[搜索框隐藏样式 13](#_Toc1780691842)

[页面如何播放视频 13](#_Toc2136508962)

[手机查看电脑本地环境网站 13](#_Toc1864421387)

[css单位 13](#_Toc1301712244)

[BOM和DOM的区别 14](#_Toc173018309)

[CSS 用 position: absolute 与 transform 来居中块级元素的问题 15](#_Toc1318480303)

[Js实现表单验证 16](#_Toc1050674803)

[Js打印页面属性 26](#_Toc145508210)

[jquery 之 $().hover(func1, funct2)使用方法 26](#_Toc1156344871)

[Bootstrap的自适应 27](#_Toc117687412)

[后端 27](#_Toc537471482)

[= == ===的区别 27](#_Toc762726274)

[Mysql使用命令 28](#_Toc766225199)

[Linux下文件属性 29](#_Toc822640521)

[万网的虚拟主机本地目录 30](#_Toc899464812)

[入侵 32](#_Toc1930715037)

[安装kali后要做的事 32](#_Toc1339558160)

[更新源 32](#_Toc423789470)

[增加中文输入法 34](#_Toc1766719419)

[安装WPS for Linux 40](#_Toc1137393715)

[安装unrar 41](#_Toc309560599)

[安装chrome浏览器 41](#_Toc1027771116)

[无法锁定管理目录(/var/lib/dpkg/) 45](#_Toc2107155614)

[Github 上Everything up-to-data 错误 45](#_Toc1598041408)

[Linux下本地php环境 47](#_Toc615242199)

[linux压缩文件夹下所有文件： 48](#_Toc1747940953)

[Atom安装插件 48](#_Toc1205951017)

[Linux文件目录操作的12个常用命令 48](#_Toc816557337)

[Linux下链接服务器上传内容 67](#_Toc1298198481)

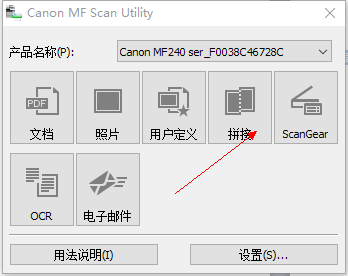
[Linux下截屏： 69](#_Toc1379883788)

# 

# 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，以佳能MF240为例：
2. 选择拼接：
3. 把身份证都统一放在右侧（左侧）打印，最后调整图像大小和位置即可。

# 电脑常识

电脑获得最高管理员权限：

命令提示符输入

net user administrator /active:yes

查看本机服务：

运行 -> services.msc

## 命令提示符cd到桌面：

命令提示符输入

cd /d %userprofile%\desktop

快捷键基本常识：

ctrl+a 全选

ctrl+c 复制

Ctrl+v 粘贴

ctrl+z是回退

ctrl+/ 是注释你选择的内容，只需要用鼠标选择你想要注释的内容 ctrl+/

局域网查看本机网站页面：

首先查出本机的局域网ip地址，Linux下输入ifconfig，windos下输入ipconfig就会出现：

其中类似192.168.31.104之类的就是你的局域网域名

用手机或者电脑浏览器登录就可以看到本机开启的网站。

修改本地虚拟域名：

C:\Windows\System32\drivers\etc 下的hosts文件

# 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。

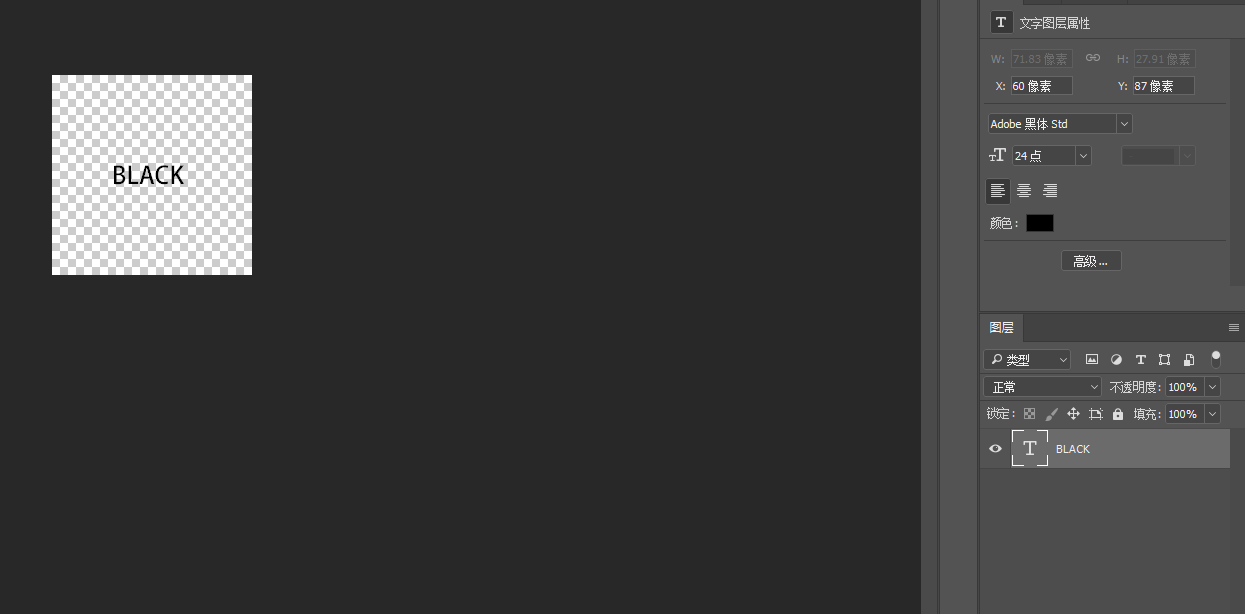
PS添加水印：

1.新建一个200像素的正方形来放置我们需要添加的水印：



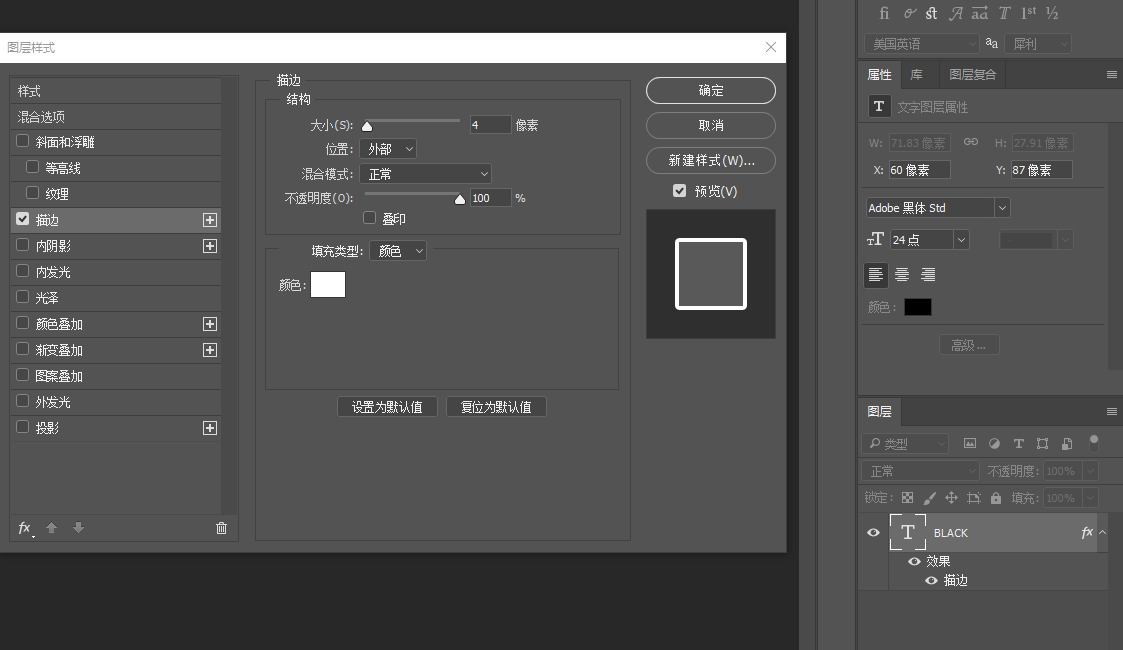
2.先点击T文本工具写入自己想要写入的字体，并可以使用ctrl+t来实现大小旋转的变换。

