



# CSS基础

CSS基础知识

---

# 目录



01

CSS简介

02

CSS引入方式

03

CSS选择器

04

CSS基本语法

05

装饰属性

06

盒子模型

07

布局属性



# — CSS简介

CSS简介

## 1.1 所有漂亮的网页都有源码

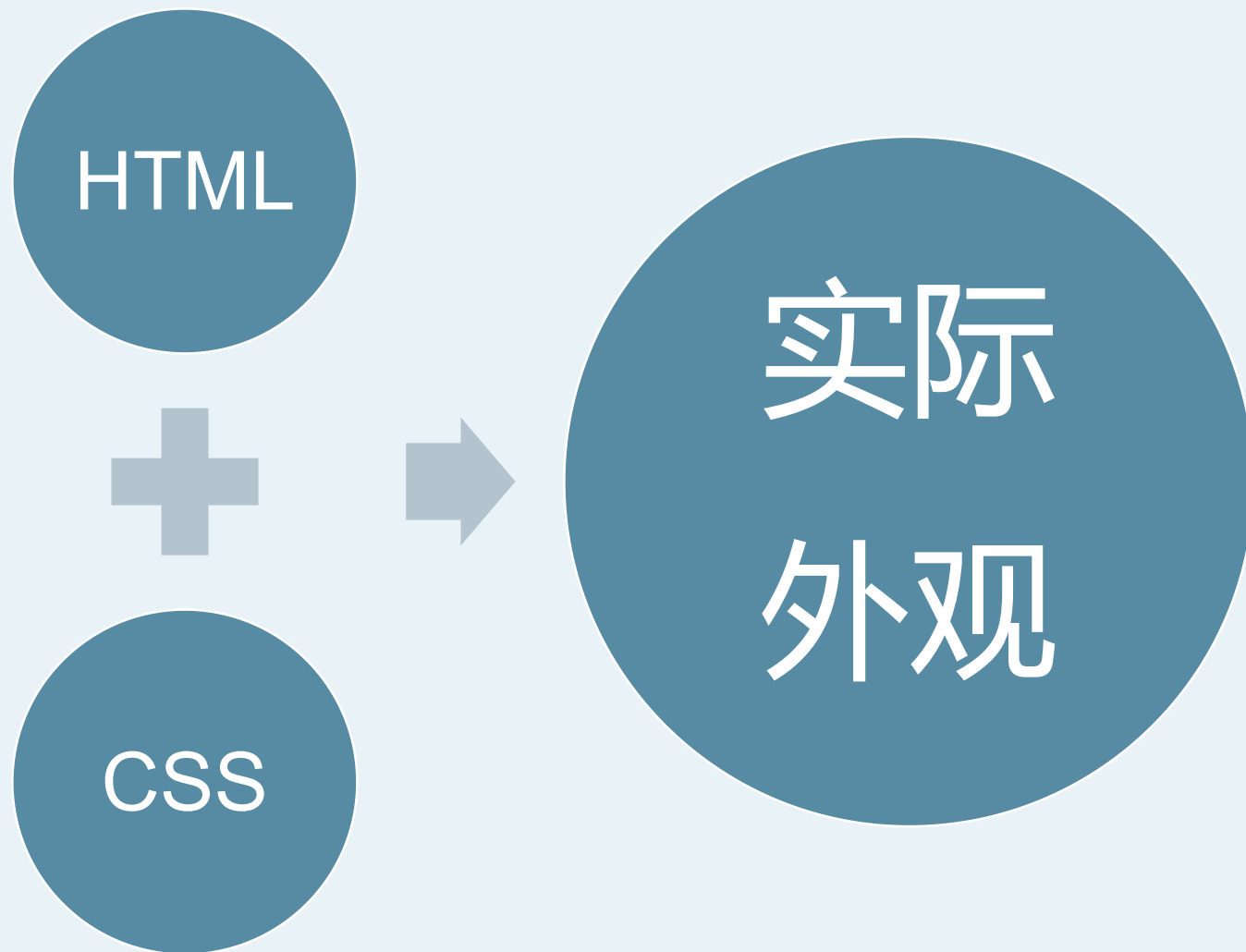
通知公告

招生招聘

19 06月	<b>【通知公告】</b> 信息学院关于进一步做好2020年夏秋季征兵工作的	微信公众号 
25 02月	<b>【外事】</b> <b>【学生交流】</b> 关于北美地区2020-2021学年校际交	
20 09月	<b>【本科生】</b> 信息学院关于组织开展2020-2021学年台湾、港澳	会议室预定
15 09月	<b>【外事】</b> 李绍滋因公出访事后公示表	大类选专业
27	<b>【研究生】</b>	

```
52 <body>
53 <script language="javascript" src="/system
54 <ul class="dice-menu">
55 <li class="wx"> <span>微信公众号</span> &nb
56 <div class="menu-img">
57  </li>
59 <li>
60 <div>
61 <em class="fa fa-lg fa-edit"></em> &nbs
62 </div><a href="http://172.27.65.113/" ta
255);">会议室预定</span></a> </li>
63 <li>
64 <div>
65 <em class="fa fa-lg fa-edit"></em> &nbs
66 </div><a href="http://10.24.81.70:8080/S
style="color: rgb(255, 255, 255);">大类选专!
67 </ul>
68 <header>
```

## 1.2 HTML + CSS = 网页实际外观



## 1.3 CSS：把丑陋的HTML渲染漂亮的工具

- HTML网页包括

网页的内容

文字内容

图片内容

各种表单元素

网页的排版布局

颜色

字体

边框线

网页内容的位置和大小尺寸

没有应用CSS

请填写下列用户注册表

用户名:

密码:

电子邮件:

提交

应用了一种CSS

请填写下列用户注册表

用户名:

密码:

电子邮件:

提交

应用了另一种CSS

请填写下列用户注册表

用户名:

密码:

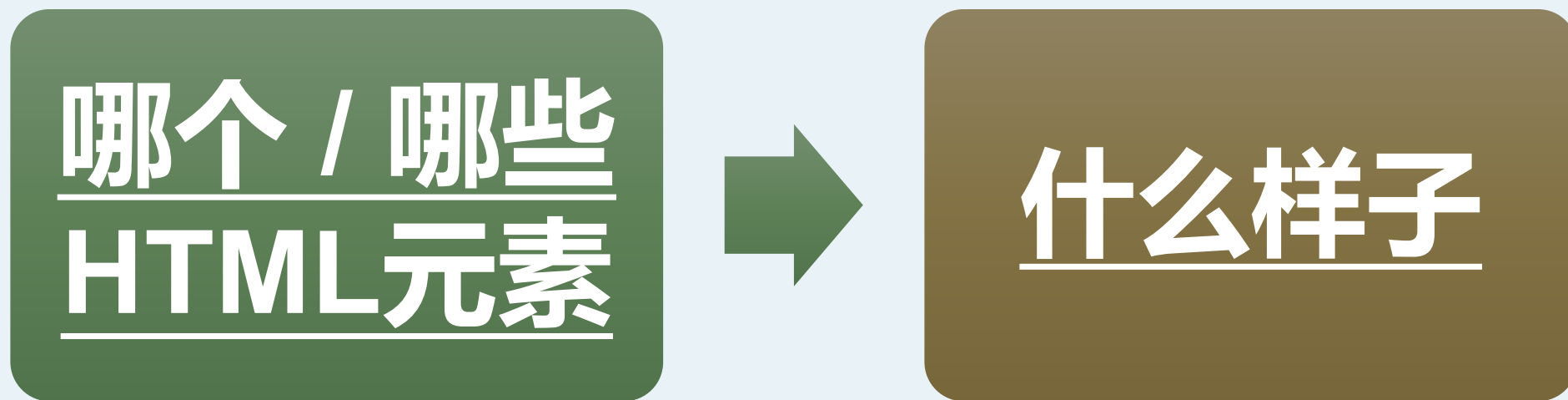
电子邮件:

提交

采用CSS技术, 可以有效地对页面布局、字体、颜色、背景和其他效果实现更加精准的控制

## 1.4 CSS小结

- CSS 指的是层叠样式表\* (Cascading Style Sheets)或层级样式表
- CSS 描述了如何在屏幕、纸张或其他媒体上显示 HTML 元素
- CSS 节省了大量工作。它可以同时控制多张网页的布局
- 外部样式表存储在 CSS 文件中





