

# CSS布局

盒子模型



黑马程序员  
[www.itheima.com](http://www.itheima.com)

传智教育旗下  
高端IT教育品牌



# 目录

Contents

- ◆ CSS三大特性
- ◆ PxCook的基本使用
- ◆ 盒子模型

# 学习目标

Learning Objectives

- ◆ 能够认识不同选择器的 **优先级** 公式
- ◆ 能够进行 **CSS 权重叠加计算**，分析并解决CSS 冲突问题
- ◆ 能够认识 **盒子模型** 的组成部分
- ◆ 能够掌握盒子模型的 **边框、内边距、外边距** 的作用及简写形式
- ◆ 能够计算盒子的 **实际大小**
- ◆ 能够了解 **外边距折叠现象**，并知道如何解决 **盒子塌陷问题**



# 目录

Contents

- ◆ CSS三大特性
- ◆ PxCook的基本使用
- ◆ 盒子模型

# 一、CSS 三大特性

目标：能认识不同选择器的 **优先级** 公式，能够进行 **CSS 权重叠加计算**，分析并解决CSS 冲突问题

学习路径：

1. 继承性
2. 层叠性
3. **优先级**

# 一、CSS 三大特性

## 3.1 优先级的介绍

- 特性：不同选择器具有不同的优先级，优先级高的选择器样式会覆盖优先级低选择器样式
- 优先级公式：
  - 继承 < 通配符选择器 < 标签选择器 < 类选择器 < id选择器 < 行内样式 < !important
- 注意点：
  1. !important写在属性值的后面，分号的前面！
  2. !important不能提升继承的优先级，**只要是继承优先级最低！**
  3. 实际开发中不建议使用 !important。

# 一、CSS 三大特性

## 小结

- 优先级比较公式是怎样的?
  - 继承 < 通配符选择器 < 标签选择器 < 类选择器 < id选择器 < 行内样式 < !important
- !important能提升继承的优先级吗?
  - 不能

## 3.2 权重叠加计算

- 场景：如果是复合选择器，此时需要通过权重叠加计算方法，判断最终哪个选择器优先级最高会生效
- 权重叠加计算公式：（每一级之间不存在进位）



- 比较规则：
  1. 先比较第一级数字，如果比较出来了，之后的统统不看
  2. 如果第一级数字相同，此时再去比较第二级数字，如果比较出来了，之后的统统不看
  3. ....
  4. 如果最终所有数字都相同，表示优先级相同，则比较层叠性（谁写在下面，谁说了算!）
- 注意点：!important如果不是继承，则权重最高，天下第一！



# 一、CSS 三大特性

## 3.3 小结

- 复合选择器权重叠加计算的公式是怎样的？



## (拓展) 权重叠加计算案例

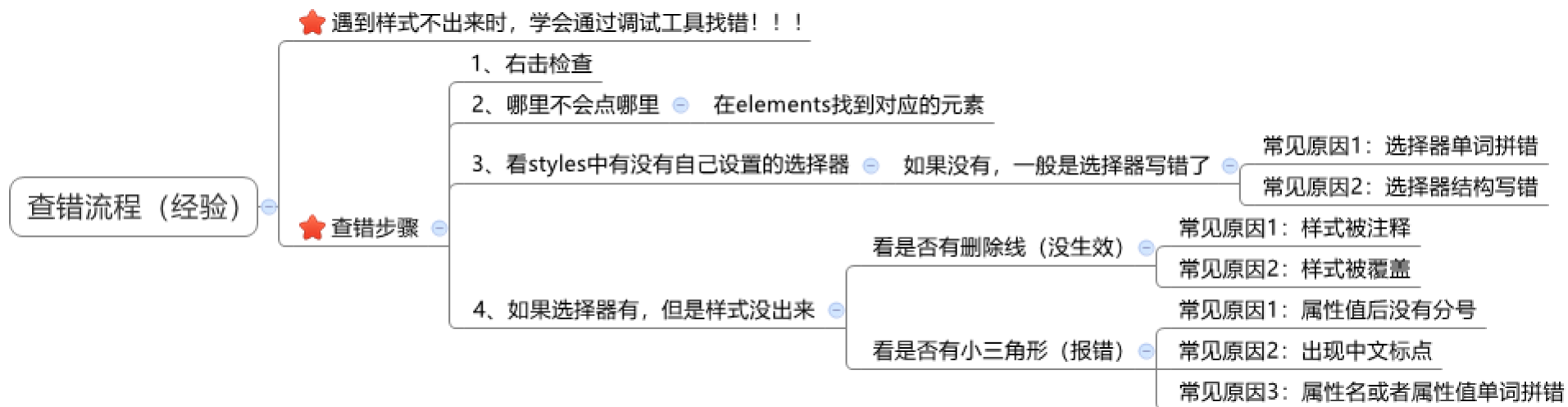
### ➤ 权重计算题解题步骤:

1. 先判断选择器是否能直接选中标签, 如果不能直接选中 → 是继承, 优先级最低 → 直接pass
2. 通过权重计算公式, 判断谁权重最高

### ➤ 注意点:

- 实际开发中选择标签需要精准, 尽量避免多个选择器同时选中一个标签的情况, 不要自己难为自己

## (拓展) 查错流程 (遇到样式出不来, 要学会通过调试工具找错)





# 目录

Contents

- ◆ CSS三大特性
- ◆ PxCook的基本使用
- ◆ 盒子模型

## 二、PxCook的基本使用

**目标：**能够使用 PxCook 工具测量设计图的 **尺寸** 和 **颜色**，能够从psd文件中直接获取数据

**学习路径：**

1. 打开设计图
2. 常用快捷键
3. 从psd文件中直接获取数据

### 1. PxCook的基本使用

#### 1. 通过软件打开设计图

① 打开软件 ② 拖拽入设计图 ③ 新建项目

#### 2. 常用快捷键

1. 放大设计图: **ctrl + +**

2. 缩小设计图: **ctrl + -**

3. 移动设计图: **空格按住不放, 鼠标拖动**

#### 3. 常用工具

1. 量尺寸

2. 吸颜色

#### 4. 从psd文件中直接获取数据

1. 切换到开发界面, 直接点击获取数据



# 目录

Contents

- ◆ CSS三大特性
- ◆ PxCook的基本使用
- ◆ 盒子模型

## 三、盒子模型

目标：能够认识 **盒子模型的组成**，能够掌握盒子模型 **边框、内边距、外边距** 的设置方法

学习路径：

1. **盒子模型的介绍**
2. 内容区域的宽度和高度
3. 边框（border）
4. 内边距（padding）
5. 外边距（margin）



### 1.1 盒子模型的介绍

#### 1. 盒子的概念

1. 页面中的每一个标签，都可看做是一个“盒子”，通过盒子的视角更方便的进行布局
2. 浏览器在渲染（显示）网页时，会将网页中的元素看做是一个个的矩形区域，我们也形象的称之为 盒子

#### 2. 盒子模型

- CSS 中规定每个盒子分别由：内容区域（content）、内边距区域（padding）、边框区域（border）、外边距区域（margin）构成，这就是 盒子模型

#### 3. 记忆：可以联想现实中的包装盒



### 1.2 小结

- 盒子模型一共有几个部分组成？分别是什么？
  1. 内容区域：content
  2. 边框区域：border
  3. 内边距区域：padding
  4. 外边距区域：margin