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：



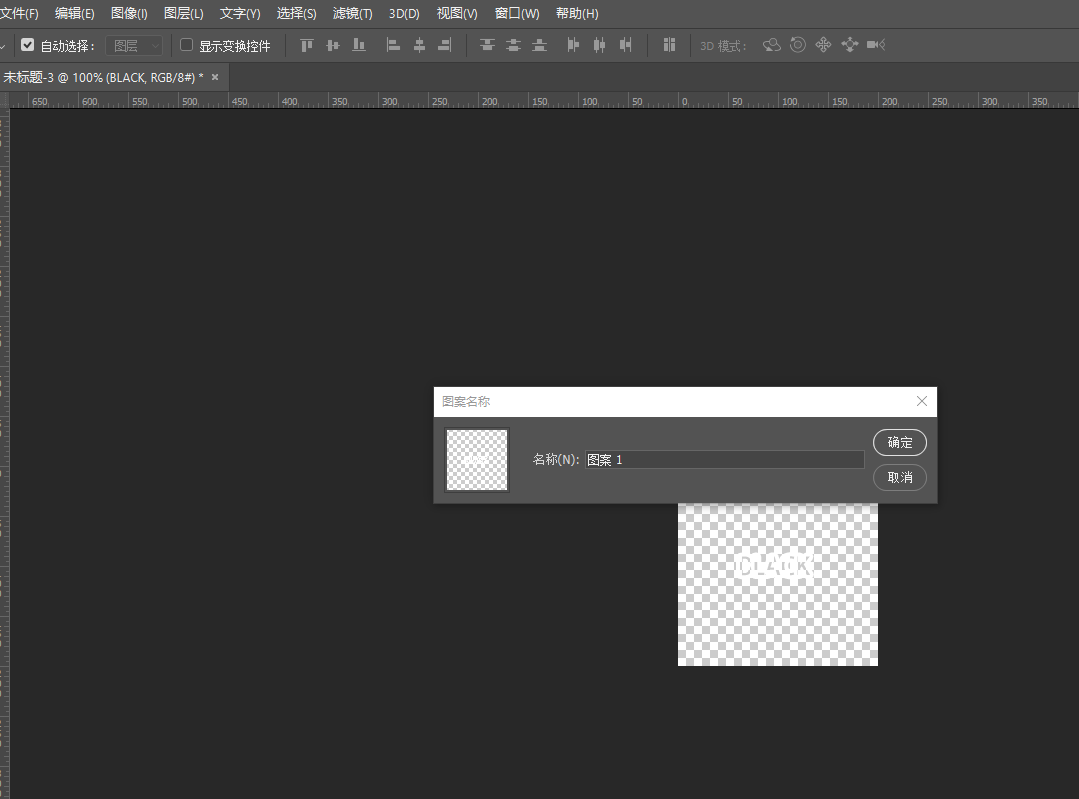
一般选择大小4像素，位置外部，其它的都不需要动，颜色你可以更换。

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：

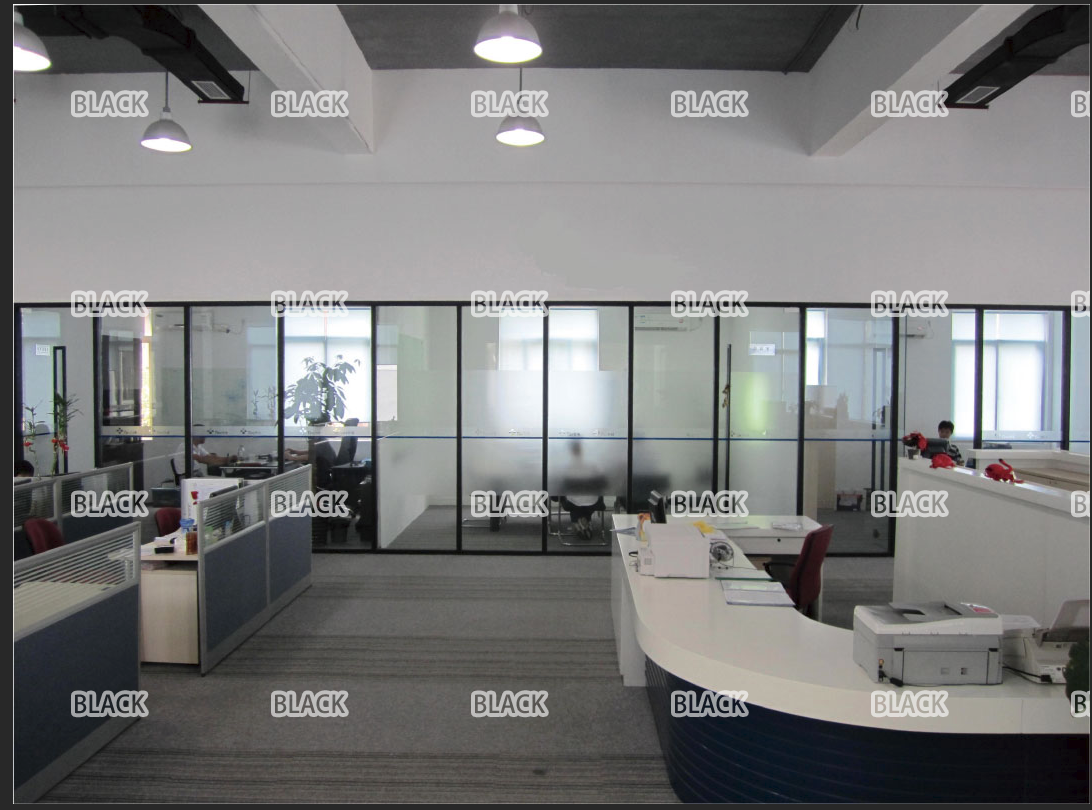


1. 调用它，我们在需要水印的大图片上

编辑→填充(一般你新建的水印在最后一个，选择它，点击确定)：



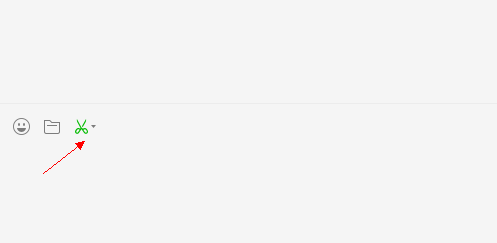
1. 完工：



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>

手机查看电脑本地环境网站：

1. 在电脑端的命令提示符中输入ipconfig就可以知道局域网的电脑ip地址
2. 在手机浏览器输入ip地址即可访问（例如：192.168.31.104）

注：(如果index不在www目录下，类似：192.168.31.104/upload即可)