## 二 CSS引入方式

CSS有三种引入方式：行内样式、内部样式表、外部样式表



## 2.1 行内样式

- 直接使用style属性引入CSS样式。

```
<h1>style属性的应用</h1>  
<p>直接在HTML标签中设置的样式</p>  
<hr/>  
<h1 style="color:red;">style属性的应用</h1>  
<p style="font-size:20px;color:green;">直接在HTML标签中设置的样式</p>
```

### style属性的应用

直接在HTML标签中设置的样式

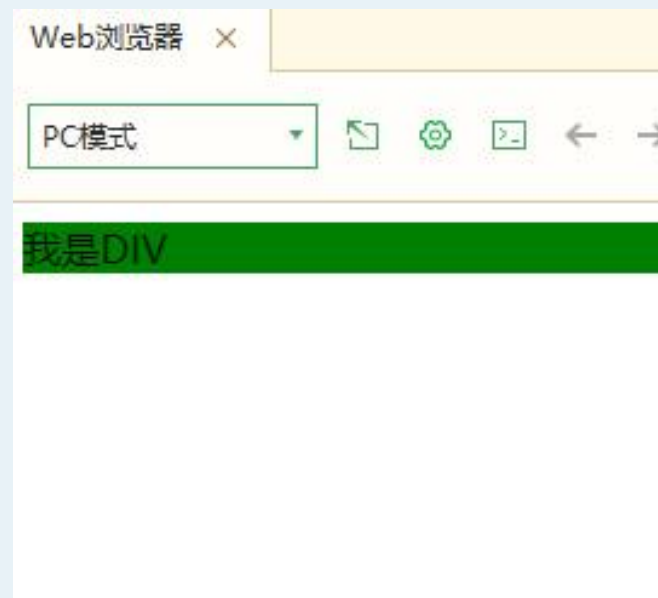
### style属性的应用

直接在HTML标签中设置的样式

## 2.2 内部样式表

- 在style标签中书写CSS代码。style标签写在head标签中。

```
1  <!DOCTYPE>
2  <html>
3  <head>
4      <meta charset="utf-8" />
5      <title>内部样式表</title>
6      <!--使用内部样式表引入CSS-->
7      <style type="text/css">
8          div{
9              background: green;
10         }
11     </style>
12 </head>
13 <body>
14     <div>我是DIV</div>
15 </body>
16 </html>
```



## 2.3 外部样式表

拢共分两步：

1. CSS代码保存在扩展名为.css的样式表文件中
2. HTML文件引用扩展名为.css的样式表文件

引用样式表文件有两种方式：

1. 链接式

```
<link type="text/css" rel="styleSheet" href="CSS文件路径" />
```

2. 导入式

```
<style type="text/css">
```

```
    @import url("css文件路径");
```

```
</style>
```

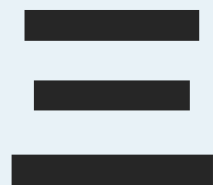
## 2.4 三种引入方式的优先级

- 就近原则：行内样式 > 内部样式 > 外部样式

```
<!DOCTYPE>
<html>
<head>
  <meta charset="utf-8" />
  <title>测试CSS引入的优先级</title>
  <link type="text/css" rel="stylesheet" href="style1.css">
  <!--使用内部样式表引入CSS-->
  <style type="text/css">
    .title{
      color:green;
      font-size: 40px;
      font-weight: bold;
    }
    .title2{
      color:green;
      font-size: 40px;
      font-weight: bold;
    }
  </style>
</head>
<body>
  <div class="title" style="font-size:30px;color:blue;font-weight:normal;">文章标题</div>
  <div class="title2">文章标题2</div>
  <div class="title3">文章标题3</div>
</body>
</html>
```

```
.title{
  color:yellow;
}
.title2{
  color:red;
}
.title3{
  color:red;
  font-size:50px;
  font-weight: bold;
}
```

文章标题  
**文章标题2**  
**文章标题3**

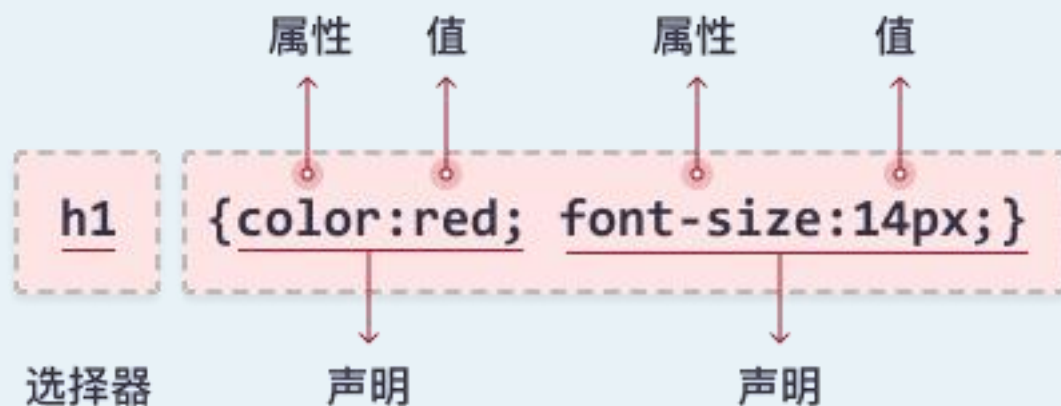


## 三 CSS选择器

将哪些HTML元素进行美颜

## 3.1 CSS规则集

- CSS 规则集 (rule-set) 由选择器和声明块组成：
  - 选择器指向您需要设置样式的 HTML 元素。
  - 声明块包含一条或多条用分号分隔的声明。
  - 每条声明都包含一个 CSS 属性名称和一个值，以冒号分隔。 即： **selector {property: value}**
  - 多条 CSS 声明用分号分隔，声明块用花括号括起来。



样例解释：

- `h1` 是 CSS 中的选择器（它指向要设置样式的 HTML 元素：`<h1>`）
- `color` 是属性，`red` 是属性值
- `font-size` 是属性，`14px` 是属性值

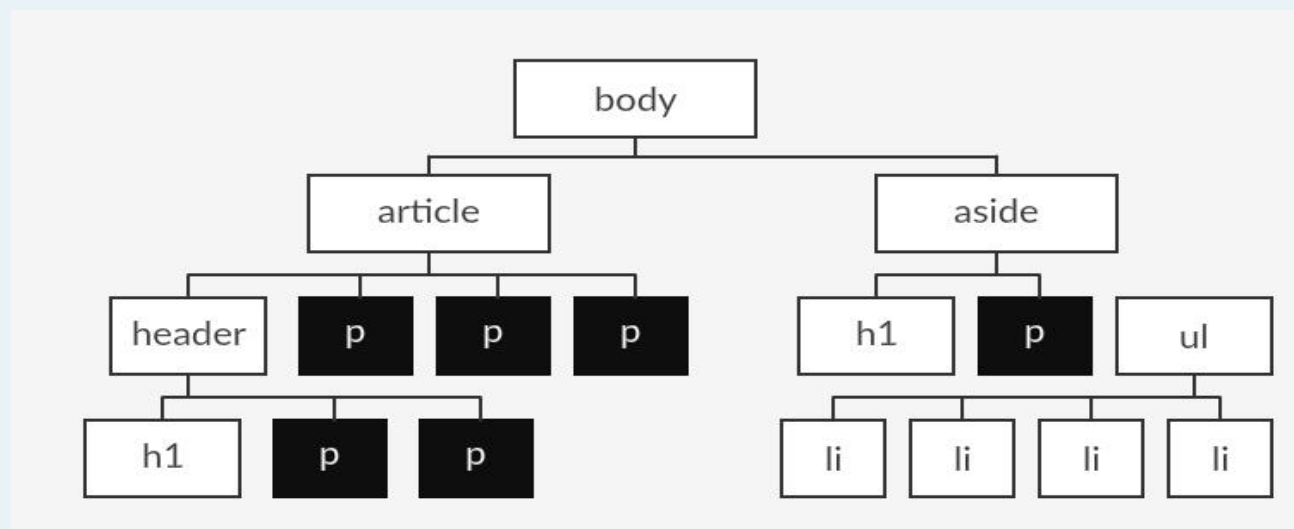
## 3.2 CSS选择器

- CSS 选择器用于“查找”（或选取）要设置样式的 HTML 元素。
- CSS 选择器分为五类：
  - 简单选择器（根据元素名称、id、类来选取元素）
  - 组合器选择器（根据它们之间的特定关系来选取元素）
  - 伪类选择器（根据特定状态选取元素）
  - 伪元素选择器（选取元素的一部分并设置其样式）
  - 属性选择器（根据属性或属性值来选取元素）

