

初识HTML

1、网页和html版本

- 1、网页：是html格式的文件，需要用浏览器来阅读
- 2、html：超文本标记语言，用来描述网页的一种语言
- 3、网页的形成：编写html代码，由浏览器来解析并生成网页
- 4、html版本：html2.0~html5.0 中间有个XHTML版本

2、常用浏览器

chrome(谷歌), ie, edge, opera(欧朋), safari

3、web标准

- 1、Web 标准是由 W3C 组织和其他标准化组织制定的一系列标准的集合
- 2、web标准包含三部分：结构 (html)、表现 (css)、行为(javascript)

4、HTML语法规范

- 1、HTML 标签是由尖括号包围的关键词，例如 <html>。
- 2、HTML 标签通常是成对出现的，例如<html> 和 </html>，我们称为双标签。
标签对中的第一个标签是开始标签，第二个标签是结束标签。
- 3、有些特殊的标签必须是单个标签（极少情况），例如
，我们称为单标签。

5、HTML结构标签

- 1、<html></html>:html标签，页面中最大的标签，称做根标签，表示页面的开始和结束
- 2、<head></head>:网页头部，在这对标签中必须要放上title标签
- 3、<title></title>:网页的标题，让文档拥有一个属于自己的网页标题
- 4、<body></body>:网页的主体，body中放置页面所有的内容，目前所有内容需放在body标签中

6、编辑工具

- 1) HBuilder：解压即用，无须安装
- 2) 用工具生成的代码中

<!DOCTYPE html>：文档类型声明标签,告诉浏览器这个页面采取html5版本来显示页面
<meta charset="UTF-8" />：必须写，表示采取 UTF-8来保存文字. 如果不写就会乱码

- 3) 其他注意点：

- a、html不区分大小写
- b、网页的主体内容需放在<body>这里放内容</body>标签的中间

一、文字控制标签

1、标题标签

<h1>~<h6>：文字会加粗加大，独占一行，h1到h6文字大小逐级递减

属性：为HTML标签提供附加信息，描述标签的特性

属性书写规范：

- a、标签可以拥有多个属性，必须写在开始标签名的后面
- b、属性之间不分先后顺序，标签名与属性用空格分开
- c、属性采取键值对的格式，属性名="属性值"，即key="value"的格式

2、段落标签p：独占一行，段落上下会有间隙

<p>段落文字</p>

常用属性

align="center"

例：

<p align="center">标题文字</p>

3、换行标签br：强制文本换行，是个单标签

 哪里需要换行放哪里即可

4、文本格式化标签

- a、字体标签font：修饰文字大小，颜色，字体

 要修饰的字体

- b、加粗标签strong：

 要加粗的文字

- c、倾斜标签em： 要倾斜的文字

- d、下划线标签ins：<ins> 要添加下划线的文字</ins>

- e、删除线标签del： 要添加删除线的文字

5、横线标签hr：添加水平线

<hr size="3" color="red" width="90%" />

常用属性：

size="3" # 线条粗细

color="gray" # 取值可以是颜色单词或十六进制颜色值

width="像素值或百分比" # 线条宽度

align="center"

6、特殊字符标记

特殊字符	描述	字符的代码
	空格	
<	小于号	<
>	大于号	>
&	和号	&
¥	人民币	¥
©	版权	©
®	注册商标	®
°	摄氏度	°
±	正负号	±
×	乘号	×
÷	除号	÷
²	平方2	²
³	立方3	³

7、注释

HTML中的注释以“<!--”开头，以“-->”结束

```
<!--
```

注释文字，内容可以换行

```
-->
```

二、图像标签

1、常用图像格式

a、gif：支持动画，支持透明，无损的图像格式，但只能处理256种颜色，常用于logo、小图标或色彩单一的图像

b、png：体积小，支持透明，颜色过渡平滑，不支持动画

c、jpg：显示颜色丰富，有损压缩（每修改一次会造成图像数据损失）

2、图像标签

```

```

注意：src 是必写属性，否则无法显示图片

常用属性：

src ="图像路径和名称"

hspace="像素值"

vspace="像素值"

alt="未显示图片时替代的文字"

title="鼠标放在图片时显示的文"

align="对齐方式" # 对齐方式 : left、right、top、bottom、center、middle

三、超链接标签

1、定义超链接标签a：用于从一个页面链接(跳转)到另一个页面

` 文字或图像 `

注意：href是必写属性，否则a不具备链接功能

常用属性：

`target="_blank"` `target`取值_self, 是默认值。_blank为在新窗口打开目标页面

2、链接分类

a、链接到外部页面：在href中指定外部页面的路径(绝对路径)

`< a href="http://www.baidu.com"> 百度`

b、网站内部页面相互链接：在href中指定内部页面路径(相对路径)