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和DOM的区别：

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。

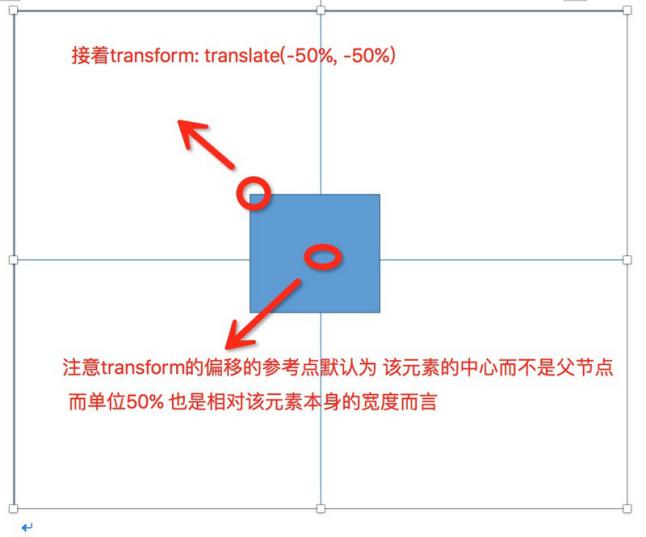
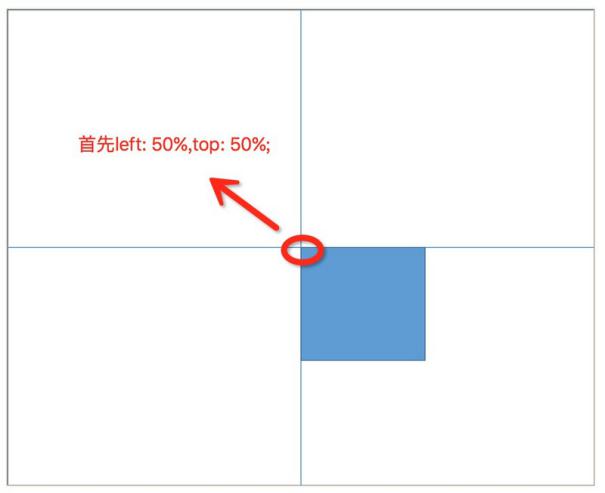
## CSS 用 position: absolute 与 transform 来居中块级元素的问题

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);



Js之debugger妙用：

测试js时加入debugger；当触发事件时，js暂停，我们就可以很清楚的看到函数是什么值，把鼠标移动到函数值时，便会显示具体值，对测试很有帮助。

网站添加ico图标：

<link rel="shortcut icon" href="/favicon.ico" />

<link rel="bookmark" href="/favicon.ico" type="image/x-icon"　/>

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);

jquery 之 $().hover(func1, funct2)使用方法：

.hover(func1, func2) 的效果等效于： mouseenter(), mouseleave()

Bootstrap的自适应:



# 后端

= == ===的区别：

= 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：

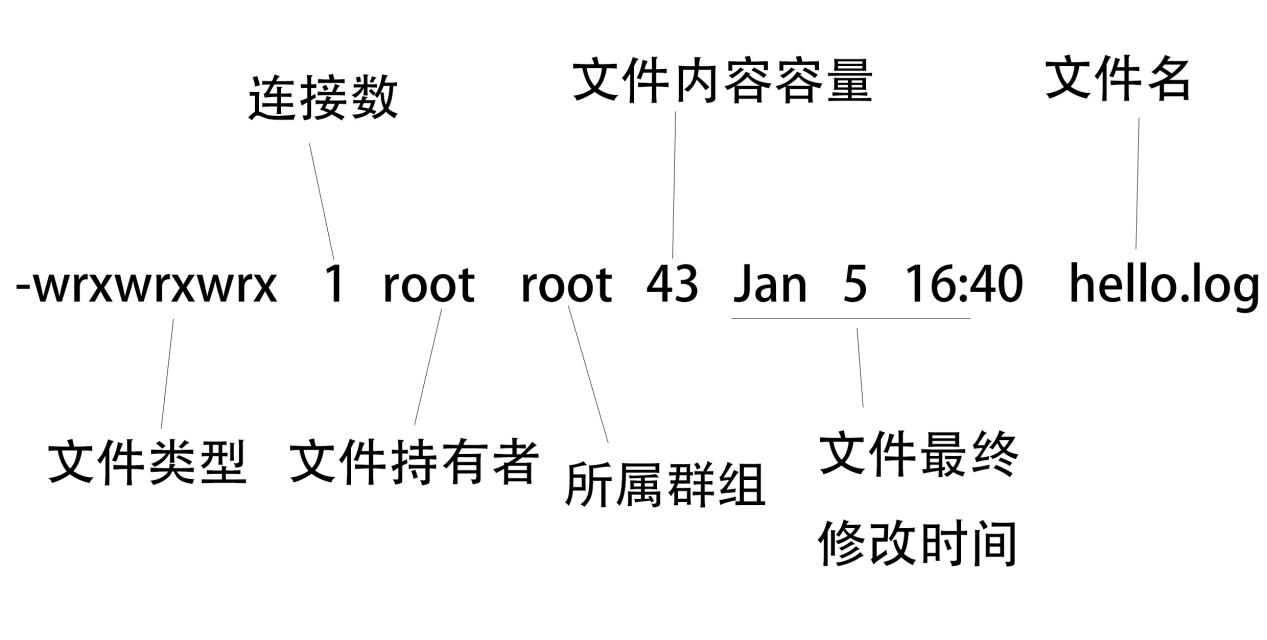
update MyClass set name='Mary' where id=1;

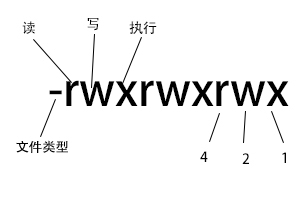
1. 删除其中一条数据：

delete from myclass where id=1;

Linux下文件属性：

以root的身份登入linux之后，在终端输入ls -al 就可以看到当前目录下的文件属性：





第一个字符代表这个文件的类型(如目录、文件或链接文件等等)：

当为[ d ]则是目录；

当为[ - ]则是文件；

若是[ l ]则表示为连结档(link file)；

若是[ b ]则表示为装置文件里面的可供储存的接口设备(可随机存取装置)；

若是[ c ]则表示为装置文件里面的串行端口设备,例如键盘、鼠标(一次性读取装置)

接下来的字符中,以三个为一组,且均为『rwx』 的三个参数的组合

< [ r ]代表可读(read)、[ w ]代表可写(write)、[ x ]代表可执行(execute) 要注意的是,这三个权限的位置不会改变,如果没有权限,就会出现减号[ - ]而已>

第一组为『文件拥有者的权限』,以『install.log』那个文件为例, 该文件的拥有者可以读写,但不可执行；

第二组为『同群组的权限』；

第三组为『其他非本群组的权限』.