## 三、盒子模型

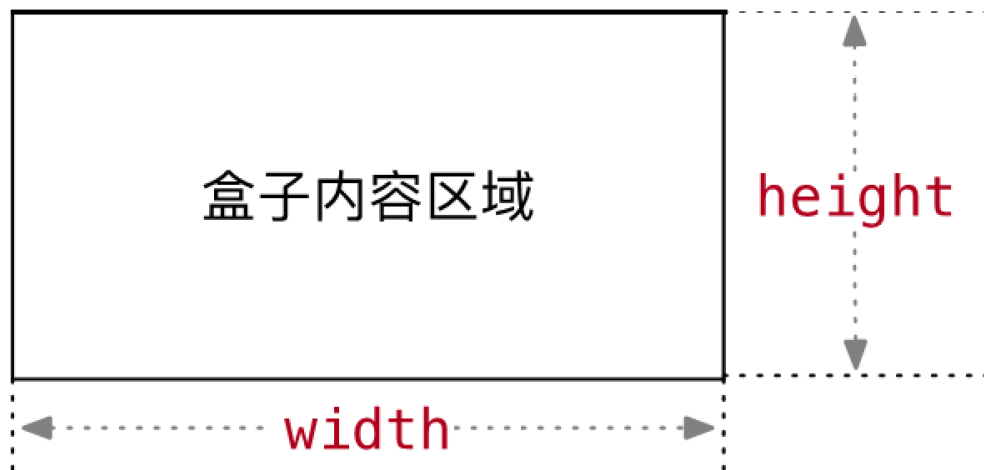
目标：能够认识 **盒子模型的组成**，能够掌握盒子模型 **边框、内边距、外边距** 的设置方法

学习路径：

1. 盒子模型的介绍
2. **内容区域的宽度和高度**
3. 边框（border）
4. 内边距（padding）
5. 外边距（margin）

### 2.1 内容的宽度和高度

- 作用：利用 `width` 和 `height` 属性默认设置是盒子 **内容区域** 的大小
- 属性：`width` / `height`
- 常见取值：数字+`px`



### 2.2 小结

- 设置盒子模型内容区域大小可以通过什么属性设置?
  1. 宽度: width
  2. 高度: height

## 三、盒子模型

目标：能够认识 **盒子模型的组成**，能够掌握盒子模型 **边框、内边距、外边距** 的设置方法

学习路径：

1. 盒子模型的介绍
2. 内容区域的宽度和高度
3. **边框 ( border )**
4. 内边距 ( padding )
5. 外边距 ( margin )

### 3.1 边框 (border) - 单个属性

- 作用：给设置边框粗细、边框样式、边框颜色效果
- 单个属性：

作用	属性名	属性值
边框粗细	border-width	数字+px
边框样式	border-style	实线 <code>solid</code> 、虚线 <code>dashed</code> 、点线 <code>dotted</code>
边框颜色	border-color	颜色取值

### 3.2 边框 (border) - 连写形式

- 属性名: border
- 属性值: 单个取值的连写, 取值之间以空格隔开
  - 如: border : 10px solid red;
- 快捷键: bd + tab



### 3.3 边框 (border) - 单方向设置

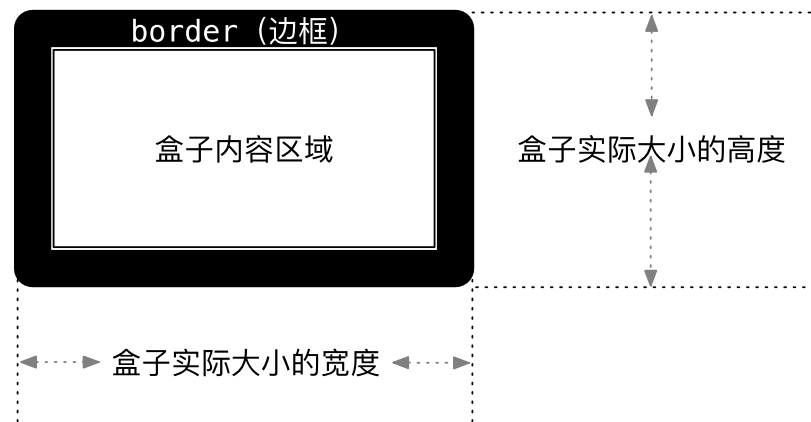
- 场景：只给盒子的某个方向单独设置边框
- 属性名：border - 方位名词
- 属性值：连写的取值

### 小结

- 给盒子设置四周 20像素、实线、蓝色的边框，属性应该如何设置？
  - border: 20px solid blue;
- 给盒子设置上边框 10像素、虚线、黄色的边框，属性应该如何设置？
  - border-top: 10px dashed yellow;

