> 第01阶段.前端基础

CSS 第二天

今天我们围绕一个 导航栏案例进行学习知识点。

今日重点：

- 复合选择器
  - 后代选择器
  - 并集选择器
- 标签显示模式
- CSS背景
  - 背景位置
- CSS三大特性
  - 优先级

1. CSS复合选择器

目标

- 理解
  - 理解css复合选择器分别的应用场景
- 应用
  - 使用后代选择器给元素添加样式
  - 使用并集选择器给元素添加样式
  - 使用伪类选择器

为什么要学习css复合选择器

CSS选择器分为 基础选择器 和 复合选择器，但是基础选择器不能满足我们实际开发中，快速高效的选择标签。

- 目的是为了可以选择更准确更精细的目标元素标签。
- 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

1.1 后代选择器（重点）

- 概念：

  后代选择器又称为包含选择器

- 作用：

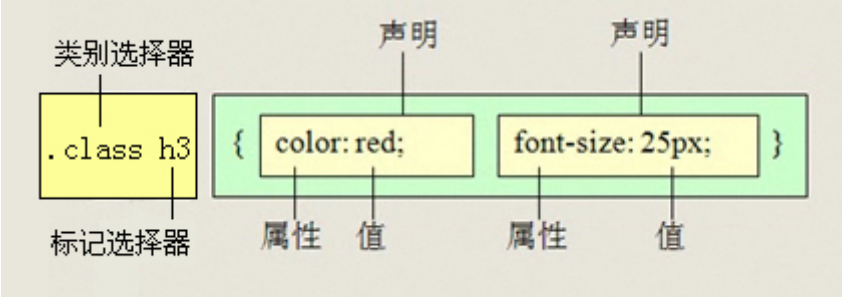
  用来选择元素或元素组的**子孙后代**

- 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，先写父亲爷爷，在写儿子孙子。

```
父级 子级{属性:属性值;属性:属性值;}
```

- 语法：

```
.class h3{color:red;font-size:16px;}
```



- 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。

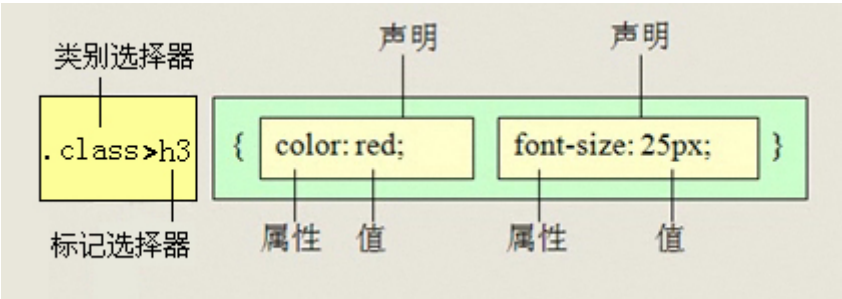
1.2 子元素选择器

- 作用：

  子元素选择器只能选择作为某元素**子元素(亲儿子)**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
- 语法：

```
.class>h3{color:red;font-size:14px;}
```



