[前置内容：基本概念 3](#_Toc450577492)

[1. HTML简介 3](#_Toc450577493)

[1.1 第1个网页 3](#_Toc450577494)

[1.2 网页的基本结构 3](#_Toc450577495)

[1.3 标签的基本结构 3](#_Toc450577496)

[2. HTML标签 4](#_Toc450577497)

[2.1 分隔标签 4](#_Toc450577498)

[2.1.1 文字上的分隔标签(换行) 4](#_Toc450577499)

[2.1.2 分隔线标签 4](#_Toc450577500)

[2.2 排版标签 5](#_Toc450577501)

[2.2.1 文字置左、置中、置右 5](#_Toc450577502)

[2.2.2 置中标签 5](#_Toc450577503)

[2.2.3 保持原始格式标签 5](#_Toc450577504)

[2.3 文字标签 6](#_Toc450577505)

[2.3.1 标题标签 6](#_Toc450577506)

[2.3.2 字体标签 6](#_Toc450577507)

[2.3.3 字型变化标签 7](#_Toc450577508)

[2.3.4 特殊字体,实体符号 8](#_Toc450577509)

[2.4 图片标签 8](#_Toc450577510)

[2.4.1 src属性 9](#_Toc450577511)

[2.4.2 height和width属性 9](#_Toc450577512)

[2.4.2 alt属性 9](#_Toc450577513)

[2.5 超链接标签 10](#_Toc450577514)

[2.5.1 外部链接 10](#_Toc450577515)

[2.5.2 内部链接 11](#_Toc450577516)

[2.6 序列 12](#_Toc450577517)

[2.6.1 无序序列 12](#_Toc450577518)

[2.6.2 有序序列 12](#_Toc450577519)

[2.7 容器标签 14](#_Toc450577520)

[2.7.1 div 14](#_Toc450577521)

[2.7.2 span 14](#_Toc450577522)

[2.8表格 14](#_Toc450577523)

[2.8.1 table 14](#_Toc450577524)

[2.8.2 tr 15](#_Toc450577525)

[2.8.3 th, td 15](#_Toc450577526)

[2.9表单 17](#_Toc450577527)

[2.9.1 Input标签 17](#_Toc450577528)

[2.9.2 下拉列表 19](#_Toc450577529)

[2.9.3 多行文本框 19](#_Toc450577530)

[3. CSS 21](#_Toc450577531)

[3.1页面使用CSS的几种方式 21](#_Toc450577532)

[3.1.1 行内样式 21](#_Toc450577533)

[3.1.2 内部样式 21](#_Toc450577534)

[3.1.3 外部样式 22](#_Toc450577535)

[3.2 常用选择器 23](#_Toc450577536)

[3.2.1 基本选择器 24](#_Toc450577537)

[3.2.2 复合选择器 27](#_Toc450577538)

[3.2.3 CSS的特性 30](#_Toc450577539)

[3.3常用CSS 32](#_Toc450577540)

[3.3.1 文字样式 32](#_Toc450577541)

[3.3.2 背景样式 33](#_Toc450577542)

[3.3.3. 超链接样式 33](#_Toc450577543)

[3.3.4 列表样式 35](#_Toc450577544)

[3.4 CSS布局 35](#_Toc450577545)

[3.4.1 盒子模型 35](#_Toc450577546)

[3.4.2 盒子的浮动 35](#_Toc450577547)

[3.3.3 盒子的定位 35](#_Toc450577548)

# 前置内容：基本概念

|  |  |  |  |
| --- | --- | --- | --- |
| 网页要素 | 实现技术 | 标准制定者 | 标准版本 |
| 内容 | html | W3C(万维网联盟) | 4.0/5.0 |
| 外观 | css | W3C(万维网联盟) | 2.0/3.0 |
| 行为 | javascript | ECMA（欧洲计算机制造商协会） | 1.8.1 |

# HTML简介

HTML（Hypertext Mark-up Language）即超文本标记语言，是目前网络上应用最为广泛的语言，也是构成网页文档的主要语言

HTML文档是由HTML标签组成的描述性文本，其命令可以说明文字、图形、动画、声音、表格、链接等元素

## 第1个网页

## 网页的基本结构

文件扩展名 ： .htm, 或者 .html

<HTML>  
　<HEAD>  
　　<TITLE>网页标题</TITLE>  
　　<Meta>  
　</HEAD>  
　<BODY>  
　　BODY之间则为主要语法所在，也是网页的主要呈现部分。  
　</BODY>  
</HTML>

## 1.3 标签的基本结构

在HTML中，元素由标签组成，标签包含在一对尖括号内， 如 <body>

大多数标签是成对使用的称为双标签，前一个称为开始标签，后一个称为结束标签，内容放在两个标签之间，如：  
 <b>要加粗的文字</b>

双标签语法: <开始标签 属性1名=”属性1值” 属性2名=”属性2值”> …….. </结束标签>

也有些标签是单独使用的，如 <br/>、<hr/> 等，这些称为单标签

虽然标签不区分大小写，但我们建议使用小写；

单标签语法: <开始标签/>

# 2. HTML标签

## 2.1 分隔标签

### 2.1.1 文字上的分隔标签(换行)

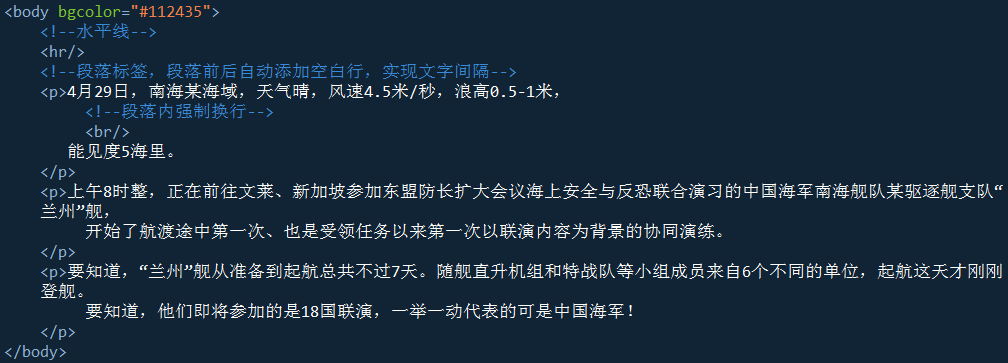
<p> 段落标签: p 元素会自动在段落前后创建一些空白

<br/>：段落内强制换行

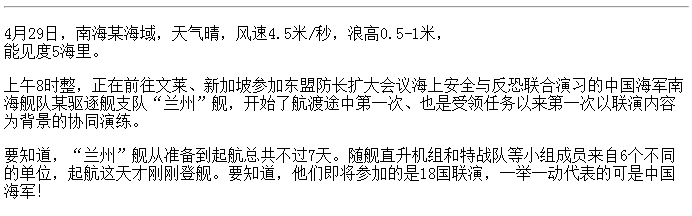
### 2.1.2 分隔线标签

<hr/> 分隔线：在 HTML 页面中创建一条水平线

代码:



