目录

**键入章标题(第 1 级)1**

键入章标题(第 2 级)2

键入章标题(第 3 级)3

**键入章标题(第 1 级)4**

键入章标题(第 2 级)5

键入章标题(第 3 级)6

1. **DIV布局**
2. **Html htm 和shtml的区别**
   1. htm與.html沒有本質上的區別，表示的是同一種文件，只是適用於不同的環境之下
   2. DOS僅能識別8+3的文件名，所以\*.htm的命名方法可以被DOS識別，而\*.html的文件命名方式不能被之識別
   3. 在UNIX系統中，網頁必須使用.html擴展名，如果是htm，在瀏覽器中打開時則出現源代碼；Windows中支持兩種擴展名。
   4. 也可以說htm和html是Windows和UNIX對抗的產物
   5. shtml是一種用於SSI（Server Side Include，服務器端包含）技術的文件，是WEB在服務器提供的一種功能，並且在服務器端執行。
3. **HTML的发展史；html和xhtml的区别；doctype的类型**
   1. Html和xhtml的区别以及Html的发展史：

<http://developer.51cto.com/art/201005/202052.htm>

* 1. Doctype有三种类型：

1. 严格的：

DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" " <http://www.w3.org/TR/html4/strict.dtd>"

1. 处于严格和带框架的中间：

DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" " http://www.w3.org/TR/html4/ Transitional.dtd"

1. 带框架的文档：

DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" " http://www.w3.org/TR/html4/frameset.dtd"

1. 通用：<!DOCTYPE html> 学习谷歌的方式 用浏览器默认的doctype
2. **CSS有几种引入方式？DIV除了可以声明id来控制，还可以声明什么控制？如何让两个DIV并排显示？**
   1. css引入方式:
3. 内联样式

内联方式指的是直接在 HTML 标签中的 style 属性中添加 CSS。

示例：

<div style="background: red"></div>

这通常是个很糟糕的书写方式，它只能改变当前标签的样式，如果想要多个 <div> 拥有相同的样式，你不得不重复地为每个 <div> 添加相同的样式，如果想要修改一种样式，又不得不修改所有的 style 中的代码。很显然，内联方式引入 CSS 代码会导致 HTML 代码变得冗长，且使得网页难以维护。

1. 嵌入方式

嵌入方式指的是在 HTML 头部中的 <style> 标签下书写 CSS 代码。

示例：

<head>

<style>

.content {

background: red;

}

</style>

</head>

嵌入方式的 CSS 只对当前的网页有效。因为 CSS 代码是在 HTML 文件中，所以会使得代码比较集中，当我们写模板网页时这通常比较有利。因为查看模板代码的人可以一目了然地查看 HTML 结构和 CSS 样式。因为嵌入的 CSS 只对当前页面有效，所以当多个页面需要引入相同的 CSS 代码时，这样写会导致代码冗余，也不利于维护。

1. 链接方式

链接方式指的是使用 HTML 头部的 <head> 标签引入外部的 CSS 文件。

示例：

<head>

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

</head>

这是最常见的也是最推荐的引入 CSS 的方式。使用这种方式，所有的 CSS 代码只存在于单独的 CSS 文件中，所以具有良好的可维护性。并且所有的 CSS 代码只存在于 CSS 文件中，CSS 文件会在第一次加载时引入，以后切换页面时只需加载 HTML 文件即可。

1. 导入方式

导入方式指的是使用 CSS 规则引入外部 CSS 文件。

示例：

<style>

@import url(style.css);

</style>

* 1. 声明的控制方式：

div除了用ID赋予它一个名称外，还可以用class来赋予它一个名称。 还可以不用赋予它一个名字，直接用style来直接设定它的样式。

唯一区别就是ID是唯一的，只能用一次。反而CLASS是可以反复使用的，可以多个div同时时候class来定义样式。

* 1. 如何让两个DIV并排显示：

1. 设置并排显示的方法

设置浮动，我们对div设置一个float浮动属性即可解决不并排显示；

行内显示，我们加入display:inline即可解决实现同行并排显示div盒子对象。

1. 两者的区别

float:left;标签会浮动起来，不会占浏览器的区域;就是说其他标签的起始位置为0.0；会遮盖住其他的标签；

display:inline标签会沉下去，不会占浏览器的区域;就是说其他标签的起始位置为0.0；会被其他的标签遮盖住；

1. **用DIV画一个田字；验证父DIV的高度会不会被浮动的子DIV给撑开；**
   1. DIV画一个田字

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

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

<title>D用css来控制</title>

<style>

div {

border-style: solid;

border-color: #000000;

border-width:5px;

margin:10px;

width:100px;

height:100px;

