第 6 天【盒模型】

主要内容

1、盒模型的概念
2、标准盒模型
3、怪异盒模型
4、弹性盒模型

学习目标

节数	知识点	要求
第一节（盒模型的概念）	概念	了解
第二节（标准盒模型）	概念	了解
	content	掌握
	padding	掌握
	border	掌握
	Margin	掌握
第三节（标准盒模型练习）	魅族官网	掌握
第四节（怪异盒模型）	概念	了解
	box-sizing	掌握
	标准盒模型和怪异盒模型的对比	掌握
第五节（弹性盒模型）	概念	了解
	display：flex；	掌握
	flex-direction:；	掌握
	justify-content:；	掌握
	align-items:；	掌握
	flex-grow:；	掌握
第六节（弹性盒模型练习）	魔笛音乐	掌握

一、CSS 盒子模型(Box Model)

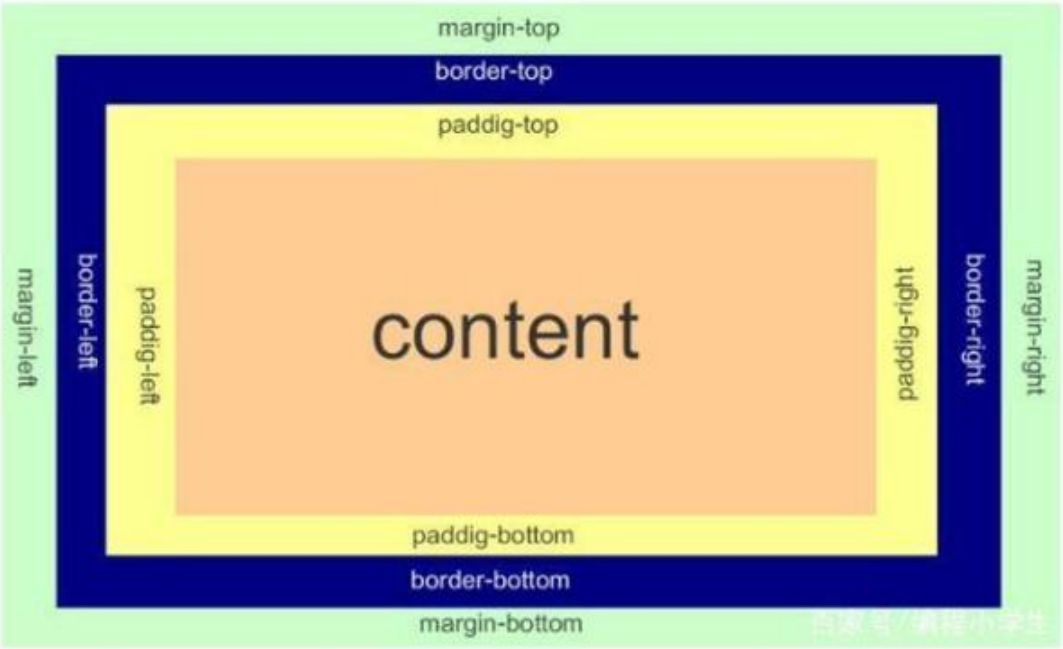
1、概念

所有 HTML 元素可以看作盒子，在 CSS 中，"box model"这一术语是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，它包括：边距（margin），边框（border），填充（padding），和实际内容（content）。

盒模型允许我们在其它元素和周围元素边框之间的空间放置元素。

下面的图片说明了盒子模型(Box Model)：



不同部分的说明：

1）Margin(外边距) - 清除边框外的区域，外边距是透明的。

2）Border(边框) - 围绕在内边距和内容外的边框。

3）Padding(内边距) - 清除内容周围的区域，内边距是透明的。

4）Content(内容) - 盒子的内容，显示文本和图像。

如果把盒子模型看作是一个生活中的快递，那么内容部分等同于你买的实物，内边距等同于快递盒子中的泡沫，边框等同于快递盒子，外边距等同于两个快递盒子之间的距离。

下面看一个例子：

【实例 1-1】

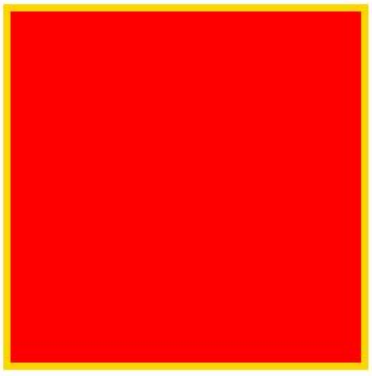
HTML 代码如下：

```
<body>
    <div class="box"></div>
</body>
```