运行效果：



## 2.2 排版标签

### 2.2.1 文字置左、置中、置右

通过p标签中的align属性实现: left, center,right，默认值为left

### 2.2.2 置中标签

center: 内容居中显示

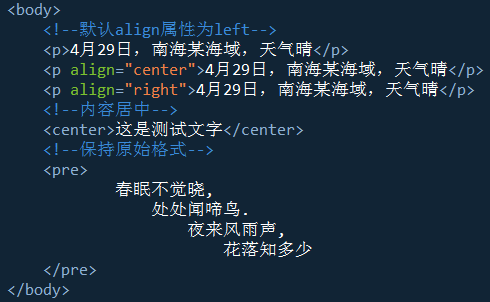
语法：<center>居中内容</center>

### 2.2.3 保持原始格式标签

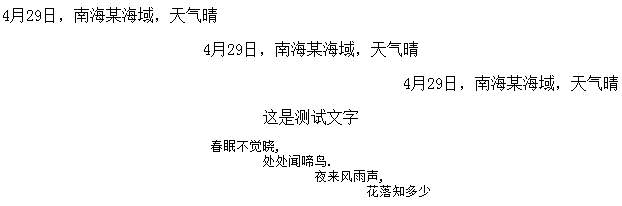
Pre:将源文件中的格式原封不动的呈现出来

语法：<pre>内容</pre>

代码：



效果：



## 2.3 文字标签

### 2.3.1 标题标签

<h1>到<h6>, h1最大，h6最小，标题自身占一行

### 2.3.2 字体标签

<font>文字内容</font>

#### 2.3.2.1 字体大小属性

Size=””,最小1，最大7，默认3

#### 2.3.2.2 字体颜色属性

Color=”颜色”，颜色可以用英文单词表示，例如：color=”red”

也可以用#后跟6位16进制的数来表示，这6位16进制的数2个为一组，分别表示在这个颜色中R(red:红),G(green:绿),B(blue:蓝)所占的比例

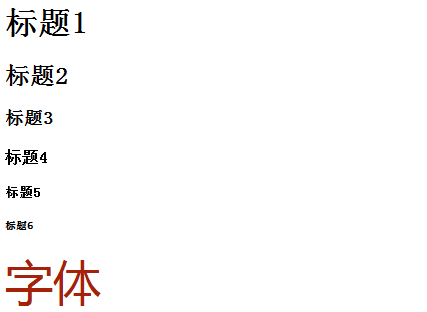
#### 2.3.2.3 字体字型属性

Face=”字体”

代码：



效果：



### 2.3.3 字型变化标签

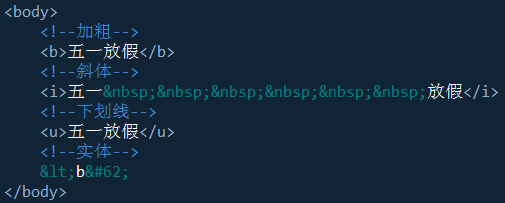


### 2.3.4 特殊字体,实体符号





代码：



效果：



## 2.4 图片标签

<img />

### 2.4.1 src属性

路径的问题:

相对路径：相对当前网页所在的位置而言

绝对路径：以/或者盘符开始



### 2.4.2 height和width属性

height,width：设置图片的宽，高

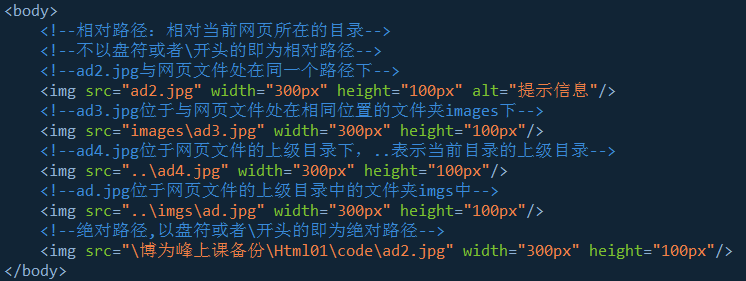
单位：px ,像素：相对长度单位

若把影像放大数倍，会发现这些连续色调其实是由许多色彩相近的小方点所组成，这些小方点就是构成影像的最小单位“像素”。这种最小的图形的单元能在屏幕上显示通常是单个的染色点。像素的大小是会“变”的，也称为“相对长度”。

### 2.4.2 alt属性

图片不存在情况的提示

代码：



效果：



## 2.5 超链接标签

<a>显示文本</a>

### 2.5.1 外部链接

#### 2.5.1.1 href属性

Href=”要转向的页面的地址”

#### 2.5.2.2 target属性

Target：设置在哪里打开新地址

\_self: 默认值: 在当前窗口

\_blank:在新窗口打开新地址

#### 2.5.2.3 不同种类的外链接



### 2.5.2 内部链接

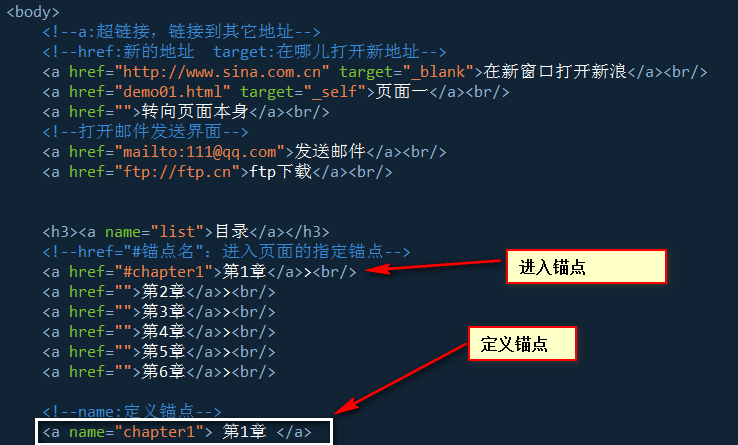
#### 2.5.1.2 name属性(锚点)

Name: 给超链接起名

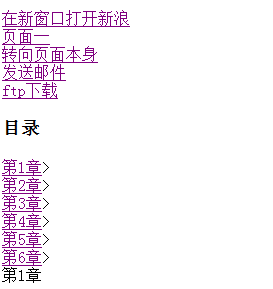
同一个页面里进入这个锚点： href=”#锚点名”

其它页面进入这个锚点: href=”页面访问路径 #锚点名”

