课程目标:

> 1. 学会使用CSS选择器
> 2. 熟记CSS样式和外观属性
> 3. 熟练掌握CSS各种选择器
> 4. 熟练掌握CSS各种选择器
> 5. 熟练掌握CSS三种显示模式
> 6. 熟练掌握CSS背景属性
> 7. 熟练掌握CSS三大特性
> 8. 熟练掌握CSS盒子模型
> 9. 熟练掌握CSS浮动 10.熟练掌握CSS定位 11.熟练掌握CSS高级技巧强化CSS

CSS的发展历程

从HTML被发明开始，样式就以各种形式存在。不同的浏览器结合它们各自的样式语言为用户提供页面效果的控制。最初的HTML只包含很少的显示属性。 随着HTML的成长，为了满足页面设计者的要求，HTML添加了很多显示功能。但是随着这些功能的增加，HTML变的越来越杂乱，而且HTML页面也越来越臃肿。于是CSS便诞生了。

CSS 网页的美容师

CSS的出现，拯救了混乱的HTML，当让更加拯救了我们web开发者。 让我们的网页更加丰富多彩。

CSS的最大贡献就是： 让 HTML 从样式中解脱苦海， 实现了 HTML 专注去做 结构呈现。

而且。。。。。 CSS 做的很出色，如果JavaScript是网页的魔法师，那么CSS它是我们网页的美容师，不信，你看:

没有不好看的网页，只有不会CSS的前端。

CSS初识

CSS(Cascading Style Sheets) 美化样式

CSS通常称为CSS样式表或层叠样式表（级联样式表），主要用于设置HTML页面中的文本内容（字体、大小、对齐方式等）、图片的外形（宽高、边框样式、边距等）以及版面的布局等外观显示样式。

CSS以HTML为基础，提供了丰富的功能，如字体、颜色、背景的控制及整体排版等，而且还可以针对不同的浏览器设置不同的样式。

引入CSS样式表（书写位置）

CSS可以写到那个位置？ 是不是一定写到html文件里面呢？

内部样式表

内嵌式是将CSS代码集中写在HTML文档的head头部标签中，并且用style标签定义，其基本语法格式如下：

```
<head>
<style type="text/CSS">
    选择器 {属性1:属性值1; 属性2:属性值2; 属性3:属性值3;}
</style>
</head>
```

语法中，style标签一般位于head标签中title标签之后，也可以把他放在HTML文档的任何地方。

type="text/CSS" 在html5中可以省略， 写上也比较符合规范， 所以这个地方可以写也可以省略。

行内式（内联样式）

内联样式，又有人称行内样式、行间样式、内嵌样式。是通过标签的style属性来设置元素的样式，其基本语法格式如下：

```
<标签名 style="属性1:属性值1; 属性2:属性值2; 属性3:属性值3;"> 内容 </标签名>
```

语法中style是标签的属性，实际上任何HTML标签都拥有style属性，用来设置行内式。其中属性和值的书写规范与CSS样式规则相同，行内式只对其所在的标签及嵌套在其中的子标签起作用。

外部样式表（外链式）

链入式是将所有的样式放在一个或多个以.CSS为扩展名的外部样式表文件中，通过link标签将外部样式表文件链接到HTML文档中，其基本语法格式如下：

```
<head>
  <link href="CSS文件的路径"  rel="stylesheet" />
</head>
```

注意： link 是个单标签哦!!!

该语法中，link标签需要放在head头部标签中，并且必须指定link标签的三个属性，具体如下：

```
href：定义所链接外部样式表文件的URL，可以是相对路径，也可以是绝对路径。
type：定义所链接文档的类型，在这里需要指定为“text/CSS”，表示链接的外部文件为CSS样式表。
rel：定义当前文档与被链接文档之间的关系，在这里需要指定为“stylesheet”，表示被链接的文档是一个样式表文件。
```

三种样式表总结（位置）

样式表	优点	缺点	使用情况	控制范围
行内样式表	书写方便，权重高	没有实现样式和结构相分离	较少	控制一个标签（少）
内部样式表	部分结构和样式相分离	没有彻底分离	较多	控制一个页面（中）
外部样式表	完全实现结构和样式相分离	需要引入	最多，强烈推荐	控制整个站点（多）

CSS样式规则

使用HTML时，需要遵从一定的规范。CSS亦如此，要想熟练地使用CSS对网页进行修饰，首先需要了解CSS样式规则，具体格式如下：

```
在上面的样式规则中：

1.选择器用于指定CSS样式作用的HTML对象，花括号内是对该对象设置的具体样式。
2.属性和属性值以“键值对”的形式出现。
3.属性是对指定的对象设置的样式属性，例如字体大小、文本颜色等。
4.属性和属性值之间用英文“:”连接。
5.多个“键值对”之间用英文“;”进行区分。
可以用段落 和 表格的对齐的演示。
```

选择器（重点）

要想将CSS样式应用于特定的HTML元素，首先需要找到该目标元素。在CSS中，执行这一任务的样式规则部分被称为选择器（选择符）。

如上图所以，要把里面的小黄人分为2组，最快的方法怎办?

很多， 比如 一只眼睛的一组，剩下的一组

选择器干啥的？ 选择标签用的

这就用到基础选择器组：

CSS基础选择器

标签选择器（元素选择器）

标签选择器是指用HTML标签名称作为选择器，按标签名称分类，为页面中某一类标签指定统一的CSS样式。其基本语法格式如下：

```
标签名{属性1:属性值1; 属性2:属性值2; 属性3:属性值3; }  或者
元素名{属性1:属性值1; 属性2:属性值2; 属性3:属性值3; }
```

标签选择器最大的优点是能快速为页面中同类型的标签统一样式，同时这也是他的缺点，不能设计差异化样式。

标签选择器 可以把某一类标签全部选择出来 div span

课堂案例：

传智简介

类选择器

类选择器使用“.”（英文点号）进行标识，后面紧跟类名，其基本语法格式如下：

```
.类名{属性1:属性值1; 属性2:属性值2; 属性3:属性值3; }
```

```
 标签调用的时候用 class="类名"  即可。
```

类选择器最大的优势是可以为元素对象定义单独或相同的样式。 可以选择一个或者多个标签

小技巧:

```
1.长名称或词组可以使用中横线来为选择器命名。
2.不建议使用"_"下划线来命名CSS选择器。
```

输入的时候少按一个shift键; 浏览器兼容问题 (比如使用*tips的选择器命名，在IE6是无效的)* *能良好区分JavaScript变量命名(JS变量命名是用"")*

```
3.不要纯数字、中文等命名，  尽量使用英文字母来表示。
```

猜谜底游戏:

你猜?

命名规范: 见附件（Web前端开发规范手册.doc）

命名是我们通俗约定的，但是没有规定必须用这些常用的命名。

课堂案例：

```
<head>
        <meta charset="utf-8">
        <style>
        span {
          font-size: 100px;
        }
        .blue {
          color: blue;
        }
        .red {
          color: red;
        }
        .orange {
      color: orange;
        }
    .green {
      color: green;
    }
        </style>
    </head>
    <body>
      <span class="blue">G</span>
      <span class="red">o</span>
      <span class="orange">o</span>
      <span class="blue">g</span>
```

```
    <span class="green">l</span>
    <span class="red">e</span>
  </body>
```

多类名选择器

我们可以给标签指定多个类名，从而达到更多的选择目的。

注意：

```
1. 样式显示效果跟HTML元素中的类名先后顺序没有关系,受CSS样式书写的上下顺序有关。
2. 各个类名中间用空格隔开。
```

多类名选择器在后期布局比较复杂的情况下，还是较多使用的。

```
<div class="pink fontWeight font20">亚瑟</div>
<div class="font20">刘备</div>
<div class="font14 pink">安其拉</div>
<div class="font14">貂蝉</div>
```

类名选择器： < div class=“nav”> 这个 div 的名字 就是 nav nav 就是 div 这个 div 也是 nav

< 人 class = 刘德华 > 我们想要吧div 找到 div {} .nav {}

id选择器

id选择器使用“#”进行标识，后面紧跟id名，其基本语法格式如下：

```
#id名{属性1:属性值1; 属性2:属性值2; 属性3:属性值3; }
```

该语法中，id名即为HTML元素的id属性值，大多数HTML元素都可以定义id属性，元素的id值是唯一的，只能对应于文档中某一个具体的元素。

用法基本和类选择器相同。

id选择器和类选择器区别

W3C标准规定，在同一个页面内，不允许有相同名字的id对象出现，但是允许相同名字的class。

类选择器（class） 好比人的名字， 是可以多次重复使用的， 比如 张伟 王伟 李伟 李娜

id选择器 好比人的身份证号码， 全中国是唯一的， 不得重复。 只能使用一次。

id选择器和类选择器最大的不同在于 使用次数上。

通配符选择器

通配符 选择器用“*”号表示，他是所有选择器中作用范围最广的，能匹配页面中所有的元素。其基本语法格式如下：

```
* { 属性1:属性值1; 属性2:属性值2; 属性3:属性值3; }
```

例如下面的代码，使用通配符选择器定义CSS样式，清除所有HTML标记的默认边距。

```
* {
  margin: 0;                    /* 定义外边距*/
  padding: 0;                   /* 定义内边距*/
}
```

注意：

这个通配符选择器，就像我们的电影明星中的梦中情人， 想想它就好了，但是它不会和你过日子。

CSS字体样式属性

font-size:字号大小

font-size属性用于设置字号，该属性的值可以使用相对长度单位，也可以使用绝对长度单位。其中，相对长度单位比较常用，推荐使用像素单位px，绝对长度单位使用较少。具体如下：

相对长度单位	说明
em	相对于当前对象内文本的字体尺寸
px	像素，最常用，推荐使用
绝对长度单位	说明
in	英寸
cm	厘米
mm	毫米
pt	点

font-family:字体

font-family属性用于设置字体。网页中常用的字体有宋体、微软雅黑、黑体等，例如将网页中所有段落文本的字体设置为微软雅黑，可以使用如下CSS样式代码：

p{ font-family:"微软雅黑";}

可以同时指定多个字体，中间以逗号隔开，表示如果浏览器不支持第一个字体，则会尝试下一个，直到找到合适的字体。

> 常用技巧：

```
1. 现在网页中普遍使用14px+。
2. 尽量使用偶数的数字字号。ie6等老式浏览器支持奇数会有bug。
3. 各种字体之间必须使用英文状态下的逗号隔开。
4. 中文字体需要加英文状态下的引号，英文字体一般不需要加引号。当需要设置英文字体时，英文字体名必须位于中文字体名之前。
5. 如果字体名中包含空格、#、$等符号，则该字体必须加英文状态下的单引号或双引号，例如font-family: "Times New Roman";。
6. 尽量使用系统默认字体，保证在任何用户的浏览器中都能正确显示。
```

CSS Unicode字体

在 CSS 中设置字体名称，直接写中文是可以的。但是在文件编码（GB2312、UTF-8 等）不匹配时会产生乱码的错误。xp 系统不支持 类似微软雅黑的中文。

方案一： 你可以使用英文来替代。 比如 font-family:"Microsoft Yahei"。

方案二： 在 CSS 直接使用 Unicode 编码来写字体名称可以避免这些错误。使用 Unicode 写中文字体名称，浏览器是可以正确的解析的。 font-family: "\5FAE\8F6F\96C5\9ED1"，表示设置字体为“微软雅黑”。

可以通过escape() 来测试属于什么字体。

字体名称	英文名称	Unicode 编码
宋体	SimSun	\5B8B\4F53
新宋体	NSimSun	\65B0\5B8B\4F53
黑体	SimHei	\9ED1\4F53
微软雅黑	Microsoft YaHei	\5FAE\8F6F\96C5\9ED1
楷体_GB2312	KaiTi_GB2312	\6977\4F53_GB2312
隶书	LiSu	\96B6\4E66
幼园	YouYuan	\5E7C\5706
华文细黑	STXihei	\534E\6587\7EC6\9ED1
细明体	MingLiU	\7EC6\660E\4F53
新细明体	PMingLiU	\65B0\7EC6\660E\4F53

为了照顾不同电脑的字体安装问题，我们尽量只使用宋体和微软雅黑中文字体

font-weight:字体粗细

字体加粗除了用 b 和 strong 标签之外，可以使用CSS 来实现，但是CSS 是没有语义的。

```
font-weight属性用于定义字体的粗细，其可用属性值：normal、bold、bolder、lighter、
100~900（100的整数倍）。
```

小技巧：

```
数字 400 等价于 normal，而 700 等价于 bold。   但是我们更喜欢用数字来表示。
```

font-style:字体风格

字体倾斜除了用 i 和 em 标签之外，可以使用CSS 来实现，但是CSS 是没有语义的。

font-style属性用于定义字体风格，如设置斜体、倾斜或正常字体，其可用属性值如下：

normal：默认值，浏览器会显示标准的字体样式。

italic：浏览器会显示斜体的字体样式。

oblique：浏览器会显示倾斜的字体样式。

小技巧：

```
平时我们很少给文字加斜体，反而喜欢给斜体标签（em，i）改为普通模式。
```

font:综合设置字体样式 (重点)

font属性用于对字体样式进行综合设置，其基本语法格式如下：

```
选择器{font: font-style  font-weight  font-size/line-height  font-family;}
```

```
使用font属性时，必须按上面语法格式中的顺序书写，不能更换顺序，各个属性以空格隔开。

注意：其中不需要设置的属性可以省略（取默认值），但必须保留font-size和font-family属性，否则font属性将不起作用。
```

CSS外观属性

color:文本颜色

color属性用于定义文本的颜色，其取值方式有如下3种：

1.预定义的颜色值，如red，green，blue等。

2.十六进制，如#FF0000，#FF6600，#29D794等。实际工作中，十六进制是最常用的定义颜色的方式。

3.RGB代码，如红色可以表示为rgb(255,0,0)或rgb(100%,0%,0%)。

需要注意的是，如果使用RGB代码的百分比颜色值，取值为0时也不能省略百分号，必须写为0%。

line-height:行间距

ine-height属性用于设置行间距，就是行与行之间的距离，即字符的垂直间距，一般称为行高。line-height常用的属性值单位有三种，分别为像素px，相对值em和百分比%，实际工作中使用最多的是像素px

一般情况下，行距比字号大7.8像素左右就可以了。

text-align:水平对齐方式

```
text-align属性用于设置文本内容的水平对齐，相当于html中的align对齐属性。其可用属性值如下：
```

left：左对齐（默认值）

right：右对齐

center：居中对齐

是让盒子里面的内容水平居中， 而不是让盒子居中对齐

text-indent:首行缩进

text-indent属性用于设置首行文本的缩进，其属性值可为不同单位的数值、em字符宽度的倍数、或相对于浏览器窗口宽度的百分比%，允许使用负值, 建议使用em作为设置单位。

1em 就是一个字的宽度 如果是汉字的段落， 1em 就是一个汉字的宽度

text-decoration 文本的装饰

text-decoration 通常我们用于给链接修改装饰效果

值	描述
none	默认。定义标准的文本。
underline	定义文本下的一条线。下划线 也是我们链接自带的
overline	定义文本上的一条线。
line-through	定义穿过文本下的一条线。