注：(修改文件属性需要使用管理员权限，例：sudo chmod 777 filename)

万网的虚拟主机本地目录：

本路径下各文件夹功能和权限说明：

/ : 您的FTP的根目录，为避免您错误删除该路径下的内容，所以禁止写入，只可读取

htdocs/: 您的网站的根目录，请将您的网站文件(php, htm, html, css, js等)上传至此文件夹

htdocs/logreport/: 访问统计报告存放目录，不占用空间，只可读取。

myfolder/: 此文件夹中的内容不会被web用户看到，您可以临时存放一些不愿公开的内容，比如站点备份、数据库备份等

ftplogs/: 您的ftp访问日志存放位置，只读

wwwlogs/: 您的www访问日志存放位置，只读

backup/: 系统自动为您生成的数据库备份的存放位置，只读

# 入侵

安装kali后要做的事：

### 更新源

安装kali linux后第一件事是修改更新源sources.list，提高软件下载安装速度：

1.切换到root用户（如果已经是root用户就直接看第二步）

dnt@HackerKali:~$ su

密码：

2.用文本编辑器打开sources.list,手动添加下面的更新源

root@HackerKali:/home/dnt# leafpad /etc/apt/sources.list

#中科大

deb http://mirrors.ustc.edu.cn/kali kali-rolling main non-free contrib

deb-src http://mirrors.ustc.edu.cn/kali kali-rolling main non-free contrib

#阿里云

#deb http://mirrors.aliyun.com/kali kali-rolling main non-free contrib

#deb-src http://mirrors.aliyun.com/kali kali-rolling main non-free contrib

#清华大学

#deb http://mirrors.tuna.tsinghua.edu.cn/kali kali-rolling main contrib non-free

#deb-src https://mirrors.tuna.tsinghua.edu.cn/kali kali-rolling main contrib non-free

#浙大

#deb http://mirrors.zju.edu.cn/kali kali-rolling main contrib non-free

#deb-src http://mirrors.zju.edu.cn/kali kali-rolling main contrib non-free

#东软大学

#deb http://mirrors.neusoft.edu.cn/kali kali-rolling/main non-free contrib

#deb-src http://mirrors.neusoft.edu.cn/kali kali-rolling/main non-free contrib

#官方源

#deb http://http.kali.org/kali kali-rolling main non-free contrib

#deb-src http://http.kali.org/kali kali-rolling main non-free contrib

#重庆大学

#deb http://http.kali.org/kali kali-rolling main non-free contrib

#deb-src http://http.kali.org/kali kali-rolling main non-free contrib

3.添加完毕后执行下面的更新指令，进行系统或者工具的更新

root@HackerKali:/home/dnt# apt-get update && apt-get upgrade && apt-get dist-upgrade

增加中文输入法：

首先apt-get install fcitx

其次搜索sogou for linux 下载64位deb安装包

进入安装包目录

dpkg -i sogou(TAB补全)

安装后如果出错，安装终端提示进行修复，例如：apt --fix-configure -a

重启电脑

开机后屏幕左下角会有移动的括号，点击选择搜狗拼音即可。

1 Linux下Git和GitHub环境的搭建

第一步： 安装Git，使用命令 “sudo apt-get install git”

第二步： 到GitHub上创建GitHub帐号

第三步： 生成ssh key，使用命令 “ssh-keygen -t rsa -C "your\_email@youremail.com"”，your\_email是你的email

第四步： 回到github，进入Account Settings，左边选择SSH Keys，Add SSH Key,title随便填，粘贴key。key就是

~/.ssh/id\_rsa.pub内容(在kali linux上，你需要显示隐藏文件属性，然后在/root/.ssh/id\_rsa.pub中复制内容)。

第五步： 测试ssh key是否成功，使用命令“ssh -T git@github.com”，如果出现You’ve successfully authenticated, but GitHub does not provide shell access 。这就表示已成功连上github。

第六步： 配置Git的配置文件，username和email

git config --global user.name "your name" //配置用户名

git config --global user.email "your email" //配置email

2 利用Git从本地上传到GitHub

第一步： 进入要所要上传文件的目录输入命令 “git init”

第二步： 创建一个本地仓库origin，使用命令 “git remote add origin git@github.com:yourName/yourRepo.git”

youname是你的GitHub的用户名，yourRepo是你要上传到GitHub的仓库，这是你再GitHub上添加的仓库。

第三步： 比如你要添加一个文件xxx到本地仓库，使用命令 “git add xxx”，可以使用“git add .”自动判断添加哪些文件

然后把这个添加提交到本地的仓库，使用命令 ”git commit -m ”说明这次的提交“ “

最后把本地仓库origin提交到远程的GitHub仓库，使用命令 ”git push origin master“

3 从GitHub克隆项目到本地

第一步： 到GitHub的某个仓库，然后复制右边的有个“HTTPS clone url”

第二步： 回到要存放的目录下，使用命令 "git clone https://github.com/chenguolin/scrapy.git"，红色的url只是一个例子

第三步： 如果本地的版本不是最新的，可以使用命令 “git fetch origin”，origin是本地仓库

第四步： 把更新的内容合并到本地分支，可以使用命令 “git merge origin/master”

如果你不想手动去合并，那么你可以使用： git pull <本地仓库> master // 这个命令可以拉去最新版本并自动合并

4 GitHub的分支管理

创建

1 创建一个本地分支： git branch <新分支名字>

2 将本地分支同步到GitHub上面： git push <本地仓库名> <新分支名>

3 切换到新建立的分支： git checkout <新分支名>

4 为你的分支加入一个新的远程端： git remote add <远程端名字> <地址>

5 查看当前仓库有几个分支: git branch

删除

1 从本地删除一个分支： git branch -d <分支名称>

2 同步到GitHub上面删除这个分支： git push <本地仓库名> :

5 常见错误

1 如果出现报错为ERROR: Repository not found.fatal: The remote end hung up unexpectedly则代表你的 origin 的url 链接有误，可能是创建错误，也可能是这个 git@github.com:xxx/new-project.git url 指定不正确。重新创建。

### 安装WPS for Linux

。下载地址：http://community.wps.cn/download/。选择alpha版deb包。

dpkg安装程序安装包

#sudo dpkg -i wps-office\_10.1.0.5672~a21\_amd64.deb

提示wps-office 依赖于 libpng12-0；然而：未安装软件包 libpng12-0。该软件包已丢失可用ttf-mscorefonts-installer代替。

要先运行下面的命令将已安装的wps-office卸载：

#apt-get -f install

之后libpng12-0依赖关系还是出错，到下面的地址下载libpng12-0然后安装：

https://packages.debian.org/zh-cn/wheezy/amd64/libpng12-0/download。

#sudo dpkg -i libpng12-0\_1.2.49-1+deb7u2\_amd64.deb

安装好后到Dash中搜索WPS启动并锁定到启动器，会报错显示缺失字体：

Symbol，Wingdings，Wingdings2，Wingdings3，MT-Extra

进入home文件夹，按Ctrl+h显示所有文件夹，查看有没有.fonts文件夹，如果没有就创建一个，然后到网上下载相关字体放入该文件夹中

### 安装unrar

sudo apt-get install rar unrar

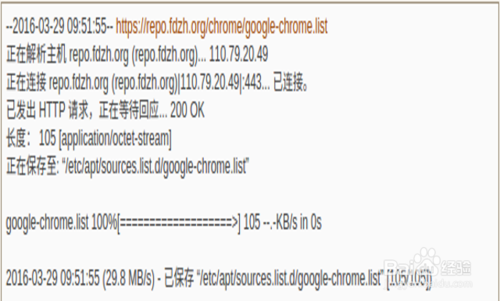
### 安装chrome浏览器

· 在终端中，输入以下命令：

·         sudo wget https://repo.fdzh.org/chrome/google-chrome.list -P /etc/apt/sources.list.d/