代码：



效果：



## 2.6 序列

### 2.6.1 无序序列

<ul>

<li></li>

<li></li>

<li></li>

<li></li>

</ul>

### 2.6.2 有序序列

<ol>

<li></li>

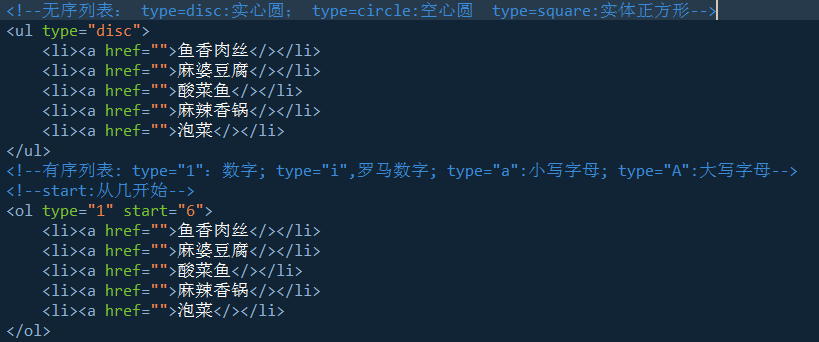
<li></li>

<li></li>

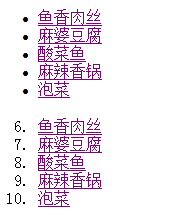
<li></li>

</ol>

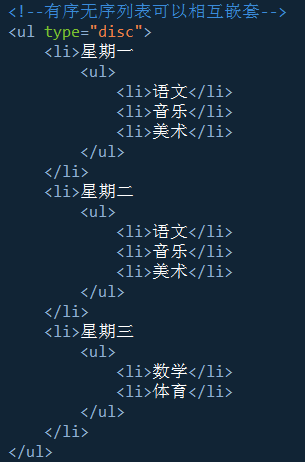
简单列表:



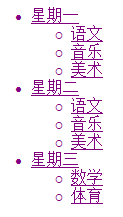
效果:



嵌套列表:



效果：



## 2.7 容器标签

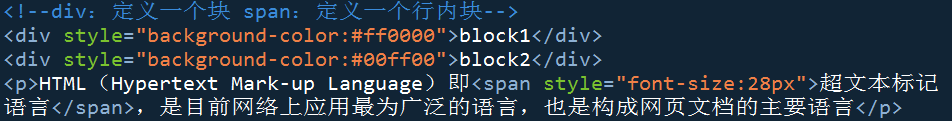
### 2.7.1 div

在网页中定义一个块，形成一个独立的区域，块的前后会有换行

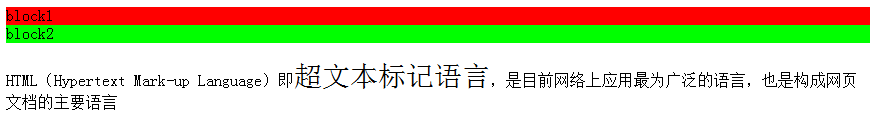
### 2.7.2 span

在行内定义一个块，此块不会换行

代码:



效果:



## 2.8表格

### 2.8.1 table

表格的最外层标签

<table>

表格内容

</table>

table常用属性:

width, height: 设置表格的宽，高; 2种设置方法，第1种固定长度，直接设置长度,

例如: width=”1000px”

第2种设置相对父容器宽度的百分比, 例如: width=”80%”

align=”center/left/right”: 设置表格水平位置居中/靠左/靠右

cellspacing: 规定单元格之间的空白

cellpadding: 规定单元边沿与其内容之间的空白

### 2.8.2 tr

table的子标签

行标签

<tr>

</tr>

常用属性：

width, height: 设置行的宽，高; 2种设置方法，同table

align: 整行的水平对其方式

valign：整行的垂直方向的位置 bottom/center/top

### 2.8.3 th, td

均为tr的子标签

th：表头标签, 表头：文本自动加粗，文本居中

td: 列标签: 默认靠左，普通文本

<tr>

<th></th>

<th></th>

</tr>

<tr>

<td></td>

<td></td>

</tr>

常用属性：

width, height: 设置单元格的宽，高; 2种设置方法，同table

align: 单元格的水平对其方式

valign：单元格的设置垂直方向的位置 bottom/center/top

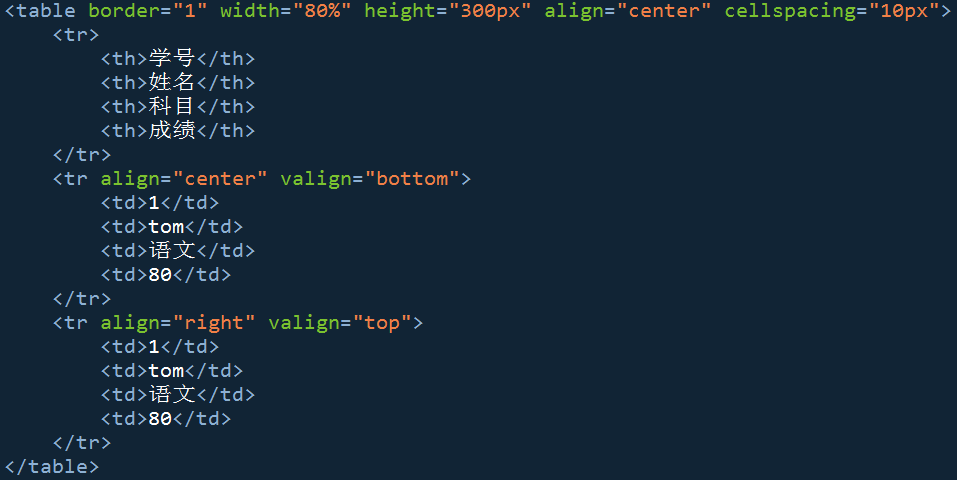
不规则单元格：

rowspan=”合并行的行数”, 例如： rowspan=”3”, 表示合并3行

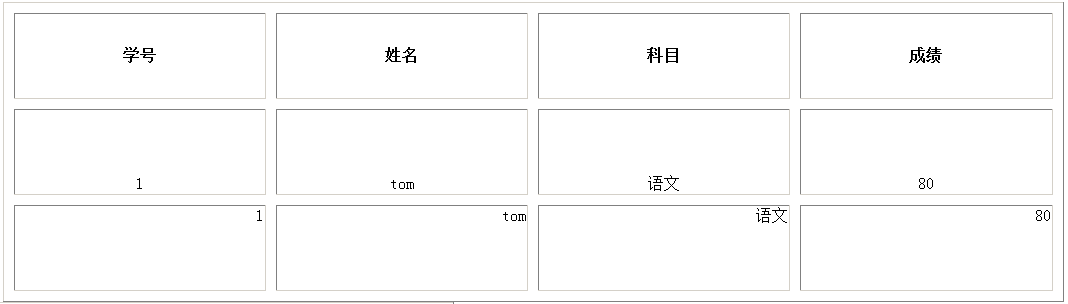
colspan=”合并列的列数” , 例如： colspan =”2”, 表示合并2列