## 3.2.1 CSS元素选择器

- 元素名对象：用于网页中的所有指定元素，例如如下样式定义，网页中所有p元素中的内容都会是绿色的。

```
p { color:green;}
```

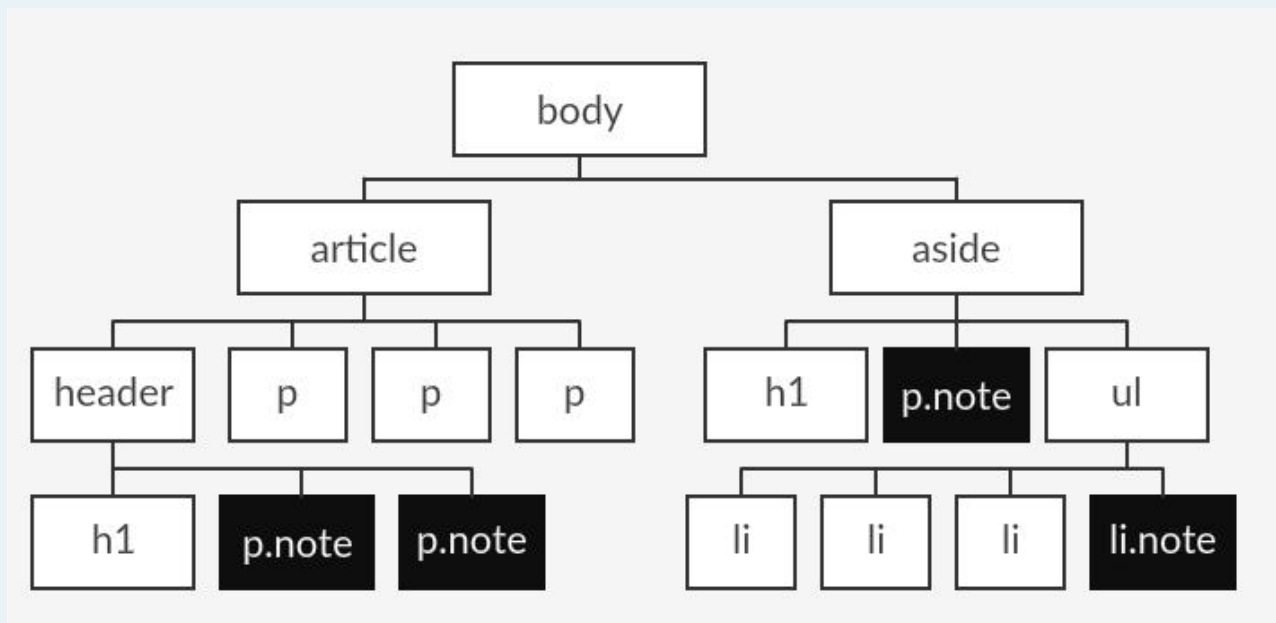




## 3.2.2 类选择器

- 类选择器：当网页中多个元素都具有相同的样式时，可以设置自定义对象。自定义对象时，对象名之前为一个点“.”，应用时使用class属性，即class="自定义对象名"。例如如下样式定义，网页中所有标有class="note"的元素中的字体都会变小。

```
.note { font-size:80%}
```



### 3.2.3 元素名和自定义对象的组合的选择器

元素名和自定义对象的组合:元素名和自定义对象的组合设置用于指定“某元素名中的自定义对象”。例如:

li.note1 只影响li元素中的“note1”样式

应用note的样式

默认不变

默认不变

- 缩小的字体
- 应用note的样式
- 这是默认情况1

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title></title>
    <style>
      .note{
        font-size:120%;
        font-style:italic;
        color:#2335FE;
      }
      li.note1{
        font-size: 80%;
        color:red;
      }
    </style>
  </head>
  <body>
    <p class="note">应用note的样式</p>
    <p class="note1">默认不变</p>
    <p>默认不变</p>
    <ul>
      <li class="note1">缩小的字体</li>
      <li class="note">应用note的样式</li>
      <li>这是默认情况1</li>
    </ul>
  </body>
</html>
```

### 3.2.4 多个自定义对象的组合的选择器

- 多个自定义对象的组合

例如示例中元素标有class="note date"的字体都会变小，并且变为斜体

```
.note { font-size:80%}  
.date { font-style:italic}
```

```
<p class="note date">2017.12.01</p>
```

## 3.2.5 ID选择器

- ID选择器: 用于网页中指定的标识名元素, 定义时标识名前有一个#。与自定义对象不同的是, 在一个网页中, 自定义对象的样式可以用于多个元素, 而标识名对象的样式只可用于一个元素。

例如示例中标识名为myHeader元素的字体颜色将是红色。

```
#myHeader { color:red;}
```

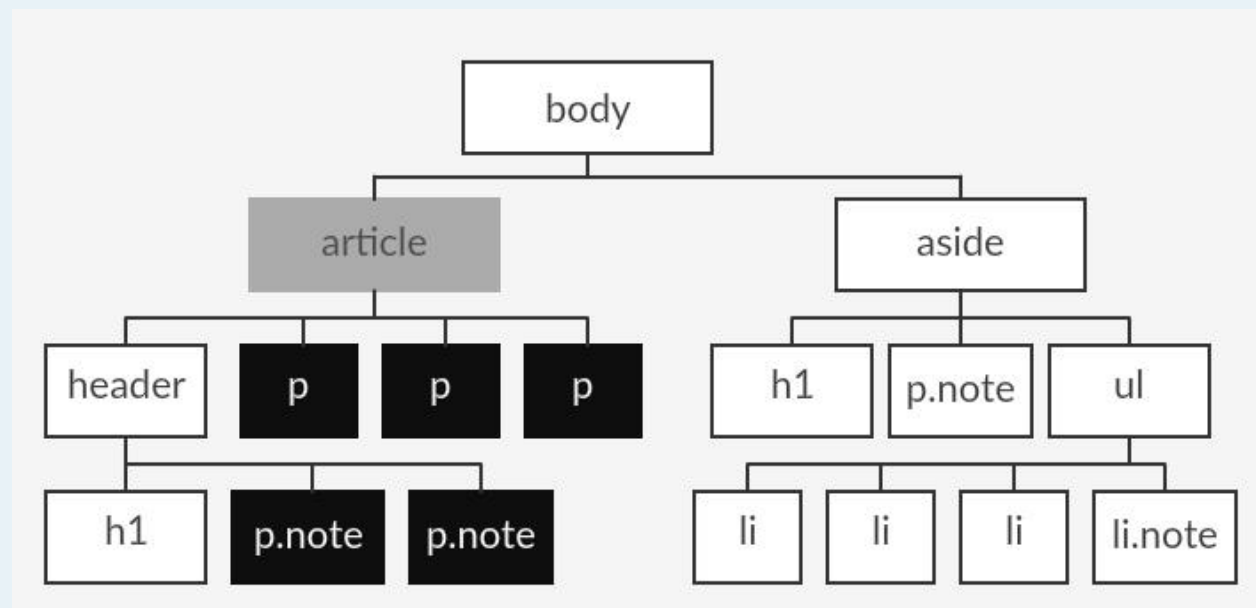
```
<header>  
  <h1 id="myHeader">我的网页</h1>  
  <p class="note">欢迎进入我的网页</p>  
  <p class="note date">2017.12.01</p>  
</header>
```

## 3.2.6 下级（后代）对象

- 下级（后代）对象：用于某一种元素中的下级（后代）元素（子子孙孙），定义时两元素名之间用空格相间。

例如示例中只有article元素中的p元素的字体颜色是蓝色。