pink老师一句话说出他们

> 这里的子 指的是 亲儿子 不包含孙子 重孙子之类。

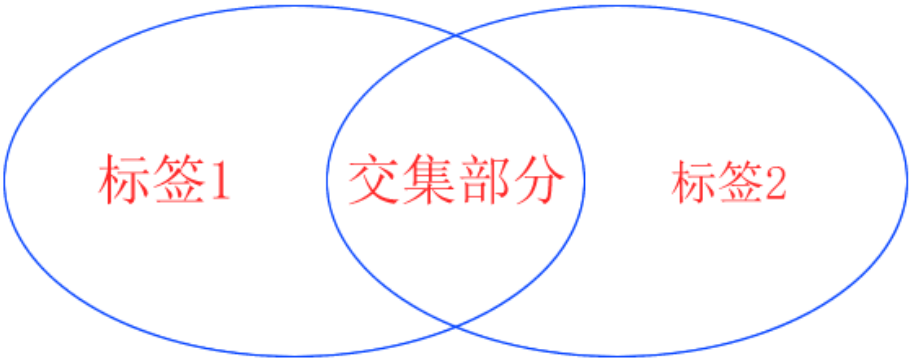
白话：

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

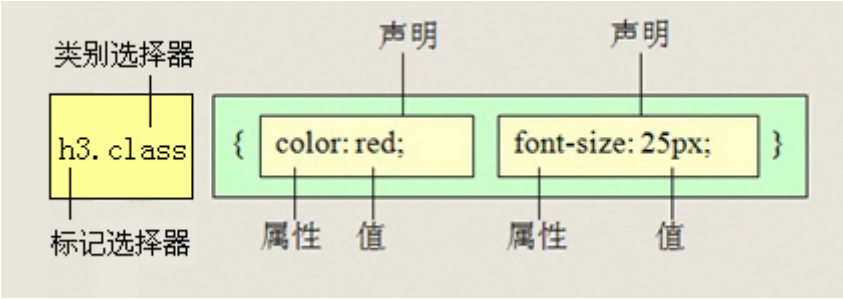
1.3 交集选择器

- 条件

  交集选择器由两个选择器构成，找到的标签必须满足：既有标签一的特点，也有标签二的特点。



- 语法：



- 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：

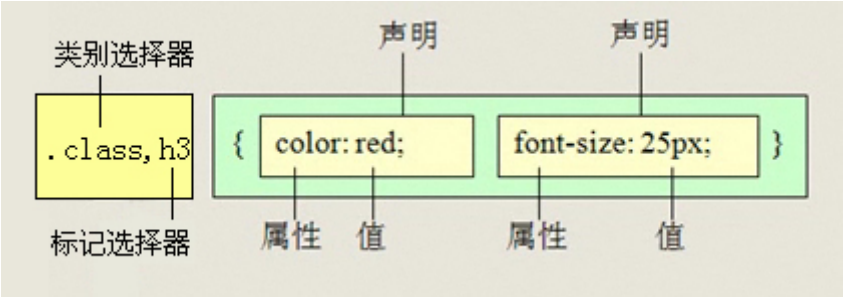
交集选择器 是 并且的意思。 即...又...的意思

```
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。

1.4 并集选择器（重点）

- 应用：
  - 如果某些选择器定义的相同样式，就可以利用并集选择器，可以让代码更简洁。
- 并集选择器（CSS选择器分组）是各个选择器通过`,`连接而成的，通常用于集体声明。
- 语法：



- 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。
- 记忆技巧：

  并集选择器通常用于集体声明 ，逗号隔开的，所有选择器都会执行后面样式，逗号可以理解为 和 的意思。

```
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。
```


他和他，在一起， 在一起 一起的意思

测试题

```
 <!-- 主导航栏 -->
<div class="nav">
  <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></li>
  </ul>
</div>
<!-- 侧导航栏 -->
<div class="sitenav">
  <div class="site-l">左侧侧导航栏</div>
  <div class="site-r"><a href="#">登录</a></div>
</div>
```

在不修改以上结构代码的前提下，完成以下任务：

1. 链接 登录 的颜色为红色
2. 主导航栏里面的所有的链接改为橙色

3. 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。

1.5 链接伪类选择器（重点）

伪类选择器：

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{} 伪娘

作用：

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。

因为伪类选择器很多，比如链接伪类，结构伪类等等。我们这里先给大家讲解链接伪类选择器。

```
```

- a:link /* 未访问的链接 */
- a:visited /* 已访问的链接 */
- a:hover /* 鼠标移动到链接上 */
- a: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。否则可能引起错误。
- 记忆法
  - **l**o**v**e **ha**te 爱上了讨厌
  - **lv** 包包 非常 **ha**o
- 因为叫链接伪类，所以都是 利用交集选择器 a:link a:hover
- 因为a链接浏览器具有默认样式，所以我们实际工作中都需要给链接单独指定样式。
- 实际工作开发中，我们很少写全四个状态，一般我们写法如下：