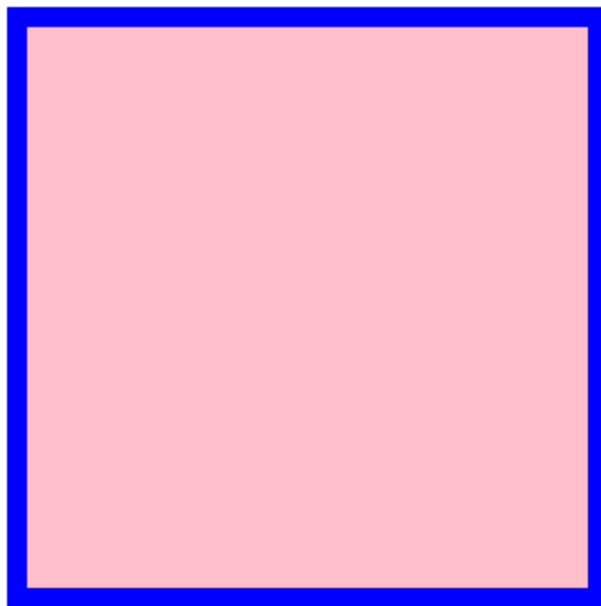
### 3.4 盒子实际大小初级计算公式

- 需求：盒子尺寸 400\*400，背景绿色，边框10px 实线 黑色，如何完成？
  - 注意点：① 设置width和height是内容的宽高！② 设置border会撑大盒子！
- 盒子实际大小初级计算公式：
  - 盒子宽度 = 左边框 + 内容宽度 + 右边框
  - 盒子高度 = 上边框 + 内容高度 + 下边框
- 解决：当盒子被border撑大后，如何满足需求？
  - 解决：计算多余大小，手动在内容中减去（手动内减）



### (案例) 盒子边框的小案例

- 需求：根据设计图，通过PxCook量取数据，通过代码在网页中完成一致的效果



### (案例) 新浪导航案例

- 需求：根据设计图，通过PxCook量取数据，通过代码在网页中完成一致的效果



- 布局顺序：
  1. 从外往内，从上往下
- 每一个盒子的样式：
  1. 宽高
  2. 辅助的背景颜色
  3. 盒子模型的部分：border、padding、margin
  4. 其他样式：color、font-、text-、.....

## 三、盒子模型

目标：能够认识 **盒子模型的组成**，能够掌握盒子模型 **边框、内边距、外边距** 的设置方法

学习路径：

1. 盒子模型的介绍
2. 内容区域的宽度和高度
3. 边框（border）
4. **内边距（padding）**
5. 外边距（margin）

### 4.1 内边距 (padding) - 取值

- 作用：设置 **边框** 与 **内容区域** 之间的距离
- 属性名：padding
- 常见取值：

取值	示例	含义
一个值	<code>padding: 10px;</code>	<b>上右下左</b> 都设置为10px
两个值	<code>padding: 10px 20px;</code>	<b>上下</b> 设置为10px、 <b>左右</b> 设置为20px
三个值	<code>padding: 10px 20px 30px;</code>	<b>上</b> 设置为10px、 <b>左右</b> 设置为20px、 <b>下</b> 设置为30px
四个值	<code>padding: 10px 20px 30px 40px;</code>	<b>上</b> 设置为10px、 <b>右</b> 设置为20px、 <b>下</b> 设置为30px、 <b>左</b> 设置为40px