规则表格：

代码：

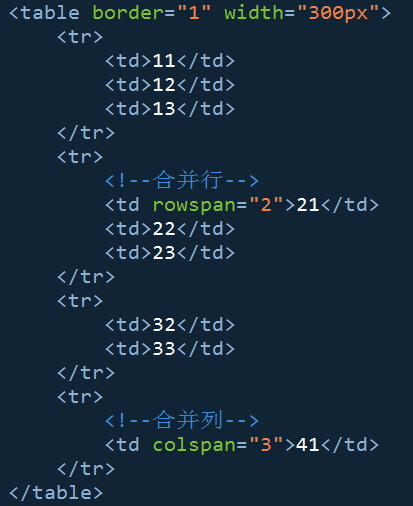


效果：

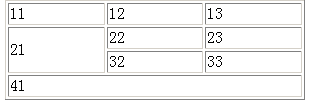


不规则表格：

代码：



效果：



## 2.9表单

作用：接受用户的数据，发送到服务器

form 标签: 表单的最外层的标签,所有表单元素都应该放在这对标签之间

<form>

表单内容

</form>

method: 设定表单的提交的方式 get/post, 默认是get

action: 设定表单提交的目的地

enctype: 设定表单数据提交时的编码的方式

### 2.9.1 Input标签

设定input的type属性，呈现不同的表单元素

所有input元素共有的属性:

name: 表单元素的名字，同时是上传数据的参数名

value: 表单元素的值，是上传数据的值

disabled: 不可用，不可用的表单数据不会上传到服务器

id: 表单元素的唯一标记，同一个网页中，所有的id应该唯一

上传数据的格式： name=value

#### 2.9.1.1 文本框

type=”text”

size=”显示尺寸”

maxLength=”允许输入最大字符数”

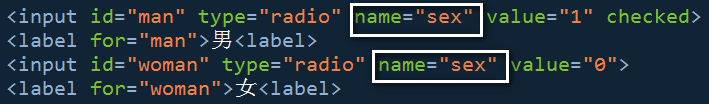
readOnly: 只读

#### 2.9.1.2 密码框

type=”password”, 其它属性同文本框

#### 2.9.1.3 单选按钮

type=”radio”， 1组单选按钮的name属性必须相同



在radio中添加id属性，然后在label中添加for=”id”,可以扩大单选按钮的选中范围，复选框，文本框也是如此

#### 2.9.1.4 复选按钮

type=”checkbox”

#### 2.9.1.5 隐藏域

type=”hidden”: 页面不可见，数据会被传送到服务器

#### 2.9.1.6 按钮

type=”submit”

type=”image” src=”图片的位置”

这2中按钮有默认行为: 提交表单

type=”reset”

有默认行为: 重置表单，将表单中的数据恢复到初始状态

type=”button”

没有默认行为，通过自己定义脚本

#### 2.9.1.7 上传文件域

type=”file”

### 2.9.2 下拉列表

<select name=””>

<option value=””>选项1</option>

<option value=””>选项2</option>

<option value=””>选项3</option>

</select>

与单选，复选类似，option标签之间的内容是显示的内容，上传到服务器的是其value值

### 2.9.3 多行文本框

<textarea rows=”几行” cols=”几列”>显示的内容

</textarea>

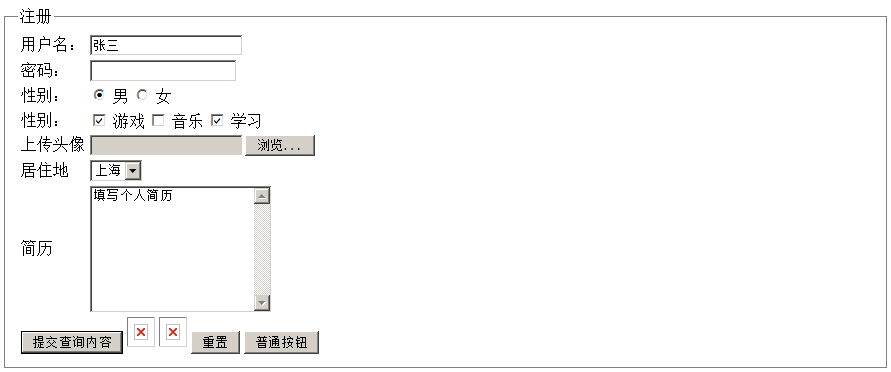
多行文本框中显示的内容为源文件中 textarea标签之间的内容，假如希望多行文本框的初始值为空，在源文件中，开始和结束标签之间不能有任何内容

例如： <textarea></textarea>

代码：



效果：



## 2.10 框架标签

### 2.10.1 frameset

1个浏览器中显示多个页面

**基本结构:**

<frameset 属性列表>

<frame src=”网页的地址”/>

<frame src=”网页的地址”/>

</frameset>

使用frameset, 不能写body

**frameset常用属性：**

rows: 设置垂直方向如何分割,

例如: rows=”120px,\*”: 第1部分高度120px, 第2部分使用剩余所有高度

rows=”120px,\*,\*”: 第1部分高度120px, 第2、3部分均分剩余高度

rows=”20%, 80%”: 第1部分高度为总高度的20%, 第2部分高度为总高度的80%

cols: 设置水平方向如何分割,设置方式同rows

frameborder: 是否显示边框； 1：显示，0:不显示

border: 边框的宽度

在frameset中还可以再嵌套frameset

**frame常用属性:**

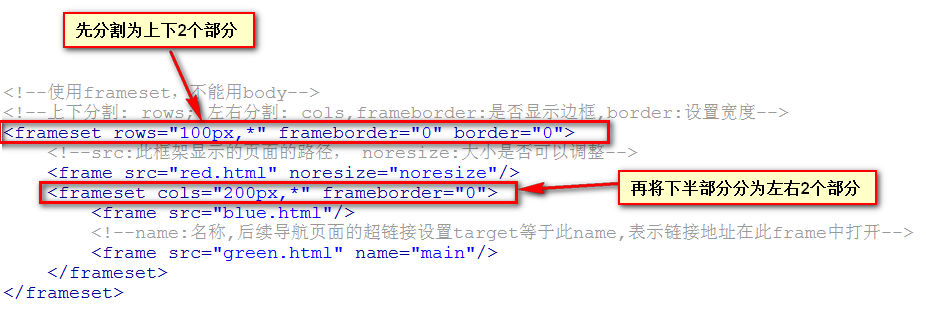
src: 显示的页面地址

noresize:是否大小可调

name:名称

导航页面的实现:

1. 框架页面：



1. 导航页面：

其中的超链接的target要与目标框架的name相一致



效果：



### iframe(内联框架)

写在body里的frame

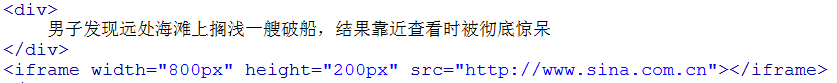
**常用属性：**

width:宽度

height:高度

src:地址

代码：



效果：



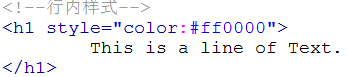
# 3. CSS

CSS: Cascading Style Sheet ， 层叠样式表

## 3.1页面使用CSS的几种方式

### 3.1.1 行内样式

直接设置网页元素的style属性



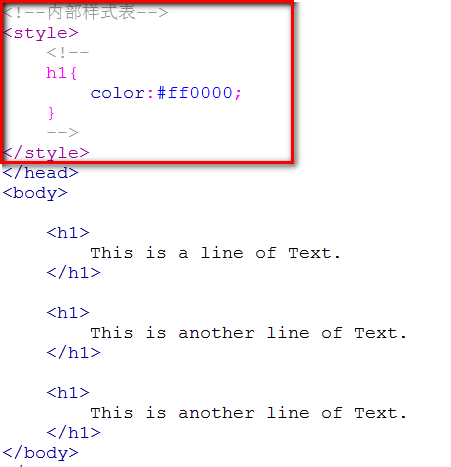
效果：



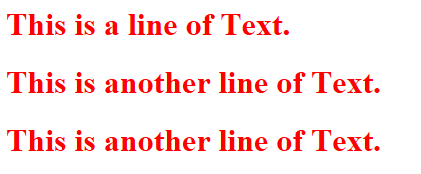
### 3.1.2 内部样式

写在网页的<head></head>的部分

写在一对 <style></style> 标签中



效果：



### 3.1.3 外部样式

目录结构：



将样式单独定义到样式文件里，样式文件的扩展名为 .css

网页文件中引用外部的样式文件

网页文件中引入外部样式文件的方式有2种：

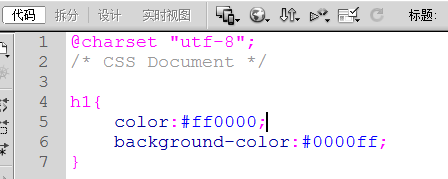
方式一: 在<head>的部分使用Html里的语法：<link>标签



方式二: 在<head>的部分使用Css里的语法: css的import指令



样式表文件内容：



### 3.2 常用选择器

样式表的基本语法:

选择器{

属性1名: 属性值；

属性2名: 属性值；

……

}

### 3.2.1 基本选择器

#### 3.2.1.1 标签选择器

以网页标签的名字作为选择器的名字，

影响的范围：网页中此类标签

代码：



效果：



#### 3.2.1.2 类选择器

.类选择器的名字

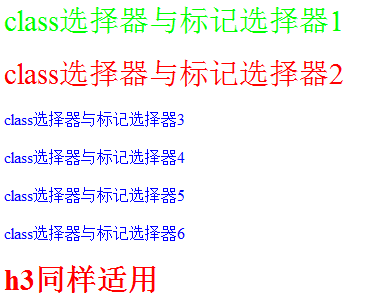
使用类选择器: 在要使用这个选择器的网页元素中使用class属性 class=”类选择器的名字”

影响的范围：网页中所有设置了class=””的标签，可以跨标签

代码：

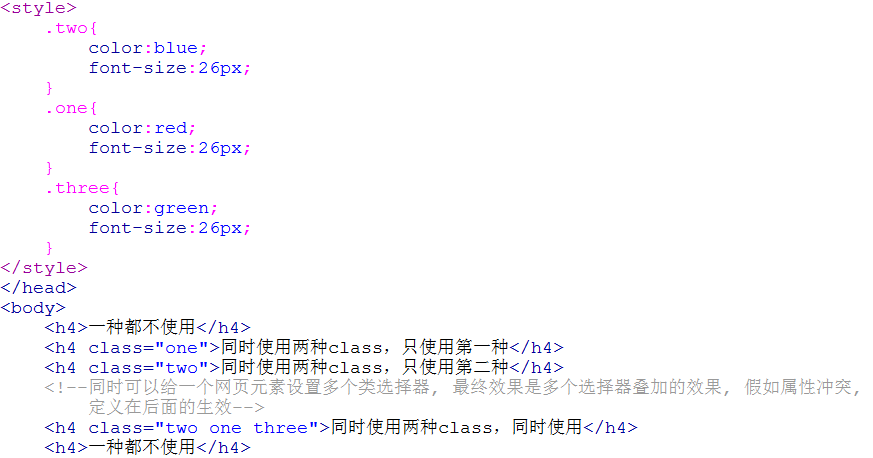


效果：



一个网页元素使用多个样式：

代码：



效果：



#### 3.2.1.3 ID选择器

#类选择器的名字

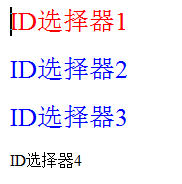
使用ID选择器: 在要使用这个选择器的网页元素中设置ID属性: id=” 类选择器的名字”

影响的范围：网页中设置了此ID的标签，通常只影响单个网页元素

代码：



效果：



### 3.2.2 复合选择器

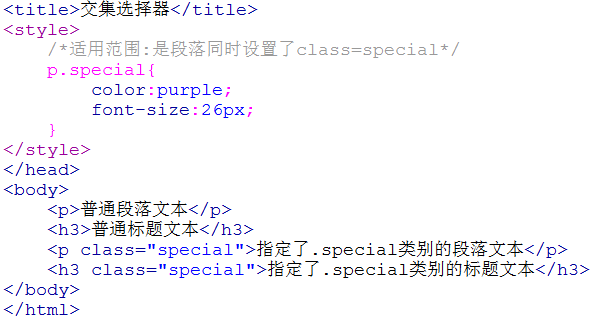
#### 3.2.2.1 交集选择器

以标签选择器开始，后面跟上类选择器或者ID选择器

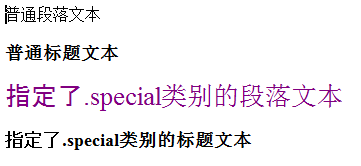
例如： p.speical{ }

影响到2同时满足2种选择器的元素

代码:



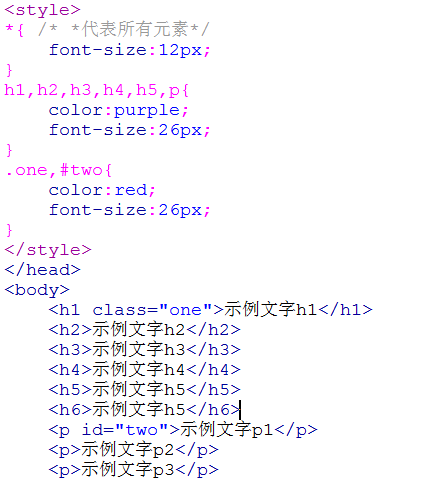
效果:



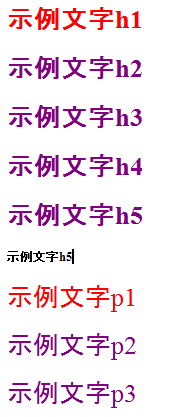
#### 3.2.2.2 并集选择器

样式相同的多个选择器的集体声明，多个选择器之间用逗号隔开

代码：



效果：

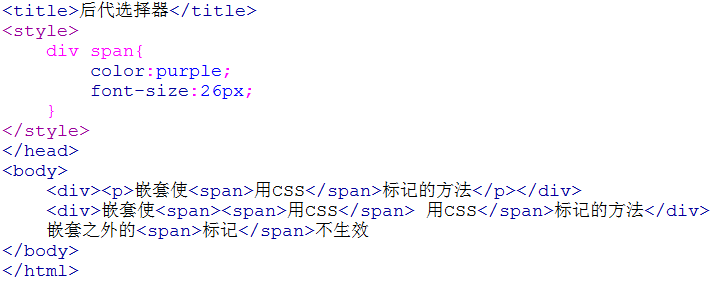


#### 3.2.2.3 后代选择器

语法：选择器1**空格**选择器2

影响范围: 选择器1的后代(包括直接后代和间接后代)中所有符合选择器2的元素

代码：



效果：

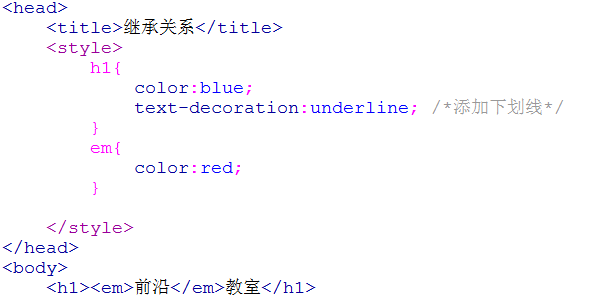


### 3.2.3 CSS的特性

#### 3.2.3.1 继承

子节点可以通过继承获得父节点的样式

代码：



效果：

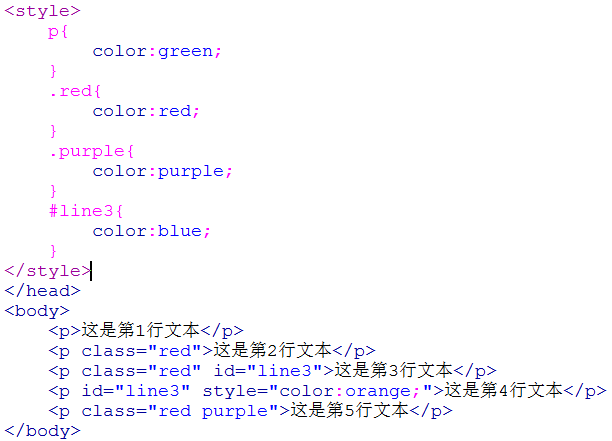


#### 3.2.3.2 层叠

同一元素上适用多个选择器，看到的是多个选择器层叠的效果

优先级：行内 〉 ID 〉 类选择器 〉 标签

代码：



效果：

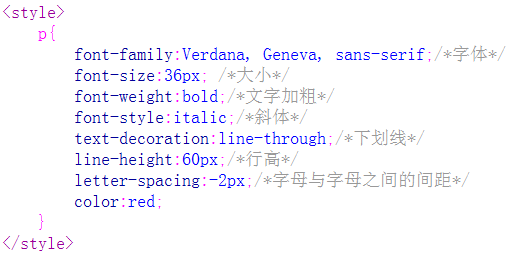


## 3.3常用CSS

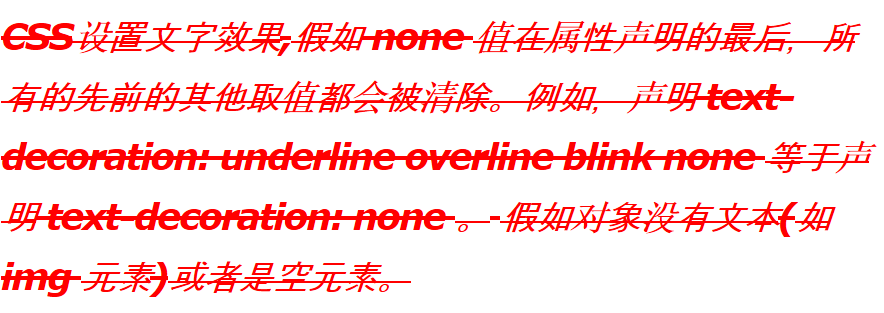
### 3.3.1 文字样式

文字常用样式：

代码：

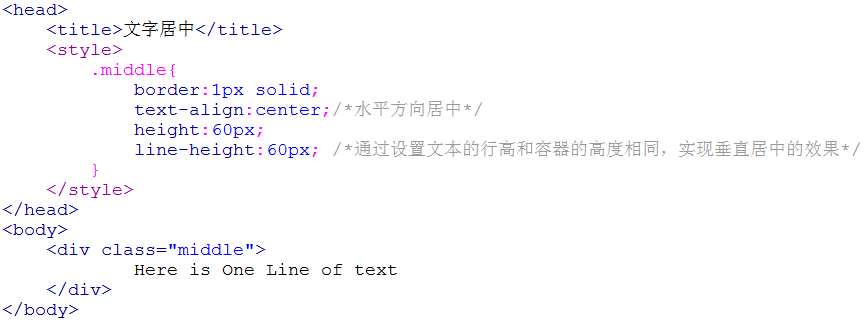


效果：

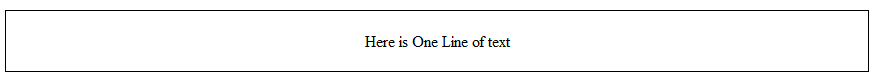


文字水平、垂直居中：

代码：

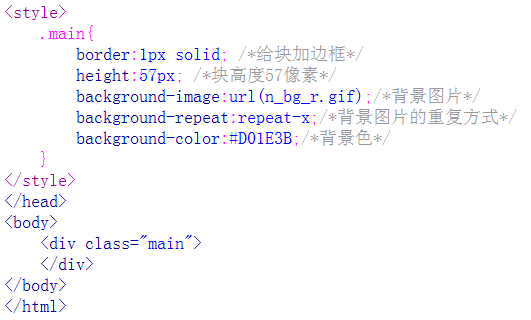


效果：



### 3.3.2 背景样式

代码：



效果：



### 3.3.3. 超链接样式

4个伪类：

：link、:visited、:hover、:active 设置超链接在4种状态下的样式

:link; 正常状态下

:visited; 访问过的

:hover; 悬停状态

: active; 激活状态

代码：

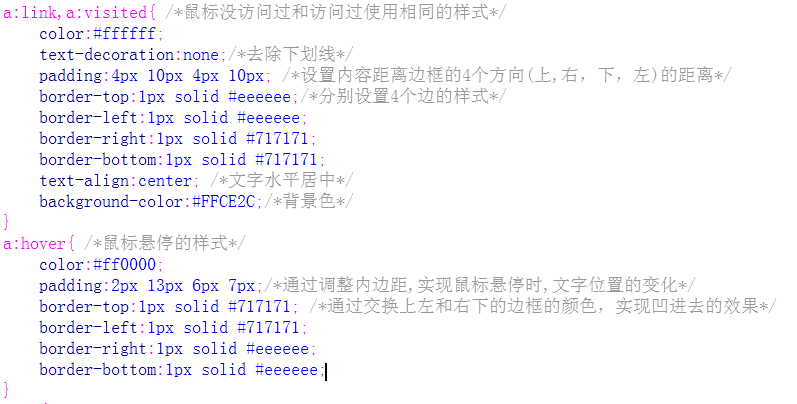


效果：



使用伪类修饰超链接：

代码：



效果：

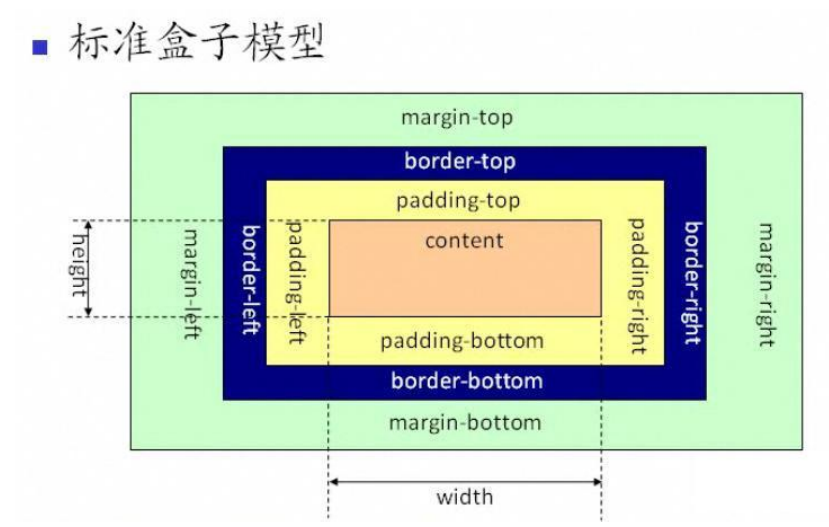


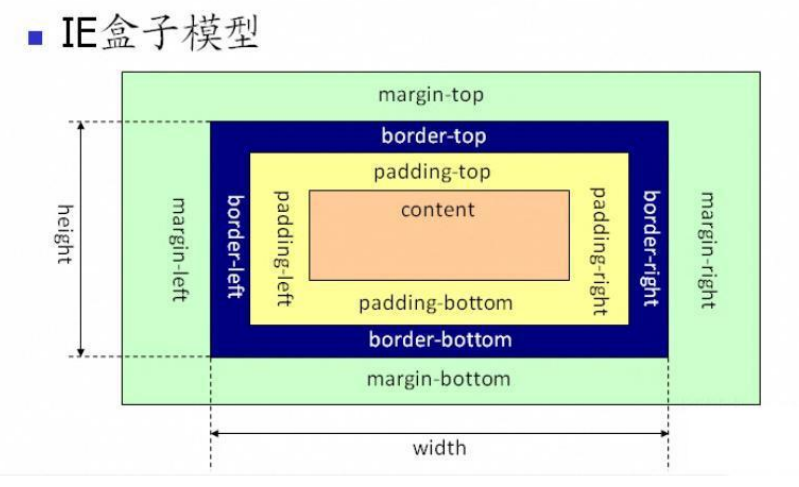
### 3.3.4 列表样式

## 3.4 CSS布局

### 3.4.1 盒子模型

#### 3.4.1.1 基本概念





在网页中添加文档类型，告知所有浏览器都采用标准 W3C 盒子模型去解释你的盒子，网页就能在各个浏览器中显示一致了。

网页标签分为2大类：

块级元素：前后会换行: div，p, ul,ol,li

行内元素：前后不会换行: span, a, 文字