开发者工具（chrome）

此工具是我们的必备工具，以后代码出了问题，我们首先第一反应就是：

“按F12”或者是 “shift+ctrl+i” 打开 开发者工具。

菜单： 右击网页空白出---查看

小技巧：

1. ctrl+滚轮 可以 放大开发者工具代码大小。
2. 左边是HTML元素结构 右边是CSS样式。
3. 右边CSS样式可以改动数值和颜色查看更改后效果。

CSS复合选择器

复合选择器是由两个或多个基础选择器，通过不同的方式组合而成的,目的是为了可以选择更准确更精细的目标元素标签。

交集选择器、、、、

交集选择器由两个选择器构成，其中第一个为标签选择器，第二个为class选择器，两个选择器之间不能有空格，如h3.special。

记忆技巧：

交集选择器 是 并且的意思。 即...又...的意思

```
比如：    p.one    选择的是：  类名为 .one  的 段落标签。
```

用的相对来说比较少，不太建议使用。

并集选择器

并集选择器（CSS选择器分组）是各个选择器通过**逗号**连接而成的，任何形式的选择器（包括标签选择器、class类选择器id选择器等），都可以作为并集选择器的一部分。如果某些选择器定义的样式完全相同，或部分相同，就可以利用并集选择器为它们定义相同的CSS样式。

记忆技巧：

并集选择器 和 的意思， 就是说，只要逗号隔开的，所有选择器都会执行后面样式。

```
比如  .one, p , #test {color: #F00;}  表示   .one 和 p  和 #test 这三个选择器都会执行颜色为红色。   通常用于集体声明。
```

后代选择器

后代选择器又称为包含选择器，用来选择元素或元素组的后代，其写法就是把外层标签写在前面，内层标签写在后面，中间用空格分隔。当标签发生嵌套时，内层标签就成为外层标签的后代。

子孙后代都可以这么选择。 或者说，它能选择任何包含在内 的标签。

子元素选择器

子元素选择器只能选择作为某元素子元素的元素。其写法就是把父级标签写在前面，子级标签写在后面，中间跟一个 > 进行连接，注意，符号左右两侧各保留一个空格。

白话： 这里的子 指的是 亲儿子 不包含孙子 重孙子之类。

```
 比如：   .demo > h3 {color: red;}   说明  h3 一定是demo 亲儿子。   demo 元素包含着 h3。
```

测试题

```
<div class="nav">    <!-- 主导航栏 -->
  <ul>
    <li><a href="#">公司首页</a></li>
  <li><a href="#">公司简介</a></li>
  <li><a href="#">公司产品</a></li>
  <li>
        <a href="#">联系我们</a>
     <ul>
            <li><a href="#">公司邮箱</a></li>
            <li><a href="#">公司电话</a></li>
     </ul>
  </li>
  </ul>
</div>
<div class="sitenav">    <!-- 侧导航栏 -->
  <div class="site-l">左侧侧导航栏</div>
  <div class="site-r"><a href="#">登录</a></div>
</div>
```

在不修改以上代码的前提下，完成以下任务：

1. 链接 登录 的颜色为红色,同时主导航栏里面的所有的链接改为橙色 (简单)

2. 主导航栏和侧导航栏里面文字都是14像素并且是微软雅黑。（中等)
3. 主导航栏里面的一级菜单链接文字颜色为绿色。（难)

伪类选择器

伪类选择器用于向某些选择器添加特殊的效果。比如给链接添加特殊效果， 比如可以选择 第1个，第n个元素。

伪娘

类 .one

伪类 :link

```
为了和我们刚才学的类选择器相区别，   类选择器是一个点 比如 .demo {}   而我们的伪类 用 2个点 就是 冒号  比如  :link{}
```

链接伪类选择器

- :link /* 未访问的链接 */
- :visited /* 已访问的链接 */
- :hover /* 鼠标移动到链接上 */
- :active /* 选定的链接 */

注意写的时候，他们的顺序尽量不要颠倒 按照 lvha 的顺序。 love hate 爱上了讨厌 记忆法 或者 lv 包包 非常 hao

```
a {   /* a是标签选择器  所有的链接 */
      font-weight: 700;
      font-size: 16px;
      color: gray;
    }
a:hover {   /* :hover 是链接伪类选择器 鼠标经过 */
      color: red; /*  鼠标经过的时候，由原来的 灰色 变成了红色 */
}
```

CSS注释

```
CSS规则是使用      /*  需要注释的内容  */  进行注释的，即在需要注释的内容前使用 “/*” 标记开始注释，在内容的结尾使用 “*/”结束。
```

例如：

```
p {
  font-size: 14px;                    /* 所有的字体是14像素大小*/
}
```

sublime快捷方式

sublime可以快速提高我们代码的书写方式

1. 生成标签 直接输入标签名 按tab键即可 比如 div 然后tab 键， 就可以生成
2. 如果想要生成多个相同标签 加上 * 就可以了 比如 div*3 就可以快速生成3个div
3. 如果有父子级关系的标签，可以用 > 比如 ul > li就可以了
4. 如果有兄弟关系的标签，用 + 就可以了 比如 div+p
5. 如果生成带有类名或者id名字的， 直接写 .demo 或者 #two tab 键就可以了

标签显示模式（display）

非洲黑人： 皮肤内黑色素含量高，以吸收阳光中的紫外线，保护皮肤内部结构免遭损害，头发象羊毛一样卷曲，使每根卷发周围都有许多空隙，空隙充满空气，卷发有隔热作用。

欧洲白人： 生活寒带或着是说常年温度较低的地缘,加上年日照时间少，身体的黑色素沉淀比较少``所以出现皮肤、发色、瞳晕都呈现浅色

最重要的总结： 是为了更好的适应环境而完成的自然选择。

同理，我们网页的标签非常多，再不同地方会用到不同类型的标签，以便更好的完成我们的网页。

标签的类型(显示模式)

HTML标签一般分为块标签和行内标签两种类型，它们也称块元素和行内元素。具体如下：

块级元素(block-level)

每个块元素通常都会独自占据一整行或多整行，可以对其设置宽度、高度、对齐等属性，常用于网页布局和网页结构的搭建。 霸道

```
常见的块元素有<h1>~<h6>、<p>、<div>、<ul>、<ol>、<li>等，其中<div>标签是最典型的块元素。
```

块级元素的特点：

（1）总是从新行开始

（2）高度，行高、外边距以及内边距都可以控制。

（3）宽度默认是容器的100%

（4）可以容纳内联元素和其他块元素。

行内元素(inline-level)

行内元素（内联元素）不占有独立的区域，仅仅靠自身的字体大小和图像尺寸来支撑结构，一般不可以设置宽度、高度、对齐等属性，常用于控制页面中文本的样式。

> 常见的行内元素有<a>、、<b>、、<i>、<del>、<s>、<ins>、<u>、等，其中标签最典型的行内元素。

我一样重要

行内元素的特点：

（1）和相邻行内元素在一行上。

（2）高、宽无效，但水平方向的padding和margin可以设置，垂直方向的无效。

（3）默认宽度就是它本身内容的宽度。

（4）行内元素只能容纳文本或则其他行内元素。（a特殊 a里面可以放块级元素 ）

注意：

1. 只有 文字才 能组成段落 因此 p 里面不能放块级元素，同理还有这些标签h1,h2,h3,h4,h5,h6,dt，他们都是文字类块级标签，里面不能放其他块级元素。
2. 链接里面不能再放链接。
3. a里面可以放块级元素

块级元素和行内元素区别

块级元素的特点：
（1）总是从新行开始
（2）高度，行高、外边距以及内边距都可以控制。
（3）宽度默认是容器的100%
（4）可以容纳内联元素和其他块元素。

行内元素的特点：
（1）和相邻行内元素在一行上。
（2）高、宽无效，但水平方向的padding和margin可以设置，垂直方向的无效。
（3）默认宽度就是它本身内容的宽度。
（4）行内元素只能容纳文本或则其他行内元素。

行内块元素（inline-block）

在行内元素中有几个特殊的标签——<img />、<input />、<td>，可以对它们设置宽高和对齐属性，有些资料可能会称它们为行内块元素。

行内块元素的特点：
（1）和相邻行内元素（行内块）在一行上,但是之间会有空白缝隙。
（2）默认宽度就是它本身内容的宽度。
（3）高度，行高、外边距以及内边距都可以控制。

标签显示模式转换 display

块转行内：display:inline;

行内转块：display:block;

块、行内元素转换为行内块： display: inline-block;

此阶段，我们只需关心这三个，其他的是我们后面的工作。

课堂练习

1.写 三个 div 给定 100 * 100 的红色盒子 -- 宽度 高度 背景色

2.三个 span 也要求 150 * 150 绿色盒子

3. 三个 a 链接 80 * 20 蓝色 盒子 要求 必须一行显示 这三个盒子
4. 鼠标经过3个a链接的时候， 背景颜色变为 橙色 hover bgc
5. 导航栏案例

CSS书写规范

开始就形成良好的书写规范，是你专业化的开始。

空格规范

【强制】 选择器 与 { 之间必须包含空格。

示例： .selector { }

【强制】 属性名 与之后的 : 之间不允许包含空格， : 与 属性值 之间必须包含空格。

示例：

font-size: 12px;

选择器规范

【强制】 当一个 rule 包含多个 selector 时，每个选择器声明必须独占一行。

示例：

```
/* good */
.post,
.page,
.comment {
    line-height: 1.5;
}


/* bad */
.post, .page, .comment {
    line-height: 1.5;
}
```

【建议】 选择器的嵌套层级应不大于 3 级，位置靠后的限定条件应尽可能精确。

示例：

```
/* good */
#username input {}
.comment .avatar {}

/* bad */
.page .header .login #username input {}
.comment div * {}
```

属性规范

【强制】 属性定义必须另起一行。

示例：

```
/* good */
.selector {
    margin: 0;
    padding: 0;
}

/* bad */
.selector { margin: 0; padding: 0; }
```

【强制】 属性定义后必须以分号结尾。

示例：

```
/* good */
.selector {
    margin: 0;
}

/* bad */
.selector {
    margin: 0
}
```

行高的测量

行高我们利用最多的一个地方是： 可以让一行文本在盒子中垂直居中对齐。

做法就是： 文字的行高等于盒子的高度。

这里情况些许复杂，开始学习，我们可以先从简单地方入手学会。

上距离和下距离总是相等的，因此文字看上去是垂直居中的。

如果 行高 等 height 高度 文字会 垂直居中

如果行高 大于 高度 文字会 偏下

如果行高小于高度 文字会 偏上

CSS 三大特性

层叠 继承 优先级 是我们学习CSS 必须掌握的三个特性。

CSS层叠性

所谓层叠性是指多种CSS样式的叠加。

是浏览器处理冲突的一个能力,如果一个属性通过两个相同选择器设置到同一个元素上，那么这个时候一个属性就会将另一个属性层叠掉

比如先给某个标签指定了内部文字颜色为红色，接着又指定了颜色为蓝色，此时出现一个标签指定了相同样式不同值的情况，这就是样式冲突。 就近原则

一般情况下，如果出现样式冲突，则会按照CSS书写的顺序，以最后的样式为准。

1. 样式冲突，遵循的原则是就近原则。 那个样式离着结构近，就执行那个样式。
2. 样式不冲突，不会层叠

```
css最后的执行口诀：    长江后浪推前浪，前浪死在沙滩上。
```

CSS继承性

所谓继承性是指书写CSS样式表时，子标签会继承父标签的某些样式，如文本颜色和字号。想要设置一个可继承的属性，只需将它应用于父元素即可。

简单的理解就是： 子承父业。

```
css最后的执行口诀：    龙生龙，凤生凤，老鼠生的孩子会打洞。
```

注意：

> 恰当地使用继承可以简化代码，降低CSS样式的复杂性。子元素可以继承父元素的样式（text-，font-，line-这些元素开头的都可以继承，以及color属性）

CSS优先级

定义CSS样式时，经常出现两个或更多规则应用在同一元素上，这时就会出现优先级的问题。

在考虑权重时，初学者还需要注意一些特殊的情况，具体如下：

> 继承样式的权重为0。即在嵌套结构中，不管父元素样式的权重多大，被子元素继承时，他的权重都为0，也就是说子元素定义的样式会覆盖继承来的样式。
>
> 行内样式优先。应用style属性的元素，其行内样式的权重非常高，可以理解为远大于100。总之，他拥有比上面提高的选择器都大的优先级。
>
> 权重相同时，CSS遵循就近原则。也就是说靠近元素的样式具有最大的优先级，或者说排在最后的样式优先级最大。
>
> CSS定义了一个!important命令，该命令被赋予最大的优先级。也就是说不管权重如何以及样式位置的远近，!important都具有最大优先级。

CSS特殊性（Specificity）

关于CSS权重，我们需要一套计算公式来去计算，这个就是 CSS Specificity，我们称为CSS 特性或称非凡性，它是一个衡量CSS值优先级的一个标准 具体规范入如下：

specificity用一个四位的数 字串(CSS2是三位)来表示，更像四个级别，值从左到右，左面的最大，一级大于一级，数位之间没有进制，级别之间不可超越。

继承或者* 的贡献值	**0,0,0,0**
每个元素（标签）贡献值为	0,0,0,1
每个类，伪类贡献值为	0,0,1,0
每个ID贡献值为	0,1,0,0
每个行内样式贡献值	1,0,0,0
每个!important贡献值 重要的	∞ 无穷大

权重是可以叠加的

比如的例子：

```
div ul  li   ------>      0,0,0,3

.nav ul li   ------>      0,0,1,2

a:hover      ------>      0,0,1,1

.nav a       ------>      0,0,1,1

#nav p       ----->       0,1,0,1
```

注意：

1.数位之间没有进制 比如说： 0,0,0,5 + 0,0,0,5 =0,0,0,10 而不是 0,0, 1, 0， 所以不会存在10个div能赶上一个类选择器的情况。

1. 继承的 权重是 0

总结优先级：

1. 使用了 !important声明的规则。
2. 内嵌在 HTML 元素的 style属性里面的声明。
3. 使用了 ID 选择器的规则。
4. 使用了类选择器、属性选择器、伪元素和伪类选择器的规则。
5. 使用了元素选择器的规则。
6. 只包含一个通用选择器的规则。
7. 同一类选择器则遵循就近原则。

```
总结：权重是优先级的算法，层叠是优先级的表现
```

CSS 背景(background)

CSS 可以添加背景颜色和背景图片，以及来进行图片设置。

background-color	**背景颜色**
background-image	背景图片地址
background-repeat	是否平铺
background-position	背景位置
background-attachment	背景固定还是滚动
背景的合写（复合属性）	
background:背景颜色 背景图片地址 背景平铺 背景滚动 背景位置	

背景图片(image)

语法：

```
background-image : none | url (url)
```

参数：

none： 无背景图（默认的） url： 使用绝对或相对地址指定背景图像

background-image 属性允许指定一个图片展示在背景中（只有CSS3才可以多背景）可以和 background-color 连用。 如果图片不重复地话，图片覆盖不到地地方都会被背景色填充。 如果有背景图片平铺，则会覆盖背景颜色。

小技巧： 我们提倡 背景图片后面的地址，url不要加引号。

背景平铺（repeat）

语法：

```
background-repeat : repeat | no-repeat | repeat-x | repeat-y
```

参数：

repeat： 背景图像在纵向和横向上平铺（默认的）

no-repeat： 背景图像不平铺

repeat-x： 背景图像在横向上平铺

repeat-y： 背景图像在纵向平铺

设置背景图片时，默认把图片在水平和垂直方向平铺以铺满整个元素。

repeat-x：　背景图像在横向上平铺

repeat-y：　背景图像在纵向平铺

设置背景图片时，默认把图片在水平和垂直方向平铺以铺满整个元素。

背景位置(position)