- 记忆规则：**从上开始赋值，然后顺时针赋值，如果设置赋值的，看对面的！！**

### 4.2 内边距 (padding) - 单方向设置

- 场景：只给盒子的某个方向单独设置内边距
- 属性名：padding - 方位名词
- 属性值：数字 + px

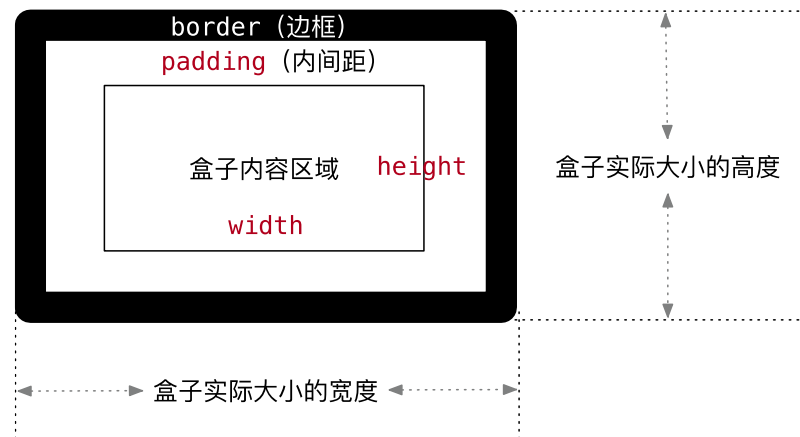


### 小结

- 给盒子设置四周 20px 的内边距可以通过什么属性设置?
  - padding : 20px ;
- 给盒子设置上下20px、左右30px的内边距可以通过什么属性设置?
  - padding : 20px 30px ;
- 给盒子设置左侧50px的内边距可以通过什么属性设置?
  - padding-left : 50px ;

### 4.3 盒子实际大小终极计算公式

- 需求：盒子尺寸300\*300，背景粉色，边框10px实线黑色，上下左右20px的内边距，如何完成？
  - 注意点：① 设置width和height是内容的宽高！② 设置border会撑大盒子 ③ 设置padding会撑大盒子
- 盒子实际大小终极计算公式：
  - 盒子宽度 = 左边框 + 左padding + 内容宽度 + 右padding + 右边框
  - 盒子高度 = 上边框 + 上padding + 内容宽度 + 下padding + 下边框
- 解决：当盒子被border和padding撑大后，如何满足需求？
  - 自己计算多余大小，手动在内容中减去（手动内减）



## 三、盒子模型

### (案例) 新浪导航的优化

- 需求：优化之前的新浪导航，如果每个导航的字数增加，如何完成效果？



### (拓展) 不会撑大盒子的特殊情况

- 不会撑大盒子的特殊情况 (块级元素)
  1. 如果子盒子没有设置宽度, 此时宽度默认是父盒子的宽度
  2. 此时给子盒子设置左右的padding或者左右的border, 此时不会撑大子盒子

### 4.5 CSS3盒模型（自动内减）

- 需求：盒子尺寸300\*300，背景粉色，边框10px实线黑色，上下左右20px的内边距，如何完成？
  - 给盒子设置border或padding时，盒子会被撑大，如果不想盒子被撑大？
- 解决方法 ①：手动内减
  - 操作：自己计算多余大小，手动在内容中减去
  - 缺点：项目中计算量太大，很麻烦
- 解决方法 ②：自动内减
  - 操作：给盒子设置属性 `box-sizing : border-box ;` 即可
  - 优点：浏览器会自动计算多余大小，自动在内容中减去

### (案例) 网页新闻列表案例

- 需求：根据设计图，通过PxCook量取数据，代码在网页中完成一致的效果
- 布局顺序：
  1. 从外往内，从上往下
- 每一个盒子的样式：
  1. 宽高
  2. 辅助的背景颜色
  3. 盒子模型的部分：border、padding、margin
  4. 其他样式：color、font-、text-、.....