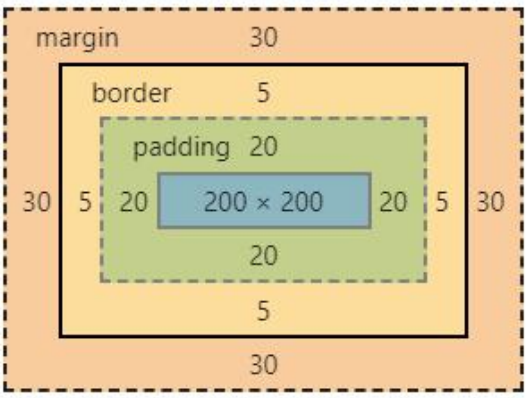
Css 代码如下：

```
<style>
    .box{
    width: 200px;
    height: 200px;
    background-color: red;
    border: 5px solid gold;
    padding: 20px;
    margin: 30px;
    }
</style>
```

效果图如下：



控制台中的盒模型如下：

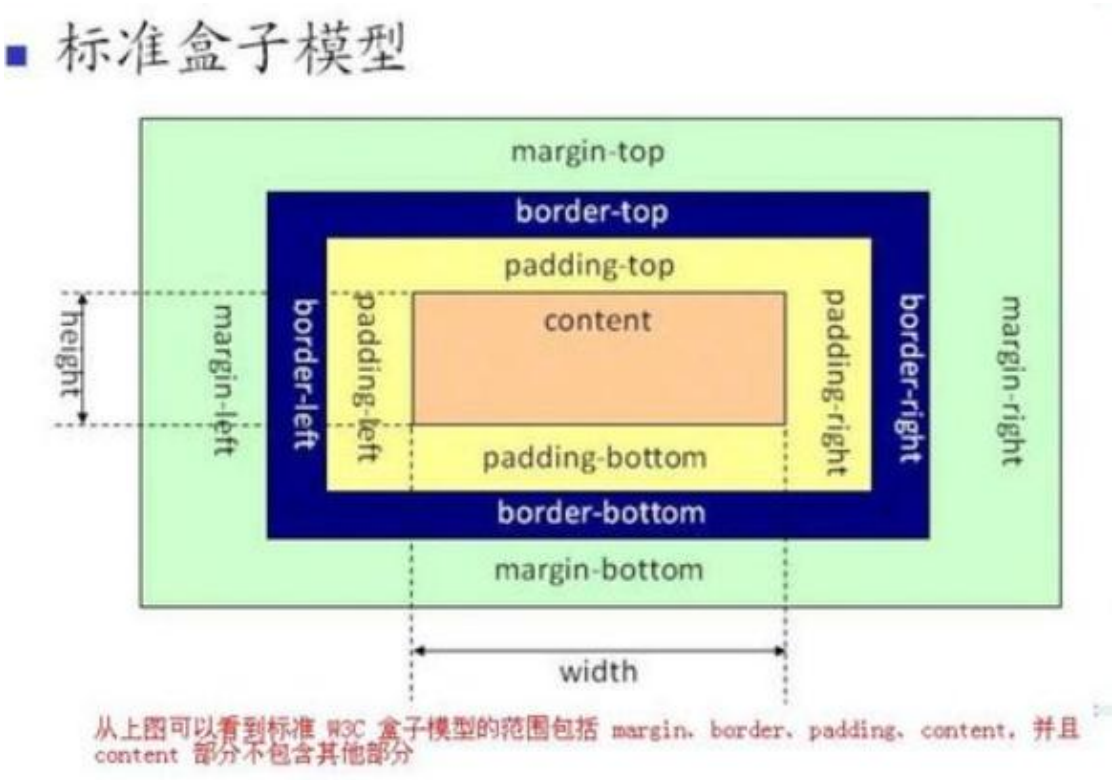


通过代码和图给大家简单讲解一下，最中间蓝色区域就是我们定义的宽度和高度，也就是 content 的宽度和高度，当我们给 padding 加一个宽度的时候整个整体也变宽了，padding 的距离就是绿色部分，在 padding 外面就是 border，浅黄色部分，由图也很明显的看出也增加了整个整体的宽度，border 外面是 margin 部分，深黄色区域。