`< a href="index.html"> 首页 `

c、空链接：暂时没有链接目标，`href="#"`

d、下载链接：`href="文件路径"`，浏览器会下载该文件

e、锚点链接：快速定位到页面中的某个位置

1) 给目标位置添加id名称，如`<h3 id="title" >电脑列表</h3>`

2) 在a链接的 href 属性中，设置属性值为 **#名字** 的形式,如 ` 第2集 `

f、发送邮件：打开系统的邮件系统，向目标邮箱发邮件

`点我发邮件`

四、列表元素

1、列表元素分类

a、无序列表：每个列表项之间没有顺序

b、有序列表：各个列表按照一定的顺序排列

c、定义列表：对于术语或名词进行解释和描述，定义列表没有项目符号

2、无序列表标签：ul和li

语法：ul中包含li，ul表示列表的开始和结束，li用于组织每项文字，一对li独占一行。ul至少包含一对li

```
<ul type="circle">
  <li>第一个列表项文字</li>
  <li>第二个列表项文字</li>
</ul>
```

type取值：disc、circle、square

3、有序列表标签：ol和li

语法：ol中包含li，至少包含一对li

```
<ol type="A">
  <li>第一个列表项文字</li>
  <li>第二个列表项文字</li>
</ol>
```

type取值：1、A、a、I、i

4、定义列表标签：dl、dt和dd

语法：dl表示定义列表的开始和结束，dt组织标题内容，dd组织描述内容

```
<dl>
  <dt>标题</dt>
  <dd>描述</dd>
</dl>
```

综合案例：

```
<body background="../img/bg-1.jpg">
  
  <ol>
    <li>游戏背景</li>
    <li>
      <a href="#">角色设定</a>
    </li>
    <li>物品道具
      <ul type="disc">
        <li>武器装备</li>
        <li>消耗品</li>
      </ul>
    </li>
    <li>特色系统
      <ul type="disc">
        <li><a href="#">玩家团队</a></li>
        <li>对战系统</li>
      </ul>
    </li>
    <li>
      <a href="#">配置要求</a>
    </li>
```

```
</ol>  
</body>
```

五、表格标签

1、表格标签

- `<table>` `</table>` 是用于定义表格的标签。
- `<tr>` `</tr>` 一对tr表示一行，必须嵌套在 `<table>` `</table>` 标签中。
- `<td>` `</td>` 一对td表示一个单元格，必须嵌套在 `<tr>` `</tr>` 标签中。
- `<th>` `</th>` 用于显示每列的标题，文字加粗居中

例：表格显示学生信息

```
<table border="1" width="300">  
  <tr>  
    <th>姓名</th>  
    <th>性别</th>  
    <th>电话</th>  
  </tr>  
  <tr>  
    <td>小王</td>  
    <td>女</td>  
    <td>110</td>  
  </tr>  
  <tr>  
    <td>小明</td>  
    <td>男</td>  
    <td>120</td>  
  </tr>  
</table>
```

2、表格结构标签

1) thead

用于组织表格的表头内容，一般是第一行

2) tbody

用于组织表格的主体内容，一般是主要数据显示总分，对于长表格可以有多个tbody。

3) tfoot

用于组织表格的页脚内容，一般是用于显示汇总信息。

提示：thead, tbody, 和 tfoot 元素默认不会影响表格的布局，只在逻辑上更好的组织表格各个部分

3、合并单元格

1) 跨行

在td中添加属性，纵向合并

```
<td rowspan="行数">单元格内容</td>
```

2) 跨列

在td中添加属性，横向合并

```
<td colspan="列数">单元格内容</td>
```

3) 步骤

- 确定是跨行还是跨列
- 找到开始跨行或列的td，添加属性
- 删除多余的单元格，跨行删除下一行的td，跨列删除本行的td

4、常用属性

属性名	属性值	描述
align	left、center、right	表格在页面的对齐方式
border	像素值或""	默认为""，无边框
cellpadding	像素值	单元格边缘与内容之间的间隙，默认1像素
cellspacing	像素值	单元格与另一单元格之间的间隙，默认2像素
width	像素值或百分比	可用在table或td上，表示宽度
bgcolor	颜色词或十六进制颜色	可用在table或tr或td上，表示背景颜色
background	图片地址和名称	可用在table或tr或td上，表示背景图片

六、表单

1、认识表单

1) 表单的作用

收集用户信息

2) 表单的构成

- 表单控件：用户输入或选择内容的元素
- 提示信息：说明性文字，提示用户操作
- 表单域：相当于一个容器，容纳表单控件和提示信息，如果没有表单域，则**表单中的数据无法传递给服务器**

2、创建表单

1) 表单域：

required：规定输入框中的内容不能为空

pattern：指定正则表达式，用于检查控件输入的内容格式。适用的控件类型是text,search,url,tel,email和password

2) 单选按钮实现选择性别

```
<input type="radio" name="gender" checked /> 男  
<input type="radio" name="gender" /> 女
```