· 将下载源加入到系统的源列表。命令的反馈结果如图。

· 如果返回“地址解析错误”等信息，可以百度搜索其他提供 Chrome 下载的源，用其地址替换掉命令中的地址。

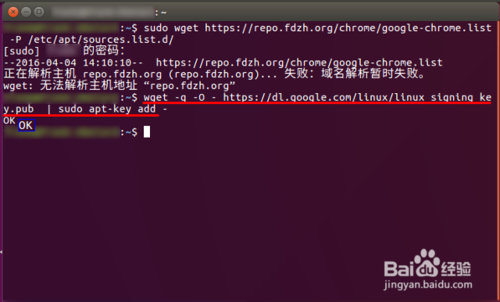
· [](http://jingyan.baidu.com/album/335530da98061b19cb41c31d.html?picindex=2)

· · 在终端中，输入以下命令：

·         wget -q -O - https://dl.google.com/linux/linux\_signing\_key.pub  | sudo apt-key add -

· 导入谷歌软件的公钥，用于下面步骤中对下载软件进行验证。

· 如果顺利的话，命令将返回“OK”，如图。

· [](http://jingyan.baidu.com/album/335530da98061b19cb41c31d.html?picindex=3)

· · 在终端中，输入以下命令：

·         sudo apt-get update

· 用于对当前系统的可用更新列表进行更新。这也是许多 Linux 发行版经常需要执行的操作，目的是随时获得最新的软件版本信息。

· 命令将会返回类似图中所示的信息。

· [](http://jingyan.baidu.com/album/335530da98061b19cb41c31d.html?picindex=4)

· · 在终端中，输入以下命令：

·         sudo apt-get install google-chrome-stable

· 执行对谷歌 Chrome 浏览器（稳定版）的安装。

· 如果一切顺利的话，命令将返回如图信息。

· [](http://jingyan.baidu.com/album/335530da98061b19cb41c31d.html?picindex=5)

· · 最后，如果一切顺利，在终端中执行以下命令：

·         /usr/bin/google-chrome-stable

· 将会启动谷歌 Chrome 浏览器，它的图标将会出现在屏幕左侧的 Launcher 上，在图标上右键——“锁定到启动器”，以后就可以简单地单击启动了。

· 解决方法：

用文件资源管理器按照这个路径打开  /usr/share/applications/  如图所示，找到 google chrome 图标，鼠标右键打开属性，在命令哪行 /usr/bin/google-chrome-stable %U 的后面再加上这个语句

- -no-sandbox

### 无法锁定管理目录(/var/lib/dpkg/)

E: 无法获得锁 /var/lib/dpkg/lock - open (11: Resource temporarily unavailable)  
E: 无法锁定管理目录(/var/lib/dpkg/)，是否有其他进程正占用它？  
  
解 决方法一：  
#:ps -aux (列出进程，形式如)  
root 5765 0.0 1.0 18204 15504 ? SN 04:02 0:00 apt-get -qq -d  
找到最后一列以apt-get 开头的进程  
#:sudo kill -9 该进程的PID  
  
解决方法二：  
#:sudo rm /var/cache/apt/archives/lock  
#:sudo rm /var/lib/dpkg/lock

### Github 上Everything up-to-data 错误

接下来先创建一个新分支提交改动  
  
  
$ git branch newbranch  
然后输入这条命令检查是否创建成功  
  
  
$ git branch  
这时终端输出  
  
  
  newbranch  
\* master  
这样就创建成功了，前面的\*代表的是当前你所在的工作分支。我们接下来就要切换工作分支。  
  
  
$ git checkout newbranch  
这样就切换完了，可以 $ git branch 确认下。然后你要将你的改动提交到新的分支上。  
  
  
$ git add .  
$ git commit -a  
此时可以 $ git status 检查下提交情况。如果提交成功，我们接下来就要回主分支了，代码和之前一样。  
  
  
$ git checkout master  
然后我们要将新分支提交的改动合并到主分支上  
  
  
$ git merge newbranch  
合并分支可能产生冲突这是正常的，虽然我们这是新建的分支不会产生冲突，但还是在这里记录下。下面的代码可以查看产生冲突的文件，然后做对应的修改再提交一次就可以了。  
  
  
$ git diff  
我们的问题就解决了，接下来就可以push代码了。  
  
  
$ git push -u origin master  
新建分支的朋友别忘了删除这个分支  
  
  
$ git branch -D newbranch

注：我的习惯动作（在当前仓库中）：

1. git add .
2. git commit -m 更新了文件修改
3. git push

### Linux下本地php环境

1.首先安装Mysql

sudo apt-get install mysql-server mysql-client

2.安装apache2

sudo apt-get install apache2

3.解析php

sudo apt-get install libapache2-mod-php7.0

## linux压缩文件夹下所有文件：

zip -r test.zip ./\*

Atom安装插件：

[autocomplete-paths](https://atom.io/packages/autocomplete-paths" \t "/root/文档\\x/_blank) 自动补全引用js/img/css路径

File-icons 图标，让文件容易区分

## **[Linux文件目录操作的12个常用命令](http://www.cnblogs.com/yolanda-lee/p/4544573.html)**

1， mkdir命令

mkdir用来创建指定名称的目录，要求创建目录的用户在当前目录中具有写权限，并且指定的目录名不能是当前目录中已有的目录。

（1）     命令格式：mkdir [选项] 目录

（2）     命令功能：在指定位置创建指定文件名命名的文件夹或目录，要创建文件夹或目录的用户必须对所创建的文件夹的父文件夹具有写权限，且同一个目录下不能有重名的。

（3）     命令参数：

-m,--mode模式 设定权限<模式>(类似chmod)，而不是rwxrwxrwx

-p,--parents可以是一个路径的名称，此时若路径中的某些目录不存在，加上此选项后系统将自动建立好那些尚不存在的目录，即一次可以建立多个目录

-v,--verbose创建当前新目录或文件时显示信息，如“已创建目录test6”

（4）     命令实例：

①创建一个空目录

             mkdir lijl\_bj1

②递归创建多个目录

             mkdir –p ./lijl\_bj1/test(./表示当前目录，../表示上级目录，/表示根目录)

③创建权限为777的目录

             mkdir –m 777 test777

④创建新目录显示信息

             mkdir –v test

⑤创建项目的目录结构

mkdir –vp

src/{com/sitech/{admin/{entity/,dao/,service,controller},common/util/},config/}

（5）     注意事项：

最后一个实例⑤对顺序是有要求的，即在-vp时才表示-v –p操作，而若是-pv则在上述语句中会默认为创建-vp目录。

2， ls命令

缺省下用来打印当前目录的清单，如果ls指定其他目录，那么就会显示指定目录的文件及文件夹清单。通过ls命令不仅可以查看linux文件夹包含的文件，而且可以查看文件权限（包括目录、文件夹、文件）、查看目录信息等。

（1）     命令格式：ls [选项] [目录名]

