盒子模型

讲师：李立超

在网页中“一切皆是盒子”

盒子模型

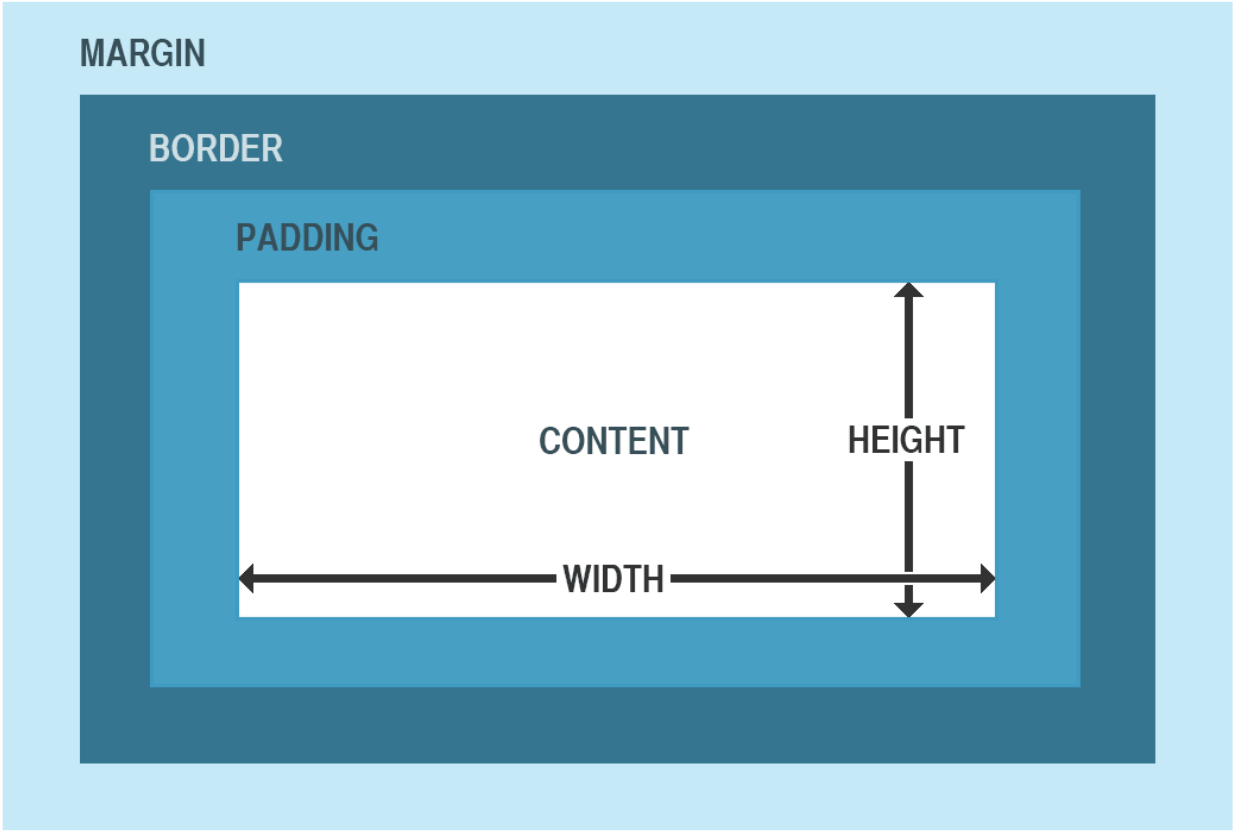
盒子

- CSS处理网页时，它认为每个元素都包含在一个不可见的盒子里。
- 为什么要想象成盒子呢？因为如果把所有的元素都想象成盒子，那么我们对网页的布局就相当于是摆放盒子。
- 我们只需要将相应的盒子摆放到网页中相应的位置即可完成网页的布局。

盒子模型

- 一个盒子我们会分成几个部分：
  - 内容区(content)
  - 内边距(padding)
  - 边框(border)
  - 外边距(margin)

盒子模型



内容区

- 内容区指的是盒子中放置内容的区域，也就是元素中的文本内容，子元素都是存在于内容区中的。
- 如果没有为元素设置内边距和边框，则内容区大小默认和盒子大小是一致的。
- 通过width和height两个属性可以设置内容区的大小。
- width和height属性只适用于块元素。

内边距

- 顾名思义，内边距指的就是元素内容区与边框以内的空间。
- 默认情况下width和height不包含padding的大小。
- 使用padding属性来设置元素的内边距。
- 例如：
  - padding:10px 20px 30px 40px
  - 这样会设置元素的上、右、下、左四个方向的内边距。

内边距

- padding:10px 20px 30px;
  - 分别指定上、左右、下四个方向的内边距
- padding:10px 20px;
  - 分别指定上下、左右四个方向的内边距
- padding:10px;
  - 同时指定上左右下四个方向的内边距
- 同时在css中还提供了padding-top、padding-right、padding-right、padding-bottom分别用来指定四个方向的内边距。

边框

- 可以在元素周围创建边框，边框是元素可见框的最外部。
- 可以使用border属性来设置盒子的边框：
  - border:1px red solid;
  - 上边的样式分别指定了边框的宽度、颜色和样式。
- 也可以使用border-top/left/right/bottom分别指定上右下左四个方向的边框。
- 和padding一样，默认width和height并包括边框的宽度。

