

Web前端开发

第11章：盒子模型

阮晓龙

13938213680 / rxl@hactcm.edu.cn

<http://web.book.51xueweb.cn>

<http://www.51xueweb.cn>

河南中医药大学管理科学与工程学科

2018.5

本章主要内容

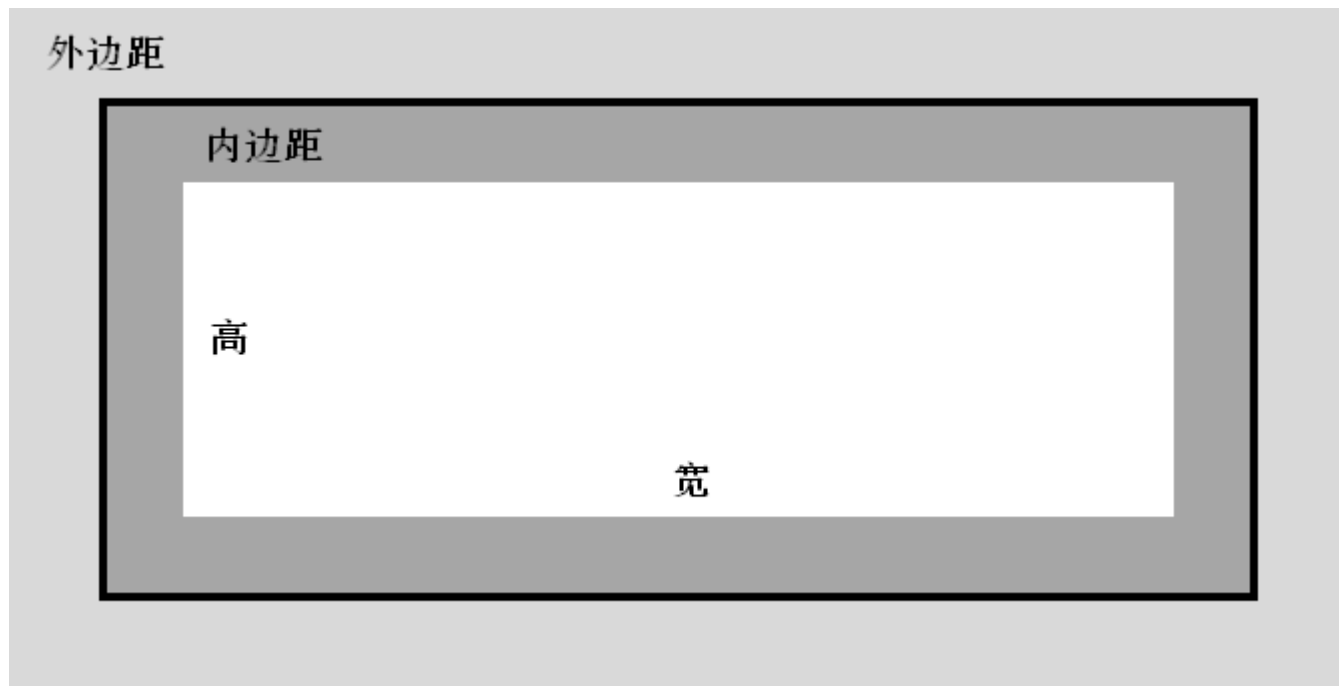
- 盒子
- 盒子类型
- 盒子的属性
- 浏览器的盒子调试



1.盒子

1.1元素盒子

- 盒子是一个概念，也可以说是容器，可以在里面放置网页中需要显示的内容，文档中每一个元素都会产生一个盒子，盒子拥有很多属性，如width、height、padding、border和margin等。
- 元素盒子的最内部分是实际的内容，直接包围内容的是内边距。内边距可以呈现元素的背景。内边距的边缘是边框，边框以外是外边距，外边距默认是透明的，因此不会遮挡其后的元素。



元素盒子的组成

1.盒子

1.1元素盒子

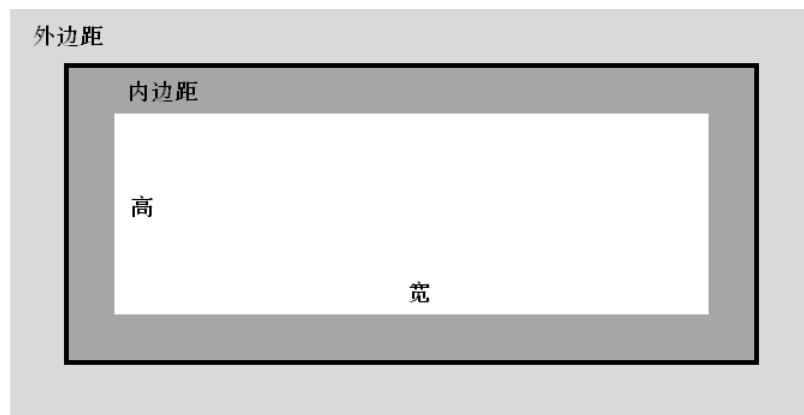
- 盒子的内边距、边框和外边距的宽度都是可选的，其默认值是零。
- 许多元素将会由浏览器自身设置不同的外边距和内边距，可以通过将元素的margin和padding设置为零来覆盖这些浏览器样式，以实现元素在不同浏览器中样式的统一。

1.盒子

1.1元素盒子

□ 外边距

- 外边距相当于文档中的页边距，是元素边框边缘与相邻元素之间的距离，主要用来分割各种元素，设置元素之间的距离。如下图中，浅灰色部分表示盒子的外边距。



1.盒子

1.1元素盒子

□ 外边距

- 使用margin属性来定义元素外边距，属性值单位可以为长度单位（px、pt、em、ex、in等）或百分比，取值可以为正值或负值。
- 同时外边距还有专门设置某一方向上外边距的属性：`margin-top`、`margin-right`、`margin-bottom`、`margin-left`五种属性。

表 11-1 外边距的属性

属性	描述
<code>margin</code>	在一个声明中设置所有外边距属性
<code>margin-bottom</code>	设置元素的下外边距
<code>margin-left</code>	设置元素的左外边距
<code>margin-right</code>	设置元素的右外边距
<code>margin-top</code>	设置元素的上外边距