二、标准盒模型

1、概念

W3C 盒子模型(标准盒模型)



根据 W3C 的规范，元素内容占据的空间是由 width 属性设置的，而内容周围的 padding 和 border 值是另外计算的；即在标准模式下的盒模型，盒子实际内容（content）

的 width/height=我们设置的 width/height;盒子总宽度/高度=width/height+padding+border+margin。

2、内容区域（content）

Content 内容区域包含宽度（width）和高度（height）两个属性。

块级元素默认宽度为 100%，行内元素默认宽度是由内容撑开，不管块级元素还是行内元素，默认高度都是由内容撑开。

块级元素可以设置宽高属性，行内元素设置宽高属性不生效。

宽度（width）和高度（height）可以取值为像素（px）和百分比（%）。

3、内边距（Padding）

3.1 定义

CSS padding（填充）是一个简写属性，定义元素边框与元素内容之间的空间，即上下左右的内边距。

3.2 可能的值

1）定义一个固定的填充(像素，pt，em,等)

2）使用百分比值定义一个填充

3.3 单边内边距属性

在 CSS 中，它可以指定不同的侧面不同的填充：

【实例 2-1】

```
padding-top:25px;

padding-bottom:25px;

padding-right:50px;

padding-left:50px;
```

分别代表：

（1）上内边距是 25px

（2）右内边距是 50px

（3）下内边距是 25px

（4）左内边距是 50px

3.4 简写属性

为了缩短代码，它可以在一个属性中指定的所有填充属性。

这就是所谓的简写属性。所有的填充属性的简写属性是 padding：

Padding 属性，可以有一到四个值。

（1） padding:25px 50px 75px 100px;

上填充为 25px

右填充为 50px

下填充为 75px

左填充为 100px

（2） padding:25px 50px 75px;

上填充为 25px

左右填充为 50px

下填充为 75px

（3）padding:25px 50px;

上下填充为 25px

左右填充为 50px

（4）padding:25px;

所有的填充都是 25px

3.5padding 需要注意的问题

padding 会撑大容器

3.6padding 实现导航条

【实例 2-2】


```
<body>

<ul class="nav">

<li><a href="#">首页</a></li>

<li><a href="#">手机</a></li>

<li><a href="#">家电</a></li>
```

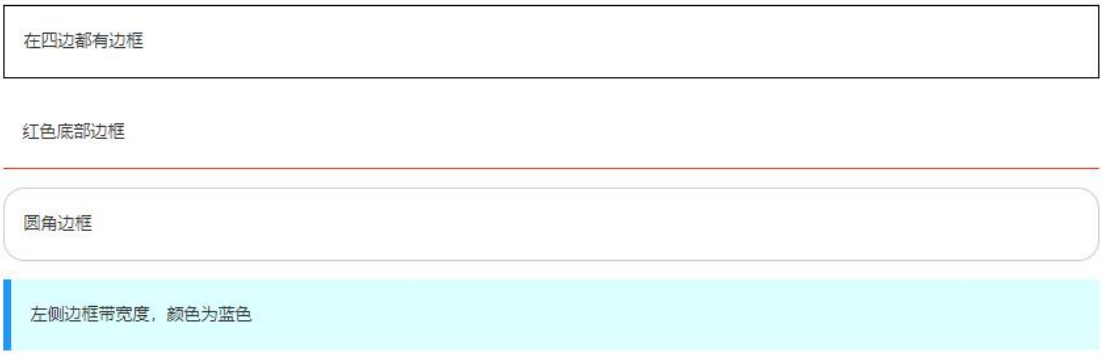
```
<li><a href="#">配件</a></li>
<li><a href="#">生活</a></li>
<li><a href="#">服务</a></li>
<li><a href="#">社区</a></li>
</ul>
</body>
```

```
<style>
        .nav>li{
            float: left;
            padding: 20px 30px;
            background-color: #1D1E22;
        }
        .nav a{
            color: #fff;
            font-size: 14px;
        }
</style>
```

