# 软件的架构

## C/S(客户端/服务器)

·一般我们使用的软件都是C/S架构的

·比如系统中的软件

- QQ

- 360

- office

·C表示客户端

- 用户通过客户端来使用软件

·S表示服务器

- 服务器负责处理软件的业务逻辑

·特点

- 使用前必须安装

- 软件更新时，服务器和客户端得同时更新

- C/S架构的软件不能跨平台使用

- C/S架构的软件客户端和服务器通信采用的是自有协议，相对来说比较安全。

## B/S（浏览器/服务器）

·B/S本质上也是C/S，只不过B/S架构的软件，使用浏览器作为软件的客户端

·B/S架构软件通过使用浏览器访问网页的形式，来使用软件

·特点

- 不需要安装，直接使用浏览器访问指定的网址即可

- 软件更新时，客户端不需要更新

- 软件可以跨平台，只要系统中有浏览器，就可以使用

- B/S架构的软件，客户端和服务器直通信采用的是通用的HTTP协议，相对来说不安全

- 可使用HTTPS安全的协议

## 软件开发流程

### 旧模式

·项目经理需求分析

·网页设计师根据需求设计网页

·前端工程师将设计做成静态网页

·后端工程师将静态网页修改为动态网页

## 新模式

·项目经理需求分析

·网页设计师根据需求设计网页

·前端工程师将设计做成静态网页

·后端工程师提供所需数据的接口

·前端工程师将功能实现

## W3C标准

·结构

- HTML用于描述页面的结构

·表现

- CSS用于控制页面中元素的样式

·行为

- JavaScript用于响应用户操作

# HTML简介

·超文本标记语言

·它负责网页三要素中的结构

·使用标签的形式来标识网页中的不同组成部分

·所谓超文本值得是超链接，使用超链接可以让我们从一个页面跳转到另一个页面

## 标准格式

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

</head>

<body>

</body>

</html>

·<!DOCTYPE html>是文档声明

- HTML4声明方式

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

- HTML5声明方式

<!DOCTYPE html>

- XHTML声明方式

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

- 如果不写文档声明，则会导致有些浏览器进入怪异模式，浏览器解析页面无法正常显示

·<html>是根标签，有且只有一个，网页中的所有内容写在这个标签里面

·<head>标签用来设置网页的头部信息，不会在网页中直接显示，帮助浏览器解析网页

·<body>，用来设置网页的主体内容，网页中所有可见的内容都写在body标签内

·<title>中的内容会显示在网页标题栏，搜索引擎在检索页面使会首先检索title中的内容，它是网页中对于搜索引擎来说最重要的内容，会影响到网页在搜索引擎中的排名

·<meta charset="UTF-8">是字符集声明

## html注释

·<!--注释内容-->

- 不会在页面中显示，可在源码中查看

- 便于自己或其他开发人员的阅读

- 要养成良好的注释习惯，并且注释要简单明了

## 标签的属性

·可以通过属性来处理标签的内容

·可以在开始标签中添加属性

·属性名 = “属性值”

·一个标签中可以同时设置多个属性，属性之间需要使用空格隔开

## 进制

·满几进一

·二进制

- 0 1

- 10 11 100 101 110 111......

·十进制

- 0 1 2 3 4 5 6 7 8 9

- 10 11 12......

·十六进制

- 0 1 2 3 4 5 6 7 8 9 A B C D E F

- 10 11 12...19 1A 1B 1C

·八进制

- 0 1 2 3 4 5 6 7

- 10 11 12 ....16 17 20 21......

## 乱码问题

### 乱码出现的原因

·计算机在底层只认识0和1

·在计算机中保存的任何内容，最终都需要转换为二进制编码，包括网页中的内容

·在读取内容时，需要将二进制编码再转换为正确的内容

·编码

- 依据一定的规则，将字符转换为二进制编码的过程

·解码

- 依据一定的规则，将二进制编码转换为字符的过程

·字符集

- 编码和解码采用的规则，我们称为字符集

- ASCII

- ISO-8859-1

- GBK

-- 中国编码，下面那个也是

- GB2312

-- 中文系统默认编码

- UTF-8

-- 万国码支持地球上所有的文字

- ANSI

-- 自动以系统的默认编码来保存文件

- 等

·产生乱码的根本原因是，编码和解码采用的字符集不同

·在中文系统的浏览器中，默认都是使用GB2312进行解码的

### 解决方法

·在head标签内添加字符集声明

<meta charset="UTF-8">

# 标签

## 常用标签

### 标题标签

·在HTML中，一共有六级标题标签

·h1~h6

·在显示效果上h1最大、h6最小，但是文字的大小我们并不关心

·使用HTML标签时，关系的是标签的语义，我们使用的标签都是语义化标签

·六级标题中，h1最重要，表示一个网页中的主要内容。h2~h6重要性依次降低

·对于搜索引擎来说，h1的重要性仅次于title，搜索引擎检索完title，会立即查看h1中的内容

·h1非常重要，他会影响到页面在搜索引擎中的排名，页面一般只写一个h1，否则会被认定为垃圾网站不被搜索

·一般页面中标题标签只使用h1 h2 h3，其他的基本不使用

<h1>这是h1标签</h1>

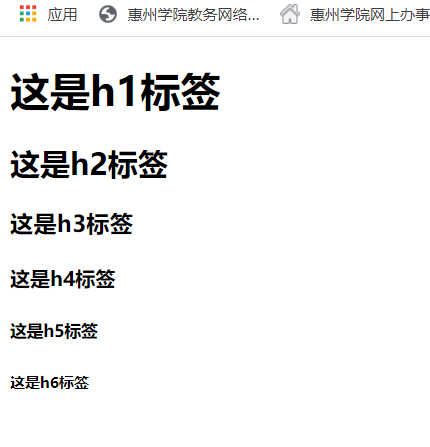
<h2>这是h2标签</h2>

<h3>这是h3标签</h3>

<h4>这是h4标签</h4>

<h5>这是h5标签</h5>

<h6>这是h6标签</h6>



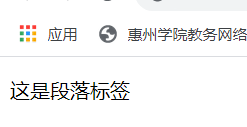
### 段落标签

·用于表示内容中的一个自然段

·使用p标签表示一个段落

·p标签中的文字，默认会独占一行，并且段与段之间会有一个间距

<p>这是段落标签</p>



### 换行标签

·在HTML中，字符之间再多的空格，浏览器也会当成一个空格解析

·换行也会当成一个空格解析

·使用br标签表示一个换行，br是一个自结束标签

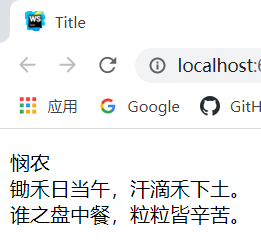
悯农

<br/>

锄禾日当午，汗滴禾下土。

<br/>

谁之盘中餐，粒粒皆辛苦。



### 分割线标签

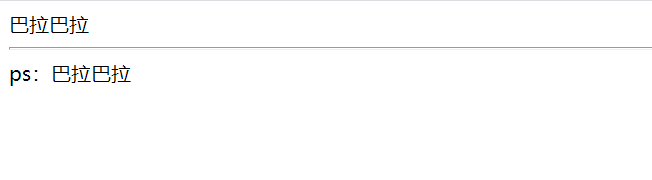
·hr标签是自结束标签

·可在在页面中生成一条水平分割线

巴拉巴拉

<hr/>

ps：巴拉巴拉



## 实体

·在HTML中，一些如<>中特色字符是不能直接使用

·需要使用一些特殊的符号来表示这些特殊字符，这些特殊符号我们称为实体

·实体语法

- &实体的名字

·浏览器解析到实体时，会自动将实体转换为其对应的字符

·常用实体

- 小于号 <

-- &lt;

- 大于号 >

-- &gt;

- 空格

-- &nbsp;

- 版权符号

-- &copy;

·详情查看w3school离线手册

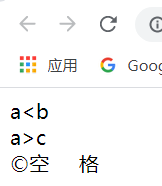
a&lt;b

<br/>

a&gt;c

<br/>

&copy;空&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;格



## 图片标签

·使用img标签来向网页中引入一个外部图片

·img标签也是一个自结束标签

·属性

- src

-- 设置一个外部图片的路径

- alt

-- 用来设置图片的描述

-- 在图片不能显示时显示描述

-- 搜索引擎通过alt来识别不同的图片

-- 如果不写alt属性，则搜索引擎不会对img中的图片进行收录

- width

-- 设置图片的宽度，一般使用px作为单位

- height

-- 设置图片的高度，一般使用px作为单位

宽度和高度只设置其中一个，另一个也会等比例调整大小，一般开发中除了自适应的页面，不建议设置width和height属性，而是让美工修改图片。

·相对路径

- src属性配置的是图片的路径，目前我们所要使用的路径全都是相对路径

- 相对于当前资源所在目录的位置

- ../是返回上级文件夹

·图片格式

- JPG

-- 支持颜色比较多，图片可以压缩，但是不支持透明

-- 一般用来保存照片等颜色丰富的图片

- GIF

-- 支持的颜色少，只支持简单的透明，支持动态图

-- 图片颜色单一或者是动态图用gif

- PNG

-- 支持的颜色多，并且支持复杂的透明

-- 用来显示颜色复杂的透明的图片

- WEBP

-- 谷歌新推出的用来表示网页中图片的一种格式

-- 具备其他格式图片的所有优点

-- 文件还特别小

-- 但是兼容性不好

- base64

-- 字符流图片

-- 通过base64转换为字符

-- 一般都是一些需要和网页一起加载的图片才会使用base64

- 图片的使用原则

-- 效果不一致，使用效果好的

-- 效果一致，使用小的

<!--引入本地图片-->

<img src="imgs/8.jpg" alt="壁纸" width="455px"/>

<!--引用外部图片-->

<img src="https://img2.baidu.com/it/u=3724265679,1678217865&fm=26&fmt=auto&gp=0.jpg" alt="iu" height="256px"/>

## meta标签

·charset

- 设置字符集

<meta charset="UTF-8">

·keywords

- 设置网页的关键字

<meta name="keywords" content="HTML5,JavaScript,前端，Java">

·description（描述）

- 指定网页的描述

<meta name="description" content="发布h5、js等前端相关信息">

·搜索引擎在检索页面时，会同时检索页面的关键词和描述，但是这两个值不会影响页面在搜索引擎中的排名

·refresh（刷新）

- 请求重定向

<!--请求重定向，5秒后打开某网址-->

<meta http-equiv="refresh" content="5;url=https://www.baidu.com">

·详情查看w3school离线手册

## 布局标签

·header

- 网页头部

<header></header>

·main

- 网页的主体部分

- 一个页面中只能有一个main

<main></main>

·footer

- 表示网页的底部

<footer></footer>

·nav

- 网页中的导航

<nav></nav>

·aside（在旁边）

- 和主体相关的其他内容（侧边栏）

<aside></aside>

·article

- 独立的文章

<article></article>

·section

- 独立的区块

<section></section>

·div

- 没有语义，一个块

- 目前主要的布局元素

- 可以代替上边的所有

## 居中标签

·center标签中的内容，会默认在页面中居中显示

·已过时，一般用css实现

<center>

<p>居中显示</p>

</center>

# xHtml的语法规范

·HTML中不区分大小写，但是一般都使用小写

·注释不能嵌套

·必须结构完整，要么成对出现，要么自结束标签

·浏览器尽最大的努力正确的解析页面，你所有不符合语法规范的内容，浏览器都会为你自动修正，但是有些情况修正会出现错误

·可以嵌套，但是不能交叉嵌套

·属性必须有值，且值必须加引号

·双引号单引号都可以

·不按规范也没问题，浏览器可以修正，但最好按规范

# 内联框架

·引入一个外部的页面

·属性

- src

-- 指向一个外部页面的路径，可以使用相对路径

- width

-- 宽度

- height

-- 高度

- name

-- 名字

·在现实开发中不推荐使用内联框架，因为内联框架中的内容不会被搜索引擎所检索

<body>

<h1>百度</h1>

<iframe src="https://www.baidu.com" name="tom"></iframe>

</body>



# 超链接

·使用超链接可以让我们从一个页面跳转到另一个页面

·使用a标签来创建一个超链接

·属性

- href

-- 指向链接跳转的目标地址，可以写一个相对路径，也可以写一个完整的地址

-- 将该属性值设置为#，点击超链接后回到页面顶部

-- 将该属性值设置为#id，会跳转到该id元素

- target

-- 指定打开链接的位置

-- \_self : 当前页面打开，默认值

-- \_blank : 在新页面中打开

-- 内联框架中的name属性值：在内联框架中打开

<body>

<a href="https://cn.vuejs.org/" target="tom">vue</a>

<iframe src="https://www.baidu.com" name="tom"></iframe>

</body>

·效果



# CSS简介

·层叠样式表（Cascading Style Sheets）

·CSS可以用来为网页创建样式表，通过样式表对网页进行装饰

·层叠是将整个网页想象成一层一层的结构，层次高的将会覆盖层低的

·CSS可以为网页的各个层次设置样式

## 内联样式

·将CSS样式写到元素的style属性当中

·内联样式只对当前的元素中的内容起作用

·内联样式不方便样式复用

·属于结构与表现耦合，不方便后期维护

·不推荐使用

<p style="color: rebeccapurple;font-size: 30px">锄禾日当午，汗滴禾下土</p>

## 内部样式

·将CSS样式写到head标签中的style标签里

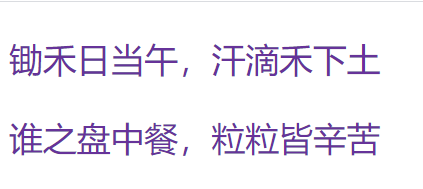
·通过CSS选择器选中指定元素

·可以同时为这些元素设置样式，使样式进一步复用

·可以使表现和结构进一步分离

·推荐的一种使用方式

·效果



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

p{

color: rebeccapurple;

font-size: 30px;

}

</style>

</head>

<body>

<p>锄禾日当午，汗滴禾下土</p>

<p>谁之盘中餐，粒粒皆辛苦</p>

</body>

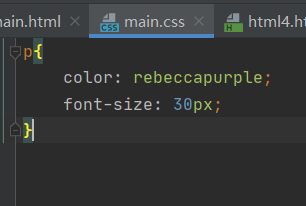
</html>

## 外部样式

·将CSS样式编写到外部CSS文件中

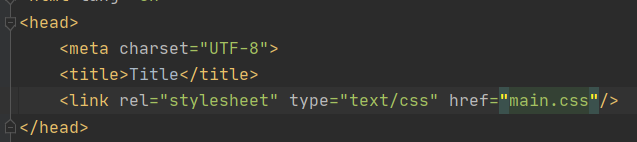
·步骤

- 编写外部CSS文件



- 通过link标签来将外部的CSS文件引入到页面中

<link rel="stylesheet" type="text/css" href="main.css"/>



·可以完全使结构和表现分离，使样式表在不同的页面中使用，最大限度的使样式进行复用

·将样式统一写在CSS文件中，然后通过link标签引入，可以利用浏览器的缓存，加快用户访问的速度，提高了用户体验。

·最推荐使用

## CSS的语法

### CSS注释

/\*这是注释\*/

### CSS选择器

·通过选择器可以选中页面中指定的元素

·将声明块中的样式应用到选择器对应的元素上

### 声明块

·写在选择器后边

·使用{}括起来

·实际上就是一组名值对结构，一组名值对称为一个声明

·多个声明之间使用;来隔开

·声明的样式名和样式值之间使用:来连接

### 块

·独占一行，无论内容有多少

·div就是一个块元素

·p、h1、h2...也是块元素

·div标签没有任何语义，就是一个纯粹的块，并且不会为它里边的元素设置任何的默认样式

·div元素主要用来对页面进行布局的

### 内联元素

·只占自身大小的元素，不会占用一行

·span是一个内联元素（行内元素）

·a、img、iframe、span

·span标签没有任何语义，专门用来选中文字，然后为文字设置样式

·块元素做页面中的布局，内联元素用来选中文本设置样式

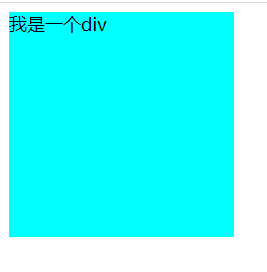
·一般只使用块元素去包含内联元素，不会用内联元素去包含块元素

·a元素可以包含任意元素，除了它本身

·p元素不能包含任何其他的块元素

·HTML5已经没有内联元素和块元素的概念了

·效果



<body>

<div style="background-color: aqua;width: 200px;height: 200px">我是一个div</div>

</body>

# 选择器

## 常用选择器

### 元素（标签）选择器

·可以选择页面中的所有指定元素

·语法

- 标签名{}

p{

color: rebeccapurple;

}

h1{

color: red;

}

### id选择器

·通过元素的id属性值选中唯一一个元素

·语法

- #id属性值{}

#p1{

color: black;

}

### 类选择器

·通过元素的class属性值选中一组元素

·与id选择器类似，但是可以重复

·拥有相同class属性值的元素，他们是一组元素

·可以同时为一个元素设置多个class属性值，多个值之间使用空格隔开

<p class="p2 hello">谁之盘中餐</p>

·语法

- .class属性值{}

.p2{

color: black;

}

### 并集选择器（选择器分组）

·同时选中多个选择器对应的元素

·语法

- 选择器1,选择器2,选择器n{}

#p1,.p2{

color: rebeccapurple;

}

### 通配选择器

·选中页面中的所有元素

·语法

- \*{}

\*{

color: rebeccapurple;

}

### 交集选择器（复合选择器）

· 选中同时满足多个选择器的元素

·对于id选择器来说，不建议使用复合选择器

·语法

- 选择器1选择器2{}

span.p2{

color: red;

}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

\*{

color: rebeccapurple;

}

p{

color: rebeccapurple;

}

h1{

color: red;

}

#p1{

color: black;

}

.p2{

color: black;

}

.hello{

color: rebeccapurple;

}

#p1,.p2{

color: rebeccapurple;

}

span.p2{

color: red;