```
article p {color: blue;}
```



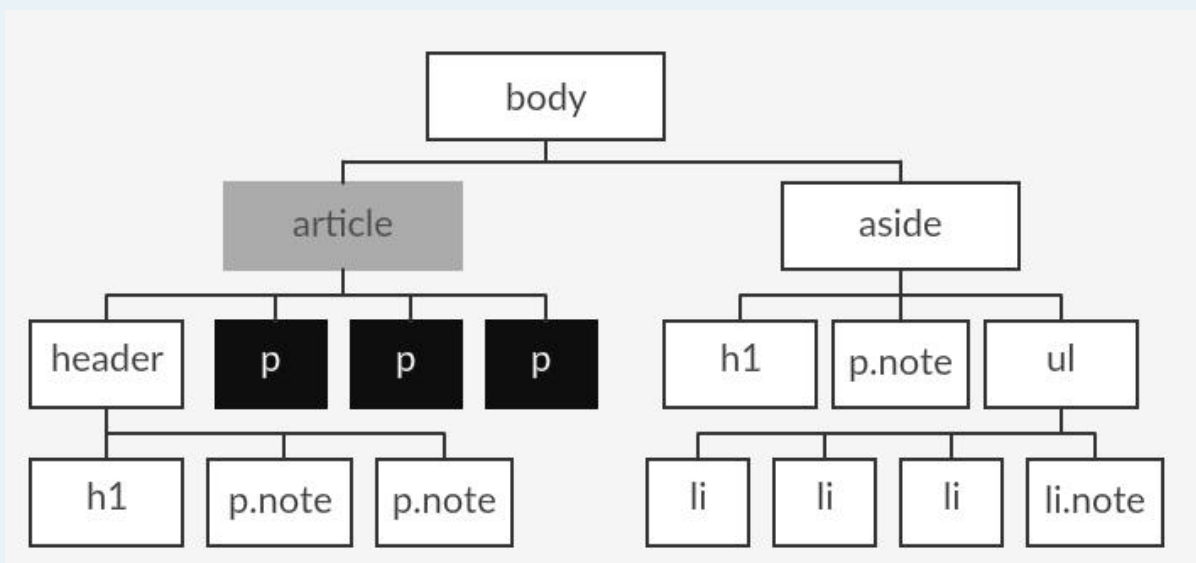
## 3.2.7 下一级对象

- 下一级对象

下一级对象用于某一种元素中的下一级元素，定义时两元素名之间用>相间。

例如示例中只有article元素中下一级的p元素起作用：背景色为浅灰色，内边距为10px，最大宽度为500px。而header下的p.note，aside下的p.note都不起作用。

```
article>p {background:#eeeeee; padding:10px; max-width:500px;}
```

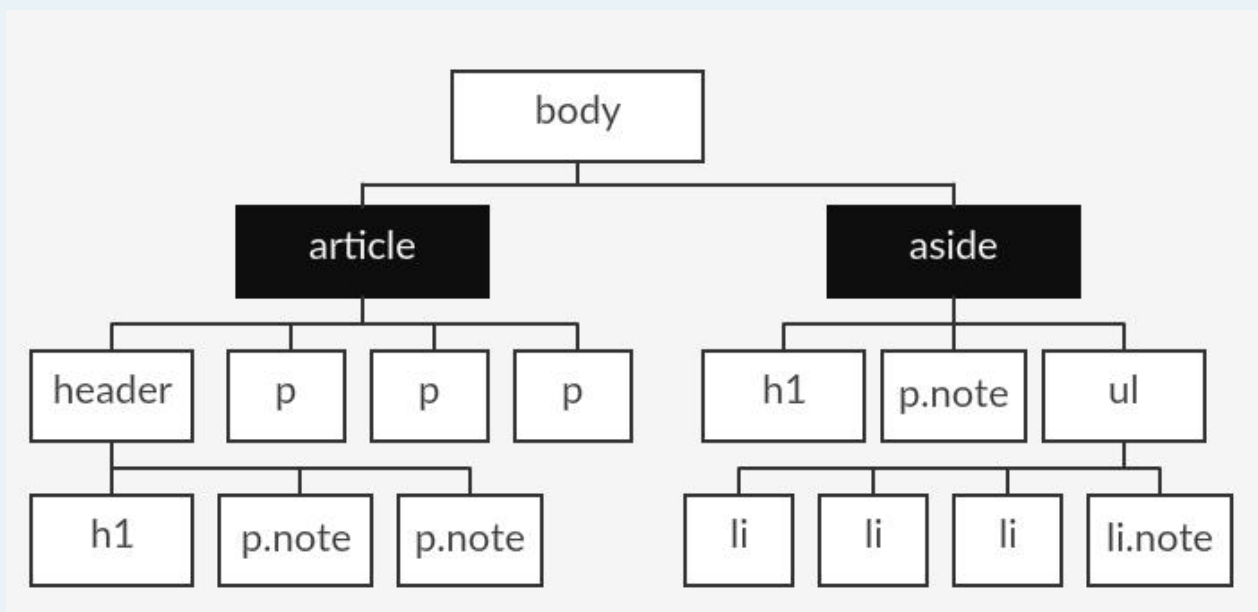


## 3.2.8 分组元素对象

- 分组元素对象：分组的元素可以具有相同的样式表，分组元素对象定义时用逗号表示。

例如示例中网页中的所有article和aside元素的边框线1px的灰色实线，内外边距为20px。

```
article,aside {border:1px solid #dddddd; padding:20px; margin:20px;}
```

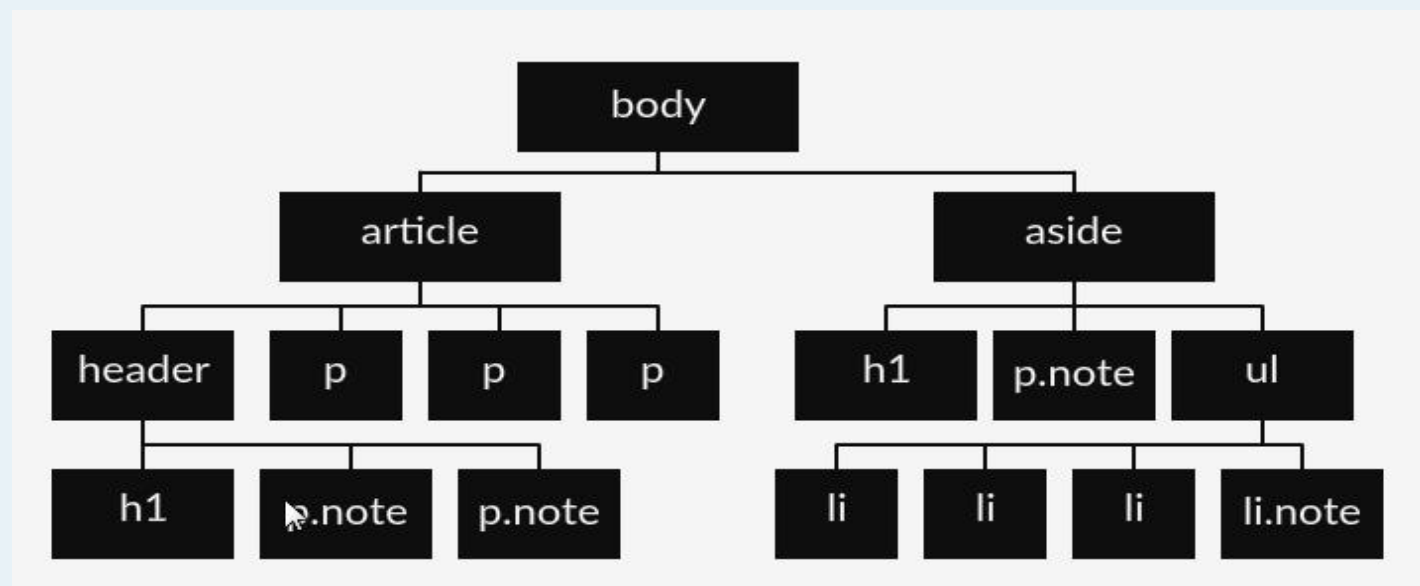


### 3.2.9 所有元素对象

- 所有元素对象:所有元素对象用于网页中的所有元素，定义时用星号\*表示。

例如示例中对网页中所有元素都起作用了，行高为150%。