语法：

```
background-position : length || length

background-position : position || position
```

参数：

length： 百分数 | 由浮点数字和单位标识符组成的长度值。请参阅长度单位 position： top | center | bottom | left | center | right

说明：

设置或检索对象的背景图像位置。必须先指定background-image属性。默认值为：(0% 0%)。 如果只指定了一个值，该值将用于横坐标。纵坐标将默认为50%。第二个值将用于纵坐标。

注意：

1. position 后面是x坐标和y坐标。 可以使用方位名词或者 精确单位。
2. 如果和精确单位和方位名字混合使用，则必须是x坐标在前，y坐标后面。比如 background-position: 15px top; 则 15px 一定是 x坐标 top是 y坐标。

实际工作用的最多的，就是背景图片居中对齐了。

背景附着

语法：

```
background-attachment : scroll | fixed
```

参数：

scroll： 背景图像是随对象内容滚动 fixed： 背景图像固定

说明：

设置或检索背景图像是随对象内容滚动还是固定的。

背景简写

background属性的值的书写顺序官方并没有强制标准的。为了可读性，建议大家如下写：

background:背景颜色 背景图片地址 背景平铺 背景滚动 背景位置

```
background: transparent url(image.jpg) repeat-y  scroll 50% 0 ;
```

背景透明(CSS3)

CSS3支持背景半透明的写法语法格式是:

```
background: rgba(0,0,0,0.3);
```

最后一个参数是alpha 透明度 取值范围 0~1之间

注意： 背景半透明是指盒子背景半透明， 盒子里面的内容不收影响。

导航栏案例

使用技巧：在一行内的盒子内，我们设定行高等于盒子的高度，就可以使文字垂直居中。

```
<head>
        <meta charset="utf-8">
        <style>
    body {
      background-color: #000;
    }
    a {
      width: 200px;
      height: 50px;
      /* background-color: orange; */
      display: inline-block;  /* 把a 行内元素转换为行内块元素 */
      text-align: center;  /* 文字水平居中 */
      line-height: 50px;  /* 我们设定行高等于盒子的高度，就可以使文字垂直居中 */
```

```
        color: #fff;
        font-size: 22px;
        text-decoration: none;  /* 取消下划线 文本装饰 */
    }
    a:hover {  /* 鼠标经过 给我们的链接添加背景图片*/
        background: url(images/h.png) no-repeat;
    }
        </style>
    </head>
    <body>
    <a href="#">专区说明</a>
    <a href="#">申请资格</a>
    <a href="#">兑换奖励</a>
    <a href="#">下载游戏</a>
    </body>
```

盒子模型（CSS重点）

其实，CSS就三个大模块： 盒子模型 、 浮动 、 定位，其余的都是细节。要求这三部分，无论如何也要学的非常精通。

所谓盒子模型就是把HTML页面中的元素看作是一个矩形的盒子，也就是一个盛装内容的容器。每个矩形都由元素的内容、内边距（padding）、边框（border）和外边距（margin）组成。

看透网页布局的本质

网页布局中，我们是如何把里面的文字，图片，按照美工给我们的效果图排列的整齐有序呢？

牛奶是怎样运输，让消费者购买的呢？

我们说过，行内元素比如 文字 类似牛奶，也需要一个盒子把他们装起来，我们前面学过的双标签都是一个盒子。有了盒子，我们就可以随意的，自由的，摆放位置了。

看透网页布局的本质： 把网页元素比如文字图片等等，放入盒子里面，然后利用CSS摆放盒子的过程，就是网页布局。

CSS 其实没有太多逻辑可言， 类似我们小时候玩的积木,我们可以自由的，随意的摆放出我们想要的效果。

盒子模型（Box Model）

这里略过 老旧的ie盒子模型（IE6以下），对不起，我都没见过IE5的浏览器。

首先，我们来看一张图，来体会下什么是盒子模型。

所有的文档元素（标签）都会生成一个矩形框，我们成为元素框（element box），它描述了一个文档元素再网页布局汇总所占的位置大小。因此，**每个盒子除了有自己大小和位置外，还影响着其他盒子的大小和位置。**

盒子边框（border）

边框就是那层皮。 橘子皮。。柚子皮。。橙子皮。。。

语法：

```
border : border-width || border-style || border-color
```

边框属性—设置边框样式（border-style）

边框样式用于定义页面中边框的风格，常用属性值如下：

```
none：没有边框即忽略所有边框的宽度（默认值）

solid：边框为单实线(最为常用的)

dashed：边框为虚线

dotted：边框为点线

double：边框为双实线
```

盒子边框写法总结表

设置内容	样式属性	常用属性值
上边框	border-top-style:样式; border-top-width:宽度;border-top-color:颜色;border-top:宽度 样式 颜色;	
下边框	border-bottom-style:样式;border- bottom-width:宽度;border- bottom-color:颜色;border-bottom:宽度 样式 颜色;	
左边框	border-left-style:样式; border-left-width:宽度;border-left-color:颜色;border-left:宽度 样式 颜色;	
右边框	border-right-style:样式;border-right-width:宽度;border-right-color:颜色;border-right:宽度 样式 颜色;	
样式综合设置	border-style:上边 [右边 下边 左边];	none无（默认）、solid单实线、dashed虚线、dotted点线、double双实线
宽度综合设置	border-width:上边 [右边 下边 左边];	像素值
颜色综合设置	border-color:上边 [右边 下边 左边];	颜色值、#十六进制、rgb(r,g,b)、rgb(r%,g%,b%)
边框综合设置	border:四边宽度 四边样式 四边颜色;	

```
border-top: 1px solid red; /*上边框*/
border-bottom: 2px solid green; /*下边框*/
border-left: 1px solid blue;
border-right: 5px solid pink;

border: 1px solid red;
```

表格的细线边框

以前学过的html表格边框很粗，这里只需要CSS一句话就可以美观起来。 让我们真的相信，CSS就是我们的白马王子（白雪公主）。

table{ border-collapse:collapse; } collapse 单词是合并的意思

border-collapse:collapse; 表示相邻边框合并在一起。

内边距（padding）

padding属性用于设置内边距。 是指 边框与内容之间的距离。

padding-top:上内边距

padding-right:右内边距

padding-bottom:下内边距

padding-left:左内边距

注意： 后面跟几个数值表示的意思是不一样的。

值的个数	表达意思
1个值	padding：上下左右边距 比如padding: 3px; 表示上下左右都是3像素
2个值	padding: 上下边距 左右边距 比如 padding: 3px 5px; 表示 上下3像素 左右 5像素
3个值	padding：上边距 左右边距 下边距 比如 padding: 3px 5px 10px; 表示 上是3像素 左右是5像素 下是10像素
4个值	padding:上内边距 右内边距 下内边距 左内边距 比如: padding: 3px 5px 10px 15px; 表示 上3px 右是5px 下 10px 左15px 顺时针

课堂案例： 新浪导航

关于盒子宽度下列正确的是（）

(A) 盒子宽：就是width的大小

(B) 盒子宽： padding-left + width + padding-right

(C) 盒子宽： border-left + width + border-right

(D) 盒子宽： border-left+ padding-left + width + padding-right + border-right

w 100 padding 10 border 5 ? 实际大小？ 130

3关于盒子高度下列正确的是（）

(A) 盒子高：就是height的大小

(B) 盒子高：padding-top +height + padding-bottom

(C) 盒子高：border-top + height + border-bottom

(D) 盒子高：border-top + padding-top +height + padding-bottom + border-bottom

4** 关于根据下列代码计算 盒子宽高下列说法正确的是（）****

div {

width: 200px;

height: 200px;

border: 1px solid #000000;

border-top: 5px solid blue;

padding: 50px;

padding-left: 100px;

}

(A) 宽度为200px 高度为200px

(B) 宽度为352px 高度为306px

(C) 宽度为302px 高度为307px

(D) 宽度为302px 高度为252px

外边距（margin）

margin属性用于设置外边距。 设置外边距会在元素之间创建“空白”， 这段空白通常不能放置其他内容。

margin-top:上外边距

margin-right:右外边距

margin-bottom:下外边距

margin-left:上外边距

margin:上外边距 右外边距 下外边距 左外边

取值顺序跟内边距相同。

外边距实现盒子居中

可以让一个盒子实现水平居中，需要满足一下两个条件：

1. 必须是块级元素。
2. 盒子必须指定了宽度（width）

然后就给**左右的外边距都设置为auto**，就可使块级元素水平居中。

实际工作中常用这种方式进行网页布局，示例代码如下：

```
.header{ width:960px; margin:0 auto;}
```

文字盒子居中图片和背景区别

1. 文字水平居中是 text-align: center
2. 盒子水平居中 左右margin 改为 auto

```
text-align: center; /*  文字居中水平 */
margin: 10px auto;  /* 盒子水平居中  左右margin 改为 auto 就阔以了 */
```

3. 插入图片 我们用的最多 比如产品展示类
4. 背景图片我们一般用于小图标背景 或者 超大背景图片

```
section img {
    width: 200px;/* 插入图片更改大小 width 和 height */
    height: 210px;
```

```
    margin-top: 30px;  /* 插入图片更改位置 可以用margin 或padding  盒模型 */
    margin-left: 50px; /* 插入当图片也是一个盒子 */
  }

aside {
    width: 400px;
    height: 400px;
    border: 1px solid purple;
    background: #fff url(images/sun.jpg) no-repeat;

    background-size: 200px 210px; /*  背景图片更改大小只能用 background-size */
    background-position: 30px 50px; /* 背景图片更该位置 我用 background-position
*/
  }
```

清除元素的默认内外边距

为了更方便地控制网页中的元素，制作网页时，可使用如下代码清除元素的默认内外边距：

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

注意： 行内元素是只有左右外边距的，是没有上下外边距的。 内边距，在ie6等低版本浏览器也会有问题。

我们尽量不要给行内元素指定上下的内外边距就好了。

外边距合并

使用margin定义块元素的垂直外边距时，可能会出现外边距的合并。

相邻块元素垂直外边距的合并

当上下相邻的两个块元素相遇时，如果上面的元素有下外边距margin-bottom，下面的元素有上外边距margin-top，则他们之间的垂直间距不是margin-bottom与margin-top之和，而是两者中的较大者。这种现象被称为相邻块元素垂直外边距的合并（也称外边距塌陷）。

解决方案： 避免就好了。

嵌套块元素垂直外边距的合并

对于两个嵌套关系的块元素，如果父元素没有上内边距及边框，则父元素的上外边距会与子元素的上外边距发生合并，合并后的外边距为两者中的较大者，即使父元素的上外边距为0，也会发生合并。

解决方案：

1. 可以为父元素定义1像素的上边框或上内边距。
2. 可以为父元素添加overflow:hidden。

待续。。。。

content宽度和高度

使用宽度属性width和高度属性height可以对盒子的大小进行控制。

width和height的属性值可以为不同单位的数值或相对于父元素的百分比%，实际工作中最常用的是像素值。

大多数浏览器，如Firefox、IE6及以上版本都采用了W3C规范，符合CSS规范的盒子模型的总宽度和总高度的计算原则是：

```
/*外盒尺寸计算（元素空间尺寸）*/
Element空间高度 = content height + padding + border + margin
Element 空间宽度 = content width + padding + border + margin
/*内盒尺寸计算（元素实际大小）*/
Element Height = content height + padding + border （Height为内容高度）
Element Width = content width + padding + border （Width为内容宽度）
```

注意：

1、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（ img 标签和 input除外）。

2、计算盒子模型的总高度时，还应考虑上下两个盒子垂直外边距合并的情况。

3、**如果一个盒子则会和父亲一样宽 占满父亲的宽度， 如果此盒子没有给定宽度 则padding 不会影响本盒子大小。**

盒子模型布局稳定性

开始学习盒子模型，同学们最大的困惑就是， 分不清内外边距的使用，什么情况下使用内边距，什么情况下使用外边距?

答案是： 其实他们大部分情况下是可以混用的。 就是说，你用内边距也可以，用外边距也可以。 你觉得哪个方便，就用哪个。

但是，总有一个最好用的吧，我们根据稳定性来分，建议如下：

按照 优先使用 宽度 （width） 其次 使用内边距（padding） 再次 外边距（margin）。

```
width >  padding  >   margin
```

原因：

1. margin 会有外边距合并 还有 ie6下面margin 加倍的bug（讨厌）所以最后使用。
2. padding 会影响盒子大小， 需要进行加减计算（麻烦） 其次使用。
3. width 没有问题（嗨皮）我们经常使用宽度剩余法 高度剩余法来做。

圆角边框(CSS3)

从此以后，我们的世界不只有矩形。radius 半径（距离）

语法格式：

```
border-radius: 50%;   让一个正方形  变成圆圈
```

盒子阴影(CSS3)

语法格式：

```
box-shadow:水平阴影  垂直阴影  模糊距离（虚实）    阴影尺寸（影子大小）    阴影颜色   内/外阴影;
```

值	描述
h-shadow	必需。水平阴影的位置。允许负值。
v-shadow	必需。垂直阴影的位置。允许负值。
blur	可选。模糊距离。
spread	可选。阴影的尺寸。
color	可选。阴影的颜色。请参阅 CSS 颜色值。
inset	可选。将外部阴影 (outset) 改为内部阴影。

1. 前两个属性是必须写的。其余的可以省略。
2. 外阴影 (outset) 但是不能写 默认 想要内阴影 inset

```
div {
      width: 200px;
      height: 200px;
      border: 10px solid red;
      /* box-shadow: 5px 5px 3px 4px rgba(0, 0, 0, .4);  */
      /* box-shadow:水平位置  垂直位置  模糊距离  阴影尺寸（影子大小）  阴影颜色   内/外阴影;
*/
      box-shadow: 0 15px 30px  rgba(0, 0, 0, .4);

}
```

浮动(float)

普通流(normal flow)

这个单词很多人翻译为 文档流 ， 字面翻译 普通流 或者标准流都可以。

前面我们说过，网页布局的核心，就是用CSS来摆放盒子位置。如何把盒子摆放到合适的位置?