}

</style>

</head>

<body>

<h1>悯农</h1>

<p class="p2">锄禾日当午</p>

<p id="p1">汗滴禾下土</p>

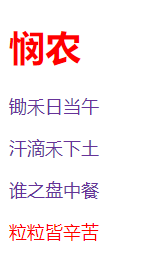
<p class="p2 hello">谁之盘中餐</p>

<span class="p2">粒粒皆辛苦</span>

</body>

</html>

·效果



## 其他选择器

### 后代元素选择器

·选中指定元素的指定后代元素

·语法

- 祖先元素 后代元素{}

#d1 span{

color: aqua;

}

#d1 p span{

color: rebeccapurple;

}

### 子元素选择器

·选中指定父元素的指定子元素

·IE6及以下的浏览器不支持子元素选择器

·语法

父元素 > 子元素

#d1 > span{

color: red;

}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

#d1 span{

color: aqua;

}

#d1 p span{

color: rebeccapurple;

}

#d1 > span{

color: red;

}

</style>

</head>

<body>

<div id="d1">

我是div标签

<span>我是div标签中的span</span>

<p>

<span>我是p标签中的span</span>

</p>

</div>

<div>

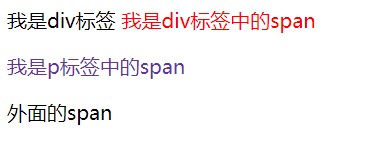
<span>外面的span</span>

</div>

</body>

</html>

·效果



### 伪类选择器

·专门用来表示元素的一种特殊的状态

·比如：访问过的超链接，普通的超链接，获取焦点的文本框

·当我们需要为处在这些特殊状态的元素设置样式时，就可以使用伪类

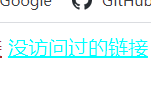
·a:link

- 正常链接（没访问过）

a:link{

color: aqua;

}



·a:visited（只能定义字体颜色）

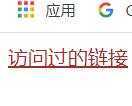
- 访问过的链接

- 浏览器通过历史记录判断链接是否被访问，由于涉及到用户的隐私问题，所以只能设置字体颜色

a:visited{

color: brown;

}



·:hover

- 鼠标划过时

- hover（悬停）

a:hover{

color: black;

}



·:active

- 正在点击时

button:active{

color: red;

}



·IE6中不支持对超链接以外的元素设置hover和active

·:focus

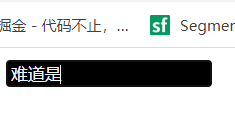
- 获取焦点

input:focus{

background-color: black;

color: aliceblue;

}



·:before

- 指定元素前

- 一般需要结合content样式一起使用

- 通过content可以向before或after添加一些内容

p:before{

content: "前面";

color: rebeccapurple;

}

·:after

- 指定元素后

p:after{

content: "后面";

color: rebeccapurple;

}

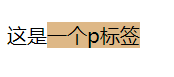
·::selection

- 选中的元素

p::selection{

background-color: burlywood;

}



- 在火狐浏览器中需要用另一种方式

-- -moz-selection{}

p::-moz-selection{

background-color: burlywood;

}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

a:link{

color: aqua;

}

a:visited{

color: brown;

}

a:hover{

color: black;

}

button:active{

color: red;

}

input:focus{

background-color: black;

color: aliceblue;

}

p::selection{

background-color: burlywood;

}

p::-moz-selection{

background-color: burlywood;

}

</style>

</head>

<body>

<a href="https://www.baidu.com">访问过的链接</a>

<a href="https://xxt.com/">没访问过的链接</a>

<button>按钮</button>

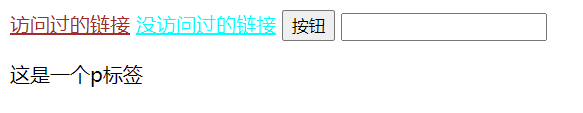
<input type="text">

<p>这是一个p标签</p>

</body>

</html>

·效果



### 伪元素选择器

·使用伪元素来表示元素中的一些特殊的位置

·:first-letter

- 首字母

p:first-letter{

color: red;

font-size: 20px;

}

·:first-line

- 首行

p:first-line{

background-color: burlywood;

}



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

p{

width: 50px;

}

p:first-letter{

color: red;

font-size: 20px;

}

p:first-line{

background-color: burlywood;

}

p:before{

content: "前面";

color: rebeccapurple;

}

p:after{

content: "后面";

color: rebeccapurple;

}

</style>

</head>

<body>

<p>这是一个p标签</p>

</body>

</html>

### 属性选择器

·根据元素中的属性或属性值来选择指定元素

·语法：

- [属性名] 选取含有指定属性名的元素

- [属性名=”属性值”] 选取含有指定属性值的元素，引号可写可不写

- [属性名^=”字符串”] 选取属性值以指定内容开头的元素

- [属性名$=”字符串”] 选取属性值以指定内容结尾的元素

- [属性名\*=”字符串”] 选取属性值包含指定内容的元素

·title属性

- 当鼠标移入到元素上时，元素中的title属性的值将会作为提示文字显示

- 可以给任何标签指定

<p title="提示">这是一个p标签</p>

·为所有具有title属性的p元素，设置一个背景颜色

p[title]{

background-color: cadetblue;

}

·为title属性值为“提示”的元素，设置一个背景颜色

p[title="提示"]{

color: aliceblue;

}

·为title属性值以”ab”开头的元素，设置一个背景颜色

p[title^="ab"]{

background-color: burlywood;

}

·为title属性值以”c”结尾的元素，设置一个背景颜色

p[title$="c"]{

background-color: red;

}

·为title属性值包含”lh”的元素，设置一个背景颜色

p[title\*="lh"]{

background-color: rebeccapurple;

}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

p[title]{

background-color: cadetblue;

}

p[title="提示"]{

color: aliceblue;

}

p[title^="ab"]{

background-color: burlywood;

}

p[title$="c"]{

background-color: red;

}

p[title\*="lh"]{

background-color: rebeccapurple;

}

</style>

</head>

<body>

<p title="提示">这是一个p标签</p>

<p title="abc">这是一个p标签</p>

<p title="abd">这是一个p标签</p>

<p title="salhfs">这是一个p标签</p>

<p title="最后">这是一个p标签</p>

</body>

</html>

·效果



### 子元素伪类

·:first-child

- 选择第一个子元素

p:first-child{

background-color: rebeccapurple;

}

- 当第一个子元素不是p元素时则无效

·:last-child

- 选择最后一个子元素

p:last-child{

background-color: rebeccapurple;

}

·:nth-child

- 选择指定位置的子元素

- 传入even表示偶数位置的子元素

- 传入odd表示偶数位置的子元素

p:nth-child(2){

background-color: rebeccapurple;

}

·:first-of-type

- 选择同类型标签第一个子元素

p:first-of-type{

background-color: burlywood;

}

·:last-of-type

- 选择最后一个子元素

·:nth-of-type

-选择指定位置的元素

·区别

- child是在所有的子元素中排列

- type是在同类型的元素中排列

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

/\*body > p:first-child{\*/

/\* background-color: rebeccapurple;\*/

/\*}\*/

p:first-child{

background-color: rebeccapurple;

}

p:last-child{

background-color: rebeccapurple;

}

p:nth-child(2){

background-color: rebeccapurple;

}

p:first-of-type{

background-color: burlywood;

}

</style>

</head>

<body>

<!-- <span>hello</span>-->

<p>我是一个p标签</p>

<p>我是一个p标签</p>

<p>我是一个p标签</p>

<div>

<p>我是一个p标签</p>

</div>

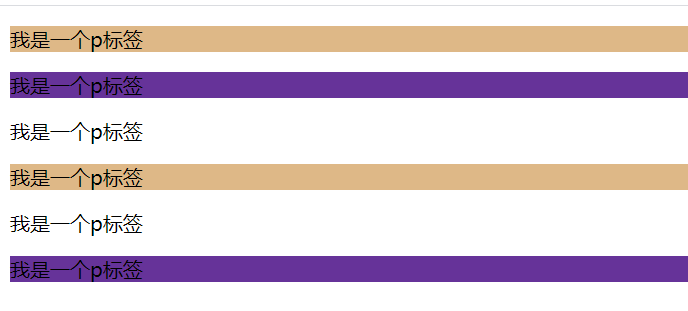
<p>我是一个p标签</p>

<p>我是一个p标签</p>

</body>

</html>

·效果



### 兄弟元素选择器

·后一个兄弟选择器

- 为span后的一个p元素设置一个背景颜色

- 语法

-- 前一个 + 后一个

·选中后边的所有兄弟元素

span ~ p{

background-color: rebeccapurple;

}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

span + p{

background-color: burlywood;

}

span ~ p{

background-color: rebeccapurple;

}

</style>

</head>

<body>

<!-- <span>hello</span>-->

<p>我是一个p标签</p>

<p>我是一个p标签</p>

<p>我是一个p标签</p>

<span>

我是一个p标签

</span>

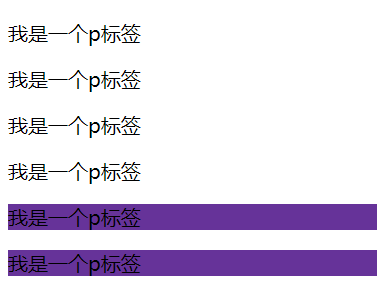
<p>我是一个p标签</p>

<p>我是一个p标签</p>

</body>

</html>

·效果



### 否定伪类

·从已选择的元素中剔除某些元素

·语法

- ：not(选择器)

·为所有的p元素设置一个背景颜色，除了class值为hello的

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

p:not(.hello){

background-color: rebeccapurple;

}

</style>

</head>

<body>

<p>我是一个p标签</p>

<p>我是一个p标签</p>

<p class="hello">我是一个p标签</p>

<p>我是一个p标签</p>

<p>我是一个p标签</p>

</body>

</html>

·效果



## 样式的继承

·后代元素会继承祖先元素上的样式

·利用继承，可以将一些基本的样式设置给祖先元素，所有的后代元素都会继承这些样式

·并不是所有的样式都会被后代元素所继承，如：背景、边框、定位相关的样式

## 选择器的优先级

·当使用不同的选择器选中同一个元素，并且设置相同的样式，这时样式之间产生了冲突，优先级高的优先显示

### 优先级规则

·内联样式

- 1000

·id选择器

- 100

·类和伪类

- 10

·元素选择器

- 1

·通配\*

- 0

·继承的样式，没有优先级

·当选择器中包含多种选择器时，需要将多种选择器的优先级相加进行比较

·但是，选择器优先级计算不会超过他的最大数量级

·如果选择器的优先级一样，则选择靠后的样式

·并集选择器的优先级是单独计算的，不会相加

·可以在样式的最后，添加一个!important，则此时该样式将会获得最高的优先级，但是在开发中尽量避免使用

### 伪类的顺序

·涉及到a的伪类一共有4个

- :link

- :visited

- :hover

- :active

·这四个选择器的优先级是一样的，但是请按以上的顺序写，不然会被覆盖

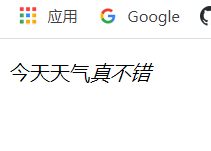
# 文本标签

·<em>

- 斜体

- 表示一段内容的着重点

<p>今天天气<em>真不错</em></p>



·<strong>

- 粗体

- 表示一个内容的重要性

<p><strong>注意：请不要靠近</strong></p>



·这两个标签可以一起使用

·em显示为斜体，而strong显示为粗体

·<i>

- 斜体

·<b>

- 粗体

<p>

<i>我是i标签的内容</i>

<b>我是b标签的内容</b>

</p>

·以上两个标签没有语义，单纯表现

·<small>

- 比父元素字号小一些

- 表示一类细则内容

- 如：合同中小字、网站版权声明等

<p>

我是p标签的内容

<small>我是small标签的内容</small>

</p>



·<cite>

- 表示参考内容

- 比如：书名、歌名、电影名等等



·<q>

- 引号

- 短引用（行内引用）

- 引号不能被选中

<p>

<cite>《论语》</cite>真不错

</p>



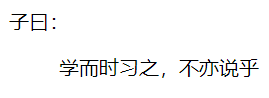
·<blockquote>

- 长引用（块级引用）

<p>

子曰：<blockquote>学而时习之，不亦说乎</blockquote>

</p>



·<sup>

- 上标

<P>2<sup>2</sup></P>



·<sub>

- 下标

<P>H<sub>2</sub>O</P>



·<ins>

- 下划线

- 表示插入的内容

<p>

11+11=<ins>22</ins>

</p>



·<del>

- 删除线

- 表示删除的内容

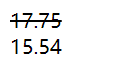
<p>

<del>17.75</del>

<br/>

15.54

</p>



·<pre>

- 预格式标签

- 保留代码原格式

·<code>

- 表示代码

<pre>

<code>

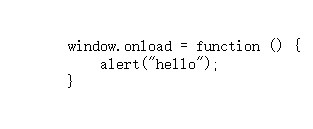
window.onload = function () {

alert("hello");

}

</code>

</pre>



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

</style>

</head>

<body>

<p>今天天气<em>真不错</em></p>

<p><strong>注意：请不要靠近</strong></p>

<p>

<i>我是i标签的内容</i>

<b>我是b标签的内容</b>

</p>

<p>

我是p标签的内容

<small>我是small标签的内容</small>

</p>

<p>

<cite>《论语》</cite>真不错

</p>

<p>

子曰：<q>学而时习之，不亦说乎</q>

</p>

<p>

子曰：<blockquote>学而时习之，不亦说乎</blockquote>

</p>

<P>2<sup>2</sup></P>

<P>H<sub>2</sub>O</P>

<p>

<del>17.75</del>

<br/>

15.54

</p>

<p>

11+11=<ins>22</ins>

</p>

<pre>

<code>

window.onload = function () {

alert("hello");

}

</code>

</pre>

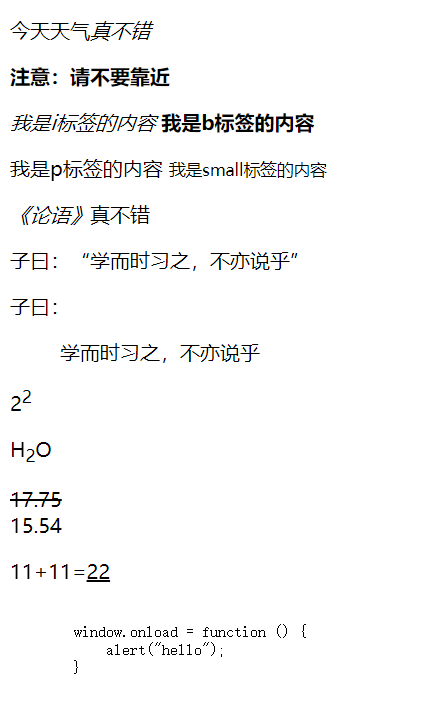
</body>

<script type="text/javascript">

</script>

</html>

·效果



# 列表

## 无序列表

·<ul>标签

- 创建一个无序列表

·<li>标签

- 创建列表项

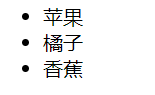
<ul><!--无序列表，默认圆点-->

<li>苹果</li><!--列表项-->

<li>橘子</li>

<li>香蕉</li>

</ul>



·属性

- type

-- 修改无序列表的项目符号

-- disc:默认实心圆点

-- square:实心方块

-- circle:空心圆圈

-- 一般不使用项目符号，如果需要设置项目符号，则可以为li设置背景图片

-- list-style: none去掉项目符号

ul{

list-style: none;

}

·ul和li都是块元素

## 有序列表

·<ol>标签

- 创建一个有序列表

·<li>标签

- 创建列表项

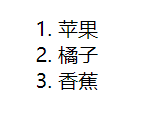
<ol><!--有序列表,默认1234-->

<li>苹果</li><!--列表项-->

<li>橘子</li>

<li>香蕉</li>

</ol>



·属性

- type

-- 指定序号的类型

-- 1 阿拉伯数字排序，默认值

-- a 小写排序

-- A 大写排序

-- i 小写罗马数字

-- I 大写罗马数字

·ol也是块元素

## 定义列表

·对一些词汇或内容进行定义

·<dl>标签

- 创建一个定义列表

·<dt>标签

- 被定义的内容

·<dd>标签

- 对定义内容的描述

<dl>

<dt>武松</dt>

<dd>打老虎很厉害</dd>

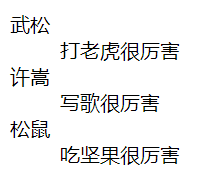
<dt>许嵩</dt>

<dd>写歌很厉害</dd>

<dt>松鼠</dt>

<dd>吃坚果很厉害</dd>

</dl>



## 嵌套列表

·ul，ol，dl列表之间可以互相嵌套

<ul><!--无序列表，默认圆点-->

<li>水果</li><!--列表项-->

<ol><!--嵌套列表-->

<li>苹果</li><!--列表项-->

<li>橘子</li>

<li>香蕉</li>

</ol>

<li>动物</li>

<ul><!--嵌套列表-->

<li>猫</li><!--列表项-->

<li>狗</li>

<li>猪</li>

<li>鱼</li>

</ul>

<dl>

<dt>武松</dt>

<dd>打老虎很厉害</dd>

<dt>许嵩</dt>

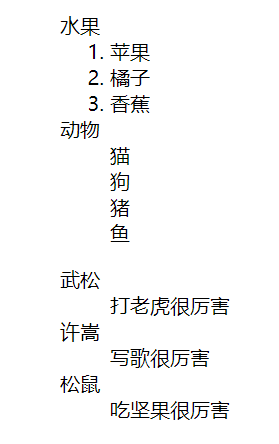
<dd>写歌很厉害</dd>

<dt>松鼠</dt>

<dd>吃坚果很厉害</dd>

</dl>

</ul>



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

ul{

list-style: none;