注意：必须将多个radio设置相同name值，这样单选才会生效。如果多个复选框属于一组，则也需设置相同的name属性

4、表单控件--label标签

label标签一般用于组织表单中的提示信息

```
<label for="username">用户名: </label>  
<input type="text" id="username" />
```

说明：label的for属性与对应控件的id属性相同。当点击label内的文字时，对应的表单元素会自动聚焦。

5、表单控件--下拉列表select元素

```
<select>  
  <option>选项1</option>  
  <option>选项2</option>  
  <option>选项n</option>  
</select>
```

说明：

select至少包含一对option。在option中添加selected属性时，该选项默认选中。

select的常用属性：

multiple：多行显示选项

6、表单控件--多行文本域

```
<textarea rows="行数" cols="字符数">  
  文本内容，通常不写。  
</textarea>
```

说明：rows和cols实际控制了文本域的高度和宽度

7、表单控件--datalist标签

用于给文本框指定选择项，实现即可输入也可选择。

```
<input type="text" list="mylist" />
<datalist id="mylist">
  <option>选项1</option>
  <option>选项2</option>
  <option>选项n</option>
</datalist>
```

说明：

- 先放置文本框，添加list属性。
- 再放置datalist标签，添加id属性，注意与input的list属性值相同

8、使用表单的方式

方式一：使用表单域form标签，在标签中组织其他控件元素

```
<form action="#">
  放置所有表单元素
</form>
```

方式二：（不建议使用）

- 使用form标签，指定form标签的id属性。
- 在控件元素中添加属性form="id值"。表单元素的form属性值与form标签的id属性值对应。

```
<form action="#" id="frm"> </form>
<input type="text" form="frm" />
```

七、布局标签

1、div与span标签

<div> 和 是没有语义的，它们就是一个盒子，用来装内容的

语法：div标签用来布局，一个div独占一行。span也可布局，多个span共享一行

```
<div>
  <h1>标题文字</h1>
  
</div>
<div>
  <span>文字</span>
</div>
```

2、有语义的布局标签

 image-20221002161920524

总结：

- 在 HTML 中，常用标签有：标签标签(<h1> ~ <h6>)、段落标签 (<p>)、换行标签 (
)、强调标签 (或)、图像标签 () 和链接标签 (<a>)

- HTML5 中常见的列表标签包括无序列表ul-li、有序列表ol-li和定义列表dl-dt-dd。
- 表格标签table-tr-td ，跨行属性rowspan ，跨列属性colspan；cellspacing设置单元格与单元格之间距离。
- 表单域form，常用表单元素有input ,select ,textarea。input标签使用type属性设置类型。
 - type='text' 单行输入框 type='password' 密码框
 - type='radio' 单选按钮 type='checkbox' 复选框
 - type='submit' 提交按钮 type='image' 图像提交按钮
 - type='email' 邮箱格式框 type='number' 数字输入框
- input常用属性：name名称，placeholder提示内容，checked单选或复选默认选中，pattern指定正则表达式
- div 标签作为组合其他 HTML 元素的容器，默认情况下，div 元素将占满一行，当存在多个 div 元素时，将按照从上到下的方式排列。
- span 标签可作为文本的容器。默认情况下，多个 span 标签可以存在于同一行，将按照从左到右的方式进行排列。

CSS3

一、CSS3核心基础

1、认识css3

CSS 是层叠样式表 (Cascading Style Sheets) 的简称

CSS 主要用于设置 HTML 页面中版面的布局和外观显示样式。

2、css语法规范

1) 通用语法

```
选择器{  
    属性1:属性值;  
    属性2:属性值;  
    ...  
}
```

- 选择器用于选择修饰的标签，使用{}组织修饰的样式
- 属性和属性值是“键:值”对的形式
- 多个属性之间用;分隔。
- 选择器、属性、属性值最好都小写

2) 注释

```
/* css的注释，允许换行 */
```

3、引入方式

1) 行内式

在元素的开始标签内使用style属性指定样式，样式语法也是“键:值”对格式

例

```
<p style="color:red;font-size:20px;">
    认识css3
</p>
```

2) 内嵌式

在<style></style>标签内组织所有样式代码，一般style标签写在</head>上方

例

```
p{
    color:red;
    font-size:20px;
}
```

3) 链接式

将所有样式代码做为单独文件(.css文件)，在html文件中使用link标签引入，一般也将link放在</head>上方

```
<link rel="stylesheet" href="文件路径" />
```

- rel 定义当前html文件与被链接文件之间的关系，这里必须指定为stylesheet
- href 指定文件路径

例：先在项目中css文件夹下新建CSS文件mysheet.css。然后在其他html文件中引入

mysheet.css 代码

```
p{ /*选择p标签，修改p的文字颜色及大小*/
    color: red;
    font-size: 20px;
}
```

index.html 代码

```
... 其他代码省略
<link rel="stylesheet" href="css/mysheet.css" />
</head>
  <body>
    <p>
      文字内容
    </p>
  </body>
... 其他代码省略
```