```
* {line-height:150%;}
```



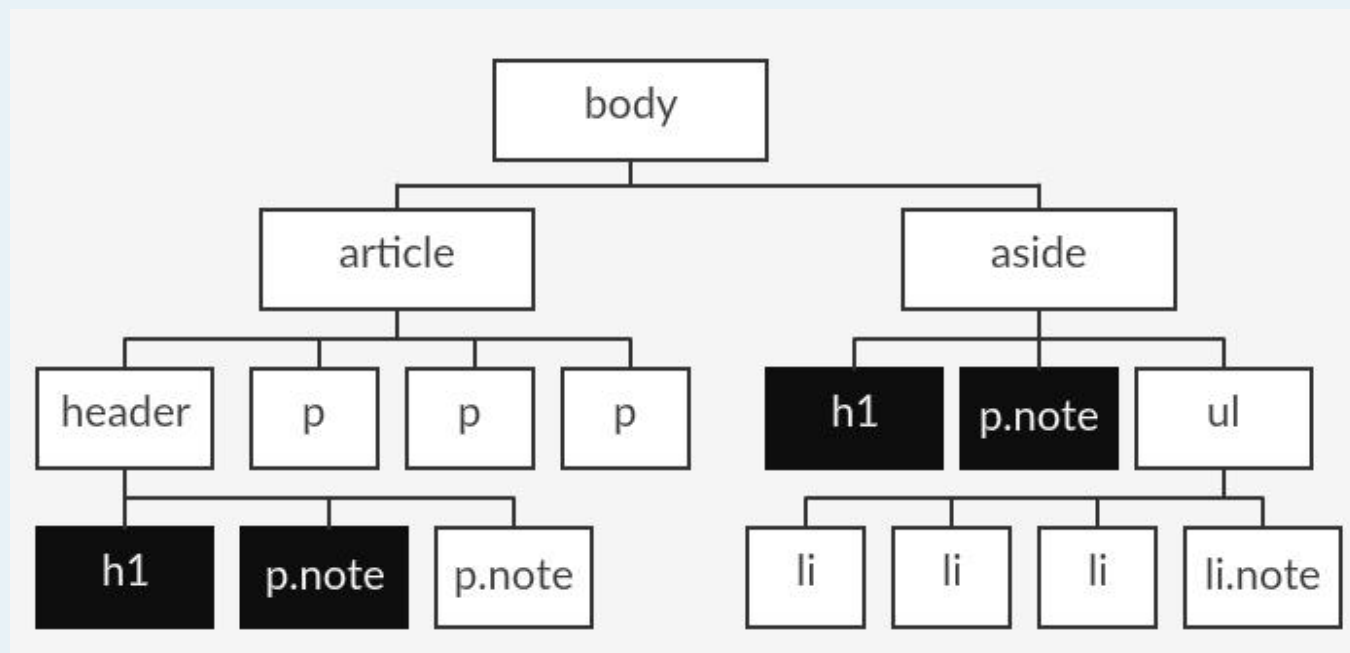


## 3.2.10 相邻对象

- 相邻对象: 用于某一种元素相邻的另一个元素, 定义时两元素名之间用加号+相间。

例如示例中只有与h1元素相邻的p元素的字体颜色是蓝色。

```
h1+p {color: blue;}
```

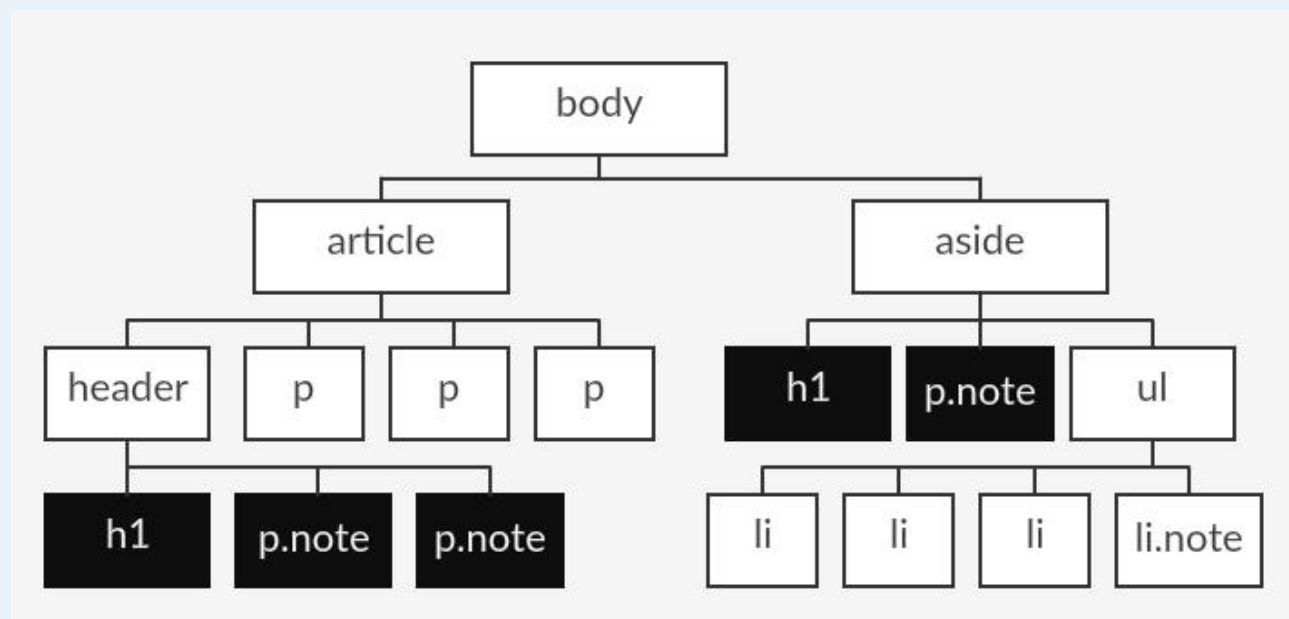


### 3.2.11 同级元素后面出现的元素对象

- 同级元素后面出现的元素对象：同级元素后面出现的元素对象定义时两元素名之间用波浪线~相间。

例如示例中h1与p同级，h1后面的所有p元素的字体颜色是蓝色。

**h1~p {color: blue;}**



## 3.2.12 其他选择器

- 属性对象：用于指定元素名中含有指定属性的元素，定义时元素名后用 “[ ]” 指定属性内容。
- 状态或级别对象：状态或级别对象用于某一种元素的状态或级别，定义时元素名与状态之间用一个或两个冒号 “:” 相间。
- 注释行：在定义样式表的语句中还可以加入注释语句，它的格式是在 /\* \*/ 之间加入注释内容。
- 更多选择器的使用详见：<https://www.runoob.com/cssref/css-selectors.html>



# 四

## 四、CSS基本语法

## 4.1 CSS基本语法

- CSS的最基本的元素就是**样式**和**样式表**
- 样式：由成对的属性名和属性值以冒号 “:” 相间组成的

属性名：属性值

- 样式表：一系列的“样式”以**分号 “;”**相间组成“样式表”

属性名1:属性值1; 属性名2:属性值2; 属性名3:属性值3; ...

- 元素中的样式表的写法：

<元素标签名 style="属性名1:属性值1; 属性名2:属性值2; ..." >

...

</元素标签名>



# 五

## 五、装饰属性

## 5.1 CSS颜色

- 颜色是由红（RED），绿（GREEN），蓝（BLUE）光线的显示结合。
- CSS RGB颜色值为红，绿，蓝值从0到255的结合，给出了总额超过1600多万不同的颜色（ $256 \times 256 \times 256$ ）。

## 5.1.1 CSS颜色表示法




- **颜色名**:万维网联盟 (W3C) 的HTML和CSS标准提供了140多种颜色名, 如red、blue、green 等。
  - 例: `color:red; /* 红色 */`
- **RGB颜色**:它是由红 (Red)、绿 (Green)、蓝 (Blue) 3种颜色组合而成的, 每种颜色用数字0 ~ 255表示, 共可表示 $255 \times 255 \times 255$  (超过1600万) 种颜色。
  - 例如: 红色: `color: rgb(255,0,0);`
- **十六进制 (hex) 表示法**: 3个两位数字的十六进制值, 以#符号开始。依次为红, 绿, 蓝的颜色值, 取值为最低值是十六进制00 (十进制0) 到最高值是十六进制FF (十进制255)。
  - 例如: 红色: `color: #ff0000;`
- **RGBA 颜色**: 类似于 `rgb()`, 但增加了 alpha 通道, 用于指定透明度, 值范围在 0 (完全透明) 到 1 (完全不透明)。
  - 例如: `color: rgba(255, 0, 0, 0.5); /* 半透明红色 */`



## 5.1 CSS颜色表示法（续）

- **HSL 颜色**：使用 `hsl()` 函数指定色调（0-360°）、饱和度（0%-100%）和亮度（0%-100%）。
  - 例如：`color: hsl(0, 100%, 50%); /* 红色 */`
  - 例如：`background-color: hsl(120,50%,50%)`
  - H：Hue(色调)。0(或360)表示红色，120表示绿色，240表示蓝色，也可取其他数值来指定颜色。取值为：0 - 360
  - S：Saturation(饱和度)。取值为：0.0% - 100.0%
  - L：Lightness(亮度)。取值为：0.0% - 100.0%
- **HSLA 颜色**：类似于 `hsl()`，但增加了 `alpha` 通道，用于透明度控制。
  - 例如：`color: hsla(0, 100%, 50%, 0.5); /* 半透明红色 */`

## 5.1.2 颜色示例

Color	Color HEX	Color RGB
	#000000	rgb(0,0,0)
	#FF0000	rgb(255,0,0)
	#00FF00	rgb(0,255,0)
	#0000FF	rgb(0,0,255)
	#FFFF00	rgb(255,255,0)
	#00FFFF	rgb(0,255,255)
	#FF00FF	rgb(255,0,255)
	#C0C0C0	rgb(192,192,192)
	#FFFFFF	rgb(255,255,255)

## 5.2 CSS长度

- CSS 有几个不同的单位用于表示长度。
- 一些设置 CSS 长度的属性有 width, margin, padding, font-size, border-width, 等。
- 长度有一个数字和单位组成如 10px, 2em, 等。数字与单位之间不能出现空格。
- 对于一些 CSS 属性, 长度可以是负数。
- 有两种类型的长度单位: 相对和绝对。

## 5.2.1 常用的CSS长度

- 像素或许被认为是最好的"设备像素",而这种像素长度和你在显示器上看到的文字屏幕像素无关。px实际上是一个按角度度量的单位。

长度单位	说明
in	英寸 英寸 (1in = 96px = 2.54cm)
cm	厘米
mm	毫米
em	相对于当前元素的字体尺寸,是相对长度单位。 假设浏览器字体大小默认为16px,则2em == 32px;
pt	点=1/72英寸 point, 大约1/72英寸; (1pt = 1/72in)
pc	pica点=12点
px	像素点 像素 (1px = 1/96th of 1in)

## 5.3 文本的样式属性

- 文字的样式属性可以用于改变文字的字体、粗细、斜体、大小、颜色、行距、对齐方式以及文字上的装饰等。
- (1) `font-family: name1,[name2],[name3]`
- (2) `font-size: length`
- (3) `color: color`
- (4) `text-align: left|center|right|justify`
- (5) `text-indent: length`

## 5.4 CSS背景的样式属性

- 背景的样式属性可以用于改变指定元素的背景颜色、背景图像及其位置等。
  - (1) background-color: color|transparent
  - (2) background-image: url()
  - (3) background-repeat: inherit|no-repeat|repeat-x|repeat-y
  - (4) background-attachment: scroll|fixed
  - (5) background-position: length|percentage|top|center|bottom|left|right
  - (6) background-size: length|percentage|cover|contain;

## 5.4.1 CSS背景的样式属性举例

- 背景图片及背景尺寸属性的设置效果: background-size
  - background-size: length|percentage|cover|contain;



## 练习

- 尝试运行以下代码，试试更改不同的css样式
- [https://www.w3school.com.cn/tiy/t.asp?f=css3\\_background-size](https://www.w3school.com.cn/tiy/t.asp?f=css3_background-size)



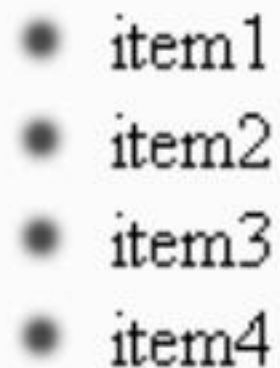


## 5.5 列表的样式属性

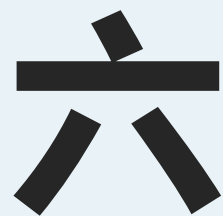
### 列表相关属性

- 列表的样式属性用于改变指定列表项的类型、图像及位置等。
- 表格的样式属性用于改变指定表格的显示方式。