}

.son {

height:200px;

float:left;

}

#Parents {

width:50%;

}

</style>

</head>

<body>

<div id="Parents">

<div class="son"></div>

<div class="son"></div>

<div class="son"></div>

</div>

</body>

</html>

* 1. 父DIV的高度自适应问题（子DIV设置了float属性）

1. 当父DIV设置了高度的时候，父DIV的高度不会被浮动的子DIV给撑开；而当父DIV没有设置高度的时候，CSS 子div设置float之后父div无法自适应高度问题（闭合(清除)浮动）导致父容器的高度塌陷了；
2. 解决父容器高度塌陷的问题最适用的解决方案是：这种方法就是向父容器的末尾再插入一个额外的标签，并令其清除浮动（clear）以撑大父容器。这种方法浏览器兼容性好，没有什么问题，缺点就是需要额外的（而且通常是无语义的）标签。我个人比较喜欢这种方法，因为它简单、实用、浏览器兼容性好，而且这种方法也是W3C推荐的方法

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

或者使用

<br style="clear:both;" />

例子：

<div id="main">

<div id="lside"></div>

<div id="rside"></div>

<br style="clear:both;"/>

</div>

1. **新手常犯的错误**
2. 忘了写Doctype标签
3. 标签的Id设置为数据；
4. 文件的编码和charset的值不一致；
5. **盒子模型**
6. **在边框画一个三角形；边框透明的方法**
7. **代码**

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

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

<title>用css画三角形</title>

<style>

#main{

border:150px solid transparent; /\*边框透明的颜色\*/

border-top-color:red;

width:0px;

height:0px;

}

</style>

</head>

<body>

<div id="main"></div>

</body>

</html>

1. **画圣诞树**
2. **代码**

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

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

<title>用css</title>

<style>

.main{

margin:-50px auto 0px;

border:150px solid transparent; /\*transparent边框透明的颜色\*/

border-top-style:hidden; /\*隐藏上面的边框，不占位置\*/

border-bottom-color:#0c7008; /\*设置下边框的颜色\*/

width:0px;

height:0px;

}

.top{

margin:0px auto;

border-width:100px

}

.cender{

border-width:125px

}

#bottom{

width:50px;

height:150px;

background:#572a1d;

margin:0px auto;

}

</style>

</head>

<body>

<div class="main top"></div>

<div class="main cender"></div>

<div class="main" style="margin-top:-80px"></div>

<div id="bottom"></div>

</body>

</html>

1. **如果盒子的宽和高各是100px,同时盒子有30px的padding值；再设置背景为灰色，灰色的面积有多大？**

解：背景会包括padding值，所以灰色的面积为(30+100+30)\*(30+100+30)

1. **如果布局完成且像素正好相容，突然增加padding,可能会导致什么情况发生？举例并说明解决办法？**

/\*加这三个属性的话说明盒子不会给padding和边框撑开（注意，marrgin还是会撑开的）\*/

-webkit-box-sizing: border-box;

-moz-box-sizing: border-box;

box-sizing: border-box;

1. **margin重叠的现象**

margin重叠现象就是相邻的两个普通元素，上下边距不是普通的相加而是取其中较大的边距值（注意是普通元素）

同时，父子元素也会发生margin重叠现象，具体情况如下

1. **内联元素的特点？哪些元素是内联元素，哪些元素是块状元素**
2. **特点：**

内联元素无法独占一行；内联元素无法设置宽高；内联元素上下方向的margin和padding值无法设置！

1. **块状元素和内联元素**

**块元素（block element）**

·address - 地址

·blockquote - 块引用

·center - 举中对齐块

·dir - 目录列表

·div - 常用块级容易，也是css layout的主要标签

·dl - 定义列表

·fieldset - form控制组

·form - 交互表单

·h1 - 大标题

·h2 - 副标题

·h3 - 3级标题

·h4 - 4级标题

·h5 - 5级标题

·h6 - 6级标题

·hr - 水平分隔线

·isindex - input prompt

·menu - 菜单列表

·noframes - frames可选内容，（对于不支持frame的浏览器显示此区块内容）

·noscript - 可选脚本内容（对于不支持script的浏览器显示此内容）

·ol - 排序表单

·p - 段落

·pre - 格式化文本

·table - 表格

·ul - 非排序列表

**内联元素（inline element）**

·a - 锚点

·abbr - 缩写

·acronym - 首字

·b - 粗体(不推荐)