引入方式优先级：

行内式>内嵌式>引入式（就近原则）

二、CSS基本选择器

1、选择器概述

选择器作用：选择应用样式的标签

选择器分类：

- 基础选择器：标签选择器、类选择器、id选择器和通配符选择器
- 复合选择器

2、基本选择器

1) 标签选择器

标签选择器（元素选择器）是指用 HTML 标签名称作为选择器，选取所有同名标签。

语法：

```
标签名 {  
    属性1:属性值1;  
    属性2:属性值2;  
    ...  
}
```

例：页面所有a标签去掉下划线

```
a {  
    text-decoration: none;  
}
```

2) 类选择器

类选择器，可以选择不同的标签。类选择器以'.'开头，类名称可自定义。

语法：

a、定义类样式

```
.类名 {  
    属性名:属性值;  
    ...  
}
```

b、在希望应用某个类样式的标签中，指定class='类名'。

```
<p class='类名'>  
    文字内容  
</p>
```

例：定义类样式cls，修饰文字大小为20px。在p和div标签中应用该类样式

```
.cls {  
    font-size:20px;  
}
```

在p和div的开始标签中使用class='cls'应用样式

```
<p class="cls">
    段落文字
</p>
<div class="cls">
    div中文字
</div>
```

3、id选择器

id选择器为有特定id属性的html标签指定样式。id选择器以'#'开头，id名与标签的id值对应

语法：

```
#id名{
    属性名:属性值;
    ...
}
```

例：将id为nav标签的文字设置为红色

```
<style>
    #nav{
        color:red;
    }
</style>
</head>
<body>
    <div id="nav">
        文字是红色
    </div>
</body>
```

4、通配符选择器

通配符选择器使用"*"定义，它表示选取页面中所有元素（标签）。

语法：

```
*{
    属性1: 属性值1;
    ...
}
```

注意：特殊情况才使用，尽量不要用

例：清除所有元素的内外边距

```
*{
    margin:0;
    padding:0;
}
```

区别：

- 标签选择器：选取同标签名的元素
- 类选择器：可跨标签，选取同一class的元素
- id选择器：选取某一个元素

选择器优先级：

id选择器>类选择器>标签选择器

三、复合选择器

1、标签指定式选择器

又称为交集选择器，由2个基本选择器组成，第1个为标签选择器，第2个为类选择器或id选择器，2个选择器之间不能有空格

例：

```
p.cls{
  color:red
}
```

说明：该选择器选择p标签，并且class属性为cls的元素。

2、后代选择器

后代选择器用来选择某个元素的后代。由2个或多个基本选择器组成，选择器之间用空格隔开。

例：选取id属性为nav下的a标签

```
#nav a{
  text-decoration: none;
}
...部分html代码略
<div id="nav">
  <a href="#">nav的直接子级a</a>
  <ul>
    <li><a href="#">首页</a> </li>
    <li><a href="#">新闻</a> </li>
    <li><a href="#">视频</a> </li>
  </ul>
</div>
```

3、并集选择器

如果某些选择器定义的样式相同，则可以使用并集选择器，并集是各个选择器通过英文逗号连接。

例：p, div同时设置文字大小为20px

```
p,div{
  font-size:20px;
}
```

四、常用属性

1、字体相关属性

属性	表示	示例	备注
font-size	字号	font-size:20px;	注意加上单位，如px或em
font-family	字体	font-family: "微软雅黑","宋体";	指定多个字体时，用逗号分隔
font-weight	字体粗细	font-weight:bold; font-weight:700;	normal常规，bold加粗，或100~900
font-style	字体样式	font-style:italic; font-style:normal;	不倾斜是normal，较常用
font	字体简写	font:normal bold 20px/40px "楷体";	简写有顺序，其中字号和字体必写

2、文字相关属性

属性	表示	示例	备注
color	文本颜色	color:red; color:#ccc;	颜色值可以是单词/十六进制/rgb值
text-align	文本对齐	text-align:center;	文本水平对齐，其他取值：left/right
text-decoration	文本修饰	text-decoration:none;	其他取值：underline下划线/line-through删除线
text-indent	缩进	text-indent:2em;	一般设置段落的首行缩进
line-height	行高	line-height:30px;	设置行间的距离
text-shadow	文字阴影	text-shadow:3px 3px 3px #ccc;	text-shadow:水平距离 垂直距离 模糊距离 颜色; 前2个值必写

例：设置超链接的下划线

```
a{
  text-decoration:none; /*设置a默认取消下划线*/
}
/* hover是悬停状态：当鼠标悬停到a上时，添加下划线*/
a:hover{
  text-decoration:underline;
}
```