```
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 */
}
```

1.6 复合选择器总结

选择器	作用	特征	使用情况	隔开符号及用法
后代选择器	用来选择元素后代	是选择所有的子孙后代	较多	符号是**空格** .nav a
子代选择器	选择 最近一级元素	只选亲儿子	较少	符号是**>** .nav>p
交集选择器	选择两个标签交集的部分	既是 又是	较少	**没有符号** p.one
并集选择器	选择某些相同样式的选择器	可以用于集体声明	较多	符号是**逗号** .nav, .header
链接伪类选择器	给链接更改状态		较多	重点记住 a{} 和 a:hover 实际开发的写法

2. 标签显示模式（display）重点

目标：

- 理解
  - 标签的三种显示模式
  - 三种显示模式的特点以及区别
  - 理解三种显示模式的相互转化
- 应用
  - 实现三种显示模式的相互转化

2.1 什么是标签显示模式

- 什么是标签的显示模式?

  标签以什么方式进行显示，比如div 自己占一行， 比如span 一行可以放很多个
- 作用：

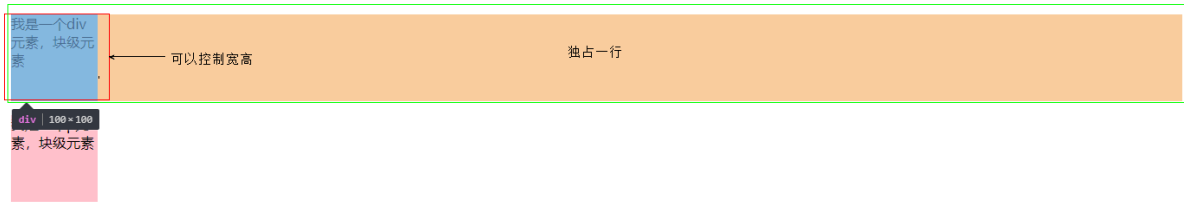
  我们网页的标签非常多，再不同地方会用到不同类型的标签，以便更好的完成我们的网页。
- 标签的类型(分类)

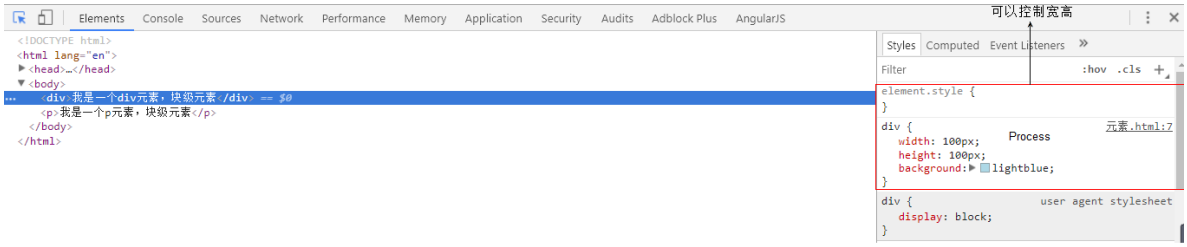
  HTML标签一般分为块标签和行内标签两种类型，它们也称块元素和行内元素。

2.2 块级元素(block-level)

- 例：

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





- 块级元素的特点

(1) 比较霸道，自己独占一行

(2) 高度，宽度、外边距以及内边距都可以控制。

(3) 宽度默认是容器（父级宽度）的100%

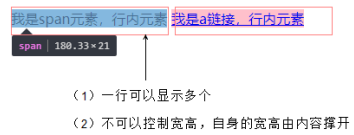
(4) 是一个容器及盒子，里面可以放行内或者块级元素。

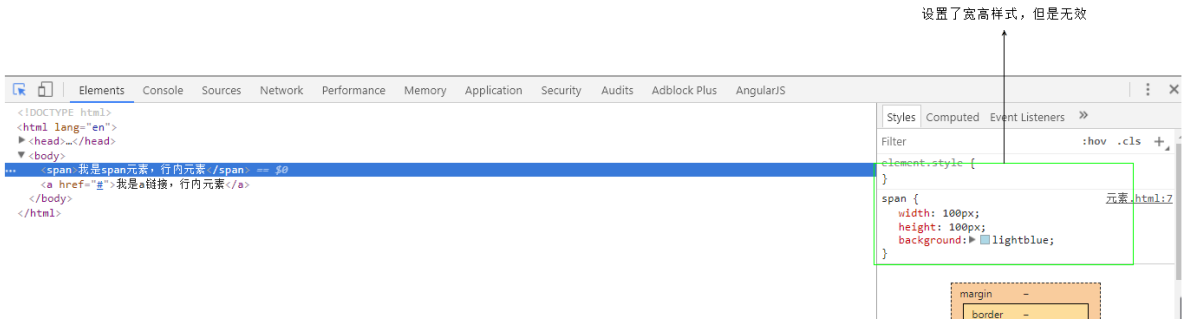
- 注意：
  - 只有 文字才 能组成段落 因此 p 里面不能放块级元素，特别是 p 不能放div
  - 同理还有这些标签h1,h2,h3,h4,h5,h6,dt，他们都是文字类块级标签，里面不能放其他块级元素。

2.3 行内元素(inline-level)

- 例：

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。有的地方也成内联元素





- 行内元素的特点：

(1) 相邻行内元素在一行上，一行可以显示多个。