CSS的定位机制有3种：普通流（标准流）、浮动和定位。

html语言当中另外一个相当重要的概念----------标准流！或者普通流。普通流实际上就是一个网页内标签元素正常从上到下，从左到右排列顺序的意思，比如块级元素会独占一行，行内元素会按顺序依次前后排列；按照这种大前提的布局排列之下绝对不会出现例外的情况叫做普通流布局。

三种布局直观理解

标准流

浮动

定位

传智播客 传智播客 传智播客

浮动(float)

浮动最早是用来控制图片，以便达到其他元素（特别是文字）实现“环绕”图片的效果。

1991年1月，发行专辑《一起走过的日子》，专辑同名主打歌获得十大劲歌金曲奖以及十大中文金曲奖[116-117]；2月，发行粤语专辑《爱不完》[118]；该专辑发行首日录音带的销量达到16万张，而激光唱片的销量则有7.2万张[119]；6月，发行国语专辑《我和我追逐的梦》；同年，获得十大劲歌金曲奖最受欢迎男歌手奖[116]。

1992年，在洛杉矶、旧金山等地举行美加巡回演唱会；9月，发行加入 华纳唱片公司以后的首张专辑《真我的风采》；该专辑发行后销量达到40万张[119]；同年，在提前录制的央视春晚视频中，刘德华与毛阿敏合唱歌曲《心中常驻芳华》[120]；此外，他还发行了专辑《谢谢你的爱》，同名主打歌曲获得第十六届十大中文金曲奖[121]。

1993年1月至2月，在香港红磡体育馆举行20场“真我的风采演唱会”；4月，推出加入飞碟唱片公司后的首张专辑《真情难收》，其中的主打歌曲《一辈子的错》由刘德华作词；11月，推出专辑粤语专辑《答案就是你》，专辑中的歌曲《永远寂寞》获得第16届十大中文金曲获奖[121]。

1994年，发行国语专辑《忘情水》，同名主打歌获得第17届十大中文金曲奖最受欢迎国语歌曲金奖以及第1届华语榜中榜Channel [V]中文Top 20榜中榜歌曲奖[122-123]；8月6日至25日，在香港红磡体育馆举行20场演唱会；11月25日，发行国语专辑《天意》；专辑中表达友情的主打歌曲《友谊历久一样浓》是刘德华个人填词的作品；同年，获得十大劲歌金曲奖最受欢迎男歌手奖[11]

刘德华（3张）

后来，我们发现浮动有个很有意思的事情：就是让任何盒子可以一行排列,因此我们就慢慢的偏离主题，用浮动的特性来布局了。（CSS3已经我们真正意义上的网页布局，具体CSS3我们会详细解释）

三个div盒子一行内显示

什么是浮动?

元素的浮动是指设置了浮动属性的元素会脱离标准标准流的控制，移动到其父元素中指定位置的过程。

在CSS中，通过float属性来定义浮动，其基本语法格式如下：

```
选择器{float:属性值;}
```

属性值	描述
left	元素向左浮动
right	元素向右浮动
none	元素不浮动（默认值）

浮动详细内幕特性

浮动脱离标准流，====脱标==== 不占位置，会影响标准流。浮动只有左右浮动。

```
1. 浮动首先创建包含块的概念（包裹）。就是说， 浮动的元素总是找理它最近的父级元素对齐。但是不会超出内边距的范围。
```

```
2.一个父盒子里面的子盒子，如果其中一个子级有浮动的，则其他子级都需要浮动。这样才能一行对齐显示。
```

```
3. 元素添加浮动后，元素会具有行内块元素的特性。元素的大小完全取决于定义的大小或者默认的内容多少浮动根据元素书写的位置来显示相应的浮动。
```

总结： 浮动 --->

浮动的目的就是为了让多个块级元素同一行上显示。 最核心的关键点就是 怎么排列的， 是否占有位置

float 浮 漏 特

浮： 加了浮动的元素盒子是浮起来的，漂浮在其他的标准流盒子上面。 漏： 加了浮动的盒子，不占位置的，它浮起来了，它原来的位置漏 给了标准流的盒子。 特： 特别注意，首先浮动的盒子需要和标准流的父级搭配使用， 其次 特别的注意浮动可以使元素显示模式体现为行内块特性。

版心和布局流程

阅读报纸时容易发现，虽然报纸中的内容很多，但是经过合理地排版，版面依然清晰、易读。同样，在制作网页时，要想使页面结构清晰、有条理，也需要对网页进行“排版”。

“版心”(可视区) 是指网页中主体内容所在的区域。一般在浏览器窗口中水平居中显示，常见的宽度值为960px、980px、1000px、1200px等。

布局流程

为了提高网页制作的效率，布局时通常需要遵守一定的布局流程，具体如下：

1、确定页面的版心（可视区）。

2、分析页面中的行模块，以及每个行模块中的列模块。

3、制作HTML结构 。

4、CSS初始化，然后开始运用盒子模型的原理，通过DIV+CSS布局来控制网页的各个模块。

一列固定宽度且居中

最普通的，最为常用的结构

两列左窄右宽型

比如小米 [小米官网]

通栏平均分布型

比如锤子 [锤子官网]

清除浮动

人生就像乘坐北京地铁一号线：

途经国贸，羡慕繁华；

途经天安门，幻想权力；

途经金融街，梦想发财；

经过公主坟，遥想华丽家族；

经过玉泉路，依然雄心勃勃...

这时，有个声音飘然入耳:乘客你好,八宝山马上就要到了！

顿时醒悟：人生苦短，有始有终。

好比我们的浮动，有浮动开始，则就应该有浮动结束。

为什么要清除浮动

我们前面说过，浮动本质是用来做一些文字混排效果的，但是被我们拿来做布局用，则会有很多的问题出现， 但是，你不能说浮动不好

。

由于浮动元素不再占用原文档流的位置，所以它会对后面的元素排版产生影响，为了解决这些问题，此时就需要在该元素中清除浮动。

准确地说，并不是清除浮动，而是**清除浮动后造成的影响**

如果浮动一开始就是一个美丽的错误，那么请用正确的方法挽救它。

清除浮动本质

清除浮动主要为了解决父级元素因为子级浮动引起内部高度为0 的问题。

正常标准流盒子

子盒子浮动

下盒子移动到下侧

清除浮动的方法

其实本质叫做闭合浮动更好一些, 记住，清除浮动就是把浮动的盒子圈到里面，让父盒子闭合出口和入口不让他们出来影响其他元素。

在CSS中，clear属性用于清除浮动，其基本语法格式如下：

```
选择器{clear:属性值;}   clear 清除
```

属性值	描述
left	不允许左侧有浮动元素（清除左侧浮动的影响）
right	不允许右侧有浮动元素（清除右侧浮动的影响）
both	同时清除左右两侧浮动的影响

额外标签法

```
是w3c推荐的做法是通过在浮动元素末尾添加一个空的标签例如 <div style="clear:both">
</div>，或则其他标签br等亦可。
```

优点： 通俗易懂，书写方便

缺点： 添加许多无意义的标签，结构化较差。 我只能说，w3c你推荐的方法我不接受，你不值得拥有。。。

父级添加overflow属性方法

可以通过触发BFC的方式，可以实现清除浮动效果。（BFC后面讲解）

```
可以给父级添加： overflow为 hidden|auto|scroll  都可以实现。
```

优点： 代码简洁

缺点： 内容增多时候容易造成不会自动换行导致内容被隐藏掉，无法显示需要溢出的元素。

使用after伪元素清除浮动

:after 方式为空元素的升级版，好处是不用单独加标签了

使用方法：

```
 .clearfix:after {  content: ""; display: block; height: 0; clear: both;
visibility: hidden;  }

 .clearfix {*zoom: 1;}   /* IE6、7 专有 */
```

优点： 符合闭合浮动思想 结构语义化正确

缺点： 由于IE6-7不支持:after，使用 zoom:1触发 hasLayout。

代表网站： 百度、淘宝网、网易等

注意： content:"" 尽量不带点

使用before和after双伪元素清除浮动

使用方法：

```
.clearfix:before,.clearfix:after {
  content:"";
  display:table;  /* 这句话可以出发BFC BFC可以清除浮动,BFC我们后面讲 */
}
.clearfix:after {
 clear:both;
}
.clearfix {
  *zoom:1;
}
```

优点： 代码更简洁

缺点： 由于IE6-7不支持:after，使用 zoom:1触发 hasLayout。

代表网站： 小米、腾讯等

Photoshop基本使用

PS界面组成：

ctrl + r 显示隐藏标尺 右击 标尺 --- 把里面的单位一律改为像素

ctrl+ d 取消选区

菜单栏、选项栏、工具栏、浮动面板（拖拽名称，可单独操作面板）、绘图窗口 窗口菜单，可显示隐藏所有面板

工作区： （新建）

调整浮动面板

图层操作(重点)

图层面板快捷键 F7 其实图层就是一张张透明的纸 可以实现叠加问题。

图层选择： 使用移动工具V

1、图层缩览图判断

2、按住CTRL,在目标图像上单击

3、将光标放置在目标图像上右键，选择图层名称

图层面板中加选图层：

1、按SHIFT，单击另一目标图层 中间所有图层被选中

2、按CTRL，单击另一目标图层 只选中目标图层

复制图层：选中目标图层后（移动工具状态下）

1、按ALT拖拽图像

2、CTRL+J （重合）

案例： 摆放一个自行车

图层编组

选中目标图层，CTRL+G

取消编组：CTRL+SHIFT+G

双击图层名称可重新命名

双击组名称，可命名组

移动工具V选择组或图层时，需设置选项栏

图层上下位置移动

1、选中目标图层，在图层面拖拽

2、CTRL+] 向上移动图层 CTRL+[向下移动图层

3、CTRL+SHIFT+] 图层置顶 CTRL+SHIFT+[图层置底

移动选区或图像时：

移动过程中，没释放鼠标，按住SHIFT，可同一水平线、同一垂线、45度移动。

ps中的撤销操作是：

ctrl+z 撤销一步

ctrl+alt+z 撤销多步

Photoshop 切图

PS切图 可以 分为 手动 利用切片切图 以及 利用PS的插件快速切图

切片工具

1. 利用切片工具手动划出
2. 图层菜单---新建基于图层的切片
3. 利用标尺 基于参考线的切片 （选择切片工具）

4. 先选个一个整个的切片， 切片选择工具-- 属性面板中有 “划分” --可以等分数平分切图

导出切片： 文件-- 存储为web设备所用格式

辅助线和切片使用及清除

视图菜单-- 清除 辅助线/ 清除切片

切图插件

Cutterman是一款运行在photoshop中的插件，能够自动将你需要的图层进行输出， 以替代传统的手工"导出web所用格式" 以及使用切片工具进行挨个切图的繁琐流程。 它支持各种各样的图片尺寸、格式、形态输出，方便你在pc、ios、Android等端上使用。 它不需要你记住一堆的语法、规则，纯点击操作，方便、快捷，易于上手。

官网: http://www.cutterman.cn/zh/cutterman

注意： cutterman插件要求你的ps 必须是完整版，不能是绿色版，所以大家需要从新安装完整版本。

项目案例： 云道页面

案例练习目的是总结以前的css和html

还有ps的使用。

制作步骤：

1. 准备相关文件。（内部样式表) html文件(index.html) 图片文件
2. 准备CSS 初始化。 书写结构和样式
3. 确定版心（是1200像素)和各个模块。

定位(position)

background-position 背景定位

如果，说浮动， 关键在一个 “浮” 字上面， 那么 我们的定位，关键在于一个 “位” 上。

PS: 定位是我们CSS算是数一数二难点的了，但是，你务必要学好它，我们CSS离不开定位，特别是后面的js特效，天天和定位打交道。不要抵触它，反而要爱上它，它可以让我们工作更加轻松哦！

为什么要用定位?

那么定位，最长运用的场景再那里呢？ 来看几幅图片，你一定会有感悟！

第一幅图， 小黄色块可以再图片上移动：

第二幅图， 左右箭头压住图片：

第三幅图, hot 再盒子外面多出一块，更加突出：

以上三个小地方，如果用标准流或者浮动，实现会比较复杂或者难以实现，此时我们用定位来做，just soso！

元素的定位属性

元素的定位属性主要包括定位模式和边偏移两部分。

1、边偏移

边偏移属性	描述
top	顶端偏移量，定义元素相对于其父元素上边线的距离
bottom	底部偏移量，定义元素相对于其父元素下边线的距离
left	左侧偏移量，定义元素相对于其父元素左边线的距离
right	右侧偏移量，定义元素相对于其父元素右边线的距离

也就说，以后定位要和这边偏移搭配使用了， 比如 top: 100px; left: 30px; 等等

2、定位模式(定位的分类)

在CSS中，position属性用于定义元素的定位模式，其基本语法格式如下：

选择器{position:属性值;}

position属性的常用值

值	描述
static	自动定位（默认定位方式）
relative	相对定位，相对于其原文档流的位置进行定位
absolute	绝对定位，相对于其上一个已经定位的父元素进行定位
fixed	固定定位，相对于浏览器窗口进行定位

静态定位(static)

静态定位是所有元素的默认定位方式，当position属性的取值为static时，可以将元素定位于静态位置。所谓静态位置就是各个元素在HTML文档流中默认的位置。

上面的话翻译成白话： 就是网页中所有元素都默认的是静态定位哦！ 其实就是标准流的特性。

在静态定位状态下，无法通过边偏移属性（top、bottom、left或right）来改变元素的位置。

PS： 静态定位其实没啥可说的。

静态定位唯一的用处： 就是 取消定位。 position: static;

相对定位relative(自恋型)

> 小笑话：
> 刚刚看到一个超级超级帅的帅哥，看得我都忍不住想和他搞基了。世间怎会有如此之完美的男人。我和他就这样一动不动的对视着，就仿佛一见钟情。时间也在这一瞬间停止了。直到我的手麻了。才恋恋不舍的放下镜子。。。。

相对定位是将元素相对于它在标准流中的位置进行定位，当position属性的取值为relative时，可以将元素定位于相对位置。

对元素设置相对定位后，可以通过边偏移属性改变元素的位置，但是它在文档流中的位置仍然保留。如下图所示，即是一个相对定位的效果展示：

注意：

1. 相对定位最重要的一点是，它可以通过边偏移移动位置，但是原来的所占的位置，继续占有。
2. 其次，每次移动的位置，是以自己的左上角为基点移动（相对于自己来移动位置）

就是说，相对定位的盒子仍在标准流中，它后面的盒子仍以标准流方式对待它。（相对定位不脱标）

如果说浮动的主要目的是 让多个块级元素一行显示，那么定位的主要价值就是 移动位置， 让盒子到我们想要的位置上去。

绝对定位absolute (拼爹型)

```
小笑话：

吃早饭时，老婆往儿子碗里放了两个煎蛋，儿子全给了我，还一本正经地说：“爸爸，多吃点，男人养家不容易。” <br/>

我一阵感动，刚想夸他两句。

儿子接着说：“以后全靠你让我拼爹了！”
```

[注意] 如果文档可滚动，绝对定位元素会随着它滚动，因为元素最终会相对于正常流的某一部分定位。

当position属性的取值为absolute时，可以将元素的定位模式设置为绝对定位。

注意： 绝对定位最重要的一点是，它可以通过边偏移移动位置，但是它完全脱标，完全不占位置。

父级没有定位

若所有父元素都没有定位，以浏览器当前屏幕为准对齐(document文档)。

父级有定位

绝对定位是将元素依据最近的已经定位（绝对、固定或相对定位）的父元素（祖先）进行定位。

子绝父相

这个“子绝父相”太重要了，是我们学习定位的口诀，时时刻刻记住的。

这句话的意思是 子级是绝对定位的话， 父级要用相对定位。

首先， 我们说下， 绝对定位是将元素依据最近的已经定位绝对、固定或相对定位）的父元素（祖先）进行定位。

就是说， 子级是绝对定位，父亲只要是定位即可（不管父亲是绝对定位还是相对定位，甚至是固定定位都可以），就是说， 子绝父绝，子绝父相都是正确的。

但是，在我们网页布局的时候， 最常说的 子绝父相是怎么来的呢？ 请看如下图：

所以，我们可以得出如下结论：

因为子级是绝对定位，不会占有位置， 可以放到父盒子里面的任何一个地方。

父盒子布局时，需要占有位置，因此父亲只能是 相对定位.