}

</style>

</head>

<body>

<ul><!--无序列表，默认圆点-->

<li>苹果</li><!--列表项-->

<li>橘子</li>

<li>香蕉</li>

</ul>

<ol><!--有序列表,默认1234-->

<li>苹果</li><!--列表项-->

<li>橘子</li>

<li>香蕉</li>

</ol>

<dl>

<dt>武松</dt>

<dd>打老虎很厉害</dd>

<dt>许嵩</dt>

<dd>写歌很厉害</dd>

<dt>松鼠</dt>

<dd>吃坚果很厉害</dd>

</dl>

<ul><!--无序列表，默认圆点-->

<li>水果</li><!--列表项-->

<ol><!--嵌套列表-->

<li>苹果</li><!--列表项-->

<li>橘子</li>

<li>香蕉</li>

</ol>

<li>动物</li>

<ul><!--嵌套列表-->

<li>猫</li><!--列表项-->

<li>狗</li>

<li>猪</li>

<li>鱼</li>

</ul>

<dl>

<dt>武松</dt>

<dd>打老虎很厉害</dd>

<dt>许嵩</dt>

<dd>写歌很厉害</dd>

<dt>松鼠</dt>

<dd>吃坚果很厉害</dd>

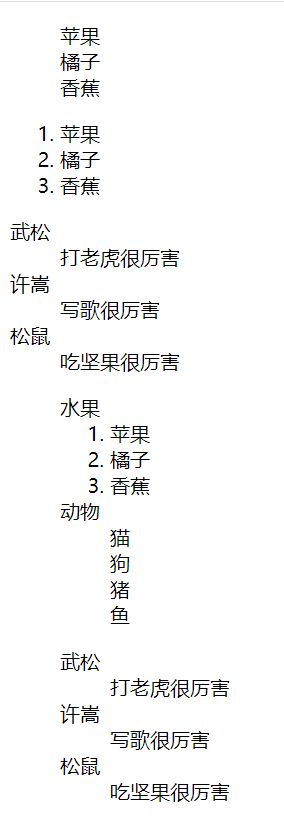
</dl>

</ul>

</body>

</html>

·效果



# 长度单位

·px

- 像素

- 最常用的单位

- 不同的显示器一个像素的大小也不同，显示效果越好越清晰，像素点越小，反之像素点越大

·%

- 百分比

- 根据其父元素的样式计算该值

- 当父元素值发生变化时子元素也会等比例变化

- 创建自适应页面时经常使用百分比作为单位

·em

- 相对于自身元素的字体大小来计算

- 1em = 1font-size

- 当字体大小发生改变，em也会随之改变

- 设置字体相关的样式时使用

·rem

- 相对于根（html）元素的字体大小

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

.box{

width: 100px;

height: 100px;

background-color: cadetblue;

}

.box1{

width: 50%;

height: 50%;

background-color: red;

}

.box2{

font-size: 50px;

width: 5em;

height: 250px;

background-color: red;

}

</style>

</head>

<body>

<div class="box">

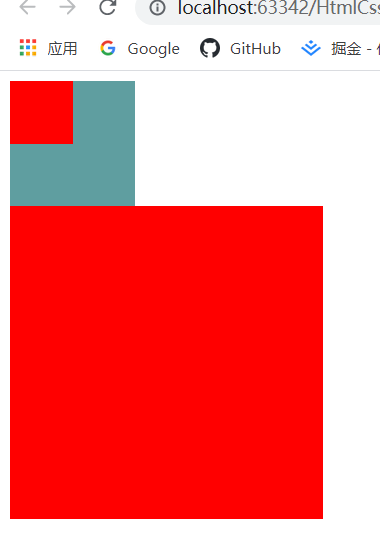
<div class="box1"></div>

</div>

<div class="box2"></div>

</body>

</html>



# 颜色单位

## 单词

·红色

- red

·蓝色

-blue

·绿色

- green

## RGB

·RGB指的是Red Green Blue三原色

- 通过这三种颜色的不同浓度，来表示不同的颜色

- 语法：

-- rgb(红色浓度,绿色浓度,蓝色浓度)

-- 颜色的浓度需要一个0~255之间的值

-- 浓度也可以采用百分数表示

## RGBA

·在RGB的基础上增加一个a表示不透明度

·需要四个值，前三个和rgb一样，第四个表示不透明度

- 透明度需要一个0~1之间的值

## 十六进制

·与RGB原理一样

·使用三组两位的十六进制数组表示一个颜色

·每组取00~ff

·语法

- #红绿蓝

·两位重复的颜色可以简写，如#ff0000可写成#f00

## HSL值 HSLA值

·H

- 色相（0~360）

·S

- 饱和度（0%~100%）

·L

- 亮度（0%~100%）

·A

- 透明度（0~1）

# 字体

## 属性

·color

- 设置字体颜色

color: red;

·font-size

- 设置字体大小

- 浏览器一般默认的文字大小都是16px

- 设置的并不是文字本身的大小

- 在页面中，每个文字都是放在一个看不见的格子中

- 设置的font-size实际上是设置格的高度

- 一般情况文字实际大小要比格子小

- 根据字体的不同，显示效果也不同

- 浏览器最小字体大写是12px

font-size: 20px;

·font-family

- 设置字体

- 当浏览器支持则使用该字体

- 如果浏览器不支持则使用默认字体

font-family: 微软雅黑;

- 同时指定多个字体，多个字体之间使用,分开，当采用多个字体时，浏览器会优先使用前边的字体，如果前边不支持，再逐一使用后边的字体

- 浏览器使用的字体就是用户计算机的字体

- 在开发中尽量不要用不常用的字体

font-family: 微软雅黑,Arial,宋体;

·font-style

- normal

-- 正常

-- 默认值

- italic

-- 斜体

- oblique

-- 倾斜

- 大部分浏览器都不会对倾斜和斜体做区分，一般使用italic

- inherit

-- 继承

-- 继承父元素字体

·font-weight

- normal

-- 默认值

- bold

-- 粗体

- bolder

-- 更粗体

- lighter

-- 细体

- inherit

-- 继承父元素字体

- 自定义

-- 100~900九个值

-- 由于用户计算机没有这么多级别的字体，所以不能达到我们想要的效果，一般不使用

·font-variant

- 是否以小型大写字体显示

- 将所有字母都以大写显示，但是小写字母变成大写后，要比正常的大写字母小

- normal

-- 默认值，不显示

- small-caps

- 显示小型大写字母

- inherit

-- 继承父元素字体

·font

- 同时设置字体所有类型的样式

- 不同的值用空格隔开

- 斜体、加粗、小型大写字母没有顺序要求，可写可不写

- 但是要求文字大小和字体必须写，而且字体必须是最后一个样式，大小必须是倒数第二个样式，否则无效

- 能够提升性能

## 字体分类

·serif（衬线字体）

·sans-serif（非衬线字体）

·monospace（等宽字体）

·cursive（草书字体）

·fantasy（虚幻字体）

·可以将字体设置为大的分类，浏览器会自动选择指定的字体样式并应用样式

·一般会将字体的大分类，指定为font-family中的最后一个字体

<p class="p1">我是p标签</p>

<p style="font-family: Serif;">衬线字体abcdEFG</p>

<p style="font-family: sans-serif;">非衬线字体abcdEFG</p>

<p style="font-family: monospace;">等宽字体abcdEFG</p>

<p style="font-family: cursive;">草书字体abcdEFG</p>

<p style="font-family: fantasy;">虚幻字体abcdEFG</p>



## 行高

·在CSS中并没有直接设置行间距的方式

·通过设置行高间接的设置行间距，行高越大，行间距越大

·使用line-height设置行高

line-height: 40px;

·网页中的文字会默认在行高中垂直居中

·行间距 = 行高 - 字体大小

·可以传一个百分数，则会相对于字体去计算行高

line-height: 200%;

·可以传一个整数，则行高会设置字体大小相应的倍数

line-height: 2;

·对于单行文本来说，将行高设置为和父元素的高度一致，可以使文本垂直居中

·在font中也可以指定行高

font: 20px/30px "微软雅黑";

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

.p1{

width: 400px;

line-height: 40px;

}

.p2{

width: 400px;

line-height: 200%;

}

.p3{

width: 400px;

line-height: 2;

}

.p4{

width: 400px;

font: 20px/30px "微软雅黑";

}

</style>

</head>

<body>

<p class="p1">我是p标签我是p标签我是p标签我是p标签我是p标签我是p标签我是p标签我是p标签</p>

<p class="p2">我是p标签我是p标签我是p标签我是p标签我是p标签我是p标签我是p标签我是p标签</p>

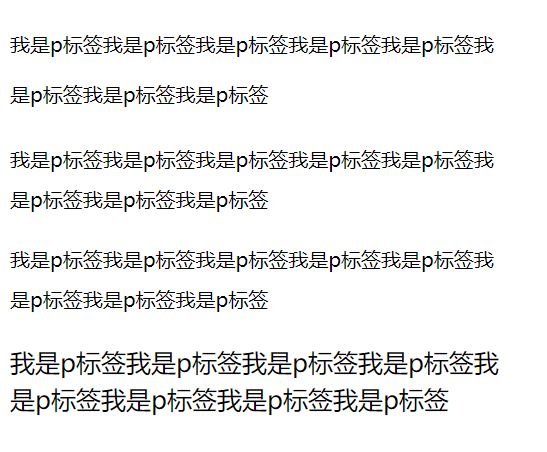
<p class="p3">我是p标签我是p标签我是p标签我是p标签我是p标签我是p标签我是p标签我是p标签</p>

<p class="p4">我是p标签我是p标签我是p标签我是p标签我是p标签我是p标签我是p标签我是p标签</p>

</body>

</html>

·效果



## 文本样式

·text-transform

- none

-- 默认值

- capitalize

-- 文本中的每个单词以大写字符开头

<p style="text-transform: capitalize">每个单词首字符大写 hello world</p>

- uppercase

-- 全部转大写

<p style="text-transform: uppercase">全部转大写 hello world</p>

- lowercase

-- 全部转小写

<p style="text-transform: lowercase">全部转小写 hELlo woRLd</p>

- inherit

-- 继承父元素

·text-decoration

- none

-- 默认值

- underline

-- 下划线

<p style="text-decoration: underline">下划线 hello world</p>

- overline

-- 上划线

<p style="text-decoration: overline">上划线 hello world</p>

- line-through

-- 删除线

<p style="text-decoration: line-through">删除线 hello world</p>

- blink

-- 闪烁文本

- inherit

-- 继承父元素

·a标签超链接默认添加下划线，如果要去除，则将text-decoration设置为none

·letter-spacing（间距）

- 字间距

<p style="letter-spacing: 10px">字间距 hello world</p>

·word-spacing

- 单词间距

<p style="word-spacing: 10px">单词间距 hello world</p>

·text-align

- left

-- 左对齐 默认值

- center

-- 居中对齐

<p style="text-align: center">男：雨后江岸天破晓 老舟新客知多少 远山见竹林芳草 晨风抚绿了芭蕉 女：寒梅落尽把冬了 衔春的燕想归巢 沿途的景 牵挂的人 两情迢迢 男：柳叶桨溅桃花浪 汀州里鹤眺远方 女：饮一盏岁月留香 唱一曲往事飞扬 和：山水间歌声回荡 回荡思念的滚烫 去年的家书两行 读来又热了眼眶 云水边静沐暖阳 烟波里久违的故乡 别来无恙 你在心上</p>

- right

-- 右对齐

- justify（使每行排齐）

-- 两端对齐

·text-indent（缩进）

- 首行缩进

<p style="text-indent: 20px">男：雨后江岸天破晓 老舟新客知多少 远山见竹林芳草 晨风抚绿了芭蕉</p>

- 缩进两个字体

<p style="text-indent: 2em">男：雨后江岸天破晓 老舟新客知多少 远山见竹林芳草 晨风抚绿了芭蕉 女：寒梅落尽把冬了 衔春的燕想归巢 沿途的景 牵挂的人 两情迢迢 男：柳叶桨溅桃花浪 汀州里鹤眺远方 女：饮一盏岁月留香 唱一曲往事飞扬 和：山水间歌声回荡 回荡思念的滚烫 去年的家书两行 读来又热了眼眶 云水边静沐暖阳 烟波里久违的故乡 别来无恙 你在心上</p>

- 可以设置负值将不想被别人看见的文字隐藏

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

</style>

</head>

<body>

<p style="text-transform: capitalize">每个单词首字符大写 hello world</p>

<p style="text-transform: uppercase">全部转大写 hello world</p>

<p style="text-transform: lowercase">全部转小写 hELlo woRLd</p>

<p style="text-decoration: underline">下划线 hello world</p>

<p style="text-decoration: overline">上划线 hello world</p>

<p style="text-decoration: line-through">删除线 hello world</p>

<p style="text-decoration: blink">闪烁文本 hello world</p>

<p style="letter-spacing: 10px">字间距 hello world</p>

<p style="word-spacing: 10px">单词间距 hello world</p>

<p style="text-align: center">男：雨后江岸天破晓 老舟新客知多少 远山见竹林芳草 晨风抚绿了芭蕉 女：寒梅落尽把冬了 衔春的燕想归巢 沿途的景 牵挂的人 两情迢迢 男：柳叶桨溅桃花浪 汀州里鹤眺远方 女：饮一盏岁月留香 唱一曲往事飞扬 和：山水间歌声回荡 回荡思念的滚烫 去年的家书两行 读来又热了眼眶 云水边静沐暖阳 烟波里久违的故乡 别来无恙 你在心上</p>

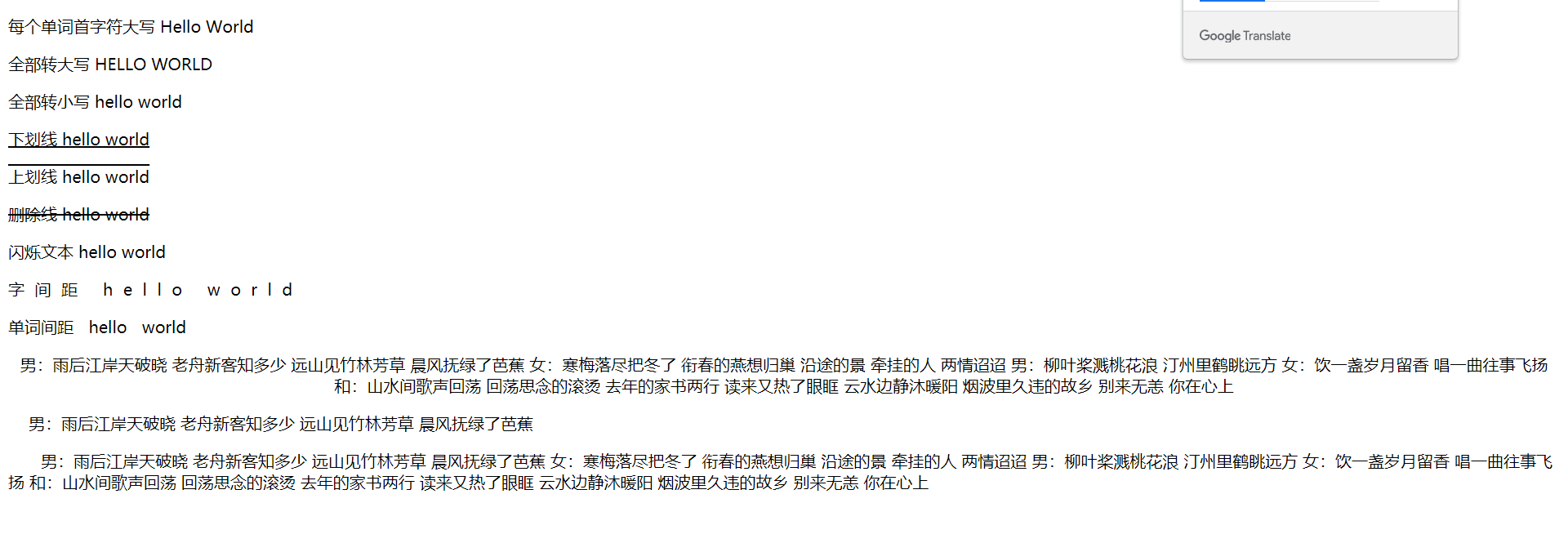
<p style="text-indent: 20px">男：雨后江岸天破晓 老舟新客知多少 远山见竹林芳草 晨风抚绿了芭蕉</p>

<p style="text-indent: 2em">男：雨后江岸天破晓 老舟新客知多少 远山见竹林芳草 晨风抚绿了芭蕉 女：寒梅落尽把冬了 衔春的燕想归巢 沿途的景 牵挂的人 两情迢迢 男：柳叶桨溅桃花浪 汀州里鹤眺远方 女：饮一盏岁月留香 唱一曲往事飞扬 和：山水间歌声回荡 回荡思念的滚烫 去年的家书两行 读来又热了眼眶 云水边静沐暖阳 烟波里久违的故乡 别来无恙 你在心上</p>

</body>

</html>

效果



# 盒模型

·CSS处理网页时，认为每个元素都包含在一个不可见的盒子里

·只需要将相应的盒子摆放到网页中相应的位置即可完成网页的布局

·一个盒子可分成4个部分

- 内容区（content）

- 内边距（padding）

- 边框（border）

- 外边距（margin）

## 边框

·要为元素设置边框必须指定三个样式

- border-width

-- 边框的宽度

-- 如果设置四个值,则会分别设置上右下左的边框宽度

-- 如果只设置三个值，则会分别设置给上 左右 下

-- 如果只设置两个值，则会分别设置给上下 左右

-- 如果只设置一个值，则四边全都是

-- 还适用于padding和margin属性

border-width: 10px;

- border-color

-- 边框的颜色

-- 如果设置四个值,则会分别设置上右下左的边框颜色

-- 如果只设置三个值，则会分别设置给上 左右 下

-- 如果只设置两个值，则会分别设置给上下 左右

-- 如果只设置一个值，则四边全都是

border-color: red;

- border-style

-- 边框的样式

-- none：没有

-- solid：实线

-- dotted：点状边框

-- dashed：虚线

-- double：双线

-- 如果设置四个值,则会分别设置上右下左的边框样式

-- 如果只设置三个值，则会分别设置给上 左右 下

-- 如果只设置两个值，则会分别设置给上下 左右

-- 如果只设置一个值，则四边全都是

border-style: solid;

·大部分的浏览器中，边框的宽度和颜色都是有默认值的，而边框的样式默认值都为none

·width和height属性只是设置盒子的内容区大小，而不是盒子的整个大小

·border-xxx-width

- xxx的值可能是top right bottom left

-专门用来设置指定边的宽度

·border-xxx-color

- xxx的值可以是top right bottom left

-专门用来设置指定边的颜色

·border-xxx-style

- xxx的值可以是top right bottom left

-专门用来设置指定边的样式

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

.box1{

/\*设置内容区的宽度和高度\*/

width: 100px;

height: 100px;