（2）高、宽直接设置是无效的。

（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。**

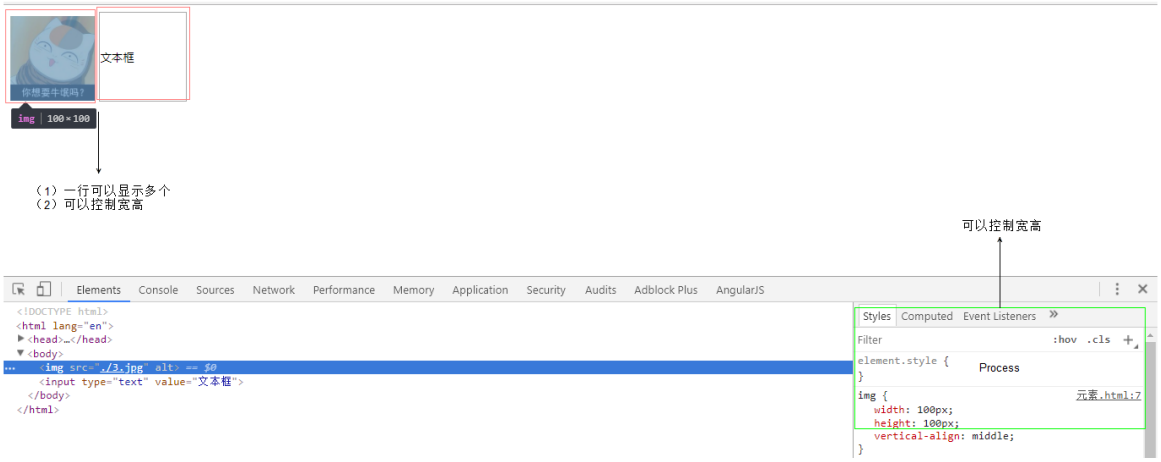
注意：

- 链接里面不能再放链接。
- 特殊情况a里面可以放块级元素，但是给a转换一下块级模式最安全。

2.4 行内块元素（inline-block）

- 例：

> 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。一行可以显示多个
  （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。

2.5 三种模式总结区别

元素模式	元素排列	设置样式	默认宽度	包含
块级元素	一行只能放一个块级元素	可以设置宽度高度	容器的100%	容器级可以包含任何标签
行内元素	一行可以放多个行内元素	不可以直接设置宽度高度	它本身内容的宽度	容纳文本或则其他行内元素
行内块元素	一行放多个行内块元素	可以设置宽度和高度	它本身内容的宽度	

2.6 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。

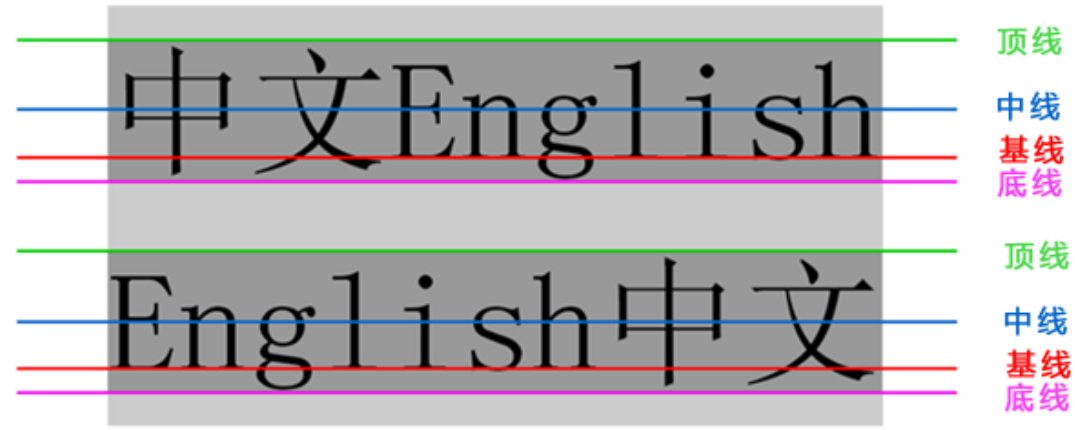
3. 行高那些事（line-height）

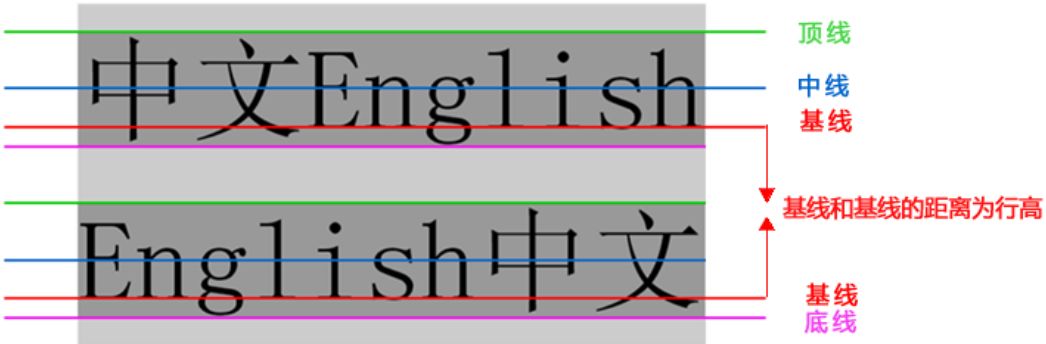
目标

- 理解
  - 能说出 行高 和 高度 三种关系
  - 能简单理解为什么行高等于高度单行文字会垂直居中
- 应用
  - 使用行高实现单行文字垂直居中
  - 能会测量行高

3.1 行高测量

行高的测量方法：





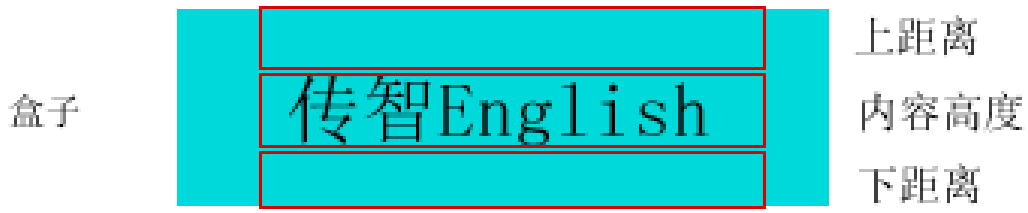
3.2 单行文本垂直居中

行高我们利用最多的一个地方是： 可以让单行文本在盒子中垂直居中对齐。

> **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。

行高 = 上距离 + 内容高度 + 下距离



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。

行高和高度的三种关系

- 如果 行高 等 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

4. CSS 背景(background)

目标

- 理解
  - 背景的作用
  - css背景图片和插入图片的区别
- 应用
  - 通过css背景属性，给页面元素添加背景样式
  - 能设置不同的背景图片位置

4.1 背景颜色(color)

- 语法：