1.盒子

1.1元素盒子

□ margin属性

- margin可以设置为auto。常见的做法是为外边距设置固定的长度值。

```
h1 {margin : 10px 0px 15px 5px;}
```

- 还可以为margin设置一个百分比数值。百分比数是相对于父元素的width计算的。

```
p {margin : 10%;}
```


1.盒子

1.1元素盒子

□ 值复制

- 在Web前端开发中，有时需要输入一些重复的值。例如：

```
p {margin: 0.5em 1em 0.5em 1em;}
```

- 这种情况下可以使用值复制，不必重复地键入相同的数字。上面的规则与下面的规则是等价的。

```
p {margin: 0.5em 1em;}
```

1.盒子

1.1元素盒子

□ 值复制的规则如下：

- 如果缺少左外边距的值，则使用右外边距的值。
- 如果缺少下外边距的值，则使用上外边距的值。
- 如果缺少右外边距的值，则使用上外边距的值。

1.盒子

1.1元素盒子

□ 值复制的规则如下：

- 如果为外边距指定了3个值，则第4个值（即左外边距）会从第2个值（右外边距）复制得到。
- 如果给定了2个值，第4个值会从第2个值复制得到，第3个值（下外边距）会从第1个值（上外边距）复制得到。
- 如果只给定一个值，那么其他3个外边距都由这个值（上外边距）复制得到。



1.盒子

1.1元素盒子

□ 值复制的规则如下：

- 利用这个简单的机制，只需指定必要的值，而不必全部都应用4个值。例如：

```
h1 {margin: 0.25em 1em 0.5em;} /* 等价于 0.25em 1em 0.5em 1em */  
h2 {margin: 0.5em 1em;} /* 等价于 0.5em 1em 0.5em 1em */  
p {margin: 1px;} /* 等价于 1px 1px 1px 1px */
```

1.盒子

1.1元素盒子

□ 单边外边距属性

- 使用单边外边距属性可以为元素单边上的外边距设置值。例如把p元素的左外边距设置为20px。

```
p {margin-left: 20px;}
```

1.盒子

1.1元素盒子

□ 外边距合并

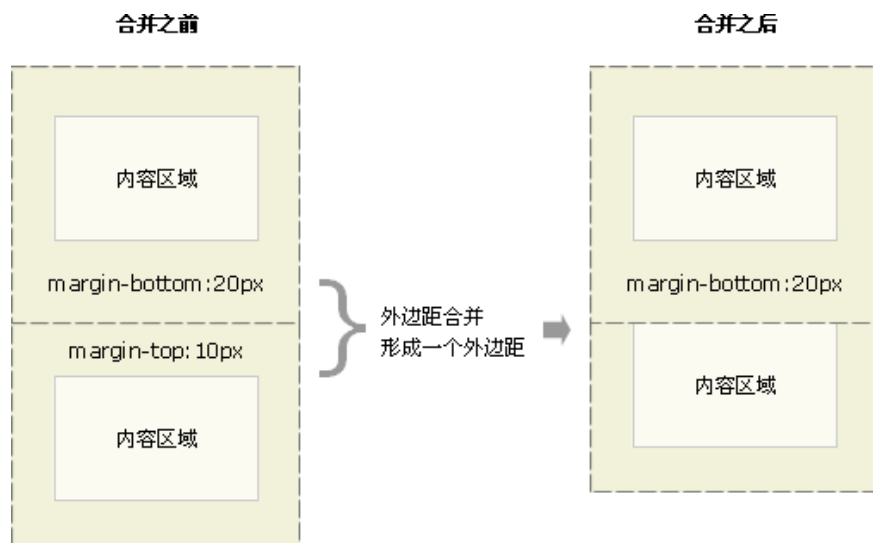
- 外边距合并指的是，当两个垂直外边距相遇时，它们将形成一个外边距。合并后的外边距的高度等于两个发生合并的外边距的高度中的较大者。