/\*设置背景颜色\*/

background-color: cadetblue;

border-width: 10px;

border-color: red;

border-style: solid;

}

.box2{

/\*设置内容区的宽度和高度\*/

width: 100px;

height: 100px;

/\*设置背景颜色\*/

background-color: cadetblue;

/\*分别设置上右下左的边框宽度

如果只设置三个值，则会分别设置给上 左右 下

如果只设置两个值，则会分别设置给上下 左右

\*/

border-width: 10px 20px 30px 40px;

border-color: red;

border-style: solid;

}

</style>

</head>

<body>

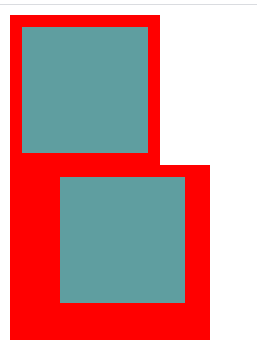
<div class="box1"></div>

<div class="box2"></div>

</body>

</html>

·效果



## 边框简写

·border

- 通过它同时设置四边的样式、宽度、颜色

- 无任何顺序要求

- 不能分别指定每个边

border: red 10px solid;

·border-xxx

- xxx的值可能是top right bottom left

- 通过它同时设置指定边的样式、宽度、颜色

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

.box1{

/\*设置内容区的宽度和高度\*/

width: 100px;

height: 100px;

/\*设置背景颜色\*/

background-color: cadetblue;

border: red 10px solid;

}

</style>

</head>

<body>

<div class="box1"></div>

</body>

</html>

## 内边距

·内容区与盒子边框之间的距离

·padding-xxx

- xxx的值可以是top right bottom left

- 通过它设置指定边的内边距

padding-top: 50px;

·padding（填充）

- 同时设置四个边的内边距

- 如果设置四个值,则会分别设置上右下左的内边距

- 如果只设置三个值，则会分别设置给上 左右 下

- 如果只设置两个值，则会分别设置给上下 左右

- 如果只设置一个值，则四边全都是

padding: 50px;

·内边距会影响盒子的可见框的大小

·元素的背景会影响内容区和内边距

·盒子的大小(可见大小) = 内容区 + 内边距 + 边框

## 外边距

·盒子与盒子之间的距离

·不会影响盒子可见框的大小，而是会影响盒子的位置

·margin-xxx

- xxx的值可能是top right bottom left

- 通过它设置指定边的外边距

margin-top: 100px;

·由于页面中的元素都是左上角摆放的

·所以设置上和左外边距时，会导致盒子的位置发生改变

·而设置右和下外边距时，会改变其他盒子的位置

·外边距也可以定为负值，往反方向移动

·margin（边缘）

- 同时设置四个边的外边距

- 如果设置四个值,则会分别设置上右下左的外边距

- 如果只设置三个值，则会分别设置给上 左右 下

- 如果只设置两个值，则会分别设置给上下 左右

- 如果只设置一个值，则四边全都是

margin: 50px auto;

·可以设置为auto

- 一般只设置水平方向的auto

- 只指定左外边距和右外边距其中一个的margin为auto则会将外边距设置为最大值

- 垂直方向设置为auto，则外边距默认就是0

- 同时将左外边距和右外边距同时设置为auto时，则会将两侧外边距设置为相同的值，即在父元素中水平居中，常用

- 当元素的宽度或高度的值是auto时，此时指定内边距不会影响可见框的大小，而是会自动修改宽度，以适应内边距

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

.box1{

/\*设置内容区的宽度和高度\*/

width: 100px;

height: 100px;

/\*设置背景颜色\*/

background-color: cadetblue;

border: red 10px solid;

padding: 50px;

/\*margin-top: 100px;\*/

margin: 50px auto;

}

</style>

</head>

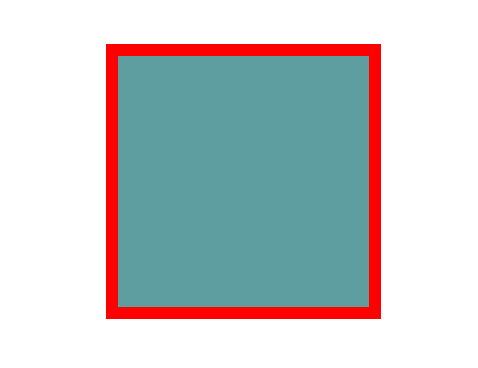
<body>

<div class="box1"></div>

</body>

</html>

·效果



## 垂直外边距的重叠

·在网页中**垂直方向**的**相邻**外边距会发生外边距的重叠

·如果兄弟元素之间的垂直方向的相邻，则外边距会取最大值

·如果父子元素之间的垂直方向的相邻，则外边距会设置给父元素

·只想移动子元素的解决方法

- 让他们不相邻

- 为父元素设置边框

- 为父元素设置内边距

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

.box1{

/\*设置内容区的宽度和高度\*/

width: 100px;

height: 100px;

/\*设置背景颜色\*/

background-color: cadetblue;

margin-bottom: 100px;

}

.box2{

/\*设置内容区的宽度和高度\*/

width: 100px;

height: 100px;

/\*设置背景颜色\*/

background-color: burlywood;

margin-top: 50px;

}

.box3{

/\*设置内容区的宽度和高度\*/

width: 200px;

height: 170px;

/\*设置背景颜色\*/

background-color: rebeccapurple;

padding-top: 30px;

}

.box4{

/\*设置内容区的宽度和高度\*/

width: 100px;

height: 100px;

/\*设置背景颜色\*/

background-color: chartreuse;

/\*为子元素设置上外边距，使子元素下移，但是父元素也跟着往下移了\*/

/\*margin-top: 30px;\*/

}

</style>

</head>

<body>

<div class="box3">

<div class="box4"></div>

</div>

<div class="box1"></div>

<div class="box2"></div>

</body>

</html>

·效果



## 浏览器默认样式

·浏览器会为页面设置默认样式，让页面在没有样式时更好的显示

·所有会为很多元素设置一些默认的margin和padding

·正常情况下不需要使用默认值

·在实际开发时会将默认的margin和padding去掉

\*{

margin: 0;

padding: 0;

}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

\*{

margin: 0;

padding: 0;

}

.box1{

/\*设置内容区的宽度和高度\*/

width: 100px;

height: 100px;

/\*设置背景颜色\*/

background-color: cadetblue;

}

p{

background-color: cadetblue;

}

</style>

</head>

<body>

<div class="box1"></div>

<p>我是p标签</p>

<ul><!--无序列表，默认圆点-->

<li>苹果</li><!--列表项-->

<li>橘子</li>

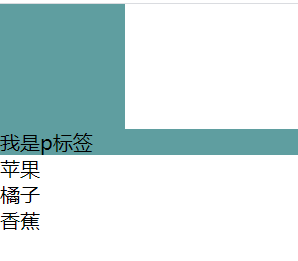
<li>香蕉</li>

</ul>

</body>

</html>

·效果



## 内联元素的盒模型

·内联元素不能设置内容区的宽度（width）和高度(height)

·内联元素可以设置水平方向的内边距

·内联元素可以设置垂直方向的内边距，但是不会影响页面布局，可能会把其他元素覆盖

·内联元素可以设置边框，但是垂直方向的边框不会影响页面布局，可能会把其他元素覆盖

·内联元素可以设置水平方向的外边距

·内联元素不能设置垂直方向的外边距

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

span{

background-color: cadetblue;

}

.box1{

width: 100px;

height: 100px;

background-color: rebeccapurple;

}

.s1{

/\*内联元素不能设置内容区的宽度和高度\*/

/\*width: 100px;\*/

/\*height: 100px;\*/

padding-left: 100px;

padding-right: 100px;

/\*padding-top: 50px;\*/

/\*padding-bottom: 50px;\*/

/\*border: 100px red solid;\*/

margin-left: 100px;

margin-right: 100px;

/\*margin-top: 100px;\*/

/\*margin-bottom: 100px;\*/

}

.s2{

margin-left: 100px;

}

</style>

</head>

<body>

<span class="s1">我是span标签</span>

<span class="s2">我是span标签</span>

<span>我是span标签</span>

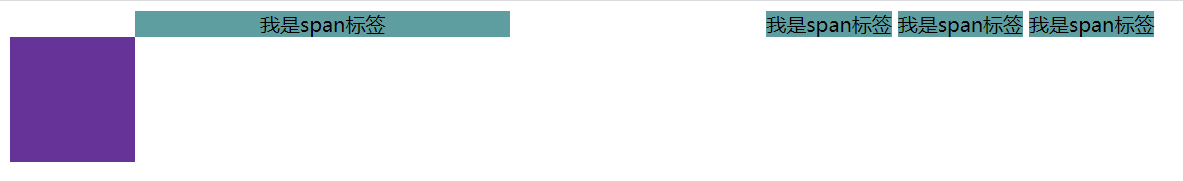
<span>我是span标签</span>

<div class="box1"></div>

</body>

</html>

·效果



# display和visibility

## display

·修改元素类型

·可以将块元素改成内联元素，也可以将内联元素改成块元素

·可选值

- inline

-- 将一个元素作为内联元素显示

- block

-- 将一个元素作为块元素显示

- inline-block

-- 将一个元素转换为行内块元素

-- 可以使一个元素既有内联元素的特点，又有块元素的特点

-- 既可以设置宽高，又不会独占一行

- table

- 将一个元素设置为表格显示

- none

-- 隐藏元素，并且元素不会占用位置

- flex

-- 将一个元素设置为弹性盒子

## visibility（可见度）

·visible

- 元素可见，默认值

·hidden

- 元素不可见

## display:none与visibility:hidden的区别

·使用display:none方式隐藏的元素，不会在页面中显示，并且不占据页面位置（直接没了，也就是消失，但是结构还在）

·使用visibility:hidden方式隐藏的元素，不会在页面中显示，但是会占据页面位置（还在，只是隐藏了）

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

a{

/\*将内联元素变成块元素\*/

display: block;

/\*设置内容区的宽度和高度\*/

width: 200px;

height: 100px;

background-color: cadetblue;

}

.box1{

/\*将元素消失\*/

display: none;

/\*设置内容区的宽度和高度\*/

width: 100px;

height: 100px;

background-color: rebeccapurple;

}

.box2{

/\*将元素隐藏\*/

visibility: hidden;

/\*设置内容区的宽度和高度\*/

width: 100px;

height: 100px;

background-color: rebeccapurple;

}

</style>

</head>

<body>

<div class="box1"></div>

<div class="box2"></div>

<a href="#">这是一个超链接</a>

</body>

</html>

·效果



# overflow（溢出）

·子元素默认是存在父元素的内容区中

·但如果子元素的大小超过父元素的内容区，则超过的大小会在父元素以外显示

·超出父元素的内容称为溢出内容

·通过overflow为父元素设置如何处理溢出内容

·visible（可见）

- 溢出内容会在父元素外显示，默认值

·hidden（隐藏）

- 溢出内容不会在父元素外显示

·scroll（滚屏）

- 溢出内容会被修剪，但是有滚动条可查看其余内容

- 该属性无论内容是否溢出，都会显示双向滚动条

·auto

- 根据需求自动添加滚动条，比scroll更加智能

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

.box1{

/\*设置内容区的宽度和高度\*/

width: 200px;

height: 200px;

background-color: rebeccapurple;

overflow: hidden;

}

.box2{

/\*设置内容区的宽度和高度\*/

width: 100px;

height: 500px;

background-color: cadetblue;

}

</style>

</head>

<body>

<div class="box1">

<div class="box2"></div>

</div>

</body>

</html>

·效果



# 文档流

·文档流处在网页的最底层，它表示的是一个页面中的位置

·我们所创建的元素默认都处在文档中

## 元素在文档流中的特点

### 块元素

·在文档流中独占一行，会默认自上向下排列

·在文档流中默认宽度是父元素的100%

·在文档流中默认高度默认被内容区撑开

### 内联元素

·在文档流中只占自身的大小，会默认从左向右排列

·会自动换行

·在文档流中的宽度和高度默认都被内容撑开

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

.box1{

/\*设置内容区的宽度和高度\*/

width: 100px;

height: 100px;

background-color: rebeccapurple;

}

</style>

</head>

<body>

<div class="box1"></div>

<div class="box1" style="background-color: cadetblue"></div>

<span style="background-color: cadetblue">我是span标签</span>

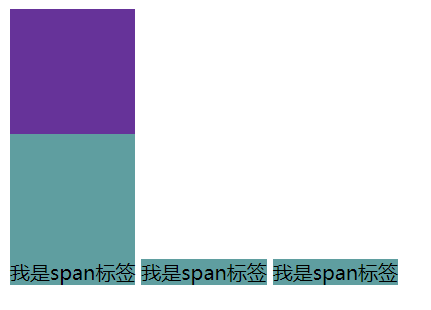
<span style="background-color: cadetblue">我是span标签</span>

<span style="background-color: cadetblue">我是span标签</span>

</body>

</html>

·效果



# 浮动

·把浮动想象成气球

·如果希望块元素在页面中水平排列，可以使块元素脱离文档流

·使用float使元素浮动，从而脱离文档流

·当为一个元素设置浮动以后，元素会立即脱离文档流，还在文档流中的元素会重新按文档流摆放位置

·元素浮动以后会尽量向左上或者右上漂浮，直到遇到**父元素的内容区边界**或**没有浮动的块元素**或**其它浮动元素**

·如果浮动元素上边是一个没有浮动的块元素，则浮动元素不会超过块元素

·浮动的元素不会超过它的相邻兄弟元素，最多顶部对齐

## float

·none

- 元素不浮动，默认值

·left

- 元素向左浮动

·right

- 元素向右浮动

·inherit

- 继承父元素

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

.box1{

/\*设置内容区的宽度和高度\*/

width: 100px;

height: 100px;

background-color: cadetblue;

float: left;

}

.box2{

/\*设置内容区的宽度和高度\*/

width: 200px;

height: 200px;

background-color: rebeccapurple;

float: left;

}

.box3{

/\*设置内容区的宽度和高度\*/

width: 100px;

height: 100px;

background-color: burlywood;

float: right;

}

</style>

</head>

<body>

<div class="box1"></div>

<div class="box2"></div>

<div class="box3"></div>

</body>

</html>

·效果



·浮动的元素不会盖住文字，文字会环绕在浮动元素的周围

·可以通过浮动来设置文字环绕图片浮动的效果

·块元素设置浮动后，元素将不再有文档流的默认宽度，高度和宽度都会被内容撑开

·内联元素设置浮动以后会变成块元素，可以直接设置宽高

# 高度塌陷问题

·在文档流中，父元素的高度默认是被子元素撑开的，也就是子元素多高，父元素就是多高

·但是当为子元素设置浮动以后，子元素会完全脱离文档流

·此时将会导致子元素无法撑起父元素的高度，从而导致父元素的高度塌陷

·由于父元素的高度塌陷，则父元素下方的所有元素都会向上移动，这样将会导致页面布局混乱

·开发中一定要避免高度塌陷的问题

·可以将父元素的高度写死，以避免塌陷问题出现，但是父元素的高度将不能自动适应子元素的高度，这种方案不推荐使用

## 解决

·根据W3C的标准，在页面中元素都有一个隐含的属性叫做Block Formatting（格式化） Context，简称BFC

·该属性可以设置打开或者关闭，默认是关闭的

·当开启元素的BFC后，元素将会具有如下特性

- 父元素的垂直外边距不会和子元素重叠

- 开启BFC的元素不会被浮动元素覆盖

- 开启BFC的父元素可以包含浮动的子元素

### 开启BFC的方式

·设置元素浮动

- 使用这种方式开启，可以撑开父元素，但是会导致父元素的默认宽度丢失

- 而且使用这种方式也会导致下边的元素上移，不能解决问题

·设置元素绝对定位

·设置元素为inline-block

- 可以解决问题，但是会导致父元素的默认宽度丢失，不推荐使用

·将元素的overflow设置为一个非visible的值

- 副作用最小的开启BFC方式，推荐使用

overflow: hidden;

·但是ie6及以下浏览器并不支持BFC

·在ie6及以下浏览器中虽然没有BFC，但是具有另一个隐含的属性hasLayout，该属性的作用和BFC类似，可以通过开启hasLayout来解决问题

·开启hasLayout的方式

- 直接将元素的zoom设置为1

- zoom表示方法，写几就将元素方法几倍

- 如果元素指定了一个宽度，则会默认开启hasLayout

zoom: 1;

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

.box1{

border: 10px red solid;

/\*height: 100px;\*/

/\*float: left;\*/

/\*display: inline-block;\*/

overflow: hidden;

zoom: 1;

}

.box2{

width: 100px;

height: 100px;

/\*为子元素设置浮动\*/

float: left;

background-color: cadetblue;

}

.box3{

height: 100px;

background-color: burlywood;

}

</style>

</head>

<body>

<div class="box1">

<div class="box2"></div>

</div>

<div class="box3"></div>

</body>

</html>

·效果



## 导航条练习

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>导航条练习</title>

<style type="text/css">

/\*清除默认样式\*/

\*{

margin: 0;

padding: 0;

}

.nav{

/\*设置元素居中\*/

margin: 50px auto;

/\*清除无序列表项目符号\*/

list-style: none;

background-color: cadetblue;

width: 1000px;

/\*解决高度塌陷\*/

overflow: hidden;

}

.nav li{

float: left;

width: 25%;

}

.nav a{

/\*将a转换为块元素\*/

display: block;

width: 100%;

/\*设置文字居中\*/

text-align: center;

/\*设置一个上下内边距\*/

padding: 5px 0;

/\*去除a的下划线\*/

text-decoration: none;

/\*设置字体加粗\*/

font-weight: bold;

/\*设置字体颜色\*/

color: white;

}

.nav a:hover{

/\*设置鼠标移入颜色改变的效果\*/

background-color: rebeccapurple;

}

</style>

</head>

<body>

<ul class="nav">

<li><a href="#">首页</a></li>

<li><a href="#">新闻</a></li>

<li><a href="#">联系</a></li>

<li><a href="#">关于</a></li>

</ul>

</body>

</html>

·效果



## 解决高度塌陷的最终方案

### 兄弟元素之间

·希望清除掉其他元素浮动对当前元素产生的影响，可以使用clear来完成功能

·clear

- none

-- 不清除浮动，默认值

- left

-- 清除左侧浮动元素对当前元素的影响

- right

-- 清除右侧浮动元素对当前元素的影响

- both

-- 清除对它影响最大的浮动元素的影响

·清除浮动以后，元素会回到浮原来的位置

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

.box1{

height: 100px;

width: 100px;

background-color: rebeccapurple;

float: left;