这就是子绝父相的由来。

绝对定位的盒子水平/垂直居中

普通的盒子是左右margin 改为 auto就可， 但是对于绝对定位就无效了

定位的盒子也可以水平或者垂直居中，有一个算法。

1. 首先left 50% 父盒子的一半大小
2. 然后走自己外边距负的一半值就可以了 margin-left。

固定定位fixed(认死理型)

固定定位是绝对定位的一种特殊形式，类似于 正方形是一个特殊的 矩形。它以浏览器窗口作为参照物来定义网页元素。当position属性的取值为fixed时，即可将元素的定位模式设置为固定定位。

当对元素设置固定定位后，它将脱离标准文档流的控制，始终依据浏览器窗口来定义自己的显示位置。不管浏览器滚动条如何滚动也不管浏览器窗口的大小如何变化，该元素都会始终显示在浏览器窗口的固定位置。

固定定位有两点：

1. 固定定位的元素跟父亲没有任何关系，只认浏览器。
2. 固定定位完全脱标，不占有位置，不随着滚动条滚动。

记忆法： 就类似于孙猴子， 无父无母，好不容易找到一个可靠的师傅（浏览器），就听的师傅的，别的都不听。

ie6等低版本浏览器不支持固定定位。

叠放次序（z-index）

当对多个元素同时设置定位时，定位元素之间有可能会发生重叠。

在CSS中，要想调整重叠定位元素的堆叠顺序，可以对定位元素应用z-index层叠等级属性，其取值可为正整数、负整数和0。

比如： z-index: 2; font-weight: 700

注意：

1. z-index的默认属性值是0，取值越大，定位元素在层叠元素中越居上。
2. 如果取值相同，则根据书写顺序，后来居上。
3. 后面数字一定不能加单位。
4. 只有相对定位，绝对定位，固定定位有此属性，其余标准流，浮动，静态定位都无此属性，亦不可指定此属性。

四种定位总结

定位模式	是否脱标占有位置	是否可以使用边偏移	移动位置基准
静态static	不脱标，正常模式	不可以	正常模式
相对定位relative	脱标，占有位置	可以	相对自身位置移动（自恋型）
绝对定位absolute	完全脱标，不占有位置	可以	相对于定位父级移动位置（拼爹型）
固定定位fixed	完全脱标，不占有位置	可以	相对于浏览器移动位置（认死理型）

定位模式转换

跟 浮动一样， 元素添加了 绝对定位和固定定位之后， 元素模式也会发生转换， 都转换为 行内块模式，

行内块 的宽度和高度 跟内容有关系

** 因此 比如 行内元素 如果添加了 绝对定位或者 固定定位后 浮动后，可以不用转换模式，直接给高度和宽度就可以了。**

CSS高级技巧

元素的显示与隐藏

在CSS中有三个显示和隐藏的单词比较常见，我们要区分开，他们分别是 display visibility 和 overflow。

他们的主要目的是让一个元素在页面中消失，但是不在文档源码中删除。 最常见的是网站广告，当我们点击类似关闭不见了，但是我们重新刷新页面，它们又会出现和你玩躲猫猫！！

display 显示

display 设置或检索对象是否及如何显示。

display : none 隐藏对象 与它相反的是 display:block 除了转换为块级元素之外，同时还有显示元素的意思。

特点： 隐藏之后，不再保留位置。

visibility 可见性

设置或检索是否显示对象。

visible :　对象可视

hidden :　对象隐藏

特点： 隐藏之后，继续保留原有位置。（停职留薪）

overflow 溢出

检索或设置当对象的内容超过其指定高度及宽度时如何管理内容。

visible :　不剪切内容也不添加滚动条。

auto :　超出自动显示滚动条，不超出不显示滚动条

hidden :　不显示超过对象尺寸的内容，超出的部分隐藏掉

scroll :　不管超出内容否，总是显示滚动条

CSS用户界面样式

所谓的界面样式， 就是更改一些用户操作样式， 比如 更改用户的鼠标样式， 表单轮廓等。但是比如滚动条的样式改动受到了很多浏览器的抵制，因此我们就放弃了。 防止表单域拖拽

鼠标样式cursor

设置或检索在对象上移动的鼠标指针采用何种系统预定义的光标形状。

```
cursor : default 小白 | pointer 小手 | move 移动 | text 文本
```

鼠标放我身上查看效果哦：

```
<ul>
  <li style="cursor:default">我是小白</li>
  <li style="cursor:pointer">我是小手</li>
  <li style="cursor:move">我是移动</li>
  <li style="cursor:text">我是文本</li>
</ul>
```

尽量不要用hand 因为 火狐不支持 pointer ie6以上都支持的尽量用

轮廓 outline

是绘制于元素周围的一条线，位于边框边缘的外围，可起到突出元素的作用。

```
outline : outline-color ||outline-style || outline-width
```

但是我们都不关心可以设置多少，我们平时都是去掉的。

最直接的写法是： outline: 0; 或者 outline: none;

```
<input  type="text"  style="outline: 0;"/>
```

防止拖拽文本域resize

resize：none 这个单词可以防止 火狐 谷歌等浏览器随意的拖动 文本域。

右下角可以拖拽：

右下角不可以拖拽：

```
<textarea  style="resize: none;"></textarea>
```

vertical-align 垂直对齐

以前我们讲过让带有宽度的块级元素居中对齐，是margin: 0 auto;

以前我们还讲过让文字居中对齐，是 text-align: center;

但是我们从来没有讲过有垂直居中的属性， 我们的妈妈一直很担心我们的垂直居中怎么做。

vertical-align 垂直对齐， 这个看上去很美好的一个属性， 实际有着不可捉摸的脾气，否则我们也不会这么晚来讲解。

```
vertical-align : baseline |top |middle |bottom
```

设置或检索对象内容的垂直对其方式。

vertical-align 不影响块级元素中的内容对齐，它只针对于 行内元素或者行内块元素，特别是行内块元素， **通常用来控制图片/表单与文字的对齐**。

模式	单词
基线对齐： 默认的是文字和图片基线对齐	vertical-align: baseline;
垂直居中： 默认的是文字和图片基线对齐	vertical-align: middle;
顶部对齐： 默认的是文字和图片基线对齐	vertical-align: top;

图片、表单和文字对齐

所以我们知道，我们可以通过vertical-align 控制图片和文字的垂直关系了。 默认的图片会和文字基线对齐。

去除图片底侧空白缝隙

有个很重要特性你要记住： 图片或者表单等行内块元素，他的底线会和父级盒子的基线对齐。这样会造成一个问题，就是图片底侧会有一个空白缝隙。

解决的方法就是：

1. 给img vertical-align:middle | top等等。 让图片不要和基线对齐。

1. 给img 添加 display：block; 转换为块级元素就不会存在问题了。

溢出的文字隐藏

white-space

white-space设置或检索对象内文本显示方式。通常我们使用于强制一行显示内容

normal： 默认处理方式 nowrap： 强制在同一行内显示所有文本，直到文本结束或者遭遇br标签对象才换行。

可以处理中文

text-overflow 文字溢出

text-overflow : clip | ellipsis

设置或检索是否使用一个省略标记（...）标示对象内文本的溢出

clip： 不显示省略标记（...），而是简单的裁切

ellipsis： 当对象内文本溢出时显示省略标记（...）

注意一定要首先强制一行内显示，再次和overflow属性 搭配使用

CSS精灵技术（sprite） 小妖精 雪碧

精灵技术产生的背景

图所示为网页的请求原理图，当用户访问一个网站时，需要向服务器发送请求，网页上的每张图像都要经过一次请求才能展现给用户。

然而，一个网页中往往会应用很多小的背景图像作为修饰，当网页中的图像过多时，服务器就会频繁地接受和发送请求，这将大大降低页面的加载速度。为了有效地减少服务器接受和发送请求的次数，提高页面的加载速度，出现了CSS精灵技术（也称CSS Sprites、CSS雪碧）。

精灵技术本质

简单地说，CSS精灵是一种处理网页背景图像的方式。它将一个页面涉及到的所有零星背景图像都集中到一张大图中去，然后将大图应用于网页，这样，当用户访问该页面时，只需向服务发送一次请求，网页中的背景图像即可全部展示出来。通常情况下，这个由很多小的背景图像合成的大图被称为精灵图（雪碧图），如下图所示为京东网站中的一个精灵图。

精灵技术的使用

CSS 精灵其实是将网页中的一些背景图像整合到一张大图中（精灵图），然而，各个网页元素通常只需要精灵图中不同位置的某个小图，要想精确定位到精灵图中的某个小图，就需要使用CSS的background-image、background-repeat和background-position属性进行背景定位，其中最关键的是使用background-position属性精确地定位。

制作精灵图

CSS 精灵其实是将网页中的一些背景图像整合到一张大图中（精灵图），那我们要做的，就是把小图拼合成一张大图。

大部分情况下，精灵图都是网页美工做。

> 我们精灵图上放的都是小的装饰性质的背景图片。 插入图片不能往上放。
> 我们精灵图的宽度取决于最宽的那个背景。
> 我们可以横向摆放也可以纵向摆放，但是每个图片之间，间隔至少隔开偶数像素合适。
> 在我们精灵图的最低端，留一片空隙，方便我们以后添加其他精灵图。

结束语： 小公司，背景图片很少的情况，没有必要使用精灵技术，维护成本太高。 如果是背景图片比较多，可以建议使用精灵技术。

滑动门

先来体会下现实中的滑动门,或者你可以叫做推拉门：

滑动门出现的背景

制作网页时，为了美观，常常需要为网页元素设置特殊形状的背景，比如微信导航栏，有凸起和凹下去的感觉，最大的问题是里面的字数不一样多，咋办？

为了使各种特殊形状的背景能够自适应元素中文本内容的多少，出现了CSS滑动门技术。它从新的角度构建页面，使各种特殊形状的背景能够自由拉伸滑动，以适应元素内部的文本内容，可用性更强。 最常见于各种导航栏的滑动门。

核心技术

核心技术就是利用CSS精灵（主要是背景位置）和盒子padding撑开宽度, 以便能适应不同字数的导航栏。

一般的经典布局都是这样的：

```
<li>
  <a href="#">
    <span>导航栏内容</span>
  </a>
</li>
```

总结：

1. a 设置 背景左侧，padding撑开合适宽度。
2. span 设置背景右侧， padding撑开合适宽度 剩下由文字继续撑开宽度。
3. 之所以a包含span就是因为 整个导航都是可以点击的。

web字体

字体格式

不同浏览器所支持的字体格式是不一样的，我们有必要了解一下有关字体格式的知识。

1、TureType(.ttf)格式

.ttf字体是Windows和Mac的最常见的字体，是一种RAW格式，支持这种字体的浏览器有IE9+、Firefox3.5+、Chrome4+、Safari3+、Opera10+、iOS Mobile、Safari4.2+；

2、OpenType(.otf)格式

.otf字体被认为是一种原始的字体格式，其内置在TureType的基础上，支持这种字体的浏览器有Firefox3.5+、Chrome4.0+、Safari3.1+、Opera10.0+、iOS Mobile、Safari4.2+；

3、Web Open Font Format(.woff)格式

woff字体是Web字体中最佳格式，他是一个开放的TrueType/OpenType的压缩版本，同时也支持元数据包的分离，支持这种字体的浏览器有IE9+、Firefox3.5+、Chrome6+、Safari3.6+、Opera11.1+；

4、Embedded Open Type(.eot)格式

.eot字体是IE专用字体，可以从TrueType创建此格式字体，支持这种字体的浏览器有IE4+；

5、SVG(.svg)格式

.svg字体是基于SVG字体渲染的一种格式，支持这种字体的浏览器有Chrome4+、Safari3.1+、Opera10.0+、iOS Mobile Safari3.2+；

了解了上面的知识后，我们就需要为不同的浏览器准备不同格式的字体，通常我们会通过字体生成工具帮我们生成各种格式的字体，因此无需过于在意字体格式间的区别差异。

字体图标

图片是有诸多优点的，但是缺点很明显，比如图片不但增加了总文件的大小，还增加了很多额外的"http请求"，这都会大大降低网页的性能的。更重要的是图片不能很好的进行“缩放”，因为图片放大和缩小会失真。 我们后面会学习移动端响应式，很多情况下希望我们的图标是可以缩放的。此时，一个非常重要的技术出现了，额不是出现了，是以前就有，是被从新"宠幸"啦。。 这就是字体图标（iconfont).

字体图标优点

```
可以做出跟图片一样可以做的事情,改变透明度、旋转度，等..
但是本质其实是文字，可以很随意的改变颜色、产生阴影、透明效果等等...
本身体积更小，但携带的信息并没有削减。
几乎支持所有的浏览器
移动端设备必备良药...
```

字体图标使用流程

总体来说，字体图标按照如下流程：

设计字体图标

假如图标是我们公司单独设计，那就需要第一步了，这个属于UI设计人员的工作， 他们在 illustrator 或 Sketch 这类矢量图形软件里创建 icon图标， 比如下图：

之后保存为svg格式，然后给我们前端人员就好了。

其实第一步，我们不需要关心，只需要给我们这些图标就可以了，如果图标是大众的，网上本来就有的，可以直接跳过第一步，进入第三步。

上传生成字体包

当UI设计人员给我们svg文件的时候，我们需要转换成我们页面能使用的字体文件， 而且需要生成的是兼容性的适合各个浏览器的。

推荐网站： http://icomoon.io

icomoon字库

IcoMoon成立于2011年，推出的第一个自定义图标字体生成器，它允许用户选择他们所需要的图标，使它们成一字型。 内容种类繁多，非常全面，唯一的遗憾是国外服务器，打开网速较慢。

推荐网站： http://www.iconfont.cn/

阿里icon font字库

http://www.iconfont.cn/

这个是阿里妈妈M2UX的一个icon font字体图标字库，包含了淘宝图标库和阿里妈妈图标库。可以使用AI制作图标上传生成。 一个字，免费，免费！！

fontello

http://fontello.com/

在线定制你自己的icon font字体图标字库，也可以直接从GitHub下载整个图标集，该项目也是开源的。

Font-Awesome

http://fortawesome.github.io/Font-Awesome/

这是我最喜欢的字库之一了，更新比较快。目前已经有369个图标了。

Glyphicon Halflings

http://glyphicons.com/

这个字体图标可以在Bootstrap下免费使用。自带了200多个图标。

Icons8

https://icons8.com/

提供PNG免费下载，像素大能到500PX

下载兼容字体包

刚才上传完毕， 网站会给我们把UI做的svg图片转换为我们的字体格式， 然后下载下来就好了

当然，我们不需要自己专门的图标，是想网上找几个图标使用，以上2步可以直接省略了， 直接到刚才的网站上找喜欢的下载使用吧。

字体引入到HTML

得到压缩包之后，最后一步，是最重要的一步了， 就是字体文件已经有了，我们需要引入到我们页面中。

1. 首先把 以下4个文件放入到 fonts文件夹里面。 通俗的做法

第一步：在样式里面声明字体： 告诉别人我们自己定义的字体

```
@font-face {
  font-family: 'icomoon';
  src:  url('fonts/icomoon.eot?7kkyc2');
  src:  url('fonts/icomoon.eot?7kkyc2#iefix') format('embedded-opentype'),
    url('fonts/icomoon.ttf?7kkyc2') format('truetype'),
    url('fonts/icomoon.woff?7kkyc2') format('woff'),
    url('fonts/icomoon.svg?7kkyc2#icomoon') format('svg');
  font-weight: normal;
  font-style: normal;
}
```