```
<ol style="list-style:outside url(dot.jpg)">  
  <li>item1 </li>  
  <li>item2 </li>  
  <li>item3 </li>  
  <li>item4 </li>  
</ol>
```



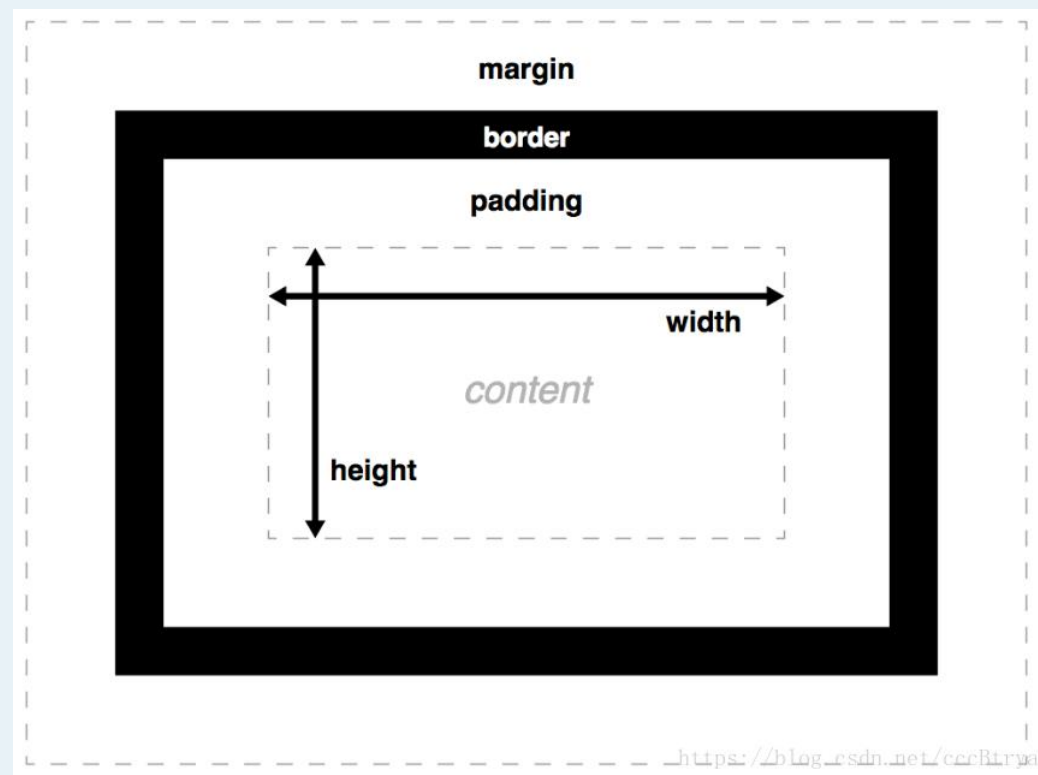
- item1
- item2
- item3
- item4



## 盒子模型

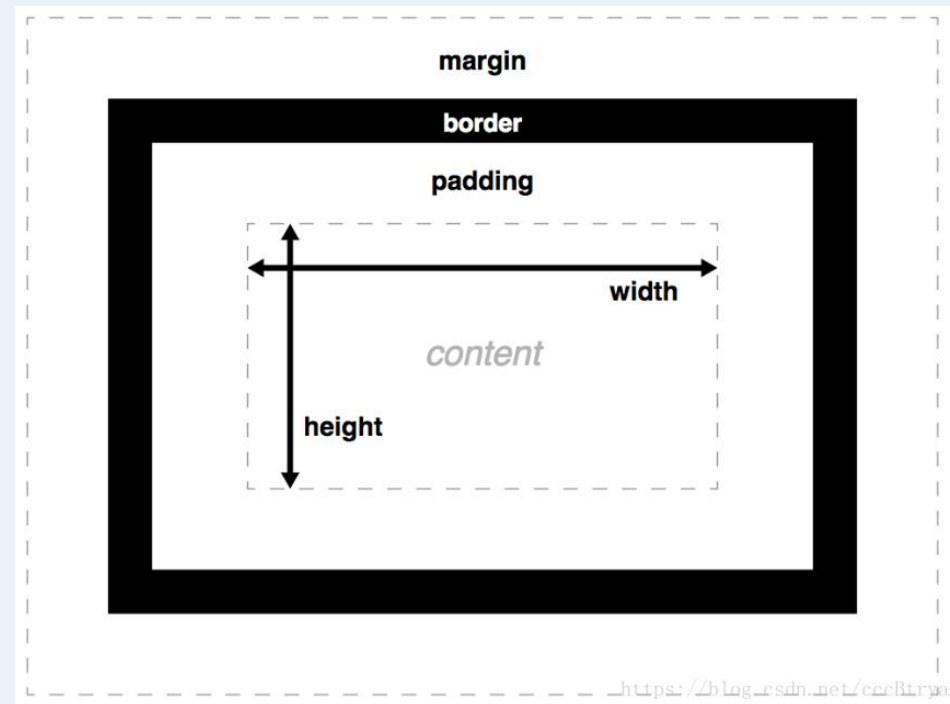
## 6.1 盒子模型基本概念

- 盒子模型 box model，用来设计和布局时使用。CSS盒模型本质上是一个盒子，封装周围的HTML元素，包括：边距（外边距），边框，填充（内边距），和实际内容。
- 所有HTML元素可以看作盒子



## 6.2 盒子模型总宽高

- 默认情况下 (`box-sizing: content-box;`)，元素的宽度、高度指的是内容的宽度和高度。
  - 元素的总宽度和总高度计算方式如下：
  - 元素总宽度 = 宽度 + 左内边距 + 右内边距 + 左边框 + 右边框 + 左外边距 + 右外边距
  - 元素总高度 = 高度 + 上内边距 + 下内边距 + 上边框 + 下边框 + 上外边距 + 下外边距
- 
- 当`box-sizing: border-box;`，`width`、`height`包括边框和内边距；
  - 元素的总宽度和总高度计算方式如下：
  - 元素总宽度 = 宽度 + 左外边距 + 右外边距
  - 元素总高度 = 高度 + 上外边距 + 下外边距



## 6.3 CSS3 box-sizing 属性

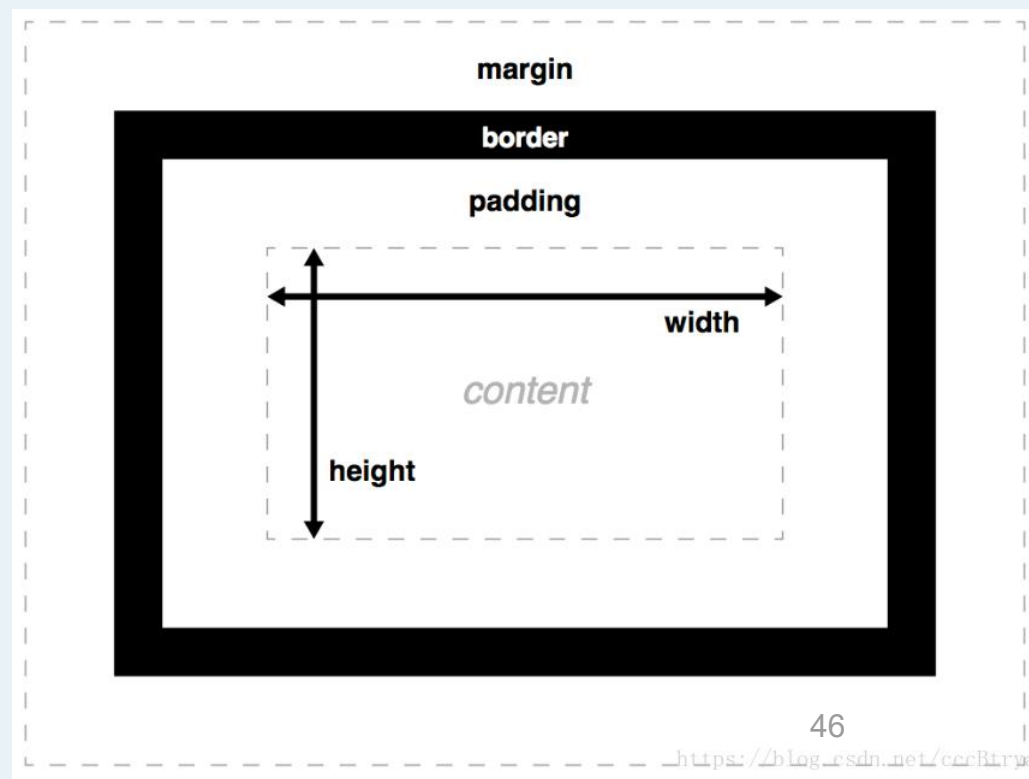
- 语法：box-sizing: content-box|border-box|inherit:

值	说明
<b>content-box</b>	默认值。如果你设置一个元素的宽为 100px，那么这个元素的内容区会有 100px 宽，并且任何边框和内边距的宽度都会被增加到最后绘制出来的元素宽度中。
<b>border-box</b>	告诉浏览器：你想要设置的边框和内边距的值是包含在 width 内的。也就是说，如果你将一个元素的 width 设为 100px，那么这 100px 会包含它的 border 和 padding，内容区的实际宽度是 width 减去 (border + padding) 的值。大多数情况下，这使得我们更容易地设定一个元素的宽高。 注：border-box 不包含 margin。
<b>inherit</b>	指定 box-sizing 属性的值，应该从父元素继承

## 6.4 CSS布局常用属性

- 高度和宽度
  - 高度和宽度的样式属性用于改变指定元素的高度和宽度。
- 内、外边距
  - 内边距 (padding) 指的是元素中的内容与边框线间的距离, 因此, 这个值不可以是负数;
  - 外边距 (margin) 指的是元素之间的距离, 这个值可以是负数
- 例如:
  - `margin-top: <length>|auto|;`
  - `margin: 5px;`
  - `margin: 5px 10px;` ←
  - `margin: 上 右 下 左;`

`margin-top: 5px;`  
`margin-right: 10px;`  
`margin-bottom: 5px;`  
`margin-left: 10px;`



## 6.5 CSS布局练习

未使用CSS



使用CSS



## 6.5 CSS布局练习

- 根据已给html 文档(练习素材→magin-padding目录下)，参考下图css样式编写CSS文档，可自行调整属性及属性值，掌握内外边距及长宽的设定方法