1.盒子

1.1元素盒子

□ 外边距合并

- 当一个元素出现在另一个元素上面时，第一个元素的下外边距与第二个元素的上外边距会发生合并。

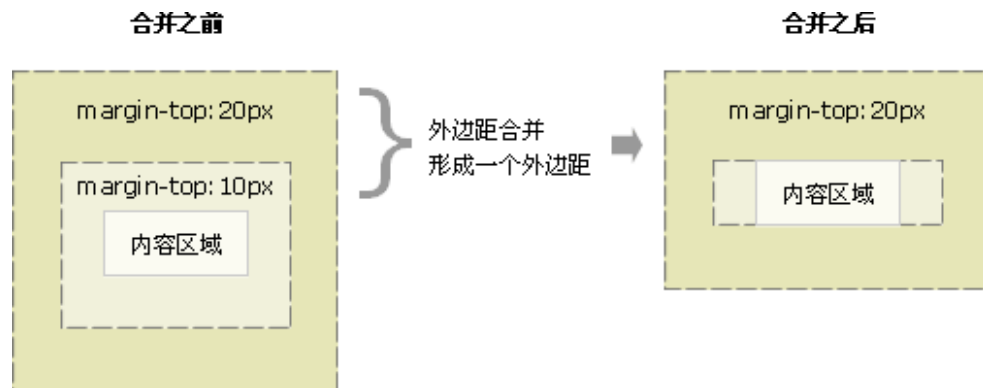


1.盒子

1.1元素盒子

□ 外边距合并

- 当一个元素包含在另一个元素中时（假设没有内边距或边框把外边距分隔开），它们的上和/或下外边距也会发生合并。

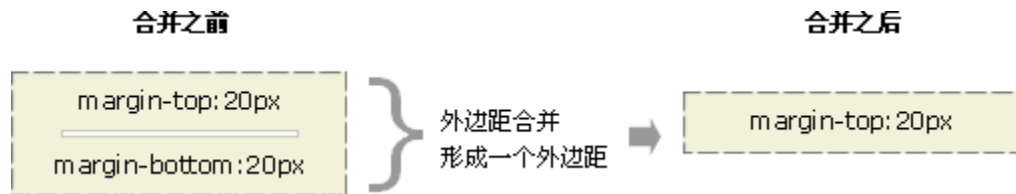


1.盒子

1.1元素盒子

□ 外边距合并

- 一个空元素，它有外边距，但是没有边框或填充，这种情况下，上外边距与下外边距就碰到了一起，就会发生合并。

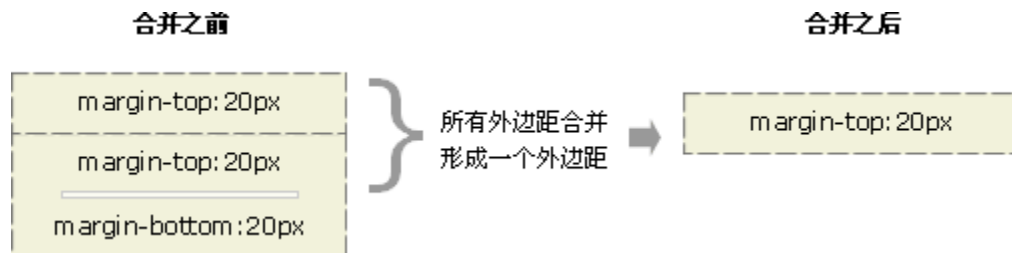


1. 盒子

1.1 元素盒子

□ 外边距合并

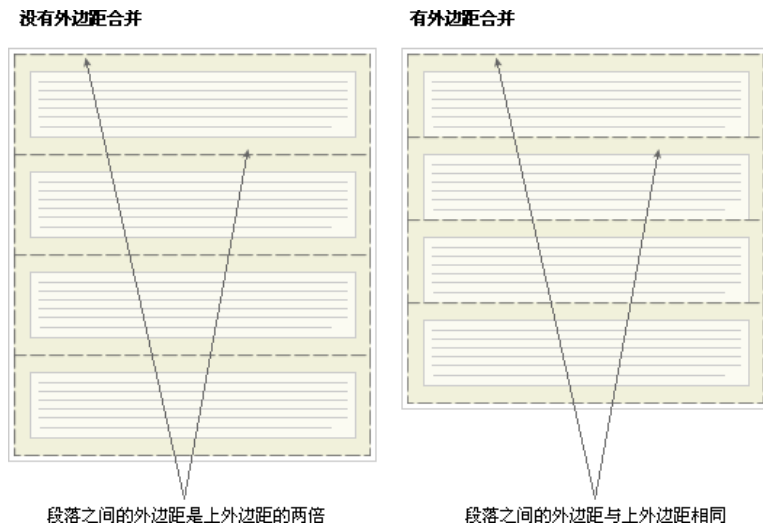
- 如果空元素外边距遇到另一个元素的外边距，也会发生合并。



1.盒子

1.1元素盒子

- 外边距合并初看上去可能有点奇怪，但实际上，它是有意义的。
 - 如下图所示，段落之间的上外边距和下外边距就合并在一起，这样各处的距离就一致了。



注：只有普通文档流中块框的垂直外边距才会发生外边距合并。行内框、浮动框或绝对定位之间的外边距不会合并。



现场演示:

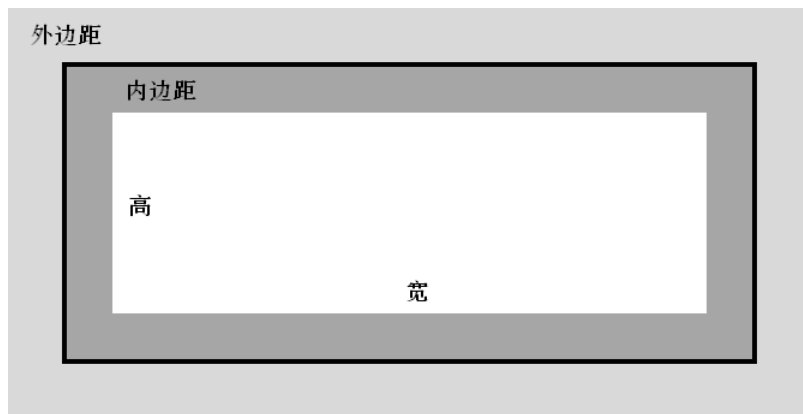
- 案例11-01: 外边距示例

1.盒子

1.1元素盒子

□ 边框：border

- 边框在网页布局中就是用来分割模块的。如下图中，黑色表示盒子的边框。



1.盒子

1.1元素盒子

- 边框可以指定样式、颜色或宽度。宽度属性值可以指定长度值，比如2px或0.1em；或者使用3个关键字thin、medium（默认值）和thick之一。
- 边框颜色可以省略，浏览器就会根据默认值来解析。当元素各边框定义不同颜色时，边角会平分来划分颜色的分布。

1.盒子

1.1元素盒子

□ border常用属性

- 边框有border、border-style、border-width、border-color、border-top、border-right、border-bottom、border-left八种常用属性。

表 11-2 边框的属性

属性	描述
border	简写属性，用于把针对四个边的属性设置在一个声明中
border-style	用于设置元素所有边框的样式，或者单独地为各边设置边框样式
border-width	简写属性，用于为元素的所有边框设置宽度，或者单独地为各边边框设置宽度
border-color	简写属性，设置元素的所有边框中可见部分的颜色，或为四个边分别设置颜色
border-bottom	简写属性，用于把下边框的所有属性设置到一个声明中
border-left	简写属性，用于把左边框的所有属性设置到一个声明中
border-right	简写属性，用于把右边框的所有属性设置到一个声明中
border-top	简写属性，用于把上边框的所有属性设置到一个声明中



现场演示:

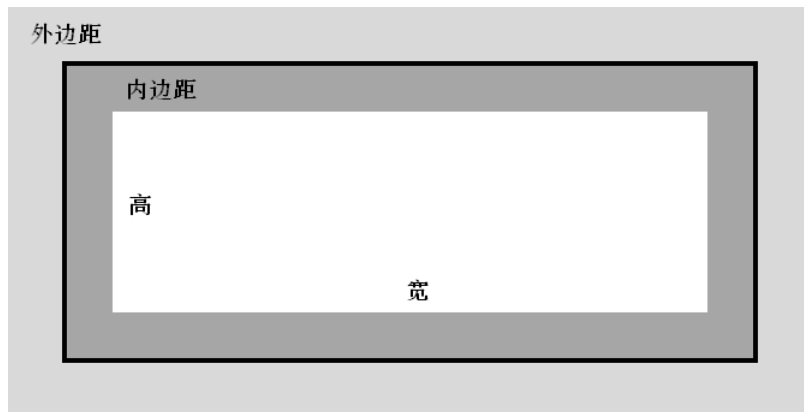
- 案例11-02: 边框

1.盒子

1.1元素盒子

□ 内边距：padding

- 内边距就是元素包含的内容与元素边框内边沿之间的距离。如下图中，深灰色表示盒子的内边距。



1.盒子

1.1元素盒子

□ 内边距：padding

- 定义内边距使用padding属性，属性值单位可以为长度单位（px、pt、em、ex等）或百分比，取值可以为正值，但不允许使用负数值。内边距有padding、padding-top、padding-right、padding-bottom、padding-left五种属性。