·bdo - bidi override

·big - 大字体

·br - 换行

·cite - 引用

·code - 计算机代码(在引用源码的时候需要)

·dfn - 定义字段

·em - 强调

·font - 字体设定(不推荐)

·i - 斜体

·img - 图片

·input - 输入框

·kbd - 定义键盘文本

·label - 表格标签

·q - 短引用

·s - 中划线(不推荐)

·samp - 定义范例计算机代码

·select - 项目选择

·small - 小字体文本

·span - 常用内联容器，定义文本内区块

·strike - 中划线

·strong - 粗体强调

·sub - 下标

·sup - 上标

·textarea - 多行文本输入框

·tt - 电传文本

·u - 下划线

·var - 定义变量

1. **内联与块状的相互转化以及应用？display属性值？**
2. 相互转化及应用：

display:block; dipplay:inline;

常用在导航栏里面，一般来说导航栏都是一个个链接来的。就好像<ul><li><a>1235</a></li></ul>。导航栏就是由一个ul多个li做出来的，这样每项都是一个超链接。但是这样的话，想要链接的话，你唯有指着“1235”这几个数字才会触发超链接，这样用户体验不好，因为不是所有人用鼠标都很精准的，这是就要把a标签转化成块状标签，宽高设成和li的宽高一样，这样鼠标一进入这个块状元素就能触发链接，而不用指中那几个字了。 还有就是页面布局

ul的一个属性 list-style-type:none; 可以让前面的小点消失

1. Display的属性值

none 此元素不会被显示。

block 此元素将显示为块级元素，此元素前后会带有换行符。

inline 默认。此元素会被显示为内联元素，元素前后没有换行符。

inline-block 行内块元素。（CSS2.1 新增的值）

list-item 此元素会作为列表显示。

run-in 此元素会根据上下文作为块级元素或内联元素显示。

compact CSS 中有值 compact，不过由于缺乏广泛支持，已经从 CSS2.1 中删除。

marker CSS 中有值 marker，不过由于缺乏广泛支持，已经从 CSS2.1 中删除。

table 此元素会作为块级表格来显示（类似 <table>），表格前后带有换行符。

inline-table 此元素会作为内联表格来显示（类似 <table>），表格前后没有换行符。

table-row-group 此元素会作为一个或多个行的分组来显示（类似 <tbody>）。

table-header-group 此元素会作为一个或多个行的分组来显示（类似 <thead>）。

table-footer-group 此元素会作为一个或多个行的分组来显示（类似 <tfoot>）。

table-row 此元素会作为一个表格行显示（类似 <tr>）。

table-column-group 此元素会作为一个或多个列的分组来显示（类似 <colgroup>）。

table-column 此元素会作为一个单元格列显示（类似 <col>）

table-cell 此元素会作为一个表格单元格显示（类似 <td> 和 <th>）

table-caption 此元素会作为一个表格标题显示（类似 <caption>）

inherit 规定应该从父元素继承 display 属性的值。

1. **CSS**
2. **CSS控制段落**

text-indent:20px 控制段落缩进

text-align:center 控制段楼居中显示

text-decoration:line-througth 划线的方式（经过文字中间）

letter-spacing:20px 文字的间隔

完整的属性：

color 设置文本颜色

direction 设置文本方向。

line-height 设置行高。

letter-spacing 设置字符间距。

text-align 对齐元素中的文本。

text-decoration 向文本添加修饰。

text-indent 缩进元素中文本的首行。

text-shadow 设置文本阴影。CSS2 包含该属性，但是 CSS2.1 没有保留该属性。

text-transform 控制元素中的字母。

unicode-bidi 设置文本方向。

white-space 设置元素中空白的处理方式。

word-spacing 设置字间距。

1. **CSS控制文字大小以及颜色**

color:red;

font-style:italic;斜体或斜体

font-weight:bold;字体的大小

font-size:27px;字体的大小

line-height:46px;行高

font-family:Simsun;字体的家族，宋体

上面的属性要按照顺序写

1. **CSS字体设置技巧**

Font-family:’哈哈’,’微软雅黑’，‘黑体’，sans-serif;