边框的样式

- 边框可以设置多种样式：
  - none（没有边框）
  - dotted（点线）
  - dashed（虚线）
  - solid（实线）
  - double（双线）
  - groove（槽线）
  - ridge（脊线）
  - inset（凹边）
  - outset（凸边）

外边距

- 外边距是元素边框与周围元素相距的空间。
- 使用margin属性可以设置外边距。
- 用法和padding类似，同样也提供了四个方向的margin-top/right/bottom/left。
- 当将左右外边距设置为auto时，浏览器会将左右外边距设置为相等，所以这行代码margin:0 auto可以使元素居中。

display

- 我们不能为行内元素设置width、height、margin-top和margin-bottom。
- 我们可以通过修改display来修改元素的性质。
- 可选值：
  - block：设置元素为块元素
  - inline：设置元素为行内元素
  - inline-block：设置元素为行内块元素
  - none：隐藏元素（元素将在页面中完全消失）

visibility

- visibility属性主要用于元素是否可见。
- 和display不同，使用visibility隐藏一个元素，隐藏后其在文档中所占的位置会依然保持，不会被其他元素覆盖。
- 可选值：
  - visible：可见的
  - hidden：隐藏的

overflow

- 当相关标签里面的内容超出了样式的宽度和高度是，就会发生一些奇怪的事情，浏览器会让内容溢出盒子。
- 可以通过overflow来控制内容溢出的情况。
- 可选值：
  - visible：默认值
  - scroll：添加滚动条
  - auto：根据需要添加滚动条
  - hidden：隐藏超出盒子的内容

文档流

- 文档流指的是文档中可现实的对象在排列时所占用的位置。
- 将窗体自上而下分成一行行，并在每行中按从左至右的顺序排放元素，即为文档流。
- 也就是说在文档流中元素默认会紧贴到上一个元素的右边，如果右边不足以放下元素，元素则会另起一行，在新的一行中继续从左至右摆放。
- 这样一来每一个块元素都会另起一行，那么我们如果想在文档流中进行布局就会变得比较麻烦。

浮动

- 所谓浮动指的是使元素脱离原来的文本流，在父元素中浮动起来。
- 浮动使用float属性。
- 可选值：
  - none：不浮动
  - left：向左浮动
  - right：向右浮动
- 块级元素和行内元素都可以浮动，当一个行内元素浮动以后将会自动变为一个块级元素。
- 当一个块级元素浮动以后，宽度会默认被内容撑开，所以当漂浮一个块级元素时我们都会为其指定一个宽度。

浮动

- 当一个元素浮动以后，其下方的元素会上移。元素中的内容将会围绕在元素的周围。
- 浮动会使元素完全脱离文本流，也就是不再在文档中在占用位置。
- 元素设置浮动以后，会一直向上漂浮直到遇到父元素的边界或者其他浮动元素。
- 元素浮动以后即完全脱离文档流，这时不会再影响父元素的高度。也就是浮动元素不会撑开父元素。
- 浮动元素默认会变为块元素，即使设置display:inline以后其依然是个块元素。

清除浮动

- clear属性可以用于清除元素周围的浮动对元素的影响。
- 也就是元素不会因为上方出现了浮动元素而改变位置。
- 可选值：
  - left：忽略左侧浮动
  - right：忽略右侧浮动
  - both：忽略全部浮动
  - none：不忽略浮动，默认值

定位

- position属性可以控制Web浏览器如何以及在何处显示特定的元素。
- 可以使用position属性把一个元素放置到网页中的任何位置。
- 可选值：
  - static
  - relative
  - absolute
  - fixed

相对定位

- 每个元素在页面的文档流中都有一个自然位置。相对于这个位置对元素进行移动就称为相对定位。周围的元素完全不受此影响。
- 当将position属性设置为relative时，则开启了元素的相对定位。
- 当开启了相对定位以后，可以使用top、right、bottom、left四个属性对元素进行定位。

相对定位的特点

- 如果不设置元素的偏移量，元素位置不会发生改变。
- 相对定位不会使元素脱离文本流。元素在文本流中的位置不会改变。
- 相对定位不会改变元素原来的特性。
- 相对定位会使元素的层级提升，使元素可以覆盖文本流中的元素。

绝对定位

- 绝对定位指使元素相对于html元素或离他最近的祖先定位元素进行定位。
- 当将position属性设置为absolute时，则开启了元素的绝对定位。
- 当开启了绝对定位以后，可以使用top、right、bottom、left四个属性对元素进行定位。

绝对定位的特点

- 绝对定位会使元素完全脱离文本流。
- 绝对定位的块元素的宽度会被其内容撑开。
- 绝对定位会使行内元素变成块元素。
- 一般使用绝对定位时会同时为其父元素指定一个相对定位，以确保元素可以相对于父元素进行定位。

固定定位

- 固定定位的元素会被锁定在屏幕的某个位置上，当访问者滚动网页时，固定元素会在屏幕上保持不动。
- 当将position属性设置为fixed时，则开启了元素的固定定位。
- 当开启了固定定位以后，可以使用top、right、bottom、left四个属性对元素进行定位。
- 固定定位的其他特性和绝对定位类似。

z-index

- 当元素开启定位以后就可以设置z-index这个属性。
- 这个属性可以提升定位元素所在的层级。
- z-index可以指定一个整数作为参数，值越大元素显示的优先级越高，也就是z-index值较大的元素会显示在网页的最上层。


尚硅谷
www.atguigu.com