例：设置文本内容在li中垂直居中

```
li{
  height:30px; /*容器的高*/
  line-height:30px; /*内容垂直居中:将内容的line-height, 设置与容器的高相同*/
}
```

3、背景相关属性

1) 常用背景属性

属性	表示	示例	备注
background-color	背景颜色	background-color:#ccc;	颜色值：单词/十六进制/rgb值
background-image	背景图片	background-image:url(img/a.jpg);	图片路径格式：url(路径)
background-repeat	背景平铺方式	background-repeat:no-repeat;	其他取值：repeat-x/repeat-y/repeat
background-position	背景坐标,偏移量	background-position:right top;	取值：关键词 或长度值或百分值
background-size	背景大小	background-size:300px 100px;	分别表示背景的宽高。仅给一个值,会等比例缩放
background	背景简写	background:url(img/a.jpg) no-repeat right top	简写顺序：颜色 图片地址 平铺 图像滚动 图片位置 至少要给颜色或图片

2) background-position背景位置

常用属性：

- 关键词，比如 top、right、bottom、left 和 center
- 长度值，比如 px、em、rem 等
- 百分值 %

例：

```
background-position: left bottom; /*背景图片在左下方*/
background-position: left; /*仅写1个值时, 另一个默认是center*/
background-position: 30px; /*背景图片水平右移30px*/
background-position: 50%; /*背景图片水平右移:(容器宽 - 图片宽)x50% */
```

3) background-size背景大小

```
background-size:auto; /* 默认值, 不改变背景图片的高度和宽度 */
background-size:100px 50px; /* 第一个值为宽, 第二个值为高, 当设置一个值时, 将其作为图片宽度来等比缩放 */
background-size:10%; /*0%~100%之间的任何值, 将背景图片宽高按百分比显示, 当设置一个值的时候同也将其作为图片宽度来等比缩放*/
background-size:cover; /* 将背景图片等比缩放填满整个容器 */
background-size:contain; /* 将背景图片等比缩放至某一边紧贴容器边缘 */
```

css应用的特性

- 1、叠加性：当多个选择器选中的是同一类元素时，如果是不一样的样式，所有选择器都会生效
- 2、层叠性：如果有多个同一样式应用在同一元素上，优先级高的会生效。如果优先级相同，后写的会生效
- 3、继承性：子元素可以继承父元素的样式(text-, font-, line-这些元素开头的可以继承，以及color属性)

五、盒子模型

1、盒子模型

- 1) 概念：将所有html元素看做是一个矩形盒子，作为容器盛装内容。
- 2) 盒子模型包括：边框border、外边距margin、内填充padding、和实际内容(weight,height)

盒子的总宽度= width+左右内边距之和+左右边框宽度之和+左右外边距之和

盒子的总高度= height+上下内边距之和+上下边框宽度之和+上下外边距之和

注意：宽度属性**width**和高度属性**height**仅适用于**块级元素**，对行内元素无效（标记和<input />除外）

2、边框border属性

- 1) border-style边框样式

常用取值：

none 无边框

solid 单实线

dashed 虚线

dotted 点线

double 双实线

```
div{
  border-width:1px; /* 四边都是1像素 */
  border-style:solid; /* 四边都是实线 */
}
```

说明：对于盒子模型的border，margin，padding属性，允许给1个值，或多个值。

示例	说明
<code>border-width:1px;</code>	四边都是1像素
<code>border-width:1px 2px;</code>	上下边1像素，左右边2像素
<code>border-width:1px 2px 3px;</code>	上边1像素，左右边2像素，下边3像素
<code>border-width:1px 2px 3px 4px;</code>	上1，右2，下3，左4

2) border是简写形式，可同时设置盒子四边的边框粗细，边框样式、边框颜色

```
div{
  border:1px solid red; /*粗细 样式 颜色*/
}
```

或仅设置某个方向的边框

```
div{
  border-bottom:1px dashed gray; /* 设置底部边框1像素 虚线 灰色 */
}
```

3) border-collapse 合并表格的边框

```
table{
  border-collapse:collapse;
}
```

4) border-radius圆角边框

属性值可以是数值或百分比

```
div{
  border-radius:25px; /*四个角的弧度为25px*/
}
```

3、外间距margin和内填充padding

1) 概念

外边距margin：盒子与盒子之间的间隔距离

内填充padding：又叫内边距，盒子的边框与内容之间的间隔距离

2) margin属性

margin是简写形式，可使用margin-left、margin-right、margin-top、margin-bottom分别设置四个方向的外间距

```
div{
  margin:20px; /*div的四个方向与其他盒子都有20像素的距离*/
}
```

margin:20px 30px; 表示外边距上下20，左右30

margin:20px 30px 40px ; 上20 左右30 下30

margin:20px 30px 40px 50px; 上20 右30 下40 左50

3) 外边距合并

a、相邻块元素垂直外边距的合并

相邻的2个块元素，如果同时设置上下外间距，会取大值。

解决：只给一个元素加垂直方向的外间距

b、嵌套块元素垂直外边距的塌陷

嵌套的2个块元素，如果父元素和子元素同时有上外间距，外间距会取较大值

解决：给父元素加边框或父元素添加属性 `overflow:hidden;`

注意：行内元素设置margin-top和margin-bottom无效

4) 清除内外边距

```
* {
  padding:0; /* 清除内边距 */
  margin:0; /* 清除外边距 */
}
```

5) 填充padding

可分别使用padding-top, padding-left, padding-right, padding-bottom分别设置4个方向的内填充。可使用使用简写形式的padding设置。

```
div{
  padding:10px; /*div的边框与内容之间，4个方向都有10像素的距离*/
}
```