第二步：给盒子使用字体

```
span {
    font-family: "icomoon";
  }
```

第三步：盒子里面添加结构

```
span::before {
     content: "\e900";
  }
或者
<span>□</span>
```

追加新图标到原来库里面

如果工作中，原来的字体图标不够用了，我们需要添加新的字体图标，但是原来的不能删除，继续使用，此时我们需要这样做

把压缩包里面的selection.json 从新上传，然后，选中自己想要新的图标，从新下载压缩包，替换原来文件即可。

京东项目(一)

京东项目介绍

项目名称：京东网 项目描述：京东首页公共部分的头部和尾部制作，京东首页中间部分。

项目背景

现阶段电商类网站很流行，很多同学毕业之后会进入电商类企业工作，同时电商类网站需要的技术也是较为复杂的，这里用京东电商网站复习、总结、提高前面所学布局技术。其实，最主要的原因还是，为啥写京东？ 因为刘强东，赚了我们的钱，抢了我们的女神， 我们也要学刘强东，赚别人的钱，抢别人..额，自己的女神。。。

设计目标

- 保证浏览器 ie7及以上, 火狐, 360, safari，chrome等。谁让我再测ie6，就跟谁急。。
- 熟悉CSS+DIV布局，页面的搭建工作
- 了解常用电商类网站的布局模式
- 为后期京东移动端做铺垫

几点思考

(1). 开发工具 sublime 、fireworks（ps）、各种浏览器(ie6.7 要测看心情)

(2). CSS Reset 类库,为跨浏览器兼容做准备(也可以直接运用jd网站的初始化)

```
normalize.css    只是一个很小的CSS文件，但它在默认的HTML元素样式上提供了跨浏览器的高度一致性。相比于传统的CSS reset，Normalize.css是一种现代的、为HTML5准备的优质替代方案。Normalize.css现在已经被用于Twitter Bootstrap、HTML5 Boilerplate、GOV.UK、Rdio、CSS Tricks 以及许许多多其他框架、工具和网站上。 你值得拥有。。

- 保护有用的浏览器默认样式而不是完全去掉它们

- 一般化的样式：为大部分HTML元素提供

- 修复浏览器自身的bug并保证各浏览器的一致性

- 优化CSS可用性：用一些小技巧

- 解释代码：用注释和详细的文档来
```

(3). 技术栈

```
HTML 结构 + CSS  布局 （因为我们就会这些。。。嘻嘻）
```

(4). 低版本浏览器 单独制作一个跳转页面 (都是孩子，也舍不得打，舍不得扔)

https://h5.m.jd.com/dev/3dm8aE4LDBNMkDfcCaRxLnVQ7rqo/index.html

目录说明

要实现结构和样式相分离的设计思想。 根目录下有这4个文件（目录）。

名称	说明
css	用于存放CSS文件
images	用于存放图片
index	京东首页 HTML
js	用于后期存放javascript文件

运用知识点

引入ico图标

```
代码：  <link rel="shortcut icon" href="favicon.ico"  type="image/x-icon"/>
```

注意：

1. 她(它)不是iconfont字体哦 也不是图片。
2. 位置是放到 head 标签中间。
3. 后面的type="image/x-icon" 属性可以省略。（我相信你也愿意省略。）
4. 为了兼容性，请将favicon.ico 这个图标放到根目录下。（我们就不要任性了，听话放位置，省很多麻烦。。你好，我也好）

转换ico图标

我们可以自己做的图片，转换为 ico图标，以便放到我们站点里面。 http://www.bitbug.net/

网站优化三大标签

SEO是由英文Search Engine Optimization缩写而来， 中文意译为“搜索引擎优化”！SEO是指通过对网站进行站内优化、网站结构调整、网站内容建设、网站代码优化等)和站外优化，从而提高网站的关键词排名以及公司产品的曝光度。 简单的说就是，把产品做好，搜索引擎就会介绍客户来。

我们现在阶段主要进行站内优化。网站优化，我们应该要懂。。。

网页title 标题

title具有不可替代性，是我们的内页第一个重要标签，是搜索引擎了解网页的入口，和对网页主题归属的最佳判断点。

建议：

首页标题：网站名（产品名）- 网站的介绍

例如：

京东(JD.COM)-综合网购首选-正品低价、品质保障、配送及时、轻松购物！

小米商城 - 小米5s、红米Note 4、小米MIX、小米笔记本官方网站

Description 网站说明

对于关键词的作用明显降低，但由于很多搜索引擎，仍然大量采用网页的MATA标签中描述部分作为搜索结果的“内容摘要”。 就是简要说明我们网站的主要做什么的。 我们提倡，Description作为网站的总体业务和主题概括，多采用“我们是...”“我们提供...”“×××网作为...”“电话：010...”之类语句。

京东网：

```
<meta name="description" content="京东JD.COM-专业的综合网上购物商城,销售家电、数码通讯、电脑、家居百货、服装服饰、母婴、图书、食品等数万个品牌优质商品.便捷、诚信的服务，为您提供愉悦的网上购物体验！" />
```

注意点：

1. 描述中出现关键词，与正文内容相关，这部分内容是给人看的，所以要写的很详细，让人感兴趣，吸引用户点击。
2. 同样遵循简短原则，字符数含空格在内不要超过 120 个汉字。
3. 补充在 title 和 keywords 中未能充分表述的说明.
4. 用英文逗号 关键词1,关键词2

```
<meta name="description" content="小米商城直营小米公司旗下所有产品，囊括小米手机系列小米MIX、小米Note 2，红米手机系列红米Note 4、红米4，智能硬件，配件及小米生活周边，同时提供小米客户服务及售后支持。" />
```

Keywords 关键字

Keywords是页面关键词，是搜索引擎关注点之一。Keywords应该限制在6～8个关键词左右，电商类网站可以多 少许。

京东网：

```
<meta name="Keywords" content="网上购物,网上商城,手机,笔记本,电脑,MP3,CD,VCD,DV,相机,数码,配件,手表,存储卡,京东" />
```

小米网：

```
<meta name="keywords" content="小米,小米6,红米Note4,小米MIX,小米商城" />
```

顶部（快捷菜单）所用知识点

知识点	说明
通栏的盒子	不用给宽度 默认为 100% 但是加了浮动和定位的盒子需要 添加 100%
盒子居中对齐	margin: auto; 注意必须有宽度的块级元素，文字水平居中对齐是 text-align:center;
行高会继承	文字性质的，比如 颜色、文字大小、字体、行高等会继承父级元素
浮动元素、固定定位，绝对定位会模式转换	具有行内块特性，比如一行放多个，有高度和宽度，如果没有指定宽度，则会根据内容多少撑开。

logo 和搜索 header 区域所用知识点

网页布局稳定性

蓝色是宽度和高度，最安全

- 宽度和高度不用进行计算
- 测量简单
- 高度剩余法和宽度剩余法

青色是padding，其次之

- 需要加减计算，考验我们的小学数学水平

橙色是margin，最后之

- 外边距合并问题，低版本浏览器还有bug

如果都可用，则优先考虑宽度高度，其次考虑padding，最后是margin

宽度剩余法：

知识点	说明
浮动元素特性	1. 浮动可以让多个元素同一行显示 2. 浮动的元素是顶部对齐
logo优化	text-indent: -20000px; 隐藏文字， 背景图片
清除浮动	清除浮动的目的就是为了解决父亲高度为0的问题
鼠标样式	cursor: pointer; 小手 cursor: move; 四角箭头 cursor: text; 插入光标 cursor: default; 小白
不允许换行	white-space: nowrap;

nav导航栏所用知识点

名称	说明
边框底侧	border-bottom: 2px solid #ccc;
定位重点	绝对定位不占位置 相对定位占有位置
标签语义化dl	dl也是块级元素 dt 是 定义标题 dd 是定义描述，dd是围绕这dt来描述的，也就是说，dd算是dt 的解释说明详细分解。
标题标签h	尽量少用h1，可以多用h2和h3等标签

页面底部所用知识点

名称	说明
绝对定位的盒子居中对齐	盒子 left 50% 然后通过 margin 负值自己的宽度一半（固定定位也是如此）

固定定位的盒子靠近版心右侧对齐

跟绝对定位的盒子居中对齐原理差不多。

left 50% 然后 margin-left 版心宽度一半。

核心算法： 小盒子left 50%　然后 margin-left 版心宽度一半。

学习目标:

- 掌握京东中间部分制作
- 理解BFC使用
- 了解优雅降级和渐进增强
- 了解CSS压缩和验证工具

typora-copy-images-to: media

京东项目(二)

nav导航栏所用知识点

名称	说明
边框底侧	border-bottom: 2px solid #ccc;
定位重点	绝对定位不占位置 相对定位占有位置
标签语义化dl	dl也是块级元素 dt 是 定义标题 dd 是定义描述，dd是围绕这dt来描述的，也就是说，dd算是dt 的解释说明详细分解。
标题标签h	尽量少用h1，可以多用h2和h3等标签

固定定位的盒子靠近版心右侧对齐

跟绝对定位的盒子居中对齐原理差不多。

left 50% 然后 margin-left 版心宽度一半。

核心算法： 小盒子left 50%　然后 margin-left 版心宽度一半。

焦点图部分所用知识点

名称	说明
圆角矩形	border-radius: 左上角 右上角 右下角 左下角。

负值自己的宽度一半（固定定位也是如此）

背景半透明

1.强烈推荐： background: rgba(r,g,b,alpha);

r,g,b 是红绿蓝的颜色， alpha 是透明度的意思，取值范围是 0~1 之间。

2.了解ie低版本浏览器 半透明

filter:Alpha(opacity=50)； // opacity值为0 到 100

但是 此属性是盒子半透明，不是背景半透明哦，因为里面的内容也一起半透明了

因此，低版本的 ie6.7浏览器，我们不需要透明了，直接采用优雅降级的做法。

background: gary;

background: rgba(0,0,0,.2);

写上两句 背景， 低版本ie只执行gray， 其他浏览器执行 半透明下面这一句。

CSS W3C 统一验证工具

CssStats 是一个在线的 CSS 代码分析工具

```
网址是：   http://www.cssstats.com/
```

如果你想要更全面的，这个神奇，你值得拥有：

W3C 统一验证工具： http://validator.w3.org/unicorn/ ☆☆☆☆☆

因为它可以检测本地文件哦！！

http://tool.chinaz.com/Tools/CssFormat.aspx css 代码压缩

HTML5新标签与特性

兼容性问题 （ie9 以上的版本）

文档类型设定

- document
  - HTML: sublime 输入 html:4s
  - XHTML: sublime 输入 html:xt
  - HTML5 sublime 输入 html:5

字符设定

- ： HTML与XHTML中建议这样去写
- ： HTML5的标签中建议这样去写

常用新标签

w3c 手册中文官网 : http://w3school.com.cn/

- header：定义文档的页眉 头部
- nav：定义导航链接的部分
- footer：定义文档或节的页脚 底部
- article：定义文章。
- section：定义文档中的节（section、区段）
- aside：定义其所处内容之外的内容 侧边

```
<header> 语义 :定义页面的头部  页眉</header>
<nav>  语义 :定义导航栏 </nav>
<footer> 语义: 定义 页面底部 页脚</footer>
<article> 语义:  定义文章</article>
<section> 语义:  定义区域</section>
<aside> 语义:  定义其所处内容之外的内容 侧边</aside>
```

- datalist 标签定义选项列表。请与 input 元素配合使用该元素

```
<input type="text" value="请输入明星" list="star"/>
  <datalist id="star">
    <option value="刘德华">刘德华</option>
    <option value="刘若英">刘若英</option>
    <option value="刘晓庆">刘晓庆</option>
    <option value="戚薇">戚薇</option>
    <option value="戚继光">戚继光</option>
  </datalist>
```

- fieldset 元素可将表单内的相关元素分组，打包 legend 搭配使用

```
<fieldset>
        <legend>用户登录</legend>  标题
        用户名：<input type="text"><br /><br />
        密  码：<input type="password">
</fieldset>
```

新增的input type属性值：

类型****	使用示例****	含义****
email****		输入邮箱格式
tel****		输入手机号码格式
url****		输入url格式
number****		输入数字格式
search****		搜索框（体现语义化）
range****		自由拖动滑块
time****		小时分钟
date****		年月日
datetime****		时间
month****		月年
week****		星期 年

常用新属性

属性****	用法****	含义****
placeholder****	请输入用户名	占位符 当用户输入的时候 里面的文字消失 删除所有文字，自动返回
autofocus****		规定当页面加载时 input 元素应该自动获得焦点
multiple****	选取文件 未选择文件	多文件上传
autocomplete****		规定表单是否应该启用自动完成功能 有2个值，一个是on 一个是off on 代表记录已经输入的值 1.autocomplete 首先需要提交按钮 2.这个表单您必须给他名字
required****		必填项 内容不能为空
accesskey****		规定激活（使元素获得焦点）元素的快捷键 采用 alt + s的形式

综合案例

```
<form action="">
  <fieldset>
    <legend>学生档案</legend>
    <label for="userName">姓名:</label>
    <input type="text" name="userName" id="userName" placeholder="请输入用户名">
<br>
    <label for="userPhone">手机号码:</label>
    <input type="tel" name="userPhone" id="userPhone" pattern="^1\d{10}$"><br>
    <label for="email">邮箱地址:</label>
    <input type="email" required name="email" id="email"><br>
    <label for="collage">所属学院:</label>
    <input type="text" name="collage" id="collage" list="cList" placeholder="请
选择"><br>
    <datalist id="cList">
      <option value="前端与移动开发学院"></option>
      <option value="java学院"></option>
      <option value="c++学院"></option>
    </datalist><br>
    <label for="score">入学成绩:</label>
    <input type="number" max="100" min="0" value="0" id="score"><br>
   <form action="">
    <fieldset>
      <legend>学生档案思密达</legend>
      <label>姓名: <input type="text" placeholder="请输入学生名字"/></label> <br
/><br />
      <label>手机号: <input type="tel" /></label> <br /><br />
      <label>邮箱: <input type="email" /></label> <br /><br />
      <label>所属学院:  <input type="text" placeholder="请选择学院"
list="xueyuan"/>
      <datalist id="xueyuan">
        <option>java学院</option>
        <option>前端学院</option>
        <option>php学院</option>
        <option>设计学院</option>
      </datalist>

      <br /><br />

      <label>出生日期:   <input type="date" /></label> <br /><br />
      <label>成绩:  <input type="number" /></label> <br /><br />
      <label>毕业时间:  <input type="date" /></label> <br /><br />
      <input type="submit" />  <input type="reset" />
    </fieldset>
    </form>
    <label for="inTime">入学日期:</label>
    <input type="date" id="inTime" name="inTime"><br>
    <label for="leaveTime">毕业日期:</label>
    <input type="date" id="leaveTime" name="leaveTime"><br>
    <input type="submit">
```

```
  </fieldset>
</form>
```

多媒体标签

- embed：标签定义嵌入的内容
- audio：播放音频
- video：播放视频

多媒体 embed（会使用）

embed可以用来插入各种多媒体，格式可以是 Midi、Wav、AIFF、AU、MP3等等。url为音频或视频文件及其路径，可以是相对路径或绝对路径。

因为兼容性问题，我们这里只讲解 插入网络视频， 后面H5会讲解 audio 和video 视频多媒体。

```
<embed src="http://player.youku.com/player.php/sid/XMTI4MzM2MDIwOA==/v.swf"
allowFullScreen="true" quality="high" width="480" height="400" align="middle"
allowScriptAccess="always" type="application/x-shockwave-flash"></embed>
```