（2）     命令功能：列出目标目录中所有的子目录和文件

（3）     命令参数：

-a,列出目录下的所有文件，包括以.开头的隐含文件

-c 配合-lt根据ctime排序及根据ctime（文件状态最后更改的时间）

  配合-l 显示ctime但根据名称排序

-l 除了文件名之外，还将文件的权限、所有者、文件大小等信息详细列出来

-t 以文件的修改顺序排序

-u 配合-lt显示访问时间，而且依据访问时间排序

     配合-l 显示访问时间，但根据名称排序

-v 依据版本进行排序

-x 逐行列出项目，而不是逐栏列出

-X 根据扩展名进行排序

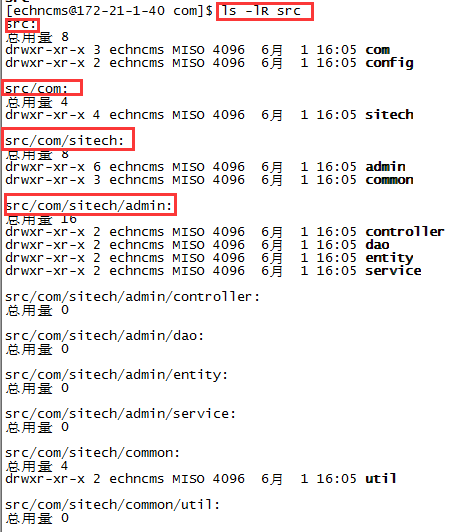
-1 每行只列出一个文件

-F 可以在显示子目录的时候在它的文件名之后加一个“/”，而文件后面的“\*”表示这是一个可执行程序

（4）     常用实例：

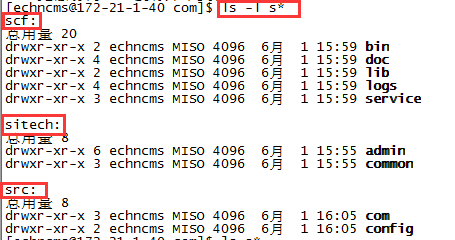
①列出 /lijl\_bj1/com/src文件夹下的所有文件和目录的详细情况

进入lijl\_bj1目录，ls –l –R(或-lR) src



②列出当前目录中所有以“s”开头的所有文件目录的信息，

           ls –l s\*



③只列出文件下的子目录：

                             ls -F ./src |grep /$ 列出src目录下面的子目录

                                   结果：com/,config/

④列出目前工作目录下所有档案及目录;目录于名称后加"/", 可执行档案名称后加"\*"

           ls –AF

3， cd命令

cd命令是Linux里面最基本的命令，很多其他命令都是建立在它的基础之上的

（1）     命令格式：cd [目录名]

（2）     命令功能：切换当前目录至dirName

（3）     常用范例：

①进入系统根目录

             cd /

②进入当前目录的上一级目录：cd ..或者cd ..//

③进入当前用户的主目录

             cd 或者 cd~

④跳转到指定目录

             cd /echncms/lijl\_bj1（根目录下进入）

跳转到指定目录，从根目录开始，目录名称前加“/”，当前目录的子目录写名称即可。

⑤返回进入此目录之前所在的目录

             cd –

⑥把上个命令的参数作为cd参数使用

             cd !$

（4）     注意事项：

跳转到指定目录，从根目录开始，目录名称前加“/”，当前目录的子目录写名称

4， pwd命令

使用“pwd”命令来查看“当前工作目录”的完整路径，每当你在终端进行操作时，都会有一个当前工作目录，在不太确定当前位置时，可以使用pwd命令来查看当前目录再文件系统里的确切位置。

（1）     命令格式：pwd [选项]

（2）     命令功能：查看“当前工作目录”的完整路径

常用参数：一般情况下不使用任何参数，如果目录是链接时，pwd –P 显示出实际路径，而非使用链接路径。若当前路径被删除了，pwd显示的仍是原路径。

5， rm命令

该命令的功能是删除一个目录中的一个或多个文件或目录，它也可以将某个目录及其以下的所有文件及目录均删除，对于链接文件，只是删除了链接，原有文件保持不变。

（1）     命令格式：rm [选项] 文件

（2）     命令功能：删除指定目录中的一个或多个文件或目录，如果没有使用-r选项，则rm不会删除目录，如果使用rm来删除文件，通常仍可以将文件恢复原状。

（3）     命令参数：

-f,--force忽略不存在的文件，从不给出提示

-i,--interactive 进行交互式地删除

-r,-R,--recursive 指示rm将参数中列出的全部目录和子目录均递归地删除

-v,--verbose 详细显示进行的步骤

（4）     命令实例：

①删除文件file，系统会先询问是否删除，rm –i log.log，输入y之后就会删除

②强行删除file，系统不再提示  rm –f log1.log

③删除任何.log文件，删除前逐一询问  rm –i \*.log

④将test1子目录及子目录中所有档案都删除 rm –r test

⑤将test2子目录及子目录中所有档案都删除，并且不用一一询问

     rm  –rf  test2（常用）

⑥删除以-f开头的文件

             rm -- -f

⑦自定义回收站功能：

myrm(){ D=/tmp/$(date +%Y%m%d%H%M%S); mkdir –p $D; mv “$@” $D && echo “moved to $D ok”; }

6， rmdir命令

rmdir是常用的命令，该命令的功能是删除空目录，一个目录被删除之前必须是空的。注意（rm –r dir命令可代替rmdir，但是有很大危险性），删除某目录时也必须对父目录有写权限。

（1）     命令格式：rmdir [选项]…目录…

（2）     命令功能：该命令从一个目录中删除一个或多个子目录项，删除某目录时也必须对父目录有写权限。

（3）     命令参数：

-p 递归删除目录dirname，当子目录删除后其父目录为空时，也一同被删除，如果整个路径被删除或者由于某种原因保留部分路径，则系统在标准输出上显示相应的信息。

-v,--verbose显示指令执行过程

（4）     命令实例：

①rmdir不能删除非空目录 rmdir lijl\_bj1,该命令不能直接删除非目录

②rmdir –p 当子目录被删除后使它也一并成为空目录的话，则顺便一起删除

7， mv命令

mv命令是move的缩写，可以用来移动文件或者将文件改名（move (rename) files），是Linux系统下常用的命令，经常用来备份文件或者目录。

（1）     命令格式：mv [选项] 源文件或目录 目标文件或目录

（2）     命令功能：视mv命令中第二个参数类型的不同（是目标文件还是目标目录），mv命令将文件重命名或将其移至一个新的目录中。当第二个参数类型是文件时，mv命令完成文件重命名，此时，源                           文件只能有一个（也可以是源目录名），它将所给的源文件或目录重命名为给定的目标文件名。当第二个参数是已存在的目录名时，源文件或目录参数可以有多个，mv命令将各参数指                             定的源文件均移至目标目录中。在跨文件系统移动文件时，mv先拷贝，再将原有文件删除，而链接该文件的链接也将丢失。