#### 最新文章/New Articles

北京招聘网页设计，平面设计，php

体验javascript的魅力

jquery世界来临

网页设计师的梦想

jquery中的链式编程是什么

## 三、盒子模型

目标：能够认识 **盒子模型的组成**，能够掌握盒子模型 **边框、内边距、外边距** 的设置方法

学习路径：

1. 盒子模型的介绍
2. 内容区域的宽度和高度
3. 边框（border）
4. 内边距（padding）
5. **外边距（margin）**

### 5.1 外边距 (margin) - 取值

- 作用：设置边框以外，**盒子与盒子之间**的距离
- 属性名：margin
- 常见取值：

取值	示例	含义
一个值	<code>margin: 10px;</code>	<b>上右下左</b> 都设置为10px
两个值	<code>margin: 10px 20px;</code>	<b>上下</b> 设置为10px、 <b>左右</b> 设置为20px
三个值	<code>margin: 10px 20px 30px;</code>	<b>上</b> 设置为10px、 <b>左右</b> 设置为20px、 <b>下</b> 设置为30px
四个值	<code>margin: 10px 20px 30px 40px;</code>	<b>上</b> 设置为10px、 <b>右</b> 设置为20px、 <b>下</b> 设置为30px、 <b>左</b> 设置为40px

- 记忆规则：**从上开始赋值，然后顺时针赋值，如果设置赋值的，看对面的！！**



### 5.2 外边距 (margin) - 单方向设置

- 场景：只给盒子的某个方向单独设置外边距
- 属性名：margin - 方位名词
- 属性值：数字 + px

### 小结

- 给盒子设置四周 20px 的外边距可以通过什么属性设置?
  - margin : 20px ;
- 给盒子设置上下20px、左右30px的外边距可以通过什么属性设置?
  - margin : 20px 30px ;
- 给盒子设置左侧50px的外边距可以通过什么属性设置?
  - margin-left : 50px ;

### 5.3 margin单方向设置的应用

➤ 应用：

方向	属性	效果
水平方向	margin-left	让当前盒子往右移动
水平方向	margin-right	让右边的盒子往右移动
垂直方向	margin-top	往当前盒子往下移动
垂直方向	margin-bottom	让下面的盒子往下移动

### 5.4 清除默认内外边距

- 场景：浏览器会默认给部分标签设置默认的margin和padding，但一般在项目开始前需要先清除这些标签默认的margin和padding，后续自己设置
  - 比如：body标签默认有margin: 8px
  - 比如：p标签默认有上下的margin
  - 比如：ul标签默认由上下的margin和padding-left
  - .....
- 解决方法：

```
blockquote, body, button, dd, dl, dt, fieldset, form, (index):49  
h1, h2, h3, h4, h5, h6, hr, input, legend, li, ol, p, pre, td,  
textarea, th, ul {  
    margin: 0;  
    padding: 0;  
}
```

淘宝网代码

```
* {  
    margin: 0;  
    padding: 0;  
} first-scrree...chunk.css:1
```

京东代码

### 5.6 外边距正常情况

- 场景：水平布局 的盒子，左右的margin正常，互不影响
- 结果：最终两者距离为左右margin的和

### 5.7 外边距折叠现象 – ① 合并现象

- 场景：垂直布局 的 块级元素，上下的margin会合并
- 结果：最终两者距离为margin的最大值
- 解决方法：避免就好
  - 只给其中一个盒子设置margin即可

### 5.8 外边距折叠现象 – ② 塌陷现象

- 场景：互相嵌套的块级元素，子元素的 `margin-top` 会作用在父元素上
- 结果：导致父元素一起往下移动
- 解决方法：
  1. 给父元素设置 `border-top` 或者 `padding-top`（分隔父子元素的 `margin-top`）
  2. 给父元素设置 `overflow: hidden`
  3. 转换成行内块元素
  4. 设置浮动

### 5.9 行内元素的margin和padding无效情况

- 场景：给行内元素设置margin和padding时
- 结果：
  1. 水平方向的margin和padding布局中有效！
  2. 垂直方向的margin和padding布局中无效！





## 总结

- ◆ 能够认识不同选择器的 **优先级** 公式
- ◆ 能够进行 **CSS 权重叠加计算**，分析并解决CSS 冲突问题
- ◆ 能够认识 **盒子模型** 的组成部分
- ◆ 能够掌握盒子模型的 **边框、内边距、外边距** 的作用及简写形式
- ◆ 能够计算盒子的 **实际大小**
- ◆ 能够了解 **外边距折叠现象**，并知道如何解决 **盒子塌陷问题**



传智教育旗下高端IT教育品牌