}

.box2{

height: 200px;

width: 200px;

background-color: cadetblue;

/\*清除浮动\*/

clear: left;

}

.box3{

height: 300px;

width: 300px;

background-color: burlywood;

}

</style>

</head>

<body>

<div class="box1"></div>

<div class="box2"></div>

<div class="box3"></div>

</body>

</html>

·效果



### 父子元素之间一

·直接在高度塌陷的父元素的最后，添加一个空白的div

·然后清除该div的浮动

·基本没有副作用

·使用这种方式虽然可以解决问题，但是会在页面中添加多余的结构

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

.box1{

border: 10px red solid;

}

.box2{

width: 100px;

height: 100px;

background-color: cadetblue;

float: left;

}

</style>

</head>

<body>

<div class="box1">

<div class="box2"></div>

<div style="clear: both"></div>

</div>

</body>

</html>

·效果



### 父子元素之间二

·通过after伪类，选中box1的后边

·添加一个空白的块元素，并清除浮动

·不会在页面添加多余的div

·最推荐的方式，几乎没有缺点

·IE6中不支持after伪类，还需要使用hasLayout来处理

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

.box1{

border: 10px red solid;

}

.box2{

width: 100px;

height: 100px;

background-color: cadetblue;

float: left;

}

/\*谱图浏览器解决方案\*/

.clearfix:after{

/\*在父元素后边添加一个内容\*/

content: "";

/\*将它转换成块元素\*/

display: block;

/\*清除两侧浮动\*/

clear: both;

}

/\*ie6解决方案\*/

.clearfix{

zoom: 1;

}

</style>

</head>

<body>

<div class="box1 clearfix">

<div class="box2"></div>

</div>

</body>

</html>

·效果



# 练习：开班信息

·效果



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>开班信息</title>

<style type="text/css">

\*{

/\*清除默认样式\*/

margin: 0;

padding: 0;

}

/\*统一页面的字体\*/

body{

font: 12px/1 宋体;

}

/\*标题区\*/

/\*设置outer的大小\*/

.outer{

width: 300px;

/\*height: 473px;\*/

/\*background-color: cadetblue;\*/

margin: 50px auto;

}

/\*设置title\*/

.title{

/\*设置顶部边框\*/

border-top: 2px #019e8b solid;

height: 36px;

background-color: #f5f5f5;

/\*将行高设置为title高度,让字体垂直居中\*/

line-height: 36px;

/\*让文字靠里面一些\*/

padding: 0px 22px 0px 16px;

}

/\*title的超链接向右浮动\*/

.title a{

float: right;

color: red;

}

.title h3{

font: 16px/36px "微软雅黑";

}

/\*内容区\*/

.content{

/\*设置内容区边框\*/

border: 1px solid #deddd9;

/\*设置内边距\*/

padding: 0px 28px 0px 20px;

}

/\*设置内容区的超链接\*/

.content a{

color: black;

/\*去除下划线\*/

text-decoration: none;

font-size: 12px;

}

/\*为超链接添加鼠标移入效果\*/

.content a:hover{

color: red;

/\*为超链接添加下划线\*/

text-decoration: underline;

}

/\*设置ul的样式\*/

.content ul{

/\*去除项目符号\*/

list-style: none;

/\*设置虚线底部边框\*/

border-bottom: 1px #deddd9 dashed;

}

/\*虚线最后一个ul的虚线底部边框\*/

.content .no-border{

border: none;

}

/\*设置内容中的红色字体\*/

.content .red{

color: red;

/\*字体加粗\*/

font-weight: bold;

}

.content .right{

/\*设置向右浮动\*/

float: right;

}

/\*内容区的标题\*/

.content h3{

margin-top: 14px;

margin-bottom: 16px;

}

/\*设置内容中的li\*/

.content li{

margin-bottom: 15px;

}

</style>

</head>

<body>

<!--创建外层div,容纳整个内容-->

<div class="outer">

<!-- 开班信息的头部 -->

<div class="title">

<a href="#">21年面授开班计划</a>

<h3>近期开班</h3>

</div>

<!--开班信息的主要内容-->

<div class="content">

<h3><a href="#">JavaEE+云计算-全程就业班</a></h3>

<ul>

<li>

<a href="#" class="right"><span class="red">预约报名</span></a>

<a href="#"> 开班时间:<span class="red">2021-04-27</span></a>

</li>

<li>

<a href="#" class="right"><span class="red">无座,名额爆满</span></a>

<a href="#"> 开班时间:<span class="red">2021-04-27</span></a>

</li>

<li>

<a href="#" class="right"><span>开班盛况</span></a>

<a href="#"> 开班时间:<span>2021-04-27</span></a>

</li>

<li>

<a href="#" class="right"><span>开班盛况</span></a>

<a href="#"> 开班时间:<span>2021-04-27</span></a>

</li>

<li>

<a href="#" class="right"><span>开班盛况</span></a>

<a href="#"> 开班时间:<span>2021-04-27</span></a>

</li>

</ul>

<h3><a href="#">Android+人工智能-全程就业班</a></h3>

<ul>

<li>

<a href="#" class="right"><span class="red">预约报名</span></a>

<a href="#"> 开班时间:<span class="red">2021-04-27</span></a>

</li>

<li>

<a href="#" class="right"><span>开班盛况</span></a>

<a href="#"> 开班时间:<span>2021-04-27</span></a>

</li>

<li>

<a href="#" class="right"><span>开班盛况</span></a>

<a href="#"> 开班时间:<span>2021-04-27</span></a>

</li>

<li>

<a href="#" class="right"><span>开班盛况</span></a>

<a href="#"> 开班时间:<span>2021-04-27</span></a>

</li>

</ul>

<h3><a href="#">前端+HTML5+全程就业班</a></h3>

<ul class="no-border">

<li>

<a href="#" class="right"><span class="red">预约报名</span></a>

<a href="#"> 开班时间:<span class="red">2021-04-27</span></a>

</li>

<li>

<a href="#" class="right"><span>开班盛况</span></a>

<a href="#"> 开班时间:<span>2021-04-27</span></a>

</li>

</ul>

</div>

</div>

</body>

</html>

# 定位

·将指定元素摆放到页面的任意位置

·通过position属性来设置元素的定位

·position

- static

-- 没有定位，默认值

- relative

-- 相对定位

- fixed

-- 固定定位，相对于浏览器窗口

- absolute

-- 绝对定位，离他最近并开启了定位的祖先元素进行定位

## 相对定位

·当开启了元素的相对定位，而不设置偏移量，元素不会发生任何变化

·相对定位是相对于元素在文档流中原来的位置进行定位

·可通过 left right top bottom四个属性来设置元素的偏移量

·相对定位的元素不会脱离文档流

·会使元素提升一个层级，可能覆盖原本层级的元素

·通常偏移量只需要设置left和top

·相对定位不会改变元素的性质，块还是块，内联还是内联

·效果



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

.box1{

height: 200px;

background-color: cadetblue;

position: relative;

}

.box2{

width: 200px;

height: 200px;

background-color: rebeccapurple;

/\*相对定位\*/

position: relative;

/\*设置偏移量\*/

left: 200px;

top: 200px;

}

.box3{

width: 200px;

height: 200px;

background-color: burlywood;

}

.s1{

position: relative;

width: 200px;

height: 200px;

background-color: chartreuse;

}

</style>

</head>

<body>

<div class="box1"></div>

<div class="box2"></div>

<div class="box3"></div>

<span class="s1">我是一个span标签</span>

</body>

</html>

## 绝对定位

·开启绝对定位会使元素脱离文档流

·开启绝对定位以后，如果不设置偏移量，则元素的位置不会发生变化

·可通过 left right top bottom四个属性来设置元素的偏移量

·离他最近并开启了定位的祖先元素进行定位

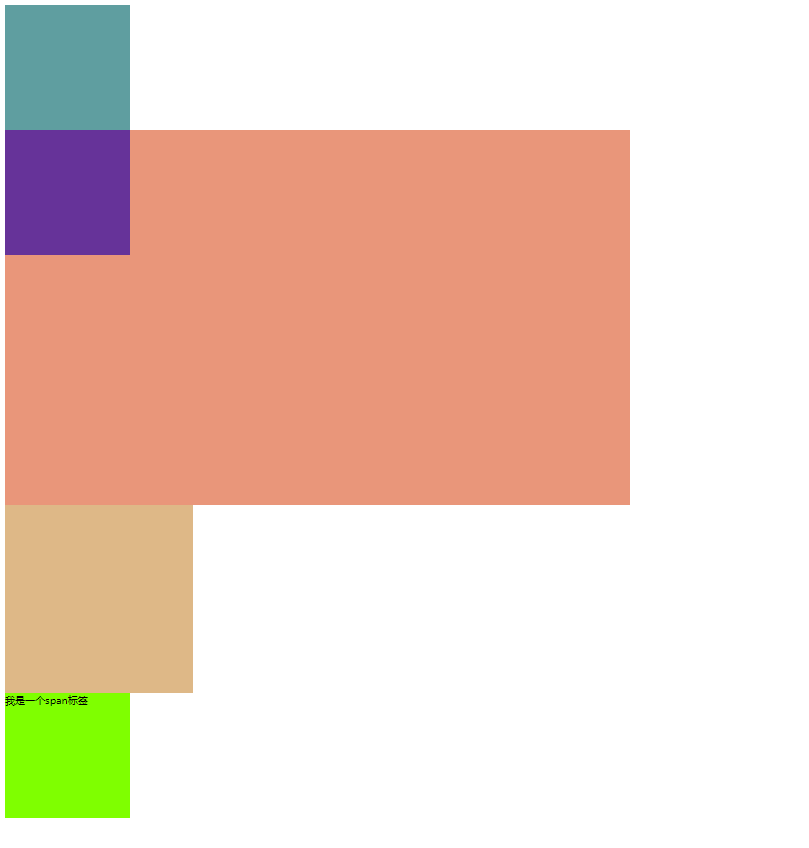
·如果所有的祖先元素都没有开启定位，则相对于浏览器窗口进行定位

·一般开启了子元素的绝对定位，都会开启父元素的相对定位

·会使元素提升一个层级，可能覆盖原本层级的元素

·绝对定位不会改变元素的性质，块元素的宽度和高度会被内容撑开，内联变为块元素

·效果



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

.box1{

width: 200px;

height: 200px;

background-color: cadetblue;

}

.box2{

width: 200px;

height: 200px;

background-color: rebeccapurple;

/\*绝对定位\*/

position: absolute;

/\*设置偏移量\*/

left: 0px;

top: 0px;

/\*top: 200px;\*/

}

.box3{

width: 300px;

height: 300px;

background-color: burlywood;

}

.box4{

width: 1000px;

height: 600px;

background-color: darksalmon;

position: relative;

}

.s1{

position: absolute;

width: 200px;

height: 200px;

background-color: chartreuse;

}

</style>

</head>

<body>

<div class="box1"></div>

<div class="box4">

<div class="box2"></div>

</div>

<div class="box3"></div>

<span class="s1">我是一个span标签</span>

</body>

</html>

## 固定定位

·固定定位也是一种绝对定位，它的大部分特点都和绝对定位一样

·区别

- 永远都会相对于浏览器窗口进行定位

·固定定位会使元素固定在浏览器窗口某个位置，不会随窗口的滚动条滚动

·IE6不支持固定定位

·效果



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

.box1{

width: 200px;

height: 200px;

background-color: cadetblue;

}

.box2{

width: 200px;

height: 200px;

background-color: rebeccapurple;

/\*固定定位\*/

position: fixed;

/\*设置偏移量\*/

left: 0px;

top: 0px;

/\*top: 200px;\*/

}

.box3{

width: 300px;

height: 300px;

background-color: burlywood;

}

.box4{

width: 1000px;

height: 600px;

background-color: darksalmon;

position: relative;

}

</style>

</head>

<body>

<div class="box1"></div>

<div class="box4">

<div class="box2"></div>

</div>

<div class="box3"></div>

</body>

</html>

## 元素的层级

·如果定位元素的层级一样，则（结构）下边的元素会盖住（结构）上边元素

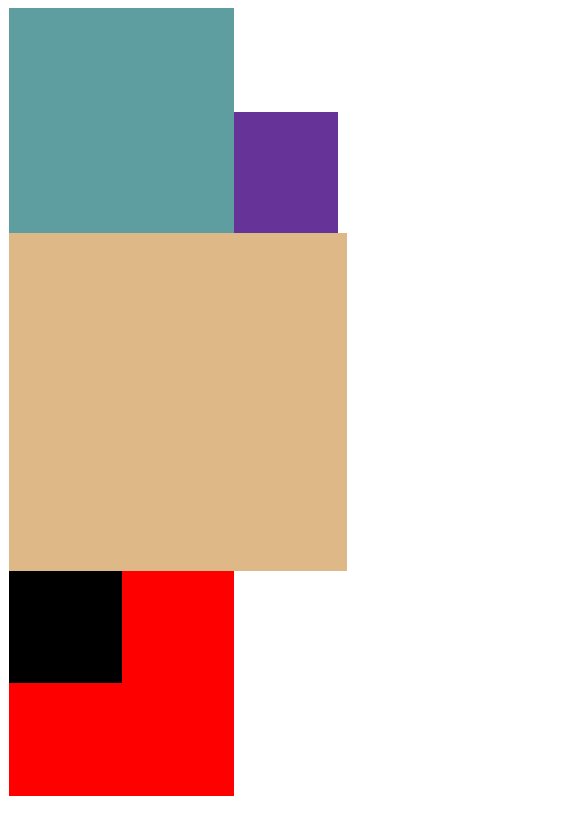
·通过z-index属性可以设置元素的层级

·可以为z-index指定一个正整数，该值会作为当前元素的层级，层级高的优先显示

·对于没有开启定位的元素不能使用z-index

·父元素的层级再高，也不会盖住子元素

·效果



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

.box1{

width: 200px;

height: 200px;

background-color: cadetblue;

position: relative;

z-index: 2;

}

.box2{

width: 200px;

height: 200px;

background-color: rebeccapurple;

/\*绝对定位\*/

position: absolute;

/\*设置偏移量\*/

left: 100px;

top: 100px;

z-index: 1;

}

.box3{

width: 300px;

height: 300px;

background-color: burlywood;

position: relative;

z-index: 3;

}

.box4{

width: 200px;

height: 200px;

background-color: red;

position: relative;

z-index: 20;

}

.box5{

width: 100px;

height: 100px;

background-color: black;

position: absolute;

z-index: 10;

}

</style>

</head>

<body>

<div class="box1"></div>

<div class="box2"></div>

<div class="box3"></div>

<div class="box4">

<div class="box5"></div>

</div>

</body>

</html>

# 背景

## 设置元素的透明背景

### Opacity（不透明度）

·用来设置透明背景

·它需要0~1之间的值，值越大越不透明

·在IE8及以下的浏览器中不支持，需要使用filter属性设置透明背景，它需要0~100之间的值，值越大越不透明

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

.box1{

width: 200px;

height: 200px;

background-color: cadetblue;

position: relative;

z-index: 2;

}

.box2{

width: 200px;

height: 200px;

background-color: rebeccapurple;

/\*相对定位\*/

position: absolute;

/\*设置偏移量\*/

left: 100px;

top: 100px;

z-index: 25;

}

.box3{

width: 300px;

height: 300px;

background-color: burlywood;

position: absolute;

top: 200px;

left: 200px;

z-index: 30;

opacity: 0.4;

filter: alpha(opacity = 40);

}

.box4{

width: 200px;

height: 200px;

background-color: red;

position: relative;

z-index: 20;

}

.box5{

width: 100px;

height: 100px;

background-color: black;

position: absolute;

z-index: 10;

}

</style>

</head>

<body>

<div class="box1"></div>

<div class="box2"></div>

<div class="box3"></div>

<div class="box4">

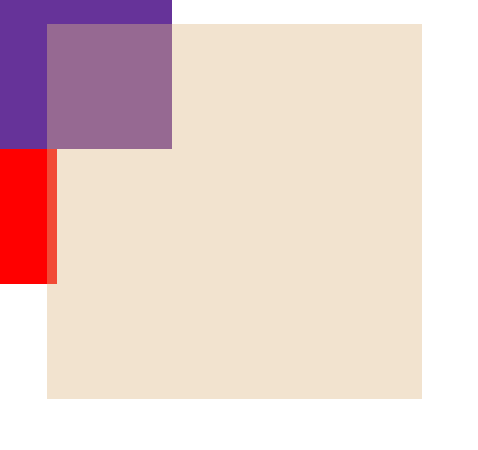
<div class="box5"></div>

</div>

</body>

</html>

·效果



## 背景有关样式

·background-color:颜色值;

- 设置背景颜色

·background-image: url("图片路径");

- 设置背景图片

- 如果背景图片大于元素,默认会显示图片左上角

- 如果背景图片小于元素,则会将背景图片平铺重复填充元素

- 相对路径写在哪，就相对于哪个文件夹

- 可以同时为一个元素指定背景颜色和背景图片,这样背景颜色将会作为背景图片的底色

- 一般情况下设置背景图片都会同时指定背景颜色

·background-repeat

- 设置背景图片的重复方式

- repeat

-- 默认值，双方向重复（平铺）

- no-repeat

-- 不重复

- repeat-x

-- 横向平铺

- repeat-y

-- 纵向平铺

·background-position

- 使用top right left bottom center中的两个值指定背景图片的位置

-- top left 左上

-- center center水平垂直居中

-- 如果只给一个值，则第二个值默认是center

background-position: center center;

- 也可以指定两个偏移量

-- 第一个值为水平偏移量

-- 第二个值为垂直偏移量

-- 如果为负值则向反方向移动

background-position: 100px 20px;

·background-attachment

- attachment（附上）

- 设置背景图片是否随页面一起滚动

- scroll（滚动）

-- 背景图片随页面滚动，默认值

- fixed

-- 背景图片会固定在某一位置，不随页面滚动

-- 背景图片会相对于浏览器窗口进行定位

-- 一般设置给body，不设置给其他元素

background-attachment: fixed;