```
h1{ /*设置标题样式*/
    padding: 30px 50px 30px 525px; /*设置标题文字的内间距*/
}

.bottom{ /*设置手机部分的整体样式*/
    height: 500px; /*设置其高度*/
}

.dl{ /*设置每一个手机部分的样式*/
    float: left; /*设置浮动为左浮动*/
    height: 511px; /*设置高度*/
    width: 394px; /*设置宽度*/
    margin-left: 4px; /*设置想做的左外边距*/
    background: #fff; /*设置背景颜色*/
}

.dd{ /*设置文字介绍部分的样式*/
    border: 1px dashed #f0f; /*添加边框样式*/
    border-radius: 10px; /*设置圆角边框*/
    padding: 10px 90px; /*设置文字的内边距*/
    margin: 5px; /*设置文字的外边距*/
}

img{ /*设置图片大小*/
    width: 250px; /*设置图片的内边距*/
    padding: 50px 65px;
}
```



## 6.6 边框

- 边框线：用于改变指定元素的边框线粗细、类型及颜色等，
- border-style: dashed|dotted|double|groove|hidden|inset|outset|ridge|solid|none;
- 下面例子中红字部分

```
<div style=  
    "background:url(images/bear.jpg) no-repeat center center;  
    background-size:contain;  
    width:300px; height:300px;  
    border: 4px solid red;  
    border-radius: 20px;  
    box-shadow: 40px 30px 20px 15px #bbbbbb;  
    padding:20px;  
    color:red">欢迎</div>
```

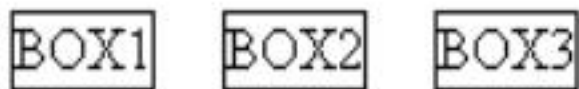


## 6.7 浮动 float

- 浮动：用于改变指定元素与其他元素之间的排列方式、显示方式等。
- float: left|right|none;

值	描述
left	元素向左浮动。
right	元素向右浮动。
none	默认值。元素不浮动，并会显示在其在文本中出现的位置。
inherit	规定应该从父元素继承 float 属性的值。

```
<div id="box1" style="float:left;margin:10px;border:1px solid black;">BOX1</div>  
<div id="box2" style="float:left;margin:10px;border:1px solid black;">BOX2</div>  
<div id="box3" style="float:left;margin:10px;border:1px solid black;">BOX3</div>
```



BOX1 BOX2 BOX3

## 练习

- 打开练习素材→ float 目录下的示例，试试浮动布局的各种效果。

```
<div class="cont">
  <p>当前表情包的浮动属性为none，当鼠标滑过本行文字时，
  <div></div>
  <div></div>
  <div></div>
  <div></div>
</div>
```