注意：

- 行内元素padding-top和padding-bottom无效
- 当给盒子设置了padding值后，会影响盒子实际大小，如果希望盒子保持原大小，则需将width或height的值减去多出来的padding值

例：原div的width为100px，设置了padding:10px后，width的值可设置为100-10-10=80

4、其他属性

1) box-shadow盒子阴影

语法：

```
#box{
  /* 横向偏移量 纵向偏移量 阴影模糊半径 阴影扩散半径 阴影颜色 阴影类型*/
  box-shadow: 2px 2px 2px 1px rgba(0, 0, 0, 0.2) inset;
}
```

2) box-sizing属性

box-sizing定义如何计算一个元素的总宽度和总高度，是否包含border和padding

```
#box{
  /*该属性默认值是content-box,width和height中不包括border和padding*/
  box-sizing:border-box; /*当定义width和height时, border和padding的值包含在width和
height中*/
}
```

5、元素显示模式

1) 块级元素

块级元素：div、h1~h6、p、ul、li、ol、dl、dt、dd、table等

特点：

- 独占一行
- 允许设置width、height、margin和padding
- 宽度默认是容器(父级)的100%

2) 行内元素

行内元素：a、span、select、textarea等

特点：

- 与其他行内元素共享一行
- 设置width、height无效，默认宽度是内容的宽度
- margin-top/margin-bottom、padding-top/padding-bottom无效果

3) 行内块元素

行内块元素：img、input、td

特点：

与其他行内元素共享一行

允许设置width、height、margin、padding属性

4) 显示模式转换

- 转换为块元素：display:block;
- 转换为行内元素：display:inline;
- 转换为行内块：display:inline-block;

六、浮动及常见布局

1、浮动属性

1) 标准文档流

标准文档流指的是元素排版布局过程中,元素会默认自动从左往右,从上往下的流式排列方式

2) 浮动

a、作用：浮动可以改变元素默认的排列方式，可以让多个块级元素一行内排列显示

b、float属性：让元素脱离标准文档流，向容器的左侧或右侧移动，直到遇到边缘或另一个浮动元素。

```
选择器{
  float:none/left/right;    /*不浮动/左浮动/右浮动*/
}
```

例：

```
div{
  float:left;    /*让div左浮动，形成一行排列*/
}
```

注意：

- 加了浮动的元素，会脱离标准文档流(释放在页面的占位)
- 浮动的元素会具有行内块元素的特性，如果浮动元素没有设置宽度，则大小由内容决定

例：

```
<style type="text/css">
  *{
    margin: 0;
    padding: 0;
    list-style: none; /*清除li默认样式*/
  }
  #nav{
    width: 50%;
    margin: 20px auto;
  }
  #nav ul li{
    float: left;    /*让多个li浮动，一行排列*/
    height: 30px;
    line-height: 30px;
    margin-right: 20px;
    /* 背景:线性渐变(to 水平 垂直,颜色1,颜色2,颜色3); */
    background: linear-gradient(to left top,#FFDFB0,red,#FFDFB0);
    /*背景:径向渐变(形状,颜色1,颜色2,颜色3,..); 形状默认椭圆型 */
    /*background: radial-gradient(circle,red 15%, yellow 45%, green 80%);*/
    border-radius: 5px;
  }
  #nav ul li a{
    color:#ddd;
    font-weight: 900;
    padding-left: 20px;
    padding-right: 20px;
    text-decoration: none;
  }
</style>

<div id="nav">
  <ul>
    <li><a href="#">百度</a></li>
    <li><a href="#">淘宝</a></li>
    <li><a href="#">京东</a></li>
    <li><a href="#">CSDN</a></li>
    <li><a href="#">小米</a></li>
  </ul>
</div>
```

```
</ul>
</div>
```