4、边框（Border）

4.1 定义

CSS 边框属性允许你指定一个元素边框的样式和颜色。



4.2 边框样式

边框样式属性指定要显示什么样的边界。

border-style 属性用来定义边框的样式

border-style 值:

none: 默认无边框

dotted: 定义一个点线边框

dashed: 定义一个虚线边框

solid: 定义实线边框

double: 定义两个边框。 两个边框的宽度和 border-width 的值相同

4.3 边框宽度

您可以通过 border-width 属性为边框指定宽度

```
.one{
border-style:solid;
border-width:5px;
}
```

4.4 边框颜色

border-color 属性用于设置边框的颜色

```
.one{
border-style:solid;
border-color: red;
}
```

您还可以设置边框的颜色为"transparent"。

注意：border-color 单独使用是不起作用的，必须得先使用 border-style 来设置边框样式。

4.5 边框单独设置各边

在 CSS 中，可以指定不同的侧面不同的边框：

上面的例子也可以设置一个单一属性：

```
p{
border-style:dotted solid;
}
```

border-style 属性可以有 1-4 个值：

（1）border-style:dotted solid double dashed;

上边框是 dotted

右边框是 solid

底边框是 double

左边框是 dashed

（2）border-style:dotted solid double;

上边框是 dotted

左、右边框是 solid

底边框是 double

（3）border-style:dotted solid;

上、底边框是 dotted

右、左边框是 solid

（4）border-style:dotted;

四面边框是 dotted

上面的例子用了 border-style。然而，它也可以和 border-width 、 border-color 一起使用。

4.6 边框简写属性

上面的例子用了很多属性来设置边框。

你也可以在一个属性中设置边框。

你可以在"border"属性中设置：

- border-width
- border-style（required)
- border-color

```
p{

border:5px solid red;

}
```

5、外边距（Margin）

5.1 定义

CSS margin(外边距)属性定义元素周围的空间。

5.2 取值

（1）Auto

（2）定义一个固定的 margin

（3）定义一个使用百分比的边距

5.3 单边外边距属性

在 CSS 中，它可以指定不同的侧面不同的边距：

```
div{

margin-top:100px;

margin-bottom:100px;

margin-right:50px;

margin-left:50px;
```

```
}
```

5.4 简写属性

为了缩短代码，有可能使用一个属性中 margin 指定的所有边距属性。这就是所谓的简写属性。

所有边距属性的简写属性是 margin：

```
div{
margin:100px 50px;
}
```

margin 属性可以有一到四个值。

（1）margin:25px 50px 75px 100px;

- 上边距为 25px
- 右边距为 50px
- 下边距为 75px
- 左边距为 100px

（2）margin:25px 50px 75px;

- 上边距为 25px
- 左右边距为 50px
- 下边距为 75px

（3）margin:25px 50px;

- 上下边距为 25px
- 左右边距为 50px

（4）margin:25px;

- 所有的 4 个边距都是 25px

5.5margin 需要注意的问题

（1）外边距合并问题

垂直方向上外边距相撞时，取较大值

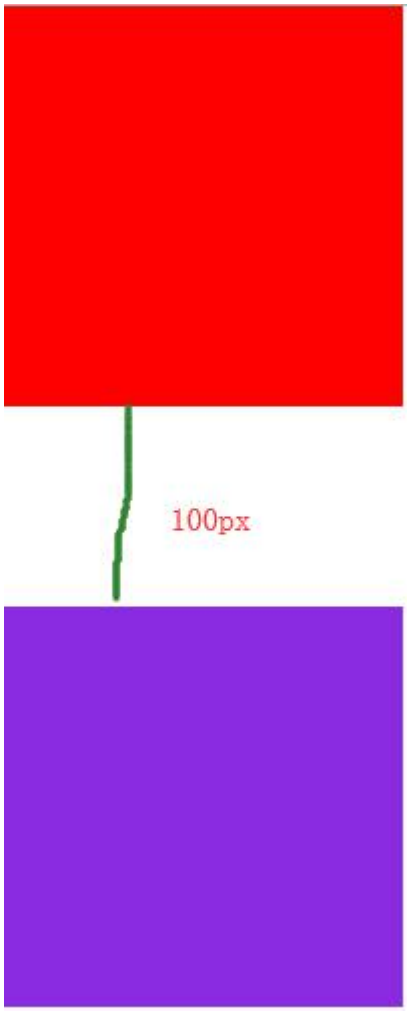
注意：浮动元素的外边距不合并

【实例 2-1】

```
<div class="one"></div>

<div class="two"></div>
```

```
<style>

.one,.two{

    width: 200px;

    height: 200px;

}

.one{

    background-color: red;

    margin-bottom: 50px;

}

.two{

    background-color: blueviolet;

    margin-top: 100px;

}

</style>
```