（3）     命令参数：

-b ：若需要覆盖文件，则覆盖前先行备份

-f ： force强制的意思，如果目标文件已存在，不会询问而直接覆盖

-i ： 若目标文件已经存在时，就会询问是否覆盖。

-t ： --target –directory，即指定mv的目标目录，该选项适用于移动多个源文件到一个目录的情况，此时目标目录在前，源文件在后。

（4）     命令实例：

①文件改名：将文件test.log 改名为test1.txt

             mv test.log test1.txt

②移动文件：将test1.log文件移至test3

             mv test1.log test3

③将文件log1.txt,log2.txt,log3.txt移至目录test中

             mv log1.txt log2.txt log3.txt test

             mv –t /opt/soft/test/ log1.txt log2.txt log3.txt

④将文件file1改名为file2，如果file2已经存在，则询问是否覆盖

             mv –i log1.txt log2.txt

⑤将文件file1改名为file2，即时file2存在，也是直接覆盖掉

             mv –f log3.txt log2.txt

log3.txt的内容直接覆盖了log2.txt内容，-f这是个危险的选项，使用时候一定要保持头脑清醒，一般情况下最好不用加上它。

⑥目录的移动

             mv dir1 dir2

如果目录dir2不存在，将目录dir1改名为dir2，否则将dir1移动到dir2中

⑦移动当前文件夹下的所有文件到上一级目录

             mv \* ../

⑧把当前目录的一个子目录里的文件移动到另一个子目录里

             mv test3/\*.txt test5

⑨文件被覆盖前做简单备份，前面加参数-b

             mv log1.txt –b log2.txt

说明：-b不接受参数，mv会去读取环境变量VERSION\_CONTROL来作为备份策略。--backup该选项指定如果目标文件存在时的动作，共有四种备份策略：

1.CONTROL=none或off : 不备份。

2.CONTROL=numbered或t：数字编号的备份

3.CONTROL=existing或nil：如果存在以数字编号的备份，则继续编号备份m+1...n：

执行mv操作前已存在以数字编号的文件log2.txt.~1~，那么再次执行将产生log2.txt~2~，以次类推。如果之前没有以数字编号的文件，则使用下面讲到的简单备份。

4.CONTROL=simple或never：使用简单备份：在被覆盖前进行了简单备份，简单备份只能有一份，再次被覆盖时，简单备份也会被覆盖。

mv不会创建新目录，只能将源目录或源文件移入已创建好的目录，若目标文件或文件夹不存在时，将只会是改名，而不是移动文件或目录

8， cp命令

cp命令用来复制文件或者目录，是Linux系统中最常用的命令之一。一般情况下，shell会设置一个别名，在命令行复制文件时，如果目标文件已存在，就会询问是否覆盖，不管你是否使用参数-i，但如果是在shell脚本中执行cp时，没有参数-i时不会询问是否覆盖，这说明命令行和shell脚本的执行方式有些不同。

（1）     命令格式： cp [选项]… [-T] 源 目的

                           或 cp [选项]… 源… 目录

                           或 cp [选项]… -t 目录  源…

（2）     命令功能：将源文件复制至目标文件，或将多个源文件复制至目标文件

（3）     命令参数：

-i ,--interactive 覆盖前询问（使前面的-n选项失效）

-n ,--no-clobber 不要覆盖已存在的文件（使前面的-i选项失效）

-R,-r,--recursive 复制目录及目录内的所有项目

（4）     命令实例：

①复制单个文件到目标目录，文件在目标文件中不存在

             cp log.log test5

在没有带-a参数时，两个文件的时间是不一样的，在带来-a参数时，两个文件的时间是一致的。

②目标文件存在时，会询问是否覆盖。

             cp log.log test5

目标文件存在时，会询问是否覆盖，这是因为cp 是cp –i 的别名，目标文件存在时，即时加了-f标志，也还会询问是否覆盖。

③复制整个目录

             cp –a test3 test5

注意目标目录存在与否结果是不一样的，目标目录存在时，整个源目录被复制到目标目录里面。

④复制的log.log建立一个链接档案log\_link.log

             cp –s log.log log\_link.log

⑤那个log\_link.log是由-s的参数造成的，建立的是一个【快捷方式】，所以会看到在文件的最右边，会显示这个文件是链接到哪里去的。

9， touch命令

Linux的touch命令不常用，一般在使用make的时候可能会用到，用来修改文件时间戳，或者新建一个不存在的文件。

（1）     命令格式： touch [选项]…文件…

（2）     命令功能：touch命令参数可更改文档或目录的日期时间，包括存取时间和更改时间。

（3）     命令参数：

-a 或--time = atime--time =access或--time=use 只更改存取时间

-c 或--no-create 不建立任何文档

-d 使用指定的日期时间，而非现在的时间

-m 或--time=mtime或--time=modify 只更改变动时间

-r 把指定文档或目录的日期时间，统统设成和参考文档和目录的日期时间相同

-t 使用指定的日期，而非现在的时间

（4）     常用实例：

①创建不存在的文件：

             touch log2015.log log2016.log

如果log2014.log不存在，则不创建文件

②更新log.log的时间和log2015.log时间戳相同

             touch –r log2015.log log.log

③设定文件的时间戳

             touch –t 201505191357.50 log.log

-t  time 使用指定的时间值 time 作为指定文件相应时间戳记的新值．此处的 time规定为如下形式的十进制数:

  [[CC]YY]MMDDhhmm[.SS]

  这里，CC为年数中的前两位，即”世纪数”；YY为年数的后两位，即某世纪中的年数．如果不给出CC的值，则touch   将把年数CCYY限定在1969--2068之内．MM为月数，DD为天将把年数CCYY限定在1969--2068之内．MM为月数，DD为天数，hh 为小时数(几点)，mm为分钟数，SS为秒数．此处秒的设定范围是0--61，这样可以处理闰秒．这些数字组成的时间是环境变量TZ指定的时区中的一个时 间．由于系统的限制，早于1970年1月1日的时间是错误的。

10，            cat命令

cat命令的用途是连接文件或标准输入并打印，这个命令常用来显示文件内容，或者将几个文件连接起来显示，或者从标准输入读取内容并显示，它常与重定向符号配合使用。

（1）     命令格式：cat [选项] [文件]…

（2）     命令功能：cat主要有三大功能：

①一次显示整个文件：cat filename

②从键盘创建一个文件：cat>filename只能创建新文件，不能编辑已有文件

③将几个文件合并为一个文件：cat file1 file2>file

（3）     命令参数：

-A,--show-all  等价于-vET

-b,--number-nonblank  对非空输出行编号

-e                                         等价于 –vE

-E,--show-ends                   在每行结束处显示$

-n,--number                        对输出的所有行编号，由1开始对所有输出的行数编号

-s,--squeeze-blank             有连续两行以上的空白行，就代换为一行的空白行

-t                                          与-vT等价

-T,--show-tabs                    将跳格字符显示为^I

（4）     使用实例：

①把log2012.log的文件内容加上行号后输入log2013.log这个文件

           cat –n log2012.log log2013.log

②把log2012.log和log2013.log的文件内容加上行号（空白行不加）之后将内容附加到log.log里。

           cat –b log2012.log log2013.log log.log

③把log2012.log的文件内容加上行号后输入log.log这个文件里

           cat –n log2012.log>log.log

④注意tac是反向列示，tac是将cat反过来，所以它的功能就跟cat相反，cat是由第一行到最后一行连续显示在荧幕上，而tac则是由最后一行到第一行反向在银幕上显示。