·background-size

- 设置背景图片大小

- 可设置像素值，也可设置百分比

background-size: 455px 256px;

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

.box1{

width: 455px;

height: 256px;

margin: 10px auto;

/\*设置背景颜色\*/

background-color: cadetblue;

/\*设置背景图片\*/

/\*如果背景图片大于元素,默认会显示图片左上角\*/

/\*如果背景图片小于元素,则会将背景图片平铺重复填充元素\*/

/\*相对路径写在哪，就相对于哪个文件夹\*/

/\*可以同时为一个元素指定背景颜色和背景图片,这样背景颜色将会作为背景图片的底色\*/

/\*一般情况下设置背景图片都会同时指定背景颜色\*/

background-image: url("imgs/8.jpg");

/\*不重复\*/

background-repeat: no-repeat;

/\*设置图片的位置\*/

/\*background-position: center center;\*/

/\*background-position: 100px 20px;\*/

/\*设置背景图片不随页面一起滚动\*/

/\*background-attachment: fixed;\*/

/\*设置背景图片大小\*/

background-size: 455px 256px;

}

</style>

</head>

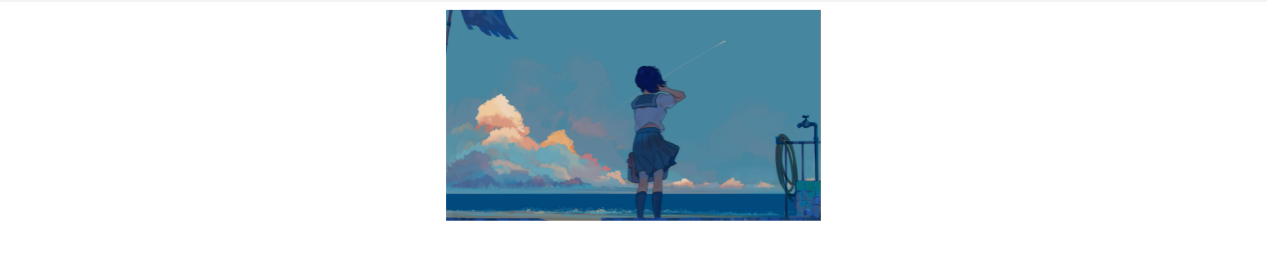
<body>

<div class="box1"></div>

</body>

</html>

·效果



## 按钮背景练习

·做完功能以后，发现在第一次切换图片时，会发现图片有一个非常快的闪烁，这个闪烁会造成一次不佳的用户体验。

·产生问题的原因：

- 背景图片是以外部资源的形式加载进网页的，浏览器每加载一个外部资源就需要单独的发送一次请求，

- 但是我们外部资源并不是同时加载，浏览器会在资源被使用才去加载资源

- 我们这个练习，一上来浏览器只会加载link.png由于hover和active的状态没有马上触发，

- 所以hover.png和active.png并不是立即加载的

- 当hover被触发时，浏览器才去加载hover.png

- 当active被触发时，浏览器才去加载active.png

- 由于加载图片需要一定的时间，所以在加载和显示过程会有一段时间，背景图片无法显示，导致出现闪烁的情况

- 为了解决该问题，可以将三个图片整合为一张图片，这样可以同时将三张图片一起加载，就不会出现闪烁的问题了，然后在通过background-position来切换要显示的图片的位置，这种技术叫做图片整合技术（CSS-Sprite）

·优点：

- 将多个图片整合为一张图片里，浏览器只需要发送一次请求，可以同时加载多个图片，提高访问效率，提高了用户体验。

- 将多个图片整合为一张图片，减小了图片的总大小，提高请求的速度，增加了用户体验

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

.btn:link{

/\*将a转换为块元素\*/

display: block;

/\*设置宽高\*/

width: 93px;

height: 29px;

/\*设置背景图片\*/

background-image: url("imgs/btn/btn2.png");

/\*设置背景图片不重复\*/

background-repeat: no-repeat;

}

.btn:hover{

/\*

\* 当是hover状态时，希望图片可以向左移动

\*/

background-position: -93px 0px;

}

.btn:active{

/\*

\* 当是active状态时，希望图片再向左移动

\*/

background-position: -186px 0px;

}

</style>

</head>

<body>

<!-- 创建一个超链接 -->

<a class="btn" href="#"></a>

</body>

</html>

·效果



## 简写属性

·background

- 通过该属性可以同时设置所有背景相关的样式

- 没有顺序的要求，谁在前谁在后都行

- 也没有数量的要求，不写的样式就使用默认值

background: cadetblue url(imgs/8.jpg) center center no-repeat fixed ;

# 表格

## 表格简介

·表格就是用来表示一些格式化的数据的，比如：课程表、银行对账单

·table标签

- 创建表格

- 块元素

·tr标签

- 表示表格中的一行，有几行就有几个tr

·th标签

- 表头，一般写在表格的第一行

- 默认居中和加粗

·td标签

- 创建单元格

- 有几个单元格就有几个td

·rowspan

- 纵向合并单元格

·colspan

- 横向合并单元格

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

</style>

</head>

<body>

<table border="1" width="40%" align="center">

<!--

在table标签中使用tr来表示表格中的一行，有几行就有几个tr

-->

<tr>

<!-- 在tr中需要使用td来创建一个单元格，有几个单元格就有几个td -->

<td>A1</td>

<td>A2</td>

<td>A3</td>

<td>A4</td>

</tr>

<tr>

<td>B1</td>

<td>B2</td>

<td>B3</td>

<!--

rowspan用来设置纵向的合并单元格

-->

<td rowspan="2">B4</td>

</tr>

<tr>

<td>C1</td>

<td>C2</td>

<td>C3</td>

</tr>

<tr>

<td>D1</td>

<td>D2</td>

<!--

colspan横向的合并单元格

-->

<td colspan="2">D3</td>

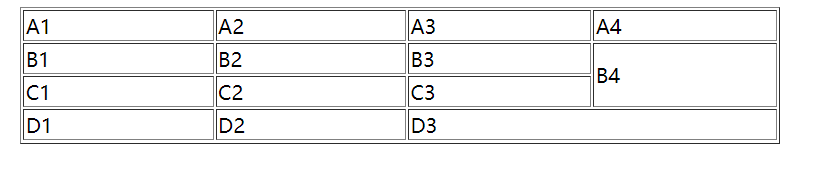
</tr>

</table>

</body>

</html>

·效果



## 表格的样式

·border-spacing

- spacing（间距）

- 设置table与td单元格之间的距离

border-spacing:5px ;

·border-collapse

- collapse（崩塌）

- 边框合并，也就是取消双线

- 如果设置了边框合并，则border-spacing自动失效

border-collapse: collapse;

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

/\*

\* 设置表格的宽度

\*/

table{

width: 300px;

/\*居中\*/

margin: 0 auto;

/\*边框\*/

/\*border:1px solid black;\*/

/\*

\* table和td边框之间默认有一个距离

\* 通过border-spacing属性可以设置这个距离

\*/

/\*border-spacing:0px ;\*/

/\*

\* border-collapse可以用来设置表格的边框合并

\* 如果设置了边框合并，则border-spacing自动失效

\*/

border-collapse: collapse;

/\*设置背景样式\*/

/\*background-color: #bfa;\*/

}

/\*

\* 设置边框

\*/

td , th{

border: 1px solid black;

}

/\*

\* 设置隔行变色

ie8不支持

\*/

tr:nth-child(even){

background-color: cadetblue;

}

/\*

\* 鼠标移入到tr以后，改变颜色

\*/

tr:hover{

background-color: #019e8b;

}

</style>

</head>

<body>

<!--

table是一个块元素

-->

<table>

<tr>

<!--

可以使用th标签来表示表头中的内容，

它的用法和td一样，不同的是它会有一些默认效果

-->

<th>学号</th>

<th>姓名</th>

<th>性别</th>

<th>住址</th>

</tr>

<tr>

<td>1</td>

<td>孙悟空</td>

<td>男</td>

<td>花果山</td>

</tr>

<tr>

<td>2</td>

<td>猪八戒</td>

<td>男</td>

<td>高老庄</td>

</tr>

<tr>

<td>3</td>

<td>沙和尚</td>

<td>男</td>

<td>流沙河</td>

</tr>

<tr>

<td>4</td>

<td>唐僧</td>

<td>男</td>

<td>女儿国</td>

</tr>

<tr>

<td>1</td>

<td>孙悟空</td>

<td>男</td>

<td>花果山</td>

</tr>

<tr>

<td>2</td>

<td>猪八戒</td>

<td>男</td>

<td>高老庄</td>

</tr>

<tr>

<td>3</td>

<td>沙和尚</td>

<td>男</td>

<td>流沙河</td>

</tr>

<tr>

<td>4</td>

<td>唐僧</td>

<td>男</td>

<td>女儿国</td>

</tr>

<tr>

<td>1</td>

<td>孙悟空</td>

<td>男</td>

<td>花果山</td>

</tr>

<tr>

<td>2</td>

<td>猪八戒</td>

<td>男</td>

<td>高老庄</td>

</tr>

<tr>

<td>3</td>

<td>沙和尚</td>

<td>男</td>

<td>流沙河</td>

</tr>

<tr>

<td>4</td>

<td>唐僧</td>

<td>男</td>

<td>女儿国</td>

</tr>

<tr>

<td>1</td>

<td>孙悟空</td>

<td>男</td>

<td>花果山</td>

</tr>

<tr>

<td>2</td>

<td>猪八戒</td>

<td>男</td>

<td>高老庄</td>

</tr>

<tr>

<td>3</td>

<td>沙和尚</td>

<td>男</td>

<td>流沙河</td>

</tr>

<tr>

<td>4</td>

<td>唐僧</td>

<td>男</td>

<td>女儿国</td>

</tr>

<tr>

<td>1</td>

<td>孙悟空</td>

<td>男</td>

<td>花果山</td>

</tr>

<tr>

<td>2</td>

<td>猪八戒</td>

<td>男</td>

<td>高老庄</td>

</tr>

<tr>

<td>3</td>

<td>沙和尚</td>

<td>男</td>

<td>流沙河</td>

</tr>

<tr>

<td>4</td>

<td>唐僧</td>

<td>男</td>

<td>女儿国</td>

</tr>

<tr>

<td>1</td>

<td>孙悟空</td>

<td>男</td>

<td>花果山</td>

</tr>

<tr>

<td>2</td>

<td>猪八戒</td>

<td>男</td>

<td>高老庄</td>

</tr>

<tr>

<td>3</td>

<td>沙和尚</td>

<td>男</td>

<td>流沙河</td>

</tr>

<tr>

<td>4</td>

<td>唐僧</td>

<td>男</td>

<td>女儿国</td>

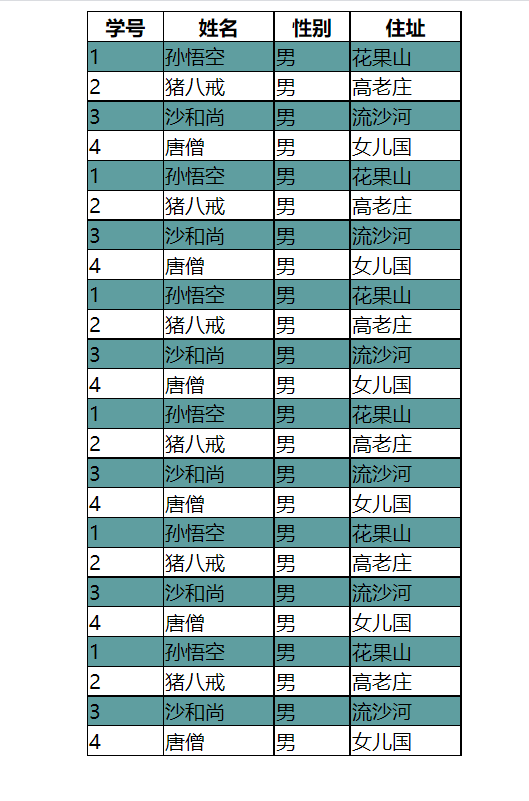
</tr>

</table>

</body>

</html>

·效果



## 长表格

·将长表格分为三个部分

- 表头

- 表格的主体

- 表格底部

·在HTML中为我们提供了三个标签：

- thead

-- 表头

- tbody

-- 表格主体

- tfoot

-- 表格底部、

- 直接写到table中

- tr/td需要写在这些标签当中

·thead中的内容，永远会显示在表格的头部

·tfoot中的内容，永远都会显示表格的底部

·tbody中的内容，永远都会显示表格的中间

·如果表格中没有写tbody，**浏览器会自动在表格中添加tbody**

·并且将所有的tr都放到tbody中，所以注意**tr并不是table的子元素，而是tbody的子元素**

·通过table > tr 无法选中行 需要通过tbody > tr选中行

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

</style>

</head>

<body>

<table>

<thead>

<tr>

<th>日期</th>

<th>收入</th>

<th>支出</th>

<th>合计</th>

</tr>

</thead>

<tfoot>

<tr>

<td></td>

<td></td>

<td>合计</td>

<td>100</td>

</tr>

</tfoot>

<tbody>

<tr>

<td>10.24</td>

<td>500</td>

<td>300</td>

<td>200</td>

</tr>

<tr>

<td>10.24</td>

<td>500</td>

<td>300</td>

<td>200</td>

</tr>

<tr>

<td>10.24</td>

<td>500</td>

<td>300</td>

<td>200</td>

</tr>

<tr>

<td>10.24</td>

<td>500</td>

<td>300</td>

<td>200</td>

</tr>

<tr>

<td>10.24</td>

<td>500</td>

<td>300</td>

<td>200</td>

</tr>

<tr>

<td>10.24</td>

<td>500</td>

<td>300</td>

<td>200</td>

</tr>

<tr>

<td>10.24</td>

<td>500</td>

<td>300</td>

<td>200</td>

</tr>

<tr>

<td>10.24</td>

<td>500</td>

<td>300</td>

<td>200</td>

</tr>

<tr>

<td>10.24</td>

<td>500</td>

<td>300</td>

<td>200</td>

</tr>

<tr>

<td>10.24</td>

<td>500</td>

<td>300</td>

<td>200</td>

</tr>

<tr>

<td>10.24</td>

<td>500</td>

<td>300</td>

<td>200</td>

</tr>

<tr>

<td>10.24</td>

<td>500</td>

<td>300</td>

<td>200</td>

</tr>

<tr>

<td>10.24</td>

<td>500</td>

<td>300</td>

<td>200</td>

</tr>

<tr>

<td>10.24</td>

<td>500</td>

<td>300</td>

<td>200</td>

</tr>

<tr>

<td>10.24</td>

<td>500</td>

<td>300</td>

<td>200</td>

</tr>

<tr>

<td>10.24</td>

<td>500</td>

<td>300</td>

<td>200</td>

</tr>

<tr>

<td>10.24</td>

<td>500</td>

<td>300</td>

<td>200</td>

</tr>

<tr>

<td>10.24</td>

<td>500</td>

<td>300</td>

<td>200</td>

</tr>

<tr>

<td>10.24</td>

<td>500</td>

<td>300</td>

<td>200</td>

</tr>

<tr>

<td>10.24</td>

<td>500</td>

<td>300</td>

<td>200</td>

</tr>

<tr>

<td>10.24</td>

<td>500</td>

<td>300</td>

<td>200</td>

</tr>

<tr>

<td>10.24</td>

<td>500</td>

<td>300</td>

<td>200</td>

</tr>

<tr>

<td>10.24</td>

<td>500</td>

<td>300</td>

<td>200</td>

</tr>

<tr>

<td>10.24</td>

<td>500</td>

<td>300</td>

<td>200</td>

</tr>

<tr>

<td>10.24</td>

<td>500</td>

<td>300</td>

<td>200</td>

</tr>

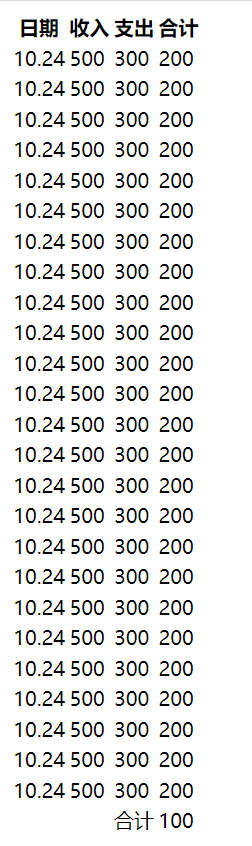
</tbody>

</table>

</body>

</html>

·效果



## 表格布局

·以前表格更多的情况实际上是用来对页面进行布局的，但是这种方式早已被CSS所淘汰了

·表格的列数由td最多的那行决定

·表格是可以嵌套，可以在td中在放置一个表格

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

</style>

</head>

<body>

<table border="1" width="100%">

<tr height="100px">

<td colspan="2"></td>

</tr>

<tr height="400px">

<td width="20%"></td>

<td width="80%">

<table border="1" width="100%" height="100%">

<tr>

<td></td>

</tr>

<tr>

<td></td>

</tr>

</table>

</td>

</tr>

<tr height="100px">

<td colspan="2"></td>

</tr>

</table>

</body>

</html>

·效果



## 完善clearfix

·解决高度塌陷和外边距重叠问题

·子元素和父元素相邻的垂直外边距会发生重叠，子元素的外边距会传递给父元素

·使用空的table标签可以隔离父子元素的外边距，阻止外边距的重叠

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

.box1{

width: 300px;

height: 300px;

background-color: #bfa;

}

.box2{

width: 200px;

height: 200px;

background-color: yellow;

/\*

\* 子元素和父元素相邻的垂直外边距会发生重叠，子元素的外边距会传递给父元素

\* 使用空的table标签可以隔离父子元素的外边距，阻止外边距的重叠

\*/

margin-top: 100px;

}

.box3{

border: 10px red solid;

}

.box4{

width: 100px;

height: 100px;

background-color: yellowgreen;

float: left;

}

/\*\*

\* 解决父子元素的外边距重叠

\*/

/\*.box1:before{

content: "";\*/

/\*

\* display:table可以将一个元素设置为表格显示

\*/

/\* display: table;