优酷，土豆，爱奇艺，腾讯、乐视等等

1. 先上传
2. 在分享

多媒体 audio

HTML5通过标签来解决音频播放的问题。

使用相当简单，如下图所示

```
<!-- 通过src指定音频文件路径即可 -->
<audio src="./music/See You Again.mp3"></audio>
```

并且可以通过附加属性可以更友好控制音频的播放，如：

autoplay 自动播放

controls 是否显不默认播放控件

loop 循环播放 如果这个属性不写 默认播放一次 loop 或者 loop = “loop” 表示无限循环

由于版权等原因，不同的浏览器可支持播放的格式是不一样的，如下图供参考

当前，audio 元素支持三种音频格式：

	IE 9	Firefox 3.5	Opera 10.5	Chrome 3.0	Safari 3.0
Ogg Vorbis		√	√	√	
MP3	√			√	√
Wav		√	√		√

多浏览器支持的方案，如下图

标签允许您规定可替换的视频/音频文件供浏览器根据它对媒体类型或者编解码器的支持进行选择

```
<audio controls>
    <!-- 通过source标签指定多格式音频文件 -->
    <source src="./music/See You Again.mp3">
    <source src="./music/See You Again.wav">
    <source src="./music/See You Again.ogg">
    您的浏览器不支持HTML音频播放功能
</audio>
```

多媒体 video

HTML5通过标签来解决音频播放的问题。

同音频播放一样，

使用也相当简单，如下图

```
<!-- 通过src属性指定视频文件的路路径 -->
<video src="./video/movie.mp4" controls="controls"></video>
```

同样，通过附加属性可以更友好的控制视频的播放

autoplay 自动播放

controls 是否显示默认播放控件

loop 循环播放

width 设置播放窗口宽度

height 设置播放窗口的高度

由于版权等原因，不同的浏览器可支持播放的格式是不一样的，如下图供参考

当前，video 元素支持三种视频格式：

格式	IE	Firefox	Opera	Chrome	Safari
Ogg	No	3.5+	10.5+	5.0+	No
MPEG 4	9.0+	No	No	5.0+	3.0+
WebM	No	4.0+	10.6+	6.0+	No

多浏览器支持的方案，如下图**

```
<video controls="controls">
    <!-- 通过source标签指定多格式视频文件 -->
    <source src="./video/movie.ogg">
    <source src="./video/movie.mp4">
    您的浏览器不支持HTML视频播放功能
</video>
```

CSS3 新增选择器

结构(位置)伪类选择器（CSS3)

- :first-child :选取属于其父元素的首个子元素的指定选择器
- :last-child :选取属于其父元素的最后一个子元素的指定选择器
- :nth-child(n)： 匹配属于其父元素的第 N 个子元素，不论元素的类型
- :nth-last-child(n)： 选择器匹配属于其元素的第 N 个子元素的每个元素，不论元素的类型，从最后一个子元素开始计数。 n 可以是数字、关键词或公式
-

```
li:first-child { /*  选择第一个孩子 */
            color: pink;
          }
li:last-child {   /* 最后一个孩子 */
            color: purple;
          }
li:nth-child(4) {   /* 选择第4个孩子  n  代表 第几个的意思 */
        color: skyblue;
          }
```

属性选择器

选取标签带有某些特殊属性的选择器 我们成为属性选择器

```
/* 获取到 拥有 该属性的元素 */
div[class^=font] { /*  class^=font 表示 font 开始位置就行了 */
      color: pink;
    }
div[class$=footer] { /*  class$=footer 表示 footer 结束位置就行了 */
      color: skyblue;
    }
div[class*=tao] { /* class*=tao  *=  表示tao 在任意位置都可以 */
      color: green;
    }
```

```
<div class="font12">属性选择器</div>
    <div class="font12">属性选择器</div>
    <div class="font24">属性选择器</div>
    <div class="font24">属性选择器</div>
    <div class="font24">属性选择器</div>
    <div class="24font">属性选择器123</div>
    <div class="sub-footer">属性选择器footer</div>
    <div class="jd-footer">属性选择器footer</div>
    <div class="news-tao-nav">属性选择器</div>
    <div class="news-tao-header">属性选择器</div>
    <div class="tao-header">属性选择器</div>
```

伪元素选择器（CSS3)

1. E::first-letter文本的第一个单词或字（如中文、日文、韩文等）
2. E::first-line 文本第一行；
3. E::selection 可改变选中文本的样式；

```
p::first-letter {
  font-size: 20px;
  color: hotpink;
}

/* 首行特殊样式 */
p::first-line {
  color: skyblue;
}

p::selection {
  /* font-size: 50px; */
  color: orange;
}
```

4、E::before和E::after

在E元素内部的开始位置和结束位创建一个元素，该元素为行内元素，且必须要结合content属性使用。

```
div::befor {
  content:"开始";
}
div::after {
  content:"结束";
}
```

E:after、E:before 在旧版本里是伪元素，CSS3的规范里“:”用来表示伪类，“::”用来表示伪元素，但是在高版本浏览器下E:after、E:before会被自动识别为E::after、E::before，这样做的目的是用来做兼容处理。

":" 与 "::" 区别在于区分伪类和伪元素

之所以被称为伪元素，是因为他们不是真正的页面元素，html没有对应的元素，但是其所有用法和表现行为与真正的页面元素一样，可以对其使用诸如页面元素一样的css样式，表面上看上去貌似是页面的某些元素来展现，实际上是css样式展现的行为，因此被称为伪元素。是伪元素在html代码机构中的展现，可以看出无法伪元素的结构无法审查

注意

伪元素:before和:after添加的内容默认是inline元素**；这个两个伪元素的`content`属性，表示伪元素的内容,设置:before和:after时必须设置其`content`属性，否则伪元素就不起作用。

CSS3盒模型

CSS3中可以通过box-sizing 来指定盒模型，即可指定为content-box、border-box，这样我们计算盒子大小的方式就发生了改变。

可以分成两种情况：

1、box-sizing: content-box 盒子大小为 width + padding + border content-box:此值为其默认值，其让元素维持W3C的标准Box Mode

2、box-sizing: border-box 盒子大小为 width 就是说 padding 和 border 是包含到width里面的

注：上面的标注的width指的是CSS属性里设置的width: length，content的值是会自动调整的。

```
div:first-child {
      width: 200px;
      height: 200px;
      background-color: pink;
      box-sizing: content-box;  /*  就是以前的标准盒模型  w3c */
      padding: 10px;
      border: 15px solid red;
      /* 盒子大小为 width + padding + border   content-box:此值为其默认值，其让元素维持W3C的标准Box Mode */
    }
    div:last-child {
      width: 200px;
      height: 200px;
      background-color: purple;
      padding: 10px;
```

```
        box-sizing: border-box;   /* padding border  不撑开盒子 */
        border: 15px solid red;
        /* margin: 10px; */
        /* 盒子大小为 width    就是说  padding 和 border 是包含到width里面的 */
}
```

学成在线综合案例

过渡(CSS3)

过渡（transition)是CSS3中具有颠覆性的特征之一，我们可以在不使用 Flash 动画或 JavaScript 的情况下，当元素从一种样式变换为另一种样式时为元素添加效果。

帧动画：通过一帧一帧的画面按照固定顺序和速度播放。如电影胶片

在CSS3里使用transition可以实现补间动画（过渡效果），并且当前元素只要有“属性”发生变化时即存在两种状态(我们用A和B代指），就可以实现平滑的过渡，为了方便演示采用hover切换两种状态，但是并不仅仅局限于hover状态来实现过渡。

语法格式:

```
transition: 要过渡的属性   花费时间   运动曲线   何时开始;
如果有多组属性变化，还是用逗号隔开。
```

属性	描述	CSS
transition	简写属性，用于在一个属性中设置四个过渡属性。	3
transition-property	规定应用过渡的 CSS 属性的名称。	3
transition-duration	定义过渡效果花费的时间。默认是 0。	3
transition-timing-function	规定过渡效果的时间曲线。默认是 "ease"。	3
transition-delay	规定过渡效果何时开始。默认是 0。	3

如果想要所有的属性都变化过渡， 写一个all 就可以

transition-duration 花费时间 单位是 秒 s 比如 0.5s 这个s单位必须写 ms 毫秒

运动曲线 默认是 ease

何时开始 默认是 0s 立马开始

运动曲线示意图：

```
div {
      width: 200px;
      height: 100px;
      background-color: pink;
      /* transition: 要过渡的属性   花费时间   运动曲线   何时开始; */
      transition: width 0.6s ease 0s, height 0.3s ease-in 1s;
      /* transtion 过渡的意思   这句话写到div里面而不是 hover里面 */


}
div:hover {  /* 鼠标经过盒子，我们的宽度变为400 */

      width: 600px;
```

```
        height: 300px
}

transition: all 0.6s;  /* 所有属性都变化用all 就可以了  后面俩个属性可以省略 */
```

2D变形(CSS3) transform

transform是CSS3中具有颠覆性的特征之一，可以实现元素的位移、旋转、倾斜、缩放，甚至支持矩阵方式，配合过渡和即将学习的动画知识，可以取代大量之前只能靠Flash才可以实现的效果。

变形转换 transform transform 变换 变形的意思 《 transformers 变形金刚》

移动 translate(x, y)

translate 移动平移的意思

```
translate(50px,50px);
```

使用translate方法来将文字或图像在水平方向和垂直方向上分别垂直移动50像素。

可以改变元素的位置，x、y可为负值；

```
 translate(x,y)水平方向和垂直方向同时移动（也就是x轴和Y轴同时移动）
 translateX(x)仅水平方向移动（x轴移动）
 translateY(Y)仅垂直方向移动（Y轴移动）
```

```
.box {
  width: 499.9999px;
  height: 400px;
  background: pink;
  position: absolute;
  left:50%;
  top:50%;
  transform:translate(-50%,-50%);  /* 走的自己的一半 */
}
```

让定位的盒子水平居中

缩放 scale(x, y)

```
transform:scale(0.8,1);
```

可以对元素进行水平和垂直方向的缩放。该语句使用scale方法使该元素在水平方向上缩小了20%，垂直方向上不缩放。

```
scale(X,Y)使元素水平方向和垂直方向同时缩放（也就是X轴和Y轴同时缩放）
scaleX(x)元素仅水平方向缩放（X轴缩放）
scaleY(y)元素仅垂直方向缩放（Y轴缩放）
```

scale()的取值默认的值为1，当值设置为0.01到0.99之间的任何值，作用使一个元素缩小；而任何大于或等于1.01的值，作用是让元素放大

旋转 rotate(deg)

可以对元素进行旋转，正值为顺时针，负值为逆时针；

```
transform:rotate(45deg);
```

注意单位是 deg 度数

transform-origin可以调整元素转换变形的原点

```
 div{transform-origin: left top;transform: rotate(45deg); }  /* 改变元素原点到左上角，然后进行顺时旋转45度 */
```

如果是4个角，可以用 left top这些，如果想要精确的位置， 可以用 px 像素。

```
 div{transform-origin: 10px 10px;transform: rotate(45deg); }  /* 改变元素原点到x为10  y 为10，然后进行顺时旋转45度 */
```

案例旋转楚乔传

```
div {
      width: 250px;
      height: 170px;
      border: 1px solid pink;
      margin: 200px auto;
      position: relative;

    }
    div img {
      width: 100%;
      height: 100%;
      position: absolute;
      top: 0;
      left: 0;
      transition: all 0.6s;
      transform-origin: top right;

    }
    div:hover img:nth-child(1) {  /* 鼠标经过div  第一张图片旋转 */
      transform: rotate(60deg);
    }
    div:hover img:nth-child(2) {
      transform: rotate(120deg);
```

```
    }
    div:hover img:nth-child(3) {
      transform: rotate(180deg);
    }
    div:hover img:nth-child(4) {
      transform: rotate(240deg);
    }
    div:hover img:nth-child(5) {
      transform: rotate(300deg);
    }
    div:hover img:nth-child(6) {
      transform: rotate(360deg);
    }
```

倾斜 skew(deg, deg)

```
transform:skew(30deg,0deg);
```

该实例通过skew方法把元素水平方向上倾斜30度，处置方向保持不变。

可以使元素按一定的角度进行倾斜，可为负值，第二个参数不写默认为0。

3D变形(CSS3) transform

2d x y

3d x y z

左手坐标系

伸出左手，让拇指和食指成“L”形，大拇指向右，食指向上，中指指向前方。这样我们就建立了一个左手坐标系，拇指、食指和中指分别代表X、Y、Z轴的正方向。如下图

CSS3中的3D坐标系与上述的3D坐标系是有一定区别的，相当于其绕着X轴旋转了180度，如下图

简单记住他们的坐标：

x左边是负的，右边是正的

y 上面是负的， 下面是正的

z 里面是负的， 外面是正的

rotateX()

就是沿着 x 立体旋转.

```
img {
  transition:all 0.5s ease 0s;
}
img:hove {

  transform:rotateX(180deg);
}
```

rotateY()

沿着y轴进行旋转

```
img {
  transition:all 0.5s ease 0s;
}
img:hove {

  transform:rotateX(180deg);
}
```

rotateZ()

沿着z轴进行旋转

```
img {
  transition:all .25s ease-in 0s;
}
img:hover {
  /* transform:rotateX(180deg); */
  /* transform:rotateY(180deg); */
  /* transform:rotateZ(180deg); */
  /* transform:rotateX(45deg) rotateY(180deg) rotateZ(90deg) skew(0,10deg); */
}
```

透视(perspective)

电脑显示屏是一个2D平面，图像之所以具有立体感（3D效果），其实只是一种视觉呈现，通过透视可以实现此目的。

透视可以将一个2D平面，在转换的过程当中，呈现3D效果。

- 透视原理： 近大远小 。
- 浏览器透视：把近大远小的所有图像，透视在屏幕上。
- perspective：视距，表示视点距离屏幕的长短。视点，用于模拟透视效果时人眼的位置

注：并非任何情况下需要透视效果，根据开发需要进行设置。

perspective 一般作为一个属性，设置给父元素，作用于所有3D转换的子元素

理解透视距离原理：

translateX(x)

仅水平方向移动**（X轴移动）

主要目的实现移动效果

translateY(y)

仅垂直方向移动（Y轴移动）

translateZ(z)

transformZ的直观表现形式就是大小变化，实质是XY平面相对于视点的远近变化（说远近就一定会说到离什么参照物远或近，在这里参照物就是perspective属性）。比如设置了perspective为200px;那么transformZ的值越接近200，就是离的越近，看上去也就越大，超过200就看不到了，因为相当于跑到后脑勺去了，我相信你正常情况下，是看不到自己的后脑勺的。

translate3d(x,y,z)