11，            more命令

more命令功能类似cat，cat命令是整个文件的内容从上到下显示在屏幕上，more会一起一页一页的显示方便使用者逐页阅读，而最基本的指令就是按空白键（space）就往下一页显示，按b键就会往回（back）一页显示，而且还有搜寻字串的功能，more命令从前向后读取文件，因此在启动时就加载整个文件。

（1）     命令格式：

                        more [-dlfpcsu][-num][+/pattern][+linenum][file…]

（2）     命令功能：more命令和cat的功能一样都是查看文件里的内容，但有所不同的是more可以按页来查看文件的内容，还支持直接跳转行等功能。

（3）     命令参数：

+n    从第n行开始显示

-n     定义屏幕大小为n行

+/pattern                  在每个档案显示前搜寻该字串（pattern），然后从该字串前两行之后开始显示

-c     从顶部清屏，然后显示

-p     通过清除窗口而不是滚屏来对文件进行换页，与-c选项相似

（4）     常用操作命令：

Enter 向下n行，需要定义，默认为1行

Ctrl+F、空格键                  向下滚动一屏

Ctrl+B                  返回上一屏

=                           输出当前行的行号

:f                 输出文件名和当前行的行号

V                          调用vi编辑器

q                           退出more

（5）     常用实例：

①显示文件中从第3行起的内容

           more +3 log2012.log

②从文件中查找第一个出现”day3”字符串的行，并从该处前两行开始显示输出

           more +/day3 log2012.log

③设定每屏显示行数

           more -5 log2012.log

最下面显示了该屏展示的内容占文件总行数的比例，按Ctrl+F或者空格键将会显示下一屏5条内容，百分比也会跟着变化。

④列一个目录下的文件，由于内容太多，我们应该学会用more来分页，这得和管道|结合起来

ls –l | more -5

每页显示5个文件信息，按Ctrl+F或者空格键将会显示下5条文件信息

12，            tail命令

tail命令从指定点开始将文件写到标准输出流，使用tail命令的-f选项可以方便的查阅正在改变的日志文件，tail –f filename会把filename里最尾部的内容显示在屏幕上，并且不但刷新，使你看到最新的文件内容。（经常用来在查看服务器上的日志文件）

（1）     命令格式：        tail   [必要参数][选择参数][文件]

（2）     命令功能：

             用于显示指定文件末尾内容，不指定文件时，作为输入信息进行处理，常用查看日志文件。

（3）     命令参数：

-f               循环读取

-q              不显示处理信息

-v              系那是详细的处理信息

-c<数目>         显示的字节数

-n<行数>                  显示行数

（4）     命令实例：

①显示文件末尾内容

             tail –n 5 log2014.log 显示文件最后5行的内容

②循环查看文件内容

             tail   -f      test.log

③从第5行开始显示文件

             tail –n +5 log2014.log

## Linux下链接服务器上传内容

1.使用ftp

sudo ftp host4668750.xincache1.cn

账号:

密码:

230 OK表示链接成功

2.常用命令

1. 下载文件

下载文件通常用get和mget这两条命令。

a) get

格式：get [remote-file] [local-file]

将文件从远端主机中传送至本地主机中。

如要获取远程服务器上/usr/your/1.htm，则

ftp> get /usr/your/1.htm 1.htm (回车)

b) mget

格式：mget [remote-files]

从远端主机接收一批文件至本地主机。

如要获取服务器上/usr/your/下的所有文件，则

ftp> cd /usr/your/

ftp> mget \*.\* (回车)

此时每下载一个文件，都会有提示。如果要除掉提示，则在mget \*.\* 命令前先执行:prompt off

注意：文件都下载到了linux主机的当前目录下。比如，在　/usr/my下运行的ftp命令，则文件都下载到了/usr/my下。

3.上传文件

a) put

格式：put local-file [remote-file]

将本地一个文件传送至远端主机中。

如要把本地的1.htm传送到远端主机/usr/your,并改名为2.htm

ftp> put 1.htm /usr/your/2.htm (回车)

b) mput

格式：mput local-files

将本地主机中一批文件传送至远端主机。

如要把本地当前目录下所有html文件上传到服务器/usr/your/ 下

ftp> cd /usr/your （回车）

ftp> mput \*.htm　（回车）

注意：上传文件都来自于主机的当前目录下。比如，在　/usr/my下运行的ftp命令，则只有在/usr/my下的文件linux才会上传到服务器/usr/your 下。

4. 断开连接

bye：中断与服务器的连接。

ftp> bye (回车)

注:1.移动当前目录 cd ..

删除当前文件 del \*.html

## Linux下截屏：

1. 使用快捷键shift+prt sc sysrq

就可以截屏。

1. 接下来打开gimp或者剪切板就可一进行修改。

yum -y install vsftpd

yum -y install vsftpd

Linux下安装go语言环境：

1. 从htt://golang.org/dl/ 下载最新版本的Go语言二进制档案包

中国人可以在中文版golang：<http://golangtc.com/download>下载

注意:根据操作系统和计算机架构正确选择档案包

2.解压   
$ tar -C /usr/local -zxf go1.3.3.linux-amd64.tar.gz

这里直接安装二进制，其它方式请自行搜索。

**2、添加环境变量**

vim /etc/profile

**添加如下内容：**

[?](http://www.jb51.net/article/115873.htm)

|  |  |
| --- | --- |
| 1  2 | export PATH=$PATH:/usr/local/go/bin  export GOROOT=/usr/local/go |

**使配置生效：**

[?](http://www.jb51.net/article/115873.htm)

|  |  |
| --- | --- |
| 1  2  3  4 | [root@localhost ~]# source /etc/profile  [root@localhost ~]# go version  go version go1.8.3 linux/amd64  [root@localhost ~]# |

**使用go**

这里以简单的示例介绍下go语言的编译、运行，更深层次的内容暂不讨论。

文件名： test1.go 代码：

[?](http://www.jb51.net/article/115873.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | package main    import "fmt"    func main() {    fmt.Println("Email : Mike\_Zhang@live.com")  } |

**常规编译运行**

1、通过go build 命令将go源码编译成二进制文件；

2、执行编译好的二进制文件即可。

示例如下：

[?](http://www.jb51.net/article/115873.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | [root@localhost src]# go build test1.go  [root@localhost src]# ls  test1 test1.go  [root@localhost src]# ./test1  Email : Mike\_Zhang@live.com  [root@localhost src]# |

**以脚本方式运行**

go语言可以直接通过go run直接运行程序，可以借助此特性以脚本方式运行go程序。

**方法1：**

[?](http://www.jb51.net/article/115873.htm)

|  |  |
| --- | --- |
| 1  2  3 | [root@localhost src]# go run test1.go  Email : Mike\_Zhang@live.com  [root@localhost src]# |

**方法2：**

在文件头部加入如下代码：

[?](http://www.jb51.net/article/115873.htm)

|  |  |
| --- | --- |
| 1 | //usr/bin/env go run $0 "$@"; exit |

然后通过chmod赋予可执行权限即可。

**示例如下：**

[?](http://www.jb51.net/article/115873.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | [root@localhost src]# cat test1.go  //usr/bin/env go run $0 "$@"; exit    package main    import "fmt"    func main() {      fmt.Println("Email : Mike\_Zhang@live.com")  }    [root@localhost src]# chmod a+x test1.go  [root@localhost src]# ./test1.go  Email : Mike\_Zhang@live.com  [root@localhost src]# |

好，就这些了，希望对你有帮助。