1.盒子

1.1元素盒子

□ 内边距：padding

表 11-3 内边距的属性

属性	描述
padding	在一个声明中设置所有内边距属性
padding-bottom	设置元素的下内边距
padding-left	设置元素的左内边距
padding-right	设置元素的右内边距
padding-top	设置元素的上内边距

1.盒子

1.1元素盒子

□ padding属性

- 设置h1元素的各边都有10像素的内边距。

```
h1 {padding: 10px;}
```

- 按照上、右、下、左的顺序分别设置各边的内边距，各边均可以使用不同的单位或百分比值，例如：

```
h1 {padding: 10px 0.25em 2ex 20%;}
```

1.盒子

1.1元素盒子

□ 单边内边距属性

- 通过使用单边内边距属性，可以分别设置上、右、下、左内边距。
例如：

```
h1 {  
  padding-top: 10px;  
  padding-right: 0.25em;  
  padding-bottom: 2ex;  
  padding-left: 20%;  
}
```

1.盒子

1.1元素盒子

□ 内边距的百分比数值

- 下面这条规则把段落的内边距设置为父元素width的10%。

```
p {padding: 10%;}
```

- 如果一个段落的父元素是div元素，那么它的内边距要根据div的width计算。例如：

```
<div style="width: 200px;">  
<p>This paragraph is contained within a DIV that has a width of 200 pixels.</p>  
</div>
```

1.盒子

1.1元素盒子

- 内边距与外边距在用法上有很大的相似性。但是在使用时应该了解内边距的几个不同的特性：
 - 当元素没有定义边框时，可以把内边距当作外边距来使用，用来调节元素与其它元素之间的距离。由于外边距相邻时会出现重叠现象，而且比较复杂，使用内边距来调节元素之间的距离往往不用考虑边距重叠的问题。
 - 当为元素定义背景时，对于外边距区域来说，背景图像是不显示的，它永远表现为透明状态；而内边距区域却可以显示背景。
 - 行内元素的内边距能够影响元素边框的大小，而外边距不存在这样的问题。



现场演示:

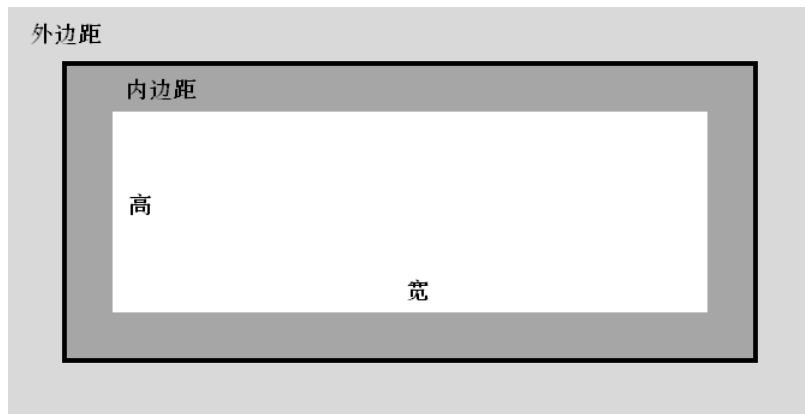
- 案例11-03: 内边距

1.盒子

1.1元素盒子

□ 宽和高：width/height

- 这里要讲的宽和高，指的是元素内容区域的宽和高，不是盒子的实际宽度和高度。如下图中，白色部分表示元素的宽和高。
- 元素的宽度指width的属性值，高度指height的属性值。属性值单位可以为长度单位（px、pt、em、ex、in等）或百分比，取值为正值。

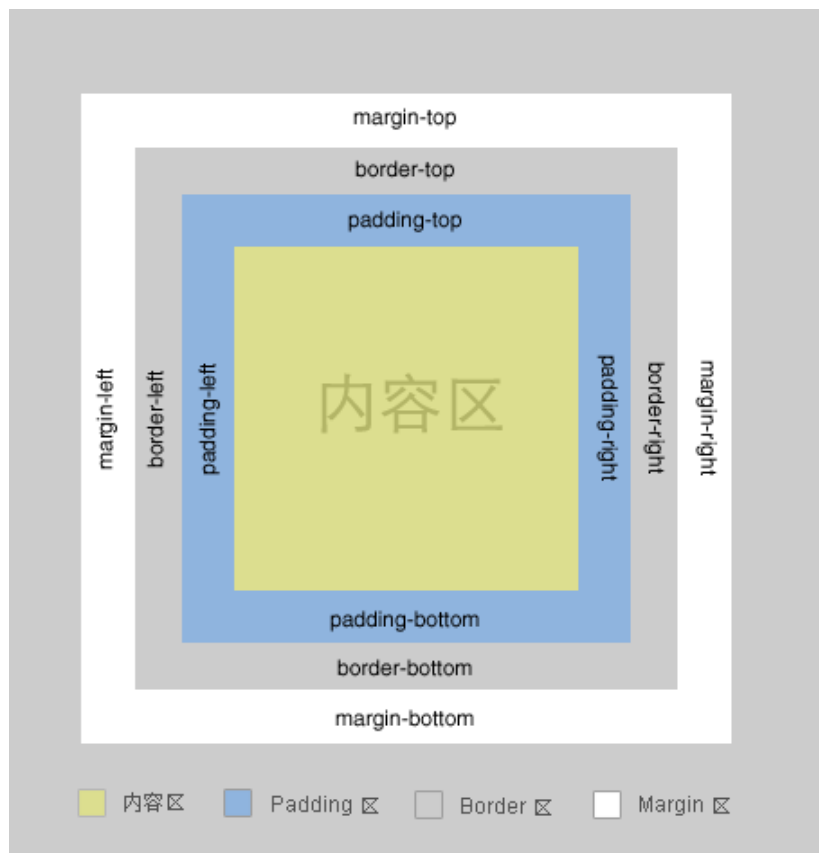


1.盒子

1.2尺寸

□ W3C标准盒子模型

- 目前浏览器大部分元素都是基于此模型，在内容区外面，依次围绕着 padding区，border区，margin区。盒子的宽度和高度计算公式如下：
 - 盒子的宽度= $\text{margin-left} + \text{border-left} + \text{padding-left} + \text{width} + \text{padding-right} + \text{border-right} + \text{margin-right}$ 。
 - 盒子的高度= $\text{margin-top} + \text{border-top} + \text{padding-top} + \text{height} + \text{padding-bottom} + \text{border-bottom} + \text{margin-bottom}$ 。