```
background-color:颜色值;   默认的值是 transparent  透明的
```

4.2 背景图片(image)

- 语法：

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

参数	作用
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 : url(images/demo.png);
```

- 小技巧： 我们提倡 背景图片后面的地址，url不要加引号。

4.3 背景平铺（repeat）

- 语法：

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

参数	作用
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

4.4 背景位置(position) 重点

- 语法：

```
background-position : length || length

background-position : position || position
```

参数	值
length	百分数 | 由浮点数字和单位标识符组成的长度值
position	top | center | bottom | left | center | right 方位名词

- 注意：
  - 必须先指定background-image属性
  - 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。
  - 如果指定两个值，两个值都是方位名字，则两个值前后顺序无关，比如left top和top left效果一致
  - 如果只指定了一个方位名词，另一个值默认居中对齐。
  - 如果position 后面是精确坐标， 那么第一个，肯定是 x 第二的一定是y
  - 如果只指定一个数值,那该数值一定是x坐标，另一个默认垂直居中
  - 如果指定的两个值是 精确单位和方位名字混合使用，则第一个值是x坐标，第二个值是y坐标

实际工作用的最多的，就是背景图片居中对齐了。

练习1：

背景大图

练习2：

小图片在盒子左侧垂直居中

4.5 背景附着

- 背景附着就是解释背景是滚动的还是固定的
- 语法：

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

参数	作用
scroll	背景图像是随对象内容滚动
fixed	背景图像固定

4.6 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;
- 语法：

```
background: transparent url(image.jpg) repeat-y  scroll center top ;
```

案例：

导航栏案例

4.7 背景透明(CSS3)