设置的话，不太常用的字体要放在前面，谨防用户的机器没有

1. **CSS设置背景图片**

Background-color:blue;背景颜色

Background-image:url(small.jpg);背景图片

Background-repeat:no-repeat/x-repeat/y-repeat背景图片的铺展

Background-attackment:fiexd;背景图片就固定住了，不论滚动条拖到哪

都有背景图片

Background-position:背景图片的位置控制

1. **精确控制背景图片的位置**

通过background-position属性来控制位置

1. **CSS选择器**

Id选择器；类选择器；标签选择器；派生选择器；

div p: div下的所有子代

div>p: div的子标签中的p元素；

1. **有语义标签**
2. **H标签和P标签**

新闻标题一般用H显示

段落一般用P标签

1. **Img标签**

<img src=”dsj.jpg” alt=”菊花台” title=”鼠标放上去会显示”/>

Img默认是有边框和margin的值，一般会在样式初始化中设置img的display属性为display和去掉这些默认的设置；IMG标签是替代元素（内联），替代的内联元素一般来说宽度和高度是由引进来的东西所决定。

1. **替换元素**

CSS 里，可替换元素是这样一些元素， 其展现不是由CSS来控制的。这些外部元素的展现不依赖于CSS规范。 典型的可替换元素有 <img>、 <object>、 <video> 以及 <textarea>、 <input> 这样的表单元素。 一些元素，比如 <audio> 和 <canvas> ，只在一些特殊情况下是可替换元素。 使用了 CSS content 属性插入的对象被称作匿名的可替换元素。

CSS在某些情况下会对可替换元素做特殊处理，比如计算外边距和处理值为 auto 的情况。

需要注意的是，一部分（并非全部）可替换元素，本身具有尺寸和基线，会被一些 CSS 属性用到，比如 vertical-align。

1. **有序列表与无序列表**

list-style-type: none;去除小图标

再自己定义小图标

1. **表格**

设置table的边框类型：

border-collapse:collspse

设置colspan rowspan 独占的行数或者列数

1. **超链接**

<a href=”dsfd.html” target=”\_blank” title=”鼠标放放上来会产生提示”>index</a>

\_blank打开新标签的方式；

1. **锚点**

<a name=”maoq”></a>

上面的方式代表的是用href链接的时候，在末尾加上”#maoq”会跳到锚点处；

http://localhost:8927/HTM%E5%88%97%E8%A1%A8.html#m3

1. **伪类**

描述a标签的状态

a:visited 访问过的

a:link 普通的状态 可以简写成a

a:hove 鼠标经过

a:active 点击的瞬间

顺序为vlha，可不能搞反，搞反了会差错的

1. **实体**

空格 &nbsp; &#160;

< 小于号 &lt; &#60;

> 大于号 &gt; &#62;

& 和号 &amp; &#38;

" 引号 &quot; &#34;

' 撇号 &apos; (IE不支持) &#39;

￠ 分 &cent; &#162;

£ 镑 &pound; &#163;

¥ 日圆 &yen; &#165;

€ 欧元 &euro; &#8364;

§ 小节 &sect; &#167;

© 版权 &copy; &#169;

® 注册商标 &reg; &#174;

™ 商标 &trade; &#8482;

× 乘号 &times; &#215;

÷ 除号 &divide; &#247;

1. **首页实战**
2. **导航栏需要块状元素转成行内块状元素**
3. **行高与文字的居中问题：设置行高的高度等于高度就能文字竖直居中**

**line-hight = hight**

1. **浏览器兼容的问题:用到再查**
2. **附加**
3. **色彩的表示**

数字；颜色的英文；3原色；

1. **尺寸的表示**

像素；百分比；em(父元素的倍数)

1. **CSS画圆角（新的浏览器）**

border-radius:5px;

可以画圆

1. **相对定位于绝对定位**

Position:relative(相对)

相对于自己本来的位置定位，原来的位置别的元素占据不了(占位置)

Position:absolution（绝对）

相对于父元素的位置来定位(如果父元素没有position属性，就相对于父元素的父元素去定位，一直找到有为止)（不占位置）

Z-Index：层叠的高度

1. **Overflow溢出处理**

Overflow:hidden;visible;scroll;auto

Overflow:hidden设置老式浏览器的元素溢出

1. **表单**
2. **框架集：**

给相应的框架集起名字，然后通过a标签的target=”name”设定该标签的内容在相应的框架集上显示