div里面可以嵌套span, span里不能嵌套div

如何判断某个元素为块级元素或者行内元素：

网页中写2个相同的元素标签，如第2个换行，则为块级元素，否则为行内元素

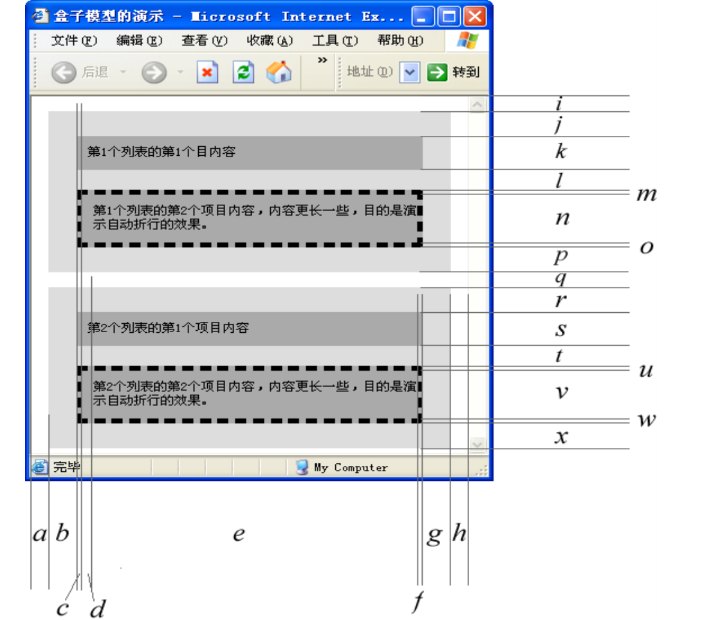
#### 3.4.1.2 盒子之间距离的计算

2个盒子水平方向的距离=左边盒子的右外边距+右边盒子的左外边距

2个盒子垂直方向的距离=max(上面盒子的下外边距, 下面盒子的上外边距之间)

嵌套盒子某个方向的距离=里边盒子此方向上的外边距+外边盒子此方向的内边距

练习:



a=body的左Padding+ul的左margin=0+15=15px

b=ul的左Padding+li的左margin=5+20=25px

c=li的border的左width=5px

d=li的左padding=10px

f= li的border的右width =5px

g=ul的右Padding+li的右margin=5+20=25px

h= body的右Padding+ul的右margin=0+15=15px

e=浏览器宽度-a-b-c-d-f-g-h

i= body的上Padding+ul的上margin=0+15=15px

j= ul的上Padding+li的上margin=5+20=25px

k=文字行高+上下padding=(文字的font-size+2px)+10+10=34px

l=max(不带边框的li的下margin, 带边框的li的上margin)=max(20,20)=20px

m= li的border的上width=5px

n=(文字行高)\*2+上下padding =14\*2+10+10=48px

o= li的border的下width=5px

p= ul的下Padding+li的下margin=5+20=25px

q=max(上面的ul的下margin,下面的ul的上margin)=max(15,15)=15px

r=j=25px

s=k=34px

t=l=20px

u=m=5px

v=n=48px

w=o=5px

x=p=25px

### 3.4.2 盒子的定位

#### 3.4.2.1 float: 浮动

没有设置float属性， 盒子处于标准流，宽度随着父容器的变化伸展，高度是容纳内容最小高度，多个盒子之间是竖直排列的

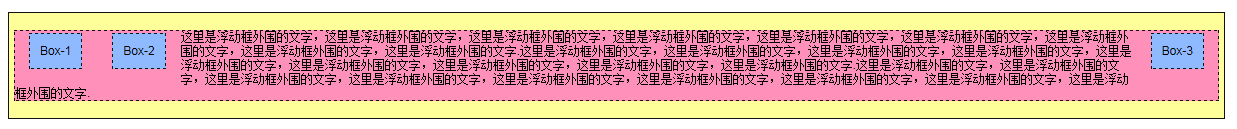
float: left|none|right: 浮动的盒子会从标准流里面脱离出来, 宽度不扩展,变成能容纳内容的最小宽度；后面的盒子视浮动盒子不存在，占据浮动盒子原来的位置；后面盒子的内容被挤成环绕浮动的盒子排列

clear: left|right|both: 清除浮动的影响：此块的内容不再环绕浮动的盒子排列，而是显示在浮动的盒子下方，利用此特性，可以撑开父容器，以容纳所有的子DIV.