- 语法：

```
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
- 我们习惯把0.3 的 0 省略掉 这样写 background: rgba(0, 0, 0, .3);
- 注意： 背景半透明是指盒子背景半透明， 盒子里面的内容不受影响
- 因为是CSS3 ， 所以 低于 ie9 的版本是不支持的。

4.8 背景总结

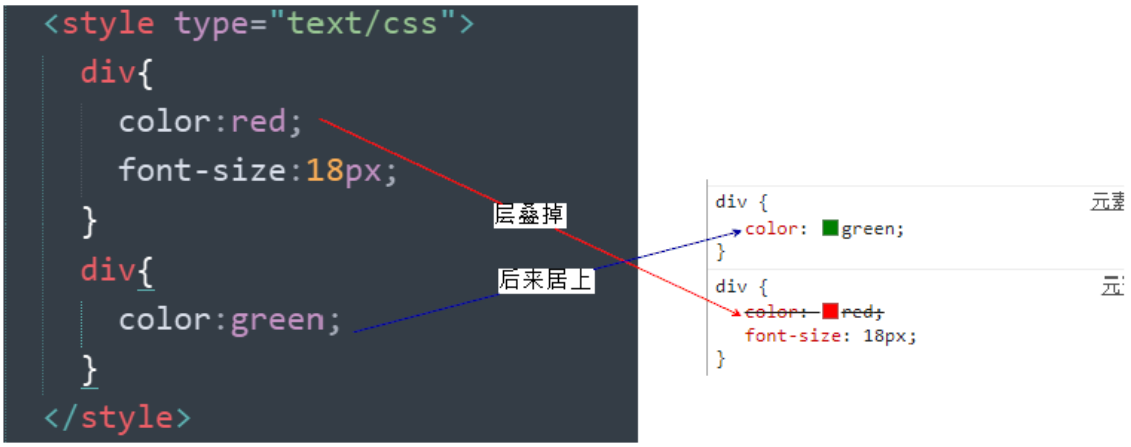
属性	作用	值
background-color	背景颜色	预定义的颜色值/十六进制/RGB代码
background-image	背景图片	url(图片路径)
background-repeat	是否平铺	repeat/no-repeat/repeat-x/repeat-y
background-position	背景位置	length/position 分别是x 和 y坐标， 切记 如果有 精确数值单位，则必须按照先X 后Y 的写法
background-attachment	背景固定还是滚动	scroll/fixed
背景简写	更简单	背景颜色 背景图片地址 背景平铺 背景滚动 背景位置; 他们没有顺序
背景透明	让盒子半透明	background: rgba(0,0,0,0.3); 后面必须是 4个值

5. CSS 三大特性

目标：

- 理解
  - 能说出css样式冲突采取的原则
  - 能说出那些常见的样式会有继承
- 应用
  - 能写出CSS优先级的算法
  - 能会计算常见选择器的叠加值

5.1 CSS层叠性



- 概念：

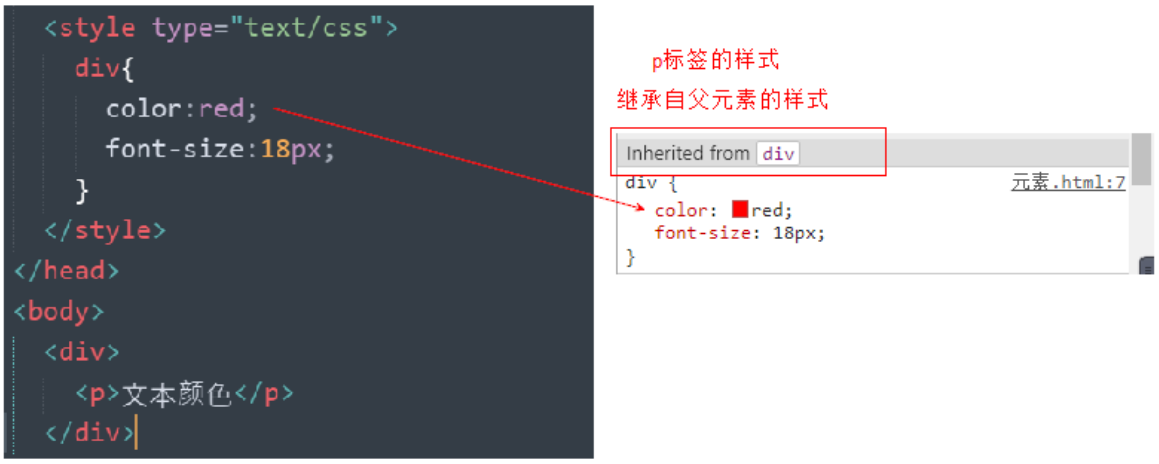
  所谓层叠性是指多种CSS样式的叠加。

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

- 原则：
  - 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。
  - 样式不冲突，不会层叠

> CSS层叠性最后的执行口诀：　长江后浪推前浪，前浪死在沙滩上。

5.2 CSS继承性



- 概念：

  子标签会继承父标签的某些样式，如文本颜色和字号。

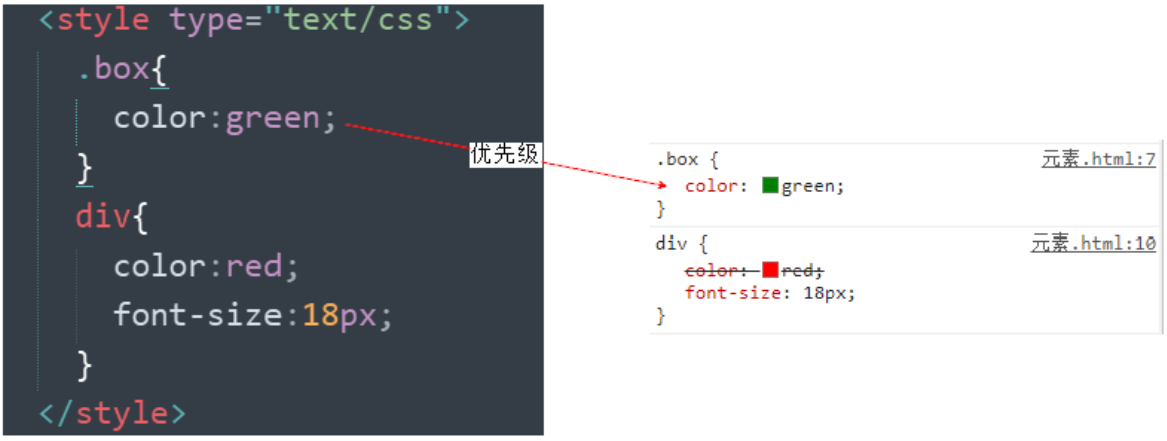
  想要设置一个可继承的属性，只需将它应用于父元素即可。