```
.cont { /*设置页面的
  background: rgb(225, 255, 255);
  width: 800px;
  height: 520px;
  margin: 0 auto;
}

p { /*设置提示文字的样
  background: #ff0;
  font-size: 20px;
  line-height: 30px;
}

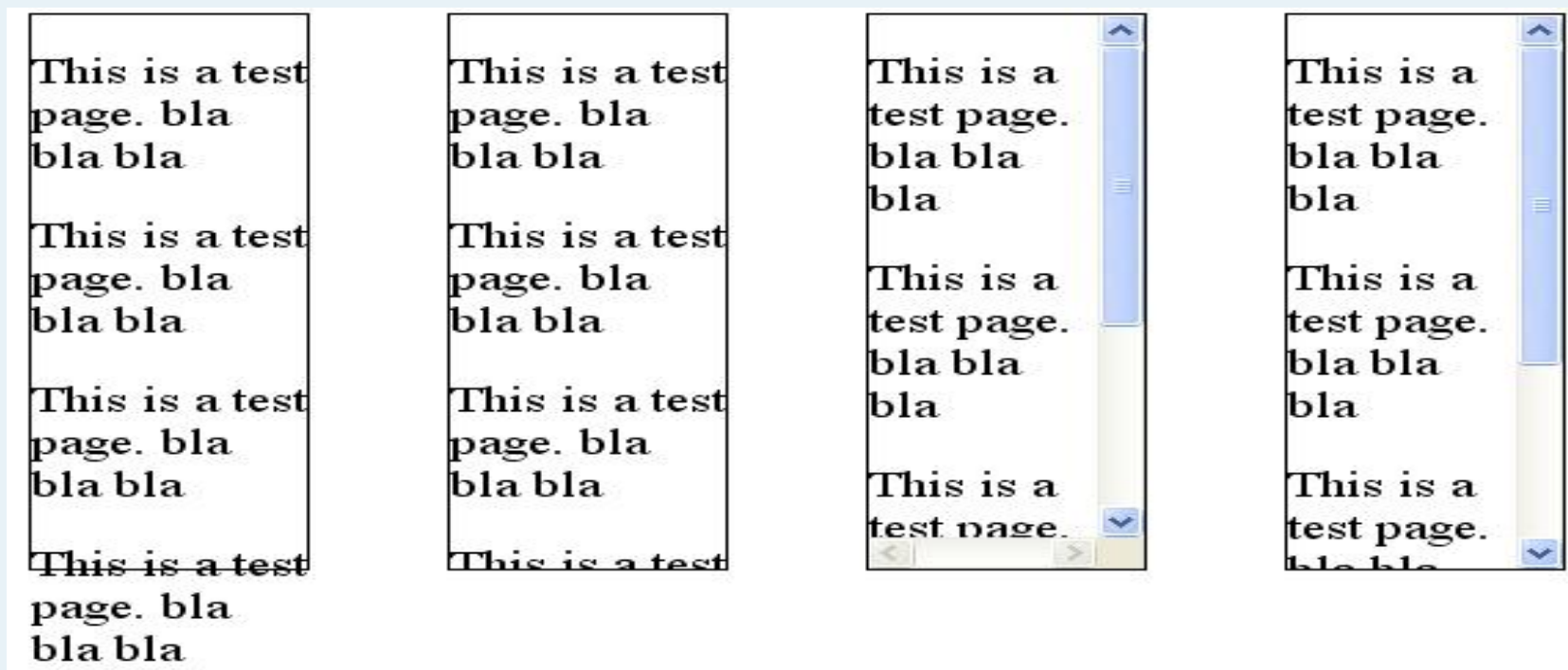
img { /*设置表情图片的
  height: 100px;
  width: 100px;
}

.cont:hover .cont div { /*
  float: left;
}

.cont:active .cont div {
  float: right;
}
```

## 6.8 overflow

- overflow: 常与width和height属性一起使用，当元素中的内容超出width或height属性的设置值时，就会按overflow的属性值决定显示方式。例如：
- 小练习：试试 [https://www.w3school.com.cn/tiy/t.asp?f=csse\\_overflow](https://www.w3school.com.cn/tiy/t.asp?f=csse_overflow)
- overflow:visible    overflow:hidden    overflow:scroll    overflow:auto



## 练习

- 利用不同display属性值，得到如下不同的菜单显示效果

display:inline

我的第一个网页 我喜爱的李白诗 我的爱好 成绩单 可爱的小海雄

display:list-item

■ 我的第一个网页

■ 我喜爱的李白诗

■ 我的爱好

■ 成绩单

■ 可爱的小海雄

display:block

我的第一个网页

我喜爱的李白诗

我的爱好

成绩单

可爱的小海雄

display:block  
float:left

我的第一个网页

我喜爱的李白诗

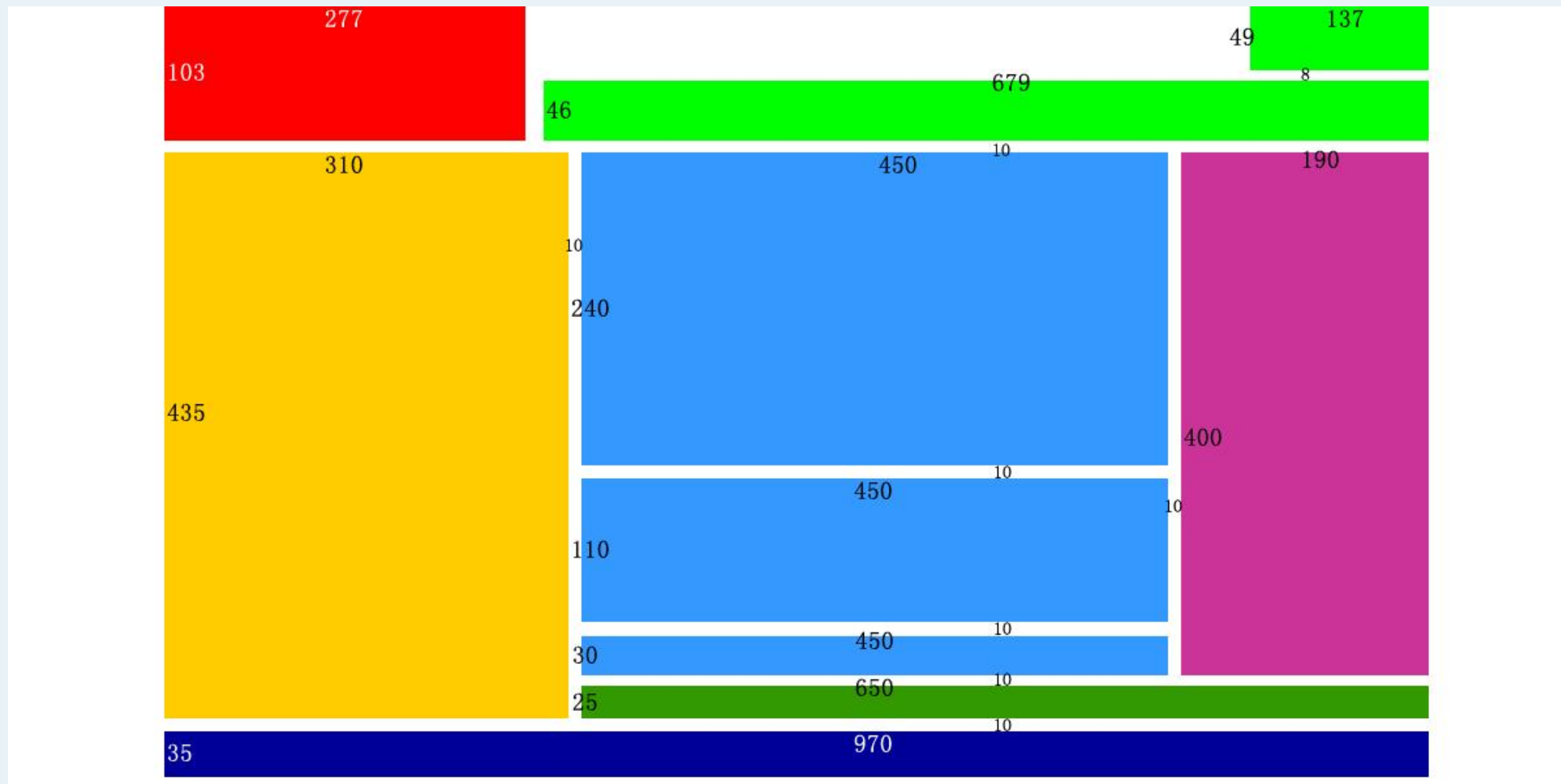
我的爱好

成绩单

可爱的小海雄

# 作业

- 1、使用div+css编写如下网页框架





# 作业

- 用CSS来美化上周的作业个人简历or试卷



2 0 2 4

谢谢！

单击此处添加副标题内容

2024-09