2、清除浮动

使用场合：上面有浮动元素，会影响下面的排版时

清除浮动方法

- 额外标签法也称为隔墙法
- 父级添加 overflow 属性
- 父级添加after伪元素

1) 额外标签法

- 在浮动元素后面添加一个空标签
- 空标签添加属性clear属性

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

clear属性

```
选择器{
  clear:both/left/right; /*清除两侧浮动/清除左侧浮动/清除右侧浮动*/
}
```

说明：也可以为浮动元素后面的元素添加clear属性

```
/*div为受到浮动影响的元素*/
div{
  clear:both;
}
```

2) 父级添加overflow

给浮动元素的父容器添加overflow属性

overflow属性:规定了当内容溢出容器时如何处理

```
选择器{
  overflow:hidden/auto/scroll; /*一般使用hidden*/
}
```

说明：

hidden 内容会被修剪，并且溢出内容不可见

auto 内容会被修剪，浏览器会显示滚动条以便查看溢出的内容

scroll 内容会被修剪，浏览器会显示滚动条以便查看溢出的内容

3) :after伪元素法

:after表示某个元素之后插入内容

在这里，使用:after在浮动元素的父元素后添加一个空元素

```
父元素:after{
  content: ""; /*元素内容为空*/
  display: block; /*元素是块级*/
  height: 0;
  clear: both; /*清除元素两侧的浮动*/
  visibility: hidden; /*元素在页面中不显示*/
}
```

3、常见网页布局

浮动布局注意点

- 浮动和标准流的父盒子搭配，先用标准流的父元素排列上下位置，之后内部子元素采取浮动排列左右位置
- 一个盒子里面有多个子盒子，如果其中一个盒子浮动，那么其他兄弟也应该浮动，以防止引起问题

七、定位

1、定位属性position

定位概念：将盒子定在某个位置。定位=定位模式+边偏移

1) 定位模式

定位模式由**position**属性的值来决定

取值	语义
position:static	静态定位模式
position:relative	相对定位
position:absolute	绝对定位
position:fixed	固定定位

2) 边偏移

边偏移就是定位的盒子移动到最终位置。有top/bottom/left/right四个属性

边偏移	示例	描述
top	top:80px;	元素顶部的偏移量，离父元素上边线的距离
bottom	bottom:80px;	元素底部的偏移量，离父元素下边线的距离
left	left:80px;	元素左侧的偏移量，离父元素左边线的距离
right	right:80px;	元素右侧的偏移量，离父元素右边线的距离

2、静态定位

静态定位是元素的默认定位方式。设置边偏移没有效果

3、相对定位

相对定位：元素移动位置时，相对于自己原来位置移动。在页面占位，未脱离标准文档流

例：让div在原位置向下移动20px

```
#box{
  position:relative;
  top:20px;
}
```

相对定位典型的应用场景是，给绝对定位的元素当父容器。

4、绝对定位

1) 绝对定位：元素移动位置时，相对于父元素移动。

注意：

- 如果元素没有父元素或父元素没有定位，则以**文档为参照物**移动
- 如果父元素**有定位(非static)**，则以**父元素为参照物**移动
- 绝对定位**释放占位**（脱离标准文档流）

例：让子div停靠在父div的底部

```
<div id="father">
  <div id="son"></div>
</div>
<style type="text/css">
  #father{
    width: 500px;
    height: 400px;
    border: 1px solid red;
    position: relative; /* 父元素定位,让子div以父元素为参照物 */
  }
  #son{
    width: 100px;
    height: 100px;
    background: yellow;
    position: absolute; /* 绝对定位 */
    bottom: 0px;
  }
</style>
```

2) 定位叠放次序z-index

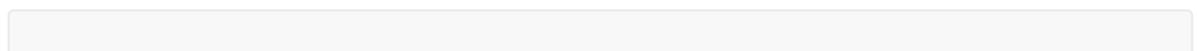
z-index控制盒子上下叠放顺序，值越大，越靠上。

注意：

只有定位的盒子才有z-index属性

z-index的值可以是正/负数或0，不能加单位

案例：祝福墙



```
<div id="father">
  <div id="div1">
    祝大家越来越顺
  </div>
  <div id="div2">祝大家越来越美</div>
  <div id="div3">
    祝大家越来越好
  </div>
  <div id="div4">
    祝大家越来越幸福
  </div>
</div>
<style type="text/css">
  #father{
    border: 1px solid black;
    width: 400px;
    height: 400px;
    margin-left: 100px;
    margin-top: 100px;
    position: relative;
  }
  #div1,#div2,#div3,#div4{
    width:200px ;
    height:200px;
    position: absolute;
    text-align: center;
    line-height: 200px;
    font-size: 20px;
  }
  /*鼠标悬停到某个div上时，让该div显示在最上方*/
  #div1:hover,#div2:hover,#div3:hover,#div4:hover{
    z-index: 100; /*z-index 值越大，则显示在上 */
  }
  #div1{
    background: pink;
    color: darkblue;
    top:20px;
    left: 20px;
    z-index: 2;
  }
  #div2{
    background: yellow;
    top:40px;
    left: 40px;
    color: blueviolet;
    z-index: 3;
  }
  #div3{
    background: purple;
    color: white;
    top:60px;
    left: 60px;
    z-index: 4;
  }
  #div4{
    background: red;
    color: gainsboro;
    top:80px;
```

```
    left: 80px;
    z-index: 5;
  }
</style>
```