盒子的尺寸

1.盒子

1.2尺寸

□ IE传统盒子模型

- 这种盒子模型主要存在于IE6以下版本浏览器（不包括IE6或QuirksMode下IE5.5+）。
- 虽然市面上用IE6以下的浏览器应该相当少，但是由于form中部分元素还基于传统的盒子模型此种模型依然被CSS3保留使用。盒子的宽度和高度计算公式如下：
 - 盒子的宽度= $\text{margin-left} + \text{width} + \text{margin-right}$ 。
 - 盒子的高度= $\text{margin-top} + \text{height} + \text{margin-bottom}$ 。

2.盒子类型

2.1盒子的基本类型

- CSS提供了display属性来控制盒子的类型，盒子的基本类型如表所示。

表 11-4 盒子的基本类型

类型	描述
none	此元素不会被显示
block	此元素将显示为块级元素，并且前后会带有换行符
inline	默认。此元素会被显示为内联元素，元素前后没有换行符
inline-block	行内块元素
list-item	此元素会作为列表显示
table	此元素会作为块级表格来显示（类似<table>），表格前后带有换行符
inline-table	此元素会作为内联表格来显示（类似<table>），表格前后没有换行符
table-row-group	此元素会作为一个或多个行的分组来显示（类似<tbody>）

表 11-4 盒子的基本类型

类型	描述
table-column-group	此元素会作为一个或多个列的分组来显示（类似<colgroup>）
table-column	此元素会作为一个单元格列显示（类似<col>）
table-cell	此元素会作为一个表格单元格显示（类似<td>和<th>）
table-caption	此元素会作为一个表格标题显示（类似<caption>）
inherit	规定应该从父元素继承 display 属性的值
run-in	分配对象为块对象或基于内容之上的内联对象

2.盒子类型

2.1盒子的基本类型

□ none

- display属性可指定为none值，用于设置目标对象隐藏，一旦该对象隐藏，其占用的页面空间也会释放。与此类似的还有visibility属性，该属性也可用于设置目标对象是否显示。与display属性不同，当通过visibility隐藏某个HTML元素后，该元素占用的页面空间依然会被保留。visibility属性的两个常用值为visible和hidden，分别用于控制目标对象的显示和隐藏。



现场演示:

- 案例11-04: `display`属性值none

2.盒子类型

2.1盒子的基本类型

□ block

- block类型的盒子会占据一行，允许通过CSS设置高度和宽度。一些元素默认就是block类型，比如div、p等。

□ inline

- inline类型的盒子不会占据一行（默认允许在一行放置多个元素），即使用CSS设置宽度和高度也不会起作用。一些元素默认就是inline类型，比如span、a。



现场演示:

- 案例11-05: display属性值block与inline

2.盒子类型

2.1盒子的基本类型

□ inline-block

- 通过为display属性设置inline-block，即可实现这种盒子类型，它是inline和block的综合体，inline-block类型既不会占据一行，也支持width和height指定宽度和高度。



现场演示:

- 案例11-06: `display`属性值`inline-block`

2.盒子类型

2.1盒子的基本类型

□ list-item

- list-item可以将目标元素转换为类似u1的列表元素，也可以同时在元素前添加列表标志。



现场演示:

- 案例11-07: display属性值list-item

2.盒子类型

2.1盒子的基本类型

□ inline-table

- 在默认情况下，table元素属于block类型，也就是说，该元素默认占据一行，它的左边不允许出现其他内容，它的右边也不允许出现其他内容。
- CSS为table元素提供了一个inline-table类型，它允许设置表格的width、height值，而且允许它的左边、右边出现其他内容。



现场演示:

- 案例11-08: display属性值inline-table

2.盒子类型

2.1盒子的基本类型

□ 表格相关的盒子类型

■ 除inline-table类型外，CSS3还提供了如下属性值：

- table：此元素会作为块级表格来显示，表格前后带有换行符。
- table-row-group：此元素会作为一个或多个行的分组来显示。
- table-header-group：此元素会作为一个或多个行的分组来显示。
- table-footer-group：此元素会作为一个或多个行的分组来显示。
- table-row：此元素会作为一个表格行显示。
- table-column-group：此元素会作为一个或多个列的分组来显示。
- table-column：此元素会作为一个单元格列显示。
- table-cell：此元素会作为一个表格单元格显示。
- table-caption：此元素会作为一个表格标题显示。



现场演示：

- 案例11-09：表格相关的盒子类型

2.盒子类型

2.1盒子的基本类型

□ run-in

- run-in类型有点相似于inline类型，run-in类型的元素希望显示在他后面的元素内部；如果run-in类型的元素后面紧跟一个block类型的元素，那么run-in类型的元素将被放入后面的元素中显示。



现场演示:

- 案例11-10: display属性值run-in

2.盒子类型

2.2 CSS3新增的类型

- CSS3对display的属性值进行了补充，CSS3新增的类型如表所示。

表 11-5 CSS3 新增类型

元素名	描述
header	标记头部区域的内容（用于整个页面或者页面中的一块区域）
footer	标记脚部区域的内容（用于整个页面或者页面中的一块区域）
ruby	Web 页面中的一块区域
article	独立的文章内容
aside	相关内容或者引文
nav	导航类辅助内容

2.盒子类型

2.2 CSS3新增的类型

□ ruby

- ruby相关的盒子类型是CSS3新增的盒子类型，常用于在文本注释或标注文本的发音，Chrome中暂不支持该类型，可在Firefox、IE8及以上版本浏览器中测试ruby相关的盒子类型。



现场演示:

- 案例11-11: ruby相关的盒子类型

2. 盒子类型

2.2 CSS3新增的类型

□ box

- box类型还没有得到Firefox、Opera、Chrome浏览器的完全支持，但可以使用它们的私有属性定义firefox(-moz-)、opera(-o-)、chrome/safari (-webkit-)。
- box类型盒子的属性说明如表所示。

表 11-6 box 类型盒子的属性

属性	描述
box-align	规定如何对齐框的子元素
box-pack	规定当框大于子元素的尺寸时，在何处放置子元素
box-direction	规定框的子元素的显示方向
box-flex	规定框的子元素是否可伸缩
box-flex-group	将可伸缩元素分配到柔性分组
box-lines	规定当超出父元素框的空间时，是否换行显示
box-ordinal-group	规定框的子元素的显示次序