[注意]其中，x和y可以是长度值，也可以是百分比，百分比是相对于其本身元素水平方向的宽度和垂直方向的高度和；z只能设置长度值

开门案例

```
body {
}
```

```
.door {
  width: 300px;
  height: 300px;
  margin: 100px auto;
  border: 1px solid gray;
  perspective: 1000px;
  background: url('images/dog.gif') no-repeat cover;
  position: relative;
}
.door > div {
  box-sizing: border-box;
  border: 1px solid black;
}
.left {
  float: left;
  width: 50%;
  height: 100%;
  background-color: brown;
  transform-origin: left center;
  transition: 1s;
  position: relative;
}
.left::before {
  content: '';
  position: absolute;
  width: 20px;
  height: 20px;
  border-radius: 50%;
  top: 50%;
  right: 0px;
  transform: translateY(-10px);
  border: 1px solid whitesmoke;
}
.right {
  width: 50%;
  height: 100%;
  float: left;
  background-color: brown;
  transform-origin: right center;
  transition: 1s;
  position: relative;
}
.right::before {
  content: '';
  position: absolute;
  width: 20px;
  height: 20px;
  border-radius: 50%;
  top: 50%;
```

```
  left: 0px;
  transform: translateY(-10px);
  border: 1px solid whitesmoke;
}
.door:hover .left {
  transform: rotateY(-130deg);
}
.door:hover .right {
  transform: rotateY(130deg);
}
```

backface-visibility

backface-visibility 属性定义当元素不面向屏幕时是否可见。

翻转盒子案例

```
div {
      width: 224px;
      height: 224px;
      margin: 100px auto;
      position: relative;
    }
    div img {
      position: absolute;
      top: 0;
      left: 0;
      transition: all 1s;
    }
    div img:first-child {
      z-index: 1;
      backface-visibility: hidden; /* 不是正面对象屏幕，就隐藏 */
    }
    div:hover img {
      transform: rotateY(180deg);
}
```

动画(CSS3) animation

动画是CSS3中具有颠覆性的特征之一，可通过设置多个节点来精确控制一个或一组动画，常用来实现复杂的动画效果。

语法格式：

```
animation:动画名称  动画时间  运动曲线   何时开始   播放次数   是否反方向;
```

CSS3 动画属性

下面的表格列出了 @keyframes 规则和所有动画属性：

属性	描述	CSS
@keyframes	规定动画。	3
animation	所有动画属性的简写属性，除了 animation-play-state 属性。	3
animation-name	规定 @keyframes 动画的名称。	3
animation-duration	规定动画完成一个周期所花费的秒或毫秒。默认是 0。	3
animation-timing-function	规定动画的速度曲线。默认是 "ease"。	3
animation-delay	规定动画何时开始。默认是 0。	3
animation-iteration-count	规定动画被播放的次数。默认是 1。	3
animation-direction	规定动画是否在下一周期逆向地播放。默认是 "normal"。	3
animation-play-state	规定动画是否正在运行或暂停。默认是 "running"。	3
animation-fill-mode	规定对象动画时间之外的状态。	3

关于几个值，除了名字，动画时间，延时有严格顺序要求其它随意r

```
@keyframes 动画名称 {
  from{ 开始位置 }  0%
  to{  结束   }  100%
}
```

```
animation-iteration-count:infinite;  无限循环播放
animation-play-state:paused;   暂停动画"
```

小汽车案例

```
body {
  background: white;
}
img {
  width: 200px;
}
.animation {
  animation-name: goback;
  animation-duration: 5s;
  animation-timing-function: ease;
  animation-iteration-count: infinite;
}
@keyframes goback {
  0%{}
  49%{
    transform: translateX(1000px);
```

```
    }
    55%{
      transform: translateX(1000px) rotateY(180deg);
    }
    95%{
      transform: translateX(0) rotateY(180deg);
    }
    100%{
      transform: translateX(0) rotateY(0deg);
    }
  }
```

伸缩布局(CSS3)

CSS3在布局方面做了非常大的改进，使得我们对块级元素的布局排列变得十分灵活，适应性非常强，其强大的伸缩性，在响应式开中可以发挥极大的作用。

主轴：Flex容器的主轴主要用来配置Flex项目，默认是水平方向

侧轴：与主轴垂直的轴称作侧轴，默认是垂直方向的

方向：默认主轴从左向右，侧轴默认从上到下

主轴和侧轴并不是固定不变的，通过flex-direction可以互换。

Flex布局的语法规范经过几年发生了很大的变化，也给Flexbox的使用带来一定的局限性，因为语法规范版本众多，浏览器支持不一致，致使Flexbox布局使用不多

2、各属性详解**

1.flex子项目在主轴的缩放比例，不指定flex属性，则不参与伸缩分配

min-width 最小值 min-width: 280px 最小宽度 不能小于 280

max-width: 1280px 最大宽度 不能大于 1280

2.flex-direction调整主轴方向（默认为水平方向）

flex-direction: column 垂直排列

flex-direction: row 水平排列

http://m.ctrip.com/html5/ 携程网手机端地址

3、justify-content调整主轴对齐（水平对齐）

子盒子如何在父盒子里面水平对齐

值	描述	白话文
flex-start	默认值。项目位于容器的开头。	让子元素从父容器的开头开始排序但是盒子顺序不变
flex-end	项目位于容器的结尾。	让子元素从父容器的后面开始排序但是盒子顺序不变
center	项目位于容器的中心。	让子元素在父容器中间显示
space-between	项目位于各行之间留有空白的容器内。	左右的盒子贴近父盒子，中间的平均分布空白间距
space-around	项目位于各行之前、之间、之后都留有空白的容器内。	相当于给每个盒子添加了左右margin外边距

4、align-items调整侧轴对齐（垂直对齐）

子盒子如何在父盒子里面垂直对齐（单行）

值	描述	白话文
stretch	默认值。项目被拉伸以适应容器。	让子元素的高度拉伸适用父容器（子元素不给高度的前提下)
center	项目位于容器的中心。	垂直居中
flex-start	项目位于容器的开头。	垂直对齐开始位置 上对齐
flex-end	项目位于容器的结尾。	垂直对齐结束位置 底对齐

5、flex-wrap控制是否换行

当我们子盒子内容宽度多于父盒子的时候如何处理

值	描述
nowrap	默认值。规定灵活的项目不拆行或不拆列。 不换行，则 收缩（压缩） 显示 强制一行内显示
wrap	规定灵活的项目在必要的时候拆行或拆列。
wrap-reverse	规定灵活的项目在必要的时候拆行或拆列，但是以相反的顺序。

6、flex-flow是flex-direction、flex-wrap的简写形式

```
flex-flow: flex-direction  flex-wrap;
```

白话记： flex-flow: 排列方向 换不换行;

两个中间用空格

例如：

```
display: flex;
/* flex-direction: row;
flex-wrap: wrap;   这两句话等价于下面的这句话*/
flex-flow: column wrap;  /* 两者的综合 */
```

7、align-content堆栈（由flex-wrap产生的独立行）多行垂直对齐方式齐

align-content是针对flex容器里面多轴(多行)的情况,align-items是针对一行的情况进行排列。

必须对父元素设置自由盒属性display:flex;，并且设置排列方式为横向排列flex-direction:row;并且设置换行，flex-wrap:wrap;这样这个属性的设置才会起作用。

值	描述	测试
stretch	默认值。项目被拉伸以适应容器。	
center	项目位于容器的中心。	
flex-start	项目位于容器的开头。	
flex-end	项目位于容器的结尾。	
space-between	项目位于各行之间留有空白的容器内。	
space-around	项目位于各行之前、之间、之后都留有空白的容器内。	

8、order控制子项目的排列顺序，正序方式排序，从小到大

用css 来控制盒子的前后顺序。 用order 就可以

用整数值来定义排列顺序，数值小的排在前面。可以为负值。 默认值是 0

```
order: 1;
```

此知识点重在理解，要明确找出主轴、侧轴、方向，各属性对应的属性值

文字阴影(CSS3)

以后我们可以给我们的文字添加阴影效果了 Shadow 影子

```
text-shadow:水平位置 垂直位置 模糊距离 阴影颜色;
```

值	描述
h-shadow	必需。水平阴影的位置。允许负值。
v-shadow	必需。垂直阴影的位置。允许负值。
blur	可选。模糊的距离。
color	可选。阴影的颜色。参阅 CSS 颜色值。

1. 前两项是必须写的。 后两项可以选写。

BFC(块级格式化上下文)

BFC(Block formatting context)

直译为"块级格式化上下文"。

元素的显示模式

我们前面讲过 元素的显示模式 display。

分为 块级元素 行内元素 行内块元素 ， 其实，它还有很多其他显示模式。

display 版本：CSS1/CSS

语法：

display : block | none | inline | compact | marker | inli

参数：

block : CSS1 块对象的默认值。用该值为对象之后添加新行
none : CSS1 隐藏对象。与visibility属性的hidden值不同，
inline : CSS1 内联对象的默认值。用该值将从对象中删除行
compact : CSS2 分配对象为块对象或基于内容之上的内联对
marker : CSS2 指定内容在容器对象之前或之后。要使用此
inline-table : CSS2 将表格显示为无前后换行的内联对象或
list-item : CSS1 将块对象指定为列表项目。并可以添加可选
run-in : CSS2 分配对象为块对象或基于内容之上的内联对象
table : CSS2 将对象作为块元素级的表格显示
table-caption : CSS2 将对象作为表格标题显示
table-cell : CSS2 将对象作为表格单元格显示
table-column : CSS2 将对象作为表格列显示
table-column-group : CSS2 将对象作为表格列组显示
table-header-group : CSS2 将对象作为表格标题组显示
table-footer-group : CSS2 将对象作为表格脚注组显示
table-row : CSS2 将对象作为表格行显示
table-row-group : CSS2 将对象作为表格行组显示

那些元素会具有BFC的条件

不是所有的元素模式都能产生BFC，w3c 规范：

display 属性为 block, list-item, table 的元素，会产生BFC.

大家有么有发现这个三个都是用来布局最为合理的元素，因为他们就是用来可视化布局。

注意其他的，display属性，比如 line 等等，他们创建的是 IFC ，我们暂且不研究。

这个BFC 有着具体的布局特性：

有宽度和高度， 有 外边距margin 有内边距padding 有边框 border。

就好比，你有了练习武术的体格了。 有潜力，有资质。

什么情况下可以让元素产生BFC

以上盒子具有BFC条件了，就是说有资质了，但是怎样触发才会产生BFC，从而创造这个封闭的环境呢?

在好比，你光有资质还不行，你需要一定额外效果才能出发的武学潜力，要么你掉到悬崖下面，捡到了一本九阴真经，要么你学习葵花宝典，欲练此功必先....

同样，要给这些元素添加如下属性就可以触发BFC。

-float属性不为none

-position为absolute或fixed

-display为inline-block, table-cell, table-caption, flex, inline-flex

-overflow不为visible。

BFC元素所具有的特性

BFC布局规则特性：

1.在BFC中，盒子从顶端开始垂直地一个接一个地排列.

2.盒子垂直方向的距离由margin决定。属于同一个BFC的两个相邻盒子的margin会发生重叠

3.在BFC中，每一个盒子的左外边缘（margin-left）会触碰到容器的左边缘(border-left)（对于从右到左的格式来说，则触碰到右边缘）。

1. BFC的区域不会与浮动盒子产生交集，而是紧贴浮动边缘。
2. 计算BFC的高度时，自然也会检测浮动或者定位的盒子高度。

它是一个独立的渲染区域，只有Block-level box参与， 它规定了内部的Block-level Box如何布局，并且与这个区域外部毫不相干。

白话文： 孩子在家里愿意怎么折腾都行，但是出了家门口，你就的乖乖的，不能影响外面的任何人。

BFC的主要用途

BFC能用来做什么?

(1) 清除元素内部浮动

只要把父元素设为BFC就可以清理子元素的浮动了，最常见的用法就是在父元素上设置overflow:hidden样式，对于IE6加上zoom:1就可以了。

主要用到

```
计算BFC的高度时，自然也会检测浮动或者定位的盒子高度。
```

(2) 解决外边距合并问题

外边距合并的问题。

主要用到

```
盒子垂直方向的距离由margin决定。属于同一个BFC的两个相邻盒子的margin会发生重叠
```

属于同一个BFC的两个相邻盒子的margin会发生重叠，那么我们创建不属于同一个BFC，就不会发生margin重叠了。

(3) 制作右侧自适应的盒子问题

主要用到

```
普通流体元素BFC后，为了和浮动元素不产生任何交集，顺着浮动边缘形成自己的封闭上下文
```

BFC 总结

BFC就是页面上的一个隔离的独立容器，容器里面的子元素不会影响到外面的元素。反之也如此。包括浮动，和外边距合并等等，因此，有了这个特性，我们布局的时候就不会出现意外情况了。

优雅降级和渐进增强

什么是渐进增强（progressive enhancement）、优雅降级（graceful degradation）呢?

渐进增强 progressive enhancement：

针对低版本浏览器进行构建页面，保证最基本的功能，然后再针对高级浏览器进行效果、交互等改进和追加功能达到更好的用户体验。

类似 爬山，由低出往高处爬

优雅降级 graceful degradation：

一开始就构建完整的功能，然后再针对低版本浏览器进行兼容。

类似蹦极，由高处往低处下落

区别：渐进增强是向上兼容，优雅降级是向下兼容。

个人建议： 现在互联网发展很快， 连微软公司都抛弃了ie浏览器，转而支持 edge这样的高版本浏览器，我们很多情况下没有必要再时刻想着低版本浏览器了，而是一开始就构建完整的效果，根据实际情况，修补低版本浏览器问题。

浏览器前缀

浏览器前缀	浏览器
-webkit-	Google Chrome, Safari, Android Browser
-moz-	Firefox
-o-	Opera
-ms-	Internet Explorer, Edge
-khtml-	Konqueror

后面我们会有 常用的解决H5和C3 的兼容解决文件， 我们这里暂且不涉及。

背景渐变

在线性渐变过程中，颜色沿着一条直线过渡：从左侧到右侧、从右侧到左侧、从顶部到底部、从底部到顶部或着沿任何任意轴。如果你曾使用过制作图件，比如说Photoshop，你对线性渐变并不会陌生。

兼容性问题很严重，我们这里之讲解线性渐变

语法格式：

```
background:-webkit-linear-gradient(渐变的起始位置，  起始颜色，  结束颜色);
```

```
background:-webkit-linear-gradient(渐变的起始位置，  颜色  位置，  颜色位置....);
```

背景缩放(CSS3)

通过background-size设置背景图片的尺寸，就像我们设置img的尺寸一样，在移动Web开发中做屏幕适配应用非常广泛。

其参数设置如下：

a) 可以设置长度单位(px)或百分比（设置百分比时，参照盒子的宽高）

b) 设置为cover时，会自动调整缩放比例，保证图片始终填充满背景区域，如有溢出部分则会被隐藏。我们平时用的cover 最多

c) 设置为contain会自动调整缩放比例，保证图片始终完整显示在背景区域。

```
background-image: url('images/gyt.jpg');
      background-size: 300px 100px;
      /* background-size: contain; */
      /* background-size: cover; */
```

多背景(CSS3)

以逗号分隔可以设置多背景，可用于自适应布局 做法就是 用逗号隔开就好了。

- 一个元素可以设置多重背景图像。
- 每组属性间使用逗号分隔。
- 如果设置的多重背景图之间存在着交集（即存在着重叠关系），前面的背景图会覆盖在后面的背景图之上。
- 为了避免背景色将图像盖住，背景色通常都定义在最后一组上，

```
background:url(test1.jpg) no-repeat scroll 10px 20px/50px 60px  ,
     url(test1.jpg) no-repeat scroll 10px 20px/70px 90px ,
     url(test1.jpg) no-repeat scroll 10px 20px/110px 130px c #aaa;
```