代码：



效果：



#### 3.4.2.2 position: 位置

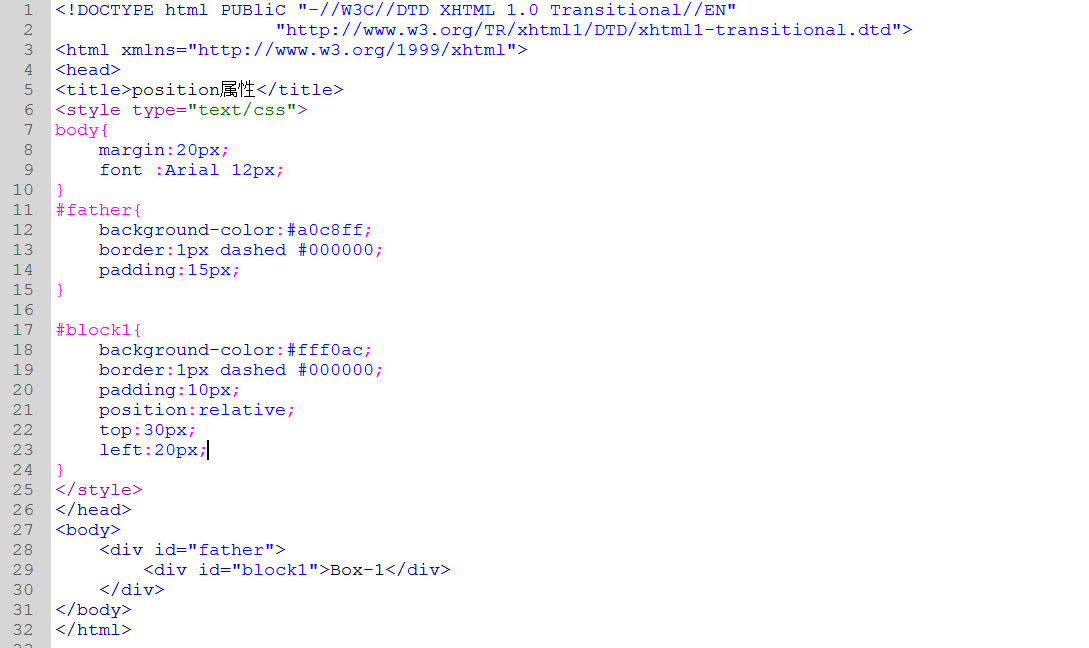
position: static|relative|absolute

static: 标准流

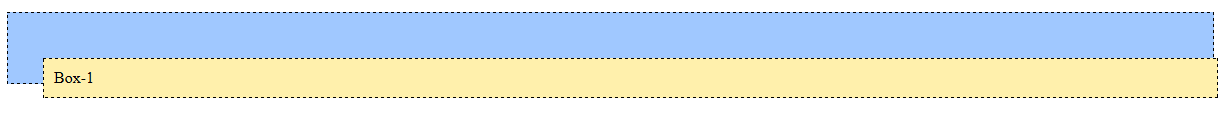
relative:相对位置，相对其在标准流中的位置

top|bottom|left|right

代码：



效果：



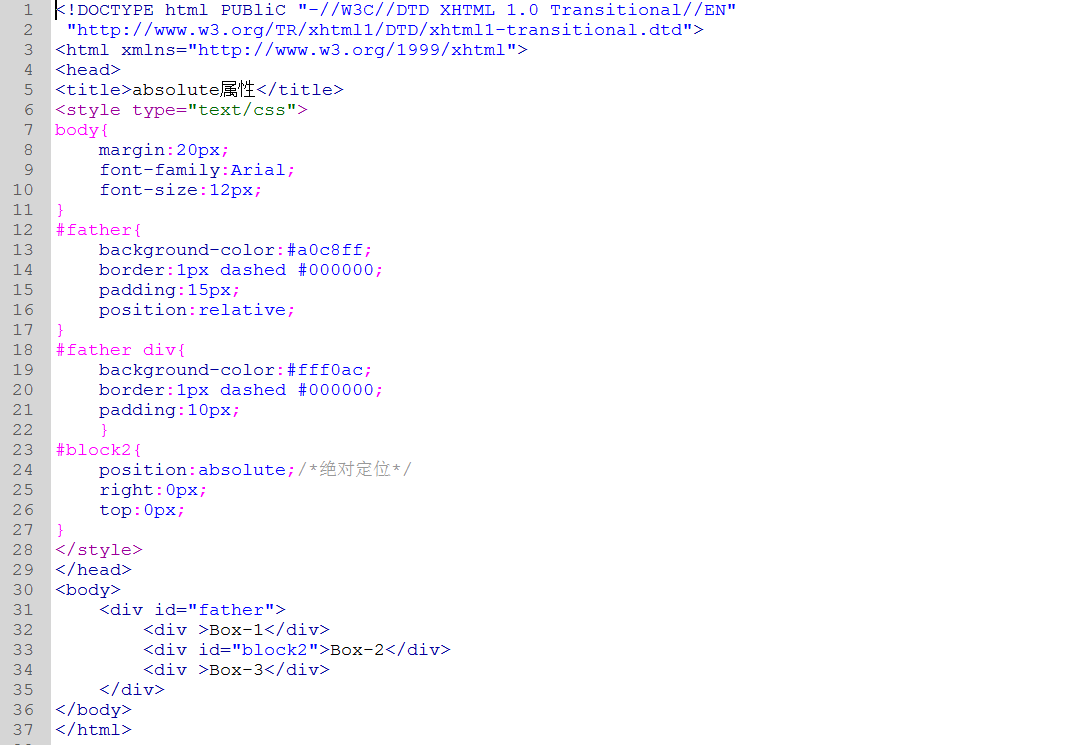
absolute:绝对位置, 以离它最近的一个 ”已经定位” 的 “祖先元素” 为基准进行定位。

已经定位：已经定义了position,并且position是除了static以外的任意一种类型

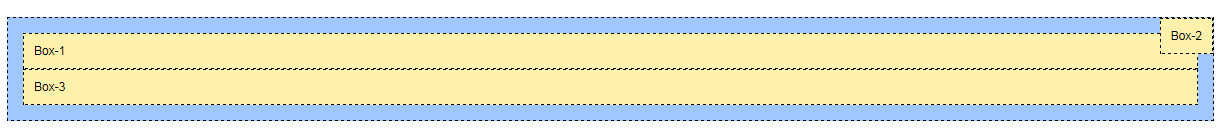
祖先元素：从此节点开始，到根节点所经过的所有节点

假如找不到已经定位的祖先元素，以浏览器窗口为基准进行绝对定位

代码：



效果：



心得笔记

<button> <input> 标签在处理盒子模型时 标准盒子模型无效，为IE模型

CSS3

transition：all 3s(时间)； //css实现动画

z-index：1； 图层级数越大 越上层

overflow：hidden;隐藏溢出部分

display：none/block/inline-block;

-webkit- 代表safari chrome浏览器

-moz 代表firefox 浏览器