}

\*/

/\*\*

\* 解决父元素高度塌陷

\*/

/\*.clearfix:after{

content: "";

display: block;

clear: both;

}\*/

/\*

\* 经过修改后的clearfix是一个多功能的

\* 既可以解决高度塌陷，又可以确保父元素和子元素的垂直外边距不会重叠

\*/

.clearfix:before,

.clearfix:after{

content: "";

display: table;

clear: both;

}

.clearfix{

zoom: 1;

}

</style>

</head>

<body>

<div class="box3 clearfix">

<div class="box4"></div>

</div>

<div class="box1 clearfix">

<div class="box2"></div>

</div>

</body>

</html>

·效果



# 表单

·表单的作用就是用来将用户信息提交给服务器的，比如：百度的搜索框 注册 登录这些操作都需要填写表单

## form标签

- 创建一个表单

- form标签中必须指定一个action属性，该属性指向的是一个服务器的地址

- 当我们提交表单时将会提交到action属性对应的地址

## fieldset标签

·为表单项进行分组

·将表单项中的同一组放到一个fieldset（字段集）中

### legend标签

·legend（传奇）

·指定组名

## input标签

·type

- text

-- 文本框

-- 通过value属性值作为文本框的默认值显示

用户名<input type="text" name="username" value="admin">

- submit

-- 提交按钮

-- 将表单中的信息提交给服务器

-- 通过value属性来指定按钮上的文字

<input type="submit" value="注册" />

- password

-- 密码框

密码<input id="pwd" type="password" name="password" />

- radio

-- 单选按钮

-- 单选按钮通过name属性进行分组，name属性相同是一组按钮

-- 通过value将属性值提交给服务器

性别 <input type="radio" name="gender" value="male" id="male" />男

<input type="radio" name="gender" value="female" checked="checked" id="female" />女

- checkbox

-- 多选框

-- 使用方法与radio相同

爱好 <input type="checkbox" name="hobby" value="zq" />足球

<input type="checkbox" name="hobby" value="lq" />篮球

<input type="checkbox" name="hobby" value="ymq" checked="checked" />羽毛球

<input type="checkbox" name="hobby" value="ppq" checked="checked"/>乒乓球

- reset

-- 重置按钮

-- 将表单所填内容恢复为默认值

<input type="reset" />

- button

-- 普通按钮

-- 可通过js绑定响应事件

·name

- 提交给服务器的属性名

- 数据会提交到服务器中，name表示提交内容的名字

- 用户填写的信息会附在url地址的后边以查询字符串的形式发送给服务器

- 属性名=属性值&属性名=属性值&属性名=属性值&属性名=属性值

·value

- text文本框

-- 作为默认值显示

- submit提交按钮

-- 作为按钮上的文字

- radio单选按钮

-- 将属性值提交给服务器

·checked

- 指定默认选中的选项

- 单选按钮或多选框

<input type="radio" name="gender" value="female" checked="checked" id="female" />女

·placeholder（占位符）

- 设置浅色提示文字

- 在ie8及以下浏览器中不支持，如果要兼容则需要使用js

<input type="text" class="txt" placeholder="your name"/>

### label标签

·该标签可以指定一个for属性，该属性的值需要指定一个表单项的id值，用来绑定元素

·当点击该文字时，相当于点击绑定的元素

<label for="um">用户名</label>

## select标签

·创建下拉列表

·下拉列表的name属性需要指定给select，而value属性需要指定给option

·name

- 提交给服务器的属性名

·multiple（倍数）

- 变为一个多选的下拉列表

<select name="star" multiple="multiple"></select>

### option标签

·option（选项）

·创建列表项

·value

- 将属性值提交给服务器

·selected

- 将选项设置为默认选中

<option value="zbs" selected="selected">赵本山</option>

### optgroup标签

·对选项进行分组

·同一个optgroup中的选项是一组

·label

- 指定分组的名字

<optgroup label="女明星"></optgroup>

## textarea标签

·文本域

·name

- 提交给服务器的属性名

## button标签

·除了使用input，也可以使用button标签来创建按钮

·这种方式和使用input类似，只不过由于它是成对出现的标签使用起来更加的灵活

<button type="submit">提交</button>

<button type="reset">重置</button>

<button type="button">按钮</button>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

</head>

<body>

<!--

表单的作用就是用来将用户信息提交给服务器的

比如：百度的搜索框 注册 登录这些操作都需要填写表单

-->

<!--

使用form标签创建一个表单

form标签中必须指定一个action属性，该属性指向的是一个服务器的地址

当我们提交表单时将会提交到action属性对应的地址

-->

<form action="target.html">

<!--

使用form创建的仅仅是一个空白的表单，

我们还需要向form中添加不同的表单项

-->

<!--

在表单中可以使用fieldset来为表单项进行分组，

可以将表单项中的同一组放到一个fieldset中

-->

<fieldset>

<!-- 在fieldset可以使用legend子标签，来指定组名 -->

<legend>用户信息</legend>

<!--

使用input来创建一个文本框，它的type属性是text

如果希望表单项中的数据会提交到服务器中，还必须给表单项指定一个name属性

name表示提交内容的名字

用户填写的信息会附在url地址的后边以查询字符串的形式发送给服务器

url地址?查询字符串

格式：

属性名=属性值&属性名=属性值&属性名=属性值&属性名=属性值

在文本框中也可以指定value属性值，该值将会作为文本框的默认值显示

-->

<!--

在html中还为我们提供了一个标签，专门用来选中表单中的提示文字的

label标签

该标签可以指定一个for属性，该属性的值需要指定一个表单项的id值

-->

<label for="um">用户名</label>

<input id="um" type="text" name="username" /> <br /><br />

<!--

密码框

- 使用input创建一个密码框，它的type属性值是password

-->

<label for="pwd">密码 </label>

<input id="pwd" type="password" name="password" /> <br /><br />

</fieldset>

<fieldset >

<legend>用户爱好</legend>

<!--

单选按钮

- 使用input来创建一个单选按钮，它的type属性使用radio

- 单选按钮通过name属性进行分组，name属性相同是一组按钮

- 像这种需要用户选择但是不需要用户直接填写内容的表单项，

还必须指定一个value属性，这样被选中的表单项的value属性值将会最终提交给服务器

如果希望在单选按钮或者是多选框中指定默认选中的选项，

则可以在希望选中的项中添加checked="checked"属性

-->

性别 <input type="radio" name="gender" value="male" id="male" /><label for="male">男</label>

<input type="radio" name="gender" value="female" checked="checked" id="female" /><label for="female">女</label>

<br /><br />

<!--

多选框

- 使用input创建一个多选框，它的type属性使用checkbox

-->

爱好 <input type="checkbox" name="hobby" value="zq" />足球

<input type="checkbox" name="hobby" value="lq" />篮球

<input type="checkbox" name="hobby" value="ymq" checked="checked" />羽毛球

<input type="checkbox" name="hobby" value="ppq" checked="checked"/>乒乓球

<br /><br />

</fieldset>

<!--

下拉列表

- 使用select来创建一个下拉列表

下拉列表的name属性需要指定给select，而value属性需要指定给option

可以通过在option中添加selected="selected"来将选项设置为默认选中

当为select添加一个multiple="multiple"，则下拉列表变为一个多选的下拉列表

-->

你喜欢的明星

<select name="star">

<!--

在select中可以使用optgroup对选项进行分组

同一个optgroup中的选项是一组

可以通过label属性来指定分组的名字

-->

<optgroup label="女明星">

<!-- 在下拉列表中使用option标签来创建一个一个列表项 -->

<option value="fbb">范冰冰</option>

<option value="lxr">林心如</option>

<option value="zw">赵薇</option>

</optgroup>

<optgroup label="男明星">

<option value="zbs" selected="selected">赵本山</option>

<option value="ldh">刘德华</option>

<option value="pcj">潘长江</option>

</optgroup>

</select>

<br /><br />

<!--

使用textarea创建一个文本域

-->

自我介绍 <textarea name="info"></textarea>

<br /><br />

<!--

提交按钮可以将表单中的信息提交给服务器

使用input创建一个提交按钮,它的type属性值是submit

在提交按钮中可以通过value属性来指定按钮上的文字

-->

<input type="submit" value="注册" />

<!--

<input type="reset" />可以创建一个重置按钮，

点击重置按钮以后表单中内容将会恢复为默认值

-->

<input type="reset" />

<!--

使用input type=button可以用来创建一个单纯的按钮，

这个按钮没有任何功能，只能被点击

-->

<input type="button" value="按钮" />

<!--

除了使用input，也可以使用button标签来创建按钮

这种方式和使用input类似，只不过由于它是成对出现的标签

使用起来更加的灵活

-->

<br /><br />

<button type="submit">提交</button>

<button type="reset">重置</button>

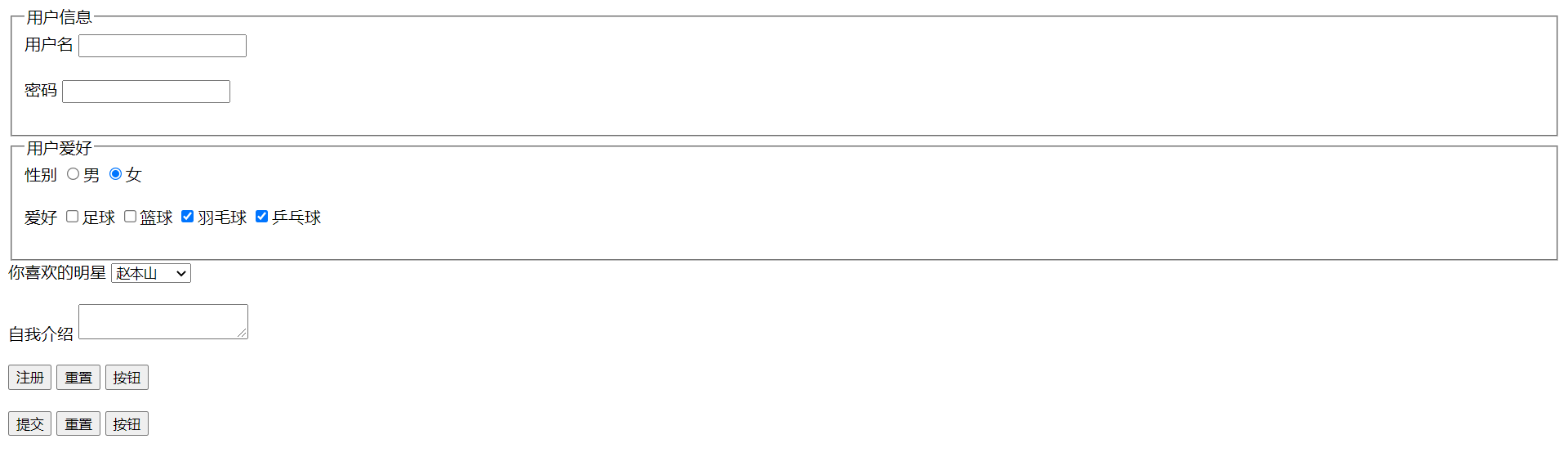
<button type="button">按钮</button>

</form>

</body>

</html>

·效果



# 框架集

·框架集和内联框架的作用类似

·用于在一个页面中引入其他的外部的页面

·框架集可以同时引入多个页面，而内联框架只能引入一个

·推荐使用框架集，而不使用内联框架

·frameset和iframe一样，它里边的内容都不会被搜索引擎所检索

·使用框架集则意味着页面中不能有自己的内容，只能引入其他的页面

·而我们每单独加载一个页面浏览器都需要重新发送一次请求，引入几个页面就需要发送几次请求，用户的体验比较差

·如果非得用建议使用frameset而不使用iframe

## frameset标签

·创建框架集

·frameset不能和body出现在同一个页面中

·rows

- 上下排列

- 按权重指定每行所占的比重

- 可以是数值，也可以是百分比

- 将其中一个值填\*可自适应比重

<frameset rows="1,3"></frameset>

·cols

- 左右排列

- 按权重指定每行所占的比重

- 可以是数值，也可以是百分比

- 将其中一个值填\*可自适应比重

<frameset cols="1,3"></frameset>

·这两个属性frameset必须选择一个，并且需要在属性中指定每一部分所占的比重

·frameset中也可以再嵌套frameset

### frame标签

·指定要引入的页面

·引入几个页面就写几个frame

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

</head>

<frameset cols="30% , \* , 30%">

<frame src="https://www.baidu.com" />

<frame src="https://gitee.com" />

<!-- 嵌套一个frameset -->

<frameset rows="30%,50%,\*">

<frame src="https://cn.vuejs.org" />

<frame src="https://www.bilibili.com/" />

<frame src="https://jquery.com" />

</frameset>

</frameset>

</html>

·效果



# Ie6png的修复

·在IE6中对图片格式png24支持度不高，如果使用的图片格式是png24，则会导致透明效果无法正常显示

## 解决方法

### 修改图片格式

·可以使用png8格式来代替png24格式，即可解决问题

·但是使用png8代替png24以后，图片的清晰图会有所下降

### JavaScript解决

·向页面中引入一个外部的JavaScript文件

·然后再写一下简单的JS代码，来处理该问题

<!-- 在body标签的最后引入外部的JS文件 -->

<script type="text/javascript" src="js/DD\_belatedPNG\_0.0.8a-min.js"></script>

<!--再创建一个新的script标签，并且编写一些js代码 -->

<script type="text/javascript">

DD\_belatedPNG.fix("\*");

</script>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

<style type="text/css">

.box1{

width: 200px;

height: 200px;

background-image: url(imgs/3.png);

background-repeat: no-repeat;

}

.box2{

width: 200px;

height: 200px;

background-image: url(imgs/4.png);

background-repeat: no-repeat;

}

</style>

</head>

<body style="background-color: #bfa;">

<!--

在IE6中对图片格式png24支持度不高，

如果使用的图片格式是png24，则会导致透明效果无法正常显示

解决方法：

1.可以使用png8来代替png24，即可解决问题，

但是使用png8代替png24以后，图片的清晰图会有所下降

2.使用JavaScript来解决该问题，需要向页面中引入一个外部的JavaScript文件

然后在写一下简单的JS代码，来处理该问题

-->

<div class="box1"></div>

<div class="box2"></div>

<img src="img/3.png"/>

<!-- 在body标签的最后引入外部的JS文件 -->

<script type="text/javascript" src="js/DD\_belatedPNG\_0.0.8a-min.js"></script>

<!--再创建一个新的script标签，并且编写一些js代码 -->

<script type="text/javascript">

***DD\_belatedPNG***.fix("\*");

</script>

</body>

</html>

# Hack

·有一些特殊的代码只需要再某些特殊的浏览器中执行，而再其他浏览器中不执行，可使用CSS Hack来解决该问题

·不到万不得已时尽量不要使用

## 条件hack

·只对ie9及以下浏览器有效，其他浏览器都会将它识别为注释

·<!--[if IE]> <![endif]-->

- 在ie浏览器中显示

·<!--[if IE 6]> <![endif]-->

- 在指定版本的ie浏览器中显示

·<!--[if lt IE 9]> <![endif]-->

- 在ie9以下的浏览器中显示

·<!--[if lte IE 9]> <![endif]-->

- 在ie9及以下的浏览器中显示

·<!--[if gt IE 9]> <![endif]-->

- 在ie9以上的浏览器中显示

·<!--[if gte IE 9]> <![endif]-->

- 在ie9及以上的浏览器中显示

·<!--[if ! IE 6]> <![endif]-->

- 在除了ie6浏览器中显示

## 属性hack

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

<style type="text/css">

body{

/\*其他浏览器中显示的样式\*/

background-color: cadetblue;

/\*在ie6及以下浏览器中显示的样式\*/

\_background-color: red;

/\*在ie7及以下浏览器中显示的样式\*/

\*background-color: blue;

/\*在ie浏览器中显示的样式\*/

background-color: black\9;

/\*在ie8及以上的浏览器中显示的样式\*/

background-color: yellow\0;