5、固定定位

1) 固定定位：元素固定于浏览器可视区的位置，以浏览器为参照物移动

例：广告漂浮在浏览器的右侧

```
<div id="box">
  
  <div id="side">
    
  </div>
</div>
<style type="text/css">
#side{
  /* position: absolute; 以文档为参照物 */
  position: fixed; /* 以浏览器为参照物 */
  right: 0px;
  bottom: 10px;
}
</style>
```

注意：

固定定位的元素跟父元素没有任何关系

不随滚动条滚动

固定定位的元素**释放占位**（脱离标准文档流）。

6、定位模式总结

定位模式	是否脱离标准流	移动位置	是否常用
static 静态定位	否	不能使用边偏移	很少
relative 相对定位	否(占有位置)	相对于自身位置移动	常用
absolute绝对定位	是(不占有位置)	相对于带有定位的父级移动	常用
fixed 固定定位	是(不占有位置)	浏览器可视区	常用

八、flex布局

1、flex介绍

1) 优点

- 操作方便，简单
- 在移动端应用广泛

2) flex布局原理

通过给父盒子添加flex属性，来控制子盒子的位置和排列方式

注意：父盒子设置为flex布局后，子元素的 float、clear 和 vertical-align 属性将失效

2、flex布局父项常用属性

属性	说明
flex-direction	设置主轴的方向，默认是水平(row)
justify-content	设置主轴上的子元素排列方式
flex-wrap	设置子元素是否换行
align-items	设置侧轴上的子元素排列方式（单行）
align-content	设置侧轴上的子元素的排列方式（多行）
flex-flow	复合属性，相当于同时设置了 flex-direction 和 flex-wrap

1) flex-direction

设置主轴的方向，默认是x轴(水平)。另一个方向即为侧轴。子元素根据主轴来排列。

主轴和侧轴不是固定的，是可以通过flex-direction属性来改变的。

属性取值：

取值	说明
flex-direction: row;	默认值，子元素从左到右排列
flex-direction: row-reverse;	从右到左
flex-direction: column;	将主轴改变为y轴，子元素从上到下排列
flex-direction: column-reverse;	从下到上排列

2) justify-content

设置**主轴上**元素的排列方式，在设置这个属性时，一定要确定主轴是哪个方向

取值	说明
justify-content: flex-start;	从主轴的开头开始排列
justify-content: flex-end;	从主轴的尾部开始排列
justify-content: center;	在主轴居中对齐
justify-content: space-around;	子元素平分剩余空间
justify-content: space-between;	左右子元素先贴两边，其他再平分剩余空间

3) flex-wrap

设置子元素是否换行。flex布局默认子元素在父容器中不换行。

取值	说明
flex-wrap: nowrap;	默认值，不换行
flex-wrap: wrap;	换行

4) align-items

设置子元素在侧轴上的排列方式（单行）

取值	说明
align-items: flex-start;	默认值，从上到下
align-items: flex-end;	从下到上
align-items: center;	在侧轴居中对齐
align-items: stretch;	拉伸（子元素有height时无效）

5) align-content

设置子元素在侧轴上的排列方式（多行）

取值	说明
align-content: flex-start;	默认值，在侧轴的头部开始排列
align-content: flex-end;	在侧轴的尾部开始排列
align-content: center;	在侧轴中间显示
align-content: space-around;	子项在侧轴平分剩余空间
align-content: space-between;	子项在侧轴先分布在两头，再平分剩余空间
stretch	设置子项元素高度平分父元素高度

6) flex-flow

flex-flow是复合属性，可同时设置flex-direction和flex-wrap

3、flex布局子项常用属性

1) flex

设置子项分配**剩余空间**，用flex来表示占多少**份数**

例：一行三个子元素，左右两个子元素分布在两头，中间子元素占剩余空间的1份

```
<div id="box">
  <span>1</span>
  <span>2</span>
  <span>3</span>
</div>
<style>
```

```
#box{
  width: 70%;
  height: 100px;
  background: blue;
  display: flex; /*设置父盒子flex布局*/
  margin: 0 auto;
}
/*过滤选择span中第1和第3个元素*/
#box span:nth-child(1),#box span:nth-child(3){
  width:100px;
  height: 100%;
  background: red;
}
#box span:nth-child(2){
  flex: 1; /*设置第2个span占剩余的份数*/
  background: yellow;
  margin-left: 5px;
  margin-right: 5px;
}
</style>
```

2) align-self

设置子项**自己**在侧轴上的排列方式

属性的取值与align-items相同

```
.item{
  align-self:flex-end;
}
```

3) order

设置子项的排列顺序。数值越小，排列越靠前，默认为0

```
.item{
  order:-1;
}
```