简单的理解就是：　子承父业。

- **注意**：
  - 恰当地使用继承可以简化代码，降低CSS样式的复杂性。比如有很多子级孩子都需要某个样式，可以给父级指定一个，这些孩子继承过来就好了。
  - 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**）

> CSS继承性口诀：　龙生龙，凤生凤，老鼠生的孩子会打洞。

5.3 CSS优先级（重点）



- 概念：

  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，此时，
  - 选择器相同，则执行层叠性
  - 选择器不同，就会出现优先级的问题。

1). 权重计算公式

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（特殊性）

标签选择器	计算权重公式
继承或者 *	0,0,0,0
每个元素（标签选择器）	0,0,0,1
每个类，伪类	0,0,1,0
每个ID	0,1,0,0
每个行内样式 style=""	1,0,0,0
每个!important 重要的	∞ 无穷大

- 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。
- 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（特殊性）
- div {
    color: pink!important;
  }

2). 权重叠加

我们经常用交集选择器，后代选择器等，是有多个基础选择器组合而成，那么此时，就会出现权重叠加。

就是一个简单的加法计算

- 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

 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。

3). 继承的权重是0

这个不难，但是忽略很容易绕晕。其实，我们修改样式，一定要看该标签有没有被选中。

1） 如果选中了，那么以上面的公式来计权重。谁大听谁的。
2） 如果没有选中，那么权重是0，因为继承的权重为0.

6. CSS注释

CSS注释规则：

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

7. 今日总结