}

</style>

</head>

<body>

</body>

</html>

# 命名规范

·命名时尽量使用英文，如果不会可以使用拼音，但是不要英文和拼音混用

·命名格式

- 驼峰命名法

-- 首字母小写，每个单词的开头字母大写

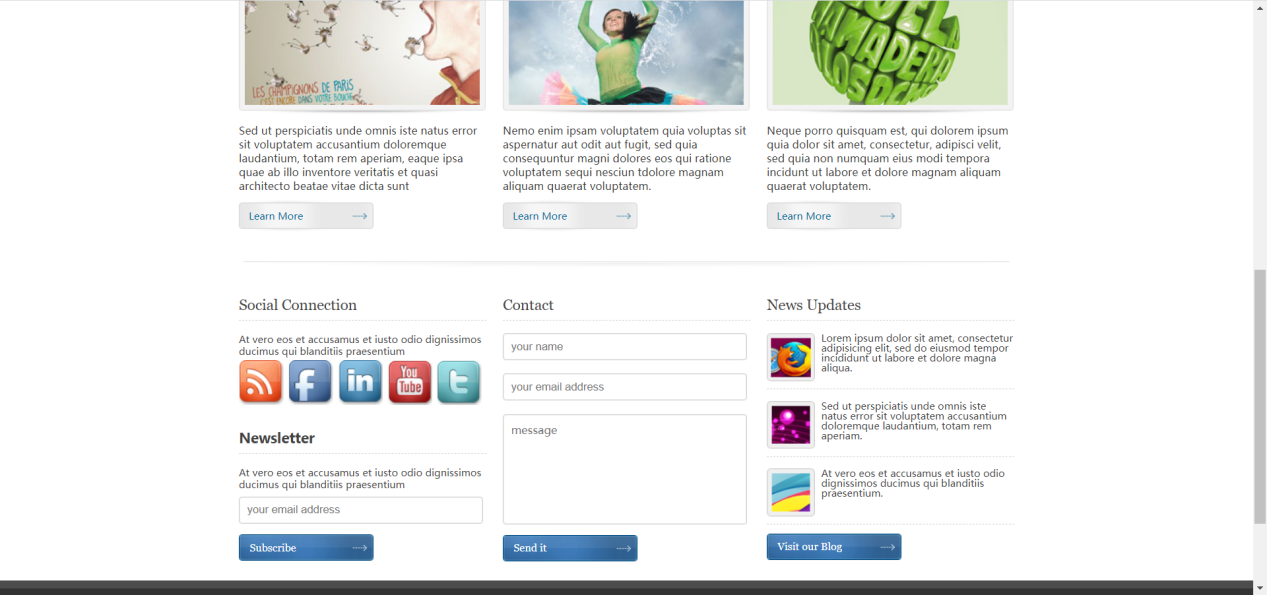
- 也可以所有字母都小写，单词之间用\_或-连接

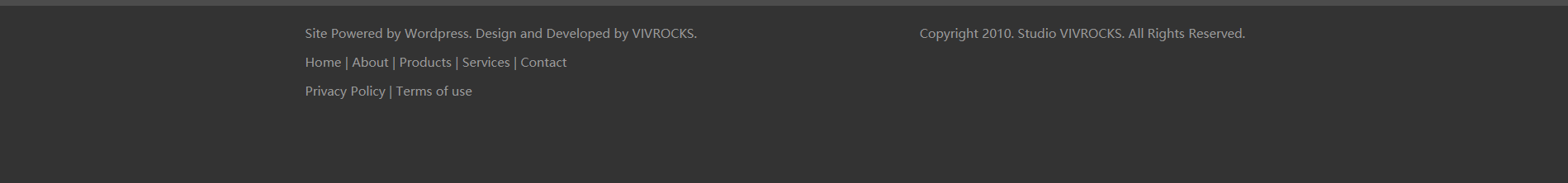
- 公司让你怎么写你就怎么写

# 项目练习

·效果







·在chrome浏览器中字体最小支持12px

·觉得页面不好可以向美工提意见，不能自己擅作主张修改

·cursor

- 设置鼠标移入时指针的样式

- pointer

--小手

- crosshair

- 十字

- help

- 可用的帮助（通常是一个问号或一个气球）。

- move

- 可移动元素指示

- wait

- 等待进度条

·ie6的双倍边距bug

- 产生的原因

-- 在ie6中，当为一个向左浮动的元素设置左外边距

-- 或者为一个向右浮动的元素设置右外边距

-- 这个外边距将会是设置值的两倍

- 解决

-- 将元素设置为行内块元素

-- display:inline

-- 对于一个浮动元素来说，设置display:inline没有意义

-- 但是该属性可解决双倍边距问题

·项目步骤

- 需求分析（1/3时间，包括ui设计）

- 编码（1/6时间）

- 测试与修改（1/2时间）

-- 性能测试（访问速度，一般不能超过3秒，有专门的软件）

-- 业务逻辑测试

·CSS代码压缩

- 原理

-- 删去不必要的空格和注释

- 工具

-- JsCssZip

- 可提升性能

·图片整合

- 做雪碧图

- 适用于背景图

- 只需要加载一次图片即可，从而提升性能

- 需要调整背景图偏移量

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>polo360</title>

<!--引入reset.css,用来清除默认样式-->

<link rel="stylesheet" type="text/css" href="css/reset-min.css">

<!--引入index.css-->

<link rel="stylesheet" type="text/css" href="css/page-index-min.css">

</head>

<body>

<!--header开始-->

<div class="header w">

<!--导航条-->

<ul class="nav">

<li>

<a href="#">HOME</a>

<p>Back to home</p>

</li>

<li>

<a href="#">PRODUCTS</a>

<p>What we have for you</p>

</li>

<li>

<a href="#">SERVICES</a>

<p>Things we do</p>

</li>

<li>

<a href="#">BLOG</a>

<p>Follow our updates</p>

</li>

<li>

<a href="#">CONTACT</a>

<p>Ways to reach us</p>

</li>

</ul>

<!--左上角logo-->

<div class="logo" title="一个非常好的网站">

<a href="#">

<img src="img/logo.png" alt="网站的logo"/>

</a>

</div>

</div>

<!--header结束-->

<!--banner开始-->

<div class="banner w">

<img src="img/banner/banner01.png" alt="轮播图">

<!--导航按钮-->

<div class="pointerDiv">

<a href="#"></a>

<a href="#" class="active"></a>

<a href="#"></a>

<a href="#"></a>

<a href="#"></a>

</div>

</div>

<!--banner结束-->

<!--content开始-->

<div class="content w clearfix">

<!--顶部标题-->

<h1>Lorem ipsum dolor sit amet, consectetur adipisicing elit</h1>

<!--放置内容的三个div-->

<div class="pl">

<!--小标题-->

<h2>Perfect Logic</h2>

<!--小标题的副标题-->

<p class="p1">All you want your website to do.</p>

<!--放图片的div-->

<div class="imgDiv">

<img src="img/pic/pic1.jpg" alt="图片1">

</div>

<!--正文-->

<p class="p2">

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt

</p>

<!--内容区按钮-->

<a href="#" class="lm">Learn More</a>

</div>

<div class="cs">

<!--小标题-->

<h2>Complete Solution</h2>

<!--小标题的副标题-->

<p class="p1">A tool anything and everything you can think</p>

<!--放图片的div-->

<div class="imgDiv">

<img src="img/pic/pic2.jpg" alt="图片2">

</div>

<!--正文-->

<p class="p2">

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciun tdolore magnam aliquam quaerat voluptatem.

</p>

<!--内容区按钮-->

<a href="#" class="lm">Learn More</a>

</div>

<div class="uc">

<!--小标题-->

<h2>Uber Culture</h2>

<!--小标题的副标题-->

<p class="p1">Fresh. Modern and ready for future</p>

<!--放图片的div-->

<div class="imgDiv">

<img src="img/pic/pic3.jpg" alt="图片3">

</div>

<!--正文-->

<p class="p2">

Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem.

</p>

<!--内容区按钮-->

<a href="#" class="lm">Learn More</a>

</div>

</div>

<!--content结束-->

<!--联系栏开始-->

<div class="contack w clearfix">

<!--创建三栏div-->

<!--联系栏左侧-->

<div class="sc">

<!--联系栏标题-->

<h2>Social Connection</h2>

<!--左侧联系栏顶部正文-->

<p class="p1">At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium</p>

<!--放置五个小图片超链接-->

<div class="icon">

<a href="#"><img src="img/rss.png"></a>

<a href="#"><img src="img/fb.png"></a>

<a href="#"><img src="img/in.png"></a>

<a href="#"><img src="img/yt.png"></a>

<a href="#"><img src="img/tw.png"></a>

</div>

<!--左侧联系栏中间标题-->

<h2 class="nl">Newsletter</h2>

<!--左侧联系栏中间正文-->

<p class="p1">At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium</p>

<!--创建表单-->

<form action="#">

<input type="text" class="txt" placeholder="your email address"/>

<button class="btn">Subscribe</button>

</form>

</div>

<!--联系栏中间-->

<div class="co">

<!--联系栏标题-->

<h2>Contact</h2>

<!--创建表单-->

<form action="#">

<input type="text" class="txt" placeholder="your name"/>

<input type="text" class="txt" placeholder="your email address"/>

<textarea class="tarea" placeholder="message"></textarea>

<button class="btn">Send it</button>

</form>

</div>

<!--联系栏右侧-->

<div class="nu">

<!--联系栏标题-->

<h2>News Updates</h2>

<!--三个图片段落-->

<p>

<img src="img/pic/1.gif"/>

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

</p>

<p>

<img src="img/pic/2.gif"/>

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam.

</p>

<p class="no-margin">

<img src="img/pic/3.gif"/>

At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium.

</p>

<button class="btn">Visit our Blog</button>

</div>

</div>

<!--联系栏结束-->

<!--底部开始-->

<div class="footer">

<div class="w">

<P class="copy">Copyright 2010. Studio VIVROCKS. All Rights Reserved.</P>

<P>Site Powered by Wordpress. Design and Developed by VIVROCKS.</P>

<P><a href="#">Home</a> | <a href="#">About</a> | <a href="#">Products</a> | <a href="#">Services</a> | <a href="#">Contact</a></P>

<P><a href="#">Privacy Policy</a> | <a href="#">Terms of use</a></P>

</div>

</div>

<!--底部结束-->

<!--ie6修复透明背景图片-->

<!--[if IE 6]>

<script src="js/DD\_belatedPNG\_0.0.8a-min.js"></script>

<script type="text/javascript">***DD\_belatedPNG***.fix('div,img,a,h1');</script>

<![endif]-->

</body>

</html>

/\*设置body\*/

body{

/\*设置背景图片水平方向重复\*/

background: url("../img/bd-bg.png") repeat-x;

/\*height: 3000px;\*/

}

/\*设置元素固定宽度和居中\*/

.w{

width: 940px;

margin: 0 auto;

}

/\*头部样式开始\*/

/\*头部样式\*/

.header{

/\*background-color: cadetblue;\*/

/\*设置上内边距\*/

padding-top: 37px;

/\*设置下内边距\*/

padding-bottom: 46px;

}

/\*左上角logo\*/

.logo{

/\*设置logo的左外边距\*/

margin-left: 15px;

}

/\*上部导航条\*/

.nav{

/\*导航条靠右\*/

float: right;

/\*设置上外边距\*/

margin-top: 22px;

}

/\*导航条li样式\*/

.nav li{

/\*li向左浮动，水平排列\*/

float: left;

/\*设置左右内边距\*/

padding: 0 10px 10px 10px;

/\*设置左边框\*/

border-left: 1px #d6d6d6 dotted;

}

/\*导航条超链接\*/

.nav a{

/\*字体样式\*/

font: bold 14px Georgia;

/\*字体颜色\*/

color: #666;

/\*去除下划线\*/

text-decoration: none;

}

/\*导航栏链接鼠标移入效果\*/

.nav a:hover{

/\*字体颜色变亮\*/

color: #a1a1a1;

/\*显示下划线\*/

text-decoration: underline;

}

/\*导航栏的副标题\*/

.nav p{

/\*字体颜色\*/

color: #b7b7b7;

/\*字体样式\*/

font: 11px Tahoma;

}

/\*头部样式结束\*/

/\*banner样式开始\*/

/\*banner样式\*/

.banner{

/\*设置高度\*/

height: 356px;

/\*设置背景图片，下部阴影，不重复，底部居中\*/

background: url("../img/banner-bg.png") no-repeat bottom center;

/\*开启相对定位\*/

position: relative;

}

/\*设置导航按钮样式\*/

.pointerDiv{

/\*开启绝对定位\*/

position: absolute;

/\*为导航点进行定位\*/

top: 314px;

left: 15px;

}

/\*超链接样式\*/

.pointerDiv a{

/\*设置水平排列，并变成块元素\*/

float: left;

/\*设置宽度和高度\*/

width: 17px;

height: 17px;

/\*background-color: cadetblue;\*/

background: url("../img/pointer.png") no-repeat;

/\*左外边距\*/

margin-left: 4px;

}

/\*激活超链接和鼠标移入超链接样式\*/

.pointerDiv .active

,.pointerDiv a:hover

{

background: url("../img/pointer-active.png") no-repeat;

}

/\*banner样式结束\*/

/\*content样式开始\*/

.content{

/\*background-color: blueviolet;\*/

}

/\*content顶部标题样式\*/

.content h1{

/\*设置文字居中\*/

text-align: center;

/\*字体样式\*/

font: bold 24px Georgia;

/\*设置上内边距\*/

padding: 6px 0 20px 0;

/\*设置背景，标题下方横线\*/

background: url("../img/line.png") no-repeat bottom center;

/\*设置下外边距\*/

margin-bottom: 38px;

}

/\*设置内容区和联系栏的三个div的样式\*/

.content .pl,.content .cs,.content .uc,

.contack .sc,.contack .co,.contack .nu

{

width: 300px;

/\*height: 100px;\*/

/\*background-color: cadetblue;\*/

/\*使三个div水平排列\*/

float: left;

}

/\*设置三个盒子间的距离\*/

.content .cs,

.contack .co

{

margin:0 20px;

}

/\*解决高度塌陷和垂直外边距重叠\*/

.clearfix:before,

.clearfix:after{

content: "";

display: table;

clear: both;

}

.clearfix{

zoom: 1;

}

/\*为小标题设置样式\*/

.content h2{

color: #117191;

font: 21px Georgia;

}

/\*小标题下方字体样式\*/

.content .p1{

color: #8c8c8c;

font: 12px 微软雅黑;

}

/\*设置放图片的div\*/

.content .imgDiv{

width: 299px;

height: 190px;

/\*设置阴影背景图片\*/

background: url("../img/img-bg.png") no-repeat;

/\*设置上下外边距\*/

margin: 16px 0 10px 0;

/\*使图片水平居中\*/

text-align: center;

/\*设置上内边距\*/

padding-top: 12px;

}

/\*正文样式\*/

.content .p2{

height: 92px;

color: #3e3e3e;

font: 13px 微软雅黑;

}

/\*设置按钮样式\*/

.content .lm{

/\*设置为块元素\*/

display: block;

width: 163px;

height: 40px;

/\*设置按钮背景\*/

background: url("../img/btn1.png") no-repeat;

/\*去掉下划线\*/

text-decoration: none;

color: #016999;

/\*字体样式\*/

font: 12px/40px 微软雅黑;

/\*首行缩进\*/

text-indent: 1em;

/\*为按钮设置外边距\*/

margin-bottom: 35px;

}

/\*content样式结束\*/

/\*联系栏样式开始\*/

/\*为联系栏设置顶部阴影\*/

.contack{

/\*需要解决高度塌陷才能设置\*/

background: url("../img/line.png") no-repeat top center;

}

/\*设置联系栏样式\*/

.contack .sc,.contack .co,.contack .nu {

/\*height: 500px;\*/

/\*background-color: cadetblue;\*/

}

/\*设置联系栏中的文本框样式\*/

.contack .txt{

width: 276px;

height: 33px;

/\*设置文本框背景图\*/

background: url("../img/input.png") no-repeat;

/\*去除表单项默认边框\*/

border: none;

/\*去除默认内外边距，并重新设置\*/

padding: 0 10px;/\*设置文字往里靠一些\*/

margin: 0 0 16px 0;

/\*文字垂直居中\*/

line-height: 33px;

/\*解决文字过多后ie6背景左移问题。将背景固定在浏览器，单独给ie6设置\*/

\_background-attachment: fixed;

}

/\*文本域样式\*/

.contack .tarea{

width: 276px;

height: 114px;

/\*设置文本框背景图\*/

background: url("../img/textarea.png") no-repeat;

/\*去除表单项默认边框\*/

border: none;

/\*ie浏览器中需要去除滚动条\*/

overflow: auto;

/\*禁止调整文本域大小\*/

resize: none;

/\*去除默认内外边距，并重新设置\*/

padding: 10px;

margin: 0;

font-family: 微软雅黑;

/\*解决文字过多后ie6背景左移问题。将背景固定在浏览器，单独给ie6设置\*/

\_background-attachment: fixed;

}

/\*按钮样式\*/

.contack .btn{

width: 163px;

height: 32px;

background: url("../img/btn2.png") no-repeat;

/\*去除表单项默认边框\*/

border: none;

/\*去除默认内外边距，并重新设置\*/

padding: 0;

margin: 11px 0 23px 0;

/\*字体样式\*/

color: white;

font: 13px Georgia;

/\*设置文字位置\*/

text-align: left;

/\*首行缩进1em\*/

/\*联系栏右侧的按钮在ie7及以下浏览器也会跟着向右移动1em，需要单独设置回去\*/

text-indent: 1em;

/\*设置鼠标移入时指针的样式\*/

cursor: pointer;

}

/\*设置联系栏标题\*/

.contack h2{

color: #444;

font: 18px/1 Georgia;

border-bottom: 1px #d3d3d3 dashed;

/\*设置内边距\*/

padding: 44px 0 10px 0;

/\*设置下外边距\*/

margin-bottom: 15px;

}

/\*左侧联系栏正文样式\*/

.sc .p1{

color: #444;

font: 12px/1.2 微软雅黑;

}

/\*五个小图标样式\*/

.sc .icon{

/\*去除空格，取消图标左右间距\*/

font-size: 0px;

/\*设置上外边距\*/

margin-top: 4px;

}

/\*为每个图标设置样式\*/

.sc .icon a{

margin-right: 6px;

}

/\*设置联系栏左侧的第二个标题\*/

.sc .nl{

/\*字体样式\*/

font: bold 18px/1 "Nirmala UI";

/\*清除内边距，并重新设置\*/

padding: 31px 0 10px;

/\*为ie7及以下浏览器设置内边距\*/

\*padding: 36px 0 10px;

margin-bottom: 15px;

}

/\*左侧联系栏的文本框\*/

.sc .txt{

/\*外边距\*/

margin: 8px 0 2px;

}

/\*右侧联系栏的图片样式\*/

.nu img{

/\*向左浮动，使文字环绕图片\*/

float: left;

margin-right: 8px;

}

/\*右侧联系栏的图片和文本样式\*/

.nu p{

/\*设置高度\*/

height: 58px;

/\*为ie7及以下浏览器设置高度\*/

\*height: 60px;

color: #444;

font: 12px/1 微软雅黑;

/\*设置底部虚线边框\*/

border-bottom: 1px dashed #d3d3d3;

margin-bottom: 14px;

padding-bottom: 9px;

}

/\*联系栏右侧的最后一行图文不需要底部外边距\*/

.nu .no-margin{

margin-bottom: 0px;

}

/\*联系栏右侧的按钮在ie7及以下浏览器也会跟着向右移动1em，需要单独设置回去\*/

.nu .btn{

\*margin-left: -1em;

}

/\*联系栏样式结束\*/

/\*底部样式开始\*/

.footer{

height: 173px;

/\*背景颜色\*/

background-color: #333333;

/\*设置上边框\*/

border-top: 10px solid #4c4c4c;

}

/\*设置底部文字样式\*/

.footer p

,.footer a{

color: #999;

font: 11px 微软雅黑;

/\*去掉下划线\*/

text-decoration: none;

}

/\*第一行文字\*/

.footer .copy{

/\*向右浮动\*/

float: right;

/\*解决ie6的双倍边距bug\*/

display: inline;

}

/\*设置底部文字的位置\*/

.footer p{

margin: 0 23px 12px 6px;

}

/\*设置底部文字的位置\*/

.footer .w{

padding-top: 18px;

}

/\*鼠标移入超链接时的效果\*/

.footer a:hover{

color: white;

text-decoration: underline;

}

/\*底部样式结束\*/