2. 盒子类型

2.2 CSS3新增的类型

- `box-align`有`start`、`end`、`center`、`baseline`、`stretch`五种属性值。

表 11-7 `box-align` 的属性值

属性值	描述
<code>start</code>	对于正常方向的框，每个子元素的上边缘沿着框的顶边放置 对于相反方向的框，每个子元素的下边缘沿着框的底边放置
<code>end</code>	对于正常方向的框，每个子元素的下边缘沿着框的底边放置 对于相反方向的框，每个子元素的上边缘沿着框的顶边放置
<code>center</code>	均等地分割多余的空间，一半位于子元素之上，另一半位于子元素之下
<code>baseline</code>	如果 <code>box-orient</code> 是 <code>inline-axis</code> 或 <code>horizontal</code> ，所有子元素均与其基线对齐
<code>stretch</code>	拉伸子元素以填充包含块

2. 盒子类型

2.2 CSS3新增的类型

□ box-pack有 start、end、center、justify四种属性值。

表 11-8 box-pack 的属性值

属性值	描述
start	对于正常方向的框，首个子元素的左边缘被放在左侧（最后子元素后是所有剩余的空间） 对于相反方向的框，最后子元素的右边缘被放在右侧（首个子元素前是所有剩余的空间）
end	对于正常方向的框，最后子元素的右边缘被放在右侧（首个子元素前是所有剩余的空间） 对于相反方向的框，首个子元素的左边缘被放在左侧（最后子元素后是所有剩余的空间）
center	均等地分割多余空间，其中一半空间被置于首个子元素前，另一半被置于最后子元素后
justify	在每个子元素之间分割多余的空间（首个子元素前和最后子元素后没有多余的空间）



现场演示:

- 案例11-12: 属性`box-align`和`box-pack`

2.盒子类型

2.2 CSS3新增的类型

□ `box-direction`有`normal`、`reverse`、`inherit`三种属性值。

表 11-9 `box-direction` 的属性值

属性值	描述
<code>normal</code>	以默认方向显示子元素
<code>reverse</code>	以反方向显示子元素
<code>inherit</code>	应该从子元素继承 <code>box-direction</code> 属性的值



现场演示:

- 案例11-13: 属性box-direction

2.盒子类型

2.2 CSS3新增的类型

- `box-flex`属性规定框的子元素是否可伸缩其尺寸。



现场演示:

- 案例11-14: 属性box-fiex

2.盒子类型

2.2 CSS3新增的类型

- `box-ordinal-group`属性规定框中子元素的显示次序，值更低的元素会显示在值更高的元素前面显示，其取值只能为整数。



现场演示:

- 案例11-15: 属性box-ordinal-group

2.盒子类型

2.3浏览器对盒子的支持情况

- 各浏览器对盒子的支持情况如表所示。

表 11-10 浏览器对各种盒子类型的支持情况

类型	Firefox	Safari	Opera	IE8	IE9	IE10	Chrome
none	✓	✓	✓	✓	✓	✓	✓
block	✓	✓	✓	✓	✓	✓	✓
inline	✓	✓	✓	✓	✓	✓	✓
inline-block	✓	✓	✓	✓	✓	✓	✓
list-item	✓	✓	✓	✓	✓	✓	✓
table	✓	✓	✓	✓	✓	✓	✓
inline-table	✓	✓	✓	✓	✓	✓	✓
table-row-group	✓	✓	✓	✓	✓	✓	✓
table-header-group	✓	✓	✓	✓	✓	✓	✓
table-footer-group	✓	✓	✓	✓	✓	✓	✓
table-row	✓	✓	✓	✓	✓	✓	✓

表 11-10 浏览器对各种盒子类型的支持情况

类型	Firefox	Safari	Opera	IE8	IE9	IE10	Chrome
table-column-group	✓	✓	×	✓	✓	✓	×
table-column	✓	✓	×	✓	✓	✓	×
table-cell	✓	✓	✓	✓	✓	✓	✓
table-caption	✓	✓	✓	✓	✓	✓	✓
inherit	✓	✓	✓	✓	✓	✓	✓
run-in	✓	✓	✓	✓	✓	✓	✓
ruby	✓	×	×	✓	✓	✓	×
ruby-base	✓	×	×	✓	✓	✓	×
ruby-text	✓	×	×	✓	✓	✓	×
ruby-base-group	✓	×	×	✓	✓	✓	×
ruby-text-group	✓	×	×	✓	✓	✓	×
box	✓	✓	✓	×	×	×	✓
inline-box	✓	✓	✓	×	×	×	✓

3.盒子的属性

3.1内容溢出

- 如在样式中指定了盒子的宽度和高度，就可能出现内容在盒子中容纳不下的情况，可以使用`overflow`属性来指定如何显示盒子中容纳不下的内容。同时，与`overflow`属性相关的还有`overflow-x`、`overflow-y`及`text-overflow`属性。

3.盒子的属性

3.1内容溢出

□ overflow

- 使用overflow属性来指定对于盒子中容纳不下的内容的显示方法。overflow有visible、hidden、scroll、auto、inherit五种属性值。

表 11-11 overflow 的属性值

属性值	描述
visible	默认值。内容不会被修剪，会呈现在元素框之外
hidden	内容会被修剪，并且其余内容是不可见的
scroll	内容会被修剪，但是浏览器会显示滚动条以便查看其余的内容
auto	如果内容被修剪，则浏览器会显示滚动条以便查看其余的内容
inherit	规定应该从父元素继承 overflow 属性的值



现场演示:

- 案例11-16: 属性overflow

3.盒子的属性

3.1内容溢出

□ overflow-x和overflow-y

- 使用overflow-x属性或overflow-y属性，可以单独指定在水平方向上或垂直方向上内容超出盒子的容纳范围时的显示方法，使用方法与overflow属性的使用方法相似。

3.盒子的属性

3.1内容溢出

□ overflow-x和overflow-y

- overflow-x和overflow-y有visible、hidden、scroll、auto、no-display、no-content六种属性值，如表所示。

表 11-11 overflow 的属性值

属性值	描述
visible	默认值。内容不会被修剪，会呈现在元素框之外
hidden	内容会被修剪，并且其余内容是不可见的
scroll	内容会被修剪，但是浏览器会显示滚动条以便查看其余的内容
auto	如果内容被修剪，则浏览器会显示滚动条以便查看其余的内容
inherit	规定应该从父元素继承 overflow 属性的值



现场演示:

- 案例11-17: 属性`overflow-x`和属性`overflow-y`

3.盒子的属性

3.1内容溢出

□ text-overflow

- 当通过把overflow属性的属性值设为“hidden”的方法，将盒子中容纳不下的内容隐藏起来时，如果使用text-overflow属性，可以在盒子的末尾显示一个代表省略的符号“...”。但是，text-overflow属性只在当盒子中内容水平方向上超出盒子的容纳范围时有效。
- text-overflow有clip、ellipsis、string三种属性值，如表所示。

表 11-13 text-overflow 的属性值

属性值	描述
clip	修剪文本
ellipsis	显示省略符号来代表被修剪的文本
string	使用给定的字符串来代表被修剪的文本



现场演示:

- 案例11-18: 属性text-overflow

3.盒子的属性

3.2自由缩放

- 为了增强用户体验，CSS增加了一个非常实用的属性`resize`，它允许用户通过拖动的方式来修改元素的尺寸，改变元素的大小。
- `resize`有`none`、`both`、`horizontal`、`vertical`、`inherit`五种属性值，如表所示。

表 11-14 `resize` 的属性值

属性值	描述
<code>none</code>	用户无法调整元素的尺寸
<code>both</code>	用户可调整元素的高度和宽度
<code>horizontal</code>	用户可调整元素的宽度
<code>vertical</code>	用户可调整元素的高度
<code>inherit</code>	继承父级元素的 <code>resize</code> 属性值



现场演示:

- 案例11-19: 属性resize

3.盒子的属性

3.3外轮廓

- 外轮廓outline在页面中呈现的效果和边框border呈现的效果极其相似，但和元素边框border完全不同，外轮廓线不占用网页布局空间，不一定是矩形，外轮廓属于一种动态样式，只有元素获取到焦点或者被激活时呈现。
- 外轮廓有outline-color、outline-style、outline-width、outline-offset、inherit五种属性，如表所示。

表 11-15 外轮廓的属性

属性值	描述
outline-color	规定边框的颜色
outline-style	规定边框的样式
outline-width	规定边框的宽度
outline-offset	规定外轮廓的偏移位置的数值
inherit	继承父级元素的 outline 属性值

3.盒子的属性

3.3外轮廓

□ outline-color

- 定义外轮廓线的颜色，默认值为黑色。属性值为CSS中定义的颜色值。可以为颜色值为颜色名称（比如 red、blue）、十六进制值（比如 #ff0000、#cccccc）、rgb 代码（比如 rgb(255,0,0)、rgb(167,167,167)）。

3.盒子的属性

3.3外轮廓

□ outline-style

- 定义外轮廓线的样式，在实际应用中，省略此参数外轮廓默认值为 none，不绘制外轮廓线。outline-style的属性值如表所示。

表 11-16 outline-style 的属性值

属性值	描述
none	默认。定义无轮廓
dotted	定义点状的轮廓
dashed	定义虚线轮廓
solid	定义实线轮廓
double	定义双线轮廓。双线的宽度等同于 outline-width 的值
groove	定义 3D 凹槽轮廓。此效果取决于 outline-color 值
ridge	定义 3D 凸槽轮廓。此效果取决于 outline-color 值
inset	定义 3D 凹边轮廓。此效果取决于 outline-color 值
outset	定义 3D 凸边轮廓。此效果取决于 outline-color 值
inherit	规定应该从父元素继承轮廓样式的设置

3.盒子的属性

3.3外轮廓

□ outline-width

- 定义外轮廓的宽度。在实际应用中，省略此参数外轮廓默认值为medium，表示绘制中等宽度的外轮廓线。outline-width的属性值如表所示。

表 11-17 outline-width 的属性值

属性值	描述
thin	规定细的轮廓
medium	默认。规定中等的轮廓
thick	规定粗的轮廓
length	允许你规定轮廓粗细的值（只能为正值）
inherit	规定应该从父元素继承轮廓宽度的设置

3.盒子的属性

3.3外轮廓

□ outline-offset

- 定义外轮廓的偏移位置，此值可以为负值。当此参数的值为正数值，表示外轮廓向外偏离多少个像素；当此参数的值为负数值，表示外轮廓向内偏移了多少像素。



现场演示：

- 案例11-20：外轮廓

3.盒子的属性

3.4阴影

- 在CSS3中，可以使用`box-shadow`属性让盒子在显示时产生阴影效果。

3.盒子的属性

3.4阴影

□ box-shadow属性

- box-shadow属性可以为所有盒子模型的元素整体增加阴影，是一个复合属性。该属性值如表所示。

表 11-18 box-shadow 属性值

属性	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（outest）改为内部阴影



现场演示:

- 案例11-21: 属性box-shadow

3.盒子的属性

3.4阴影

- 对第一个文字或第一行使用阴影
 - 使用`first-letter`选择器或`first-line`选择器可以只让第一个文字具有阴影效果。



现场演示：

- 案例11-22：对第一个文字实阴影

3.盒子的属性

3.4阴影

- 对表格及单元格使用阴影。
 - 使用`box-shadow`属性可以让表格及表格内的单元格产生阴影效果。

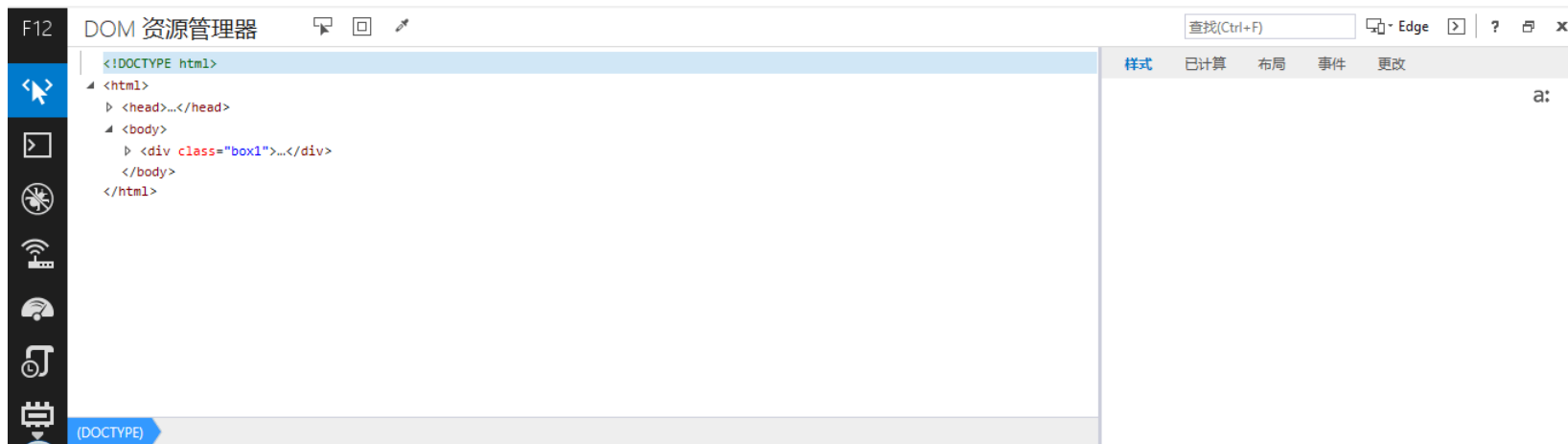


现场演示:

- 案例11-23: 对表格及单元格使用阴影

4.浏览器的盒子调试

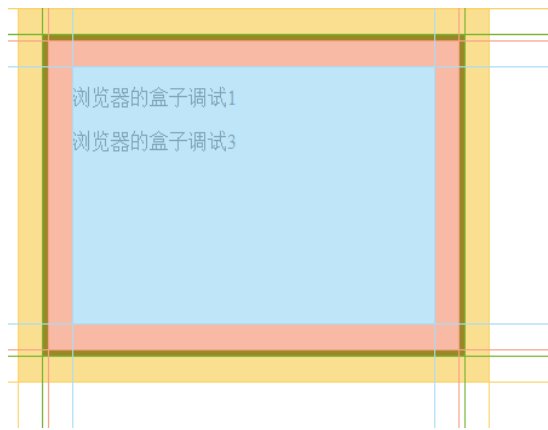
- 在Internet Explorer浏览器中进行盒子调试





```
样式 已计算 布局 事件 更改  
} a:  
13-19.html (7)  
.box1 {  
  margin-top: 20px;  
  margin-right: 20px;  
  margin-bottom: 20px;  
  margin-left: 20px;  
  border-color: red;  
  border-style: solid;  
  border-width: 5px;  
  width: 300px;  
  height: 200px;  
  padding-top: 20px;  
  padding-right: 20px;  
  padding-bottom: 20px;  
  padding-left: 20px;  
}
```

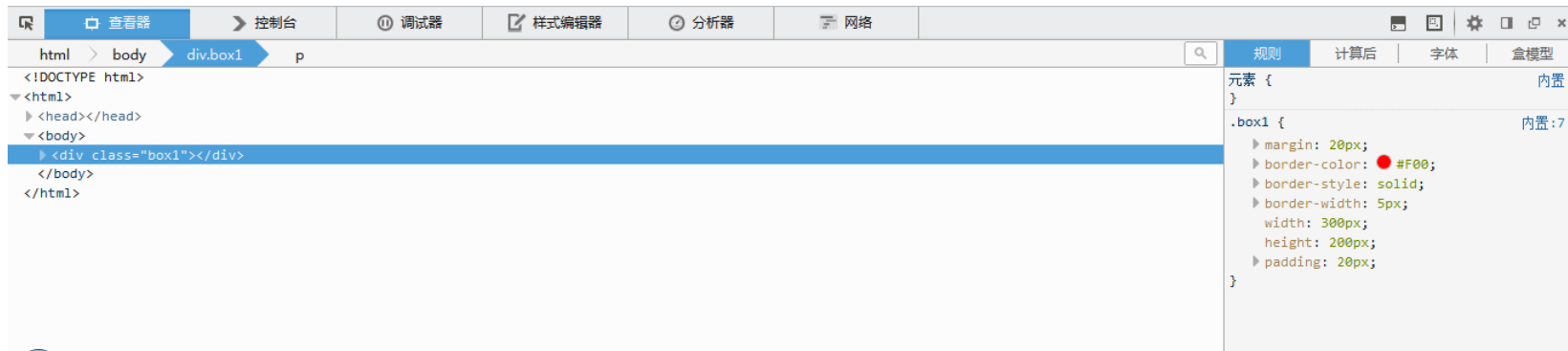
盒子的样式

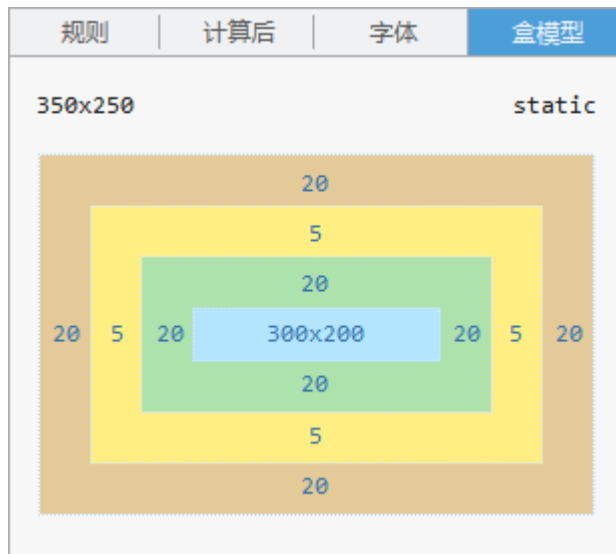


盒子的布局

4.浏览器的盒子调试

□ 在Firefox浏览器中进行盒子调试





盒子元素模型

4.浏览器的盒子调试

- 在Google Chrome浏览器中进行盒子调试



```
.box1 { 13-19.html:7  
  ✓ margin-top: 20px;  
  ✓ margin-right: 20px;  
  ✓ margin-bottom: 20px;  
  ✓ margin-left: 20px;  
  ✓ border-color: ▶ red;  
  ✓ border-style: ▶ solid;  
  ✓ border-width: ▶ 5px;  
  ✓ width: 300px;  
  ✓ height: 200px;  
  ✓ padding-top: 20px;  
  ✓ padding-right: 20px;  
  ✓ padding-bottom: 20px;  
  ✓ padding-left: 20px;  
}  
+  
div { user agent stylesheet  
  display: block;  
}
```

The diagram illustrates the box model for a 300x200px element. The element is a blue rectangle with a width of 300px and a height of 200px. It is surrounded by a 5px solid red border. Inside the border is a 20px padding area (green dashed box). The entire box is surrounded by a 20px margin area (orange dashed box). The diagram shows the margin, border, and padding values for each side of the box.

盒子元素的样式和模型



现场演示：

- 案例11-24：浏览器中的盒子调试

Thanks.