效果图如下：



（2）margin-top 问题

当给子元素设置 margin-top 时，父元素会跟着子元素一起下来：

解决方法：

①父元素设置 overflow:hidden;

②父元素或者子元素设置 float

③父元素设置 border:1px solid transparent;

④父元素设置 padding-top:1px;

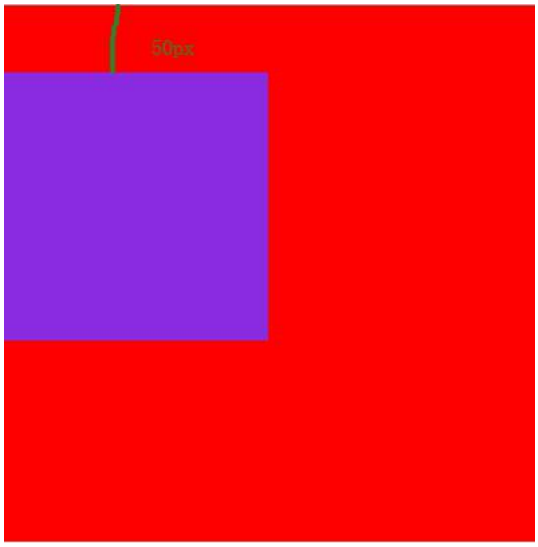
【实例 2-2】

```
<div  class="box">

<div  class="box1"></div>

</div>
```

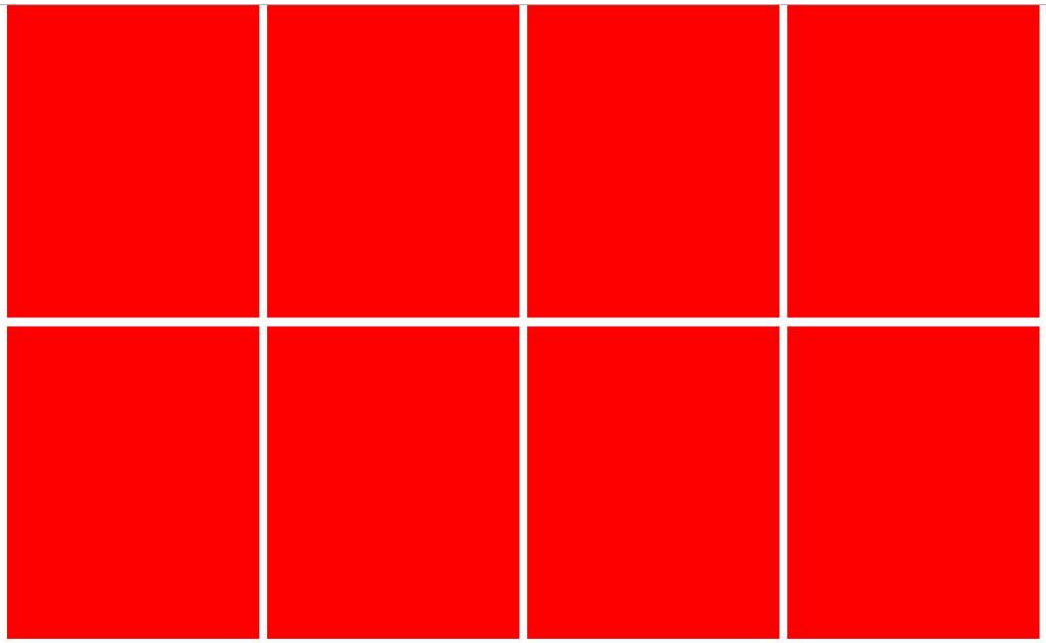
```
<style>
```

```
.box{
width: 400px;
height: 400px;
background-color: red;
overflow: hidden;
}
.box1{
width: 200px;
height: 200px;
background-color: blueviolet;
margin-top: 50px;
}
</style>
```

效果如下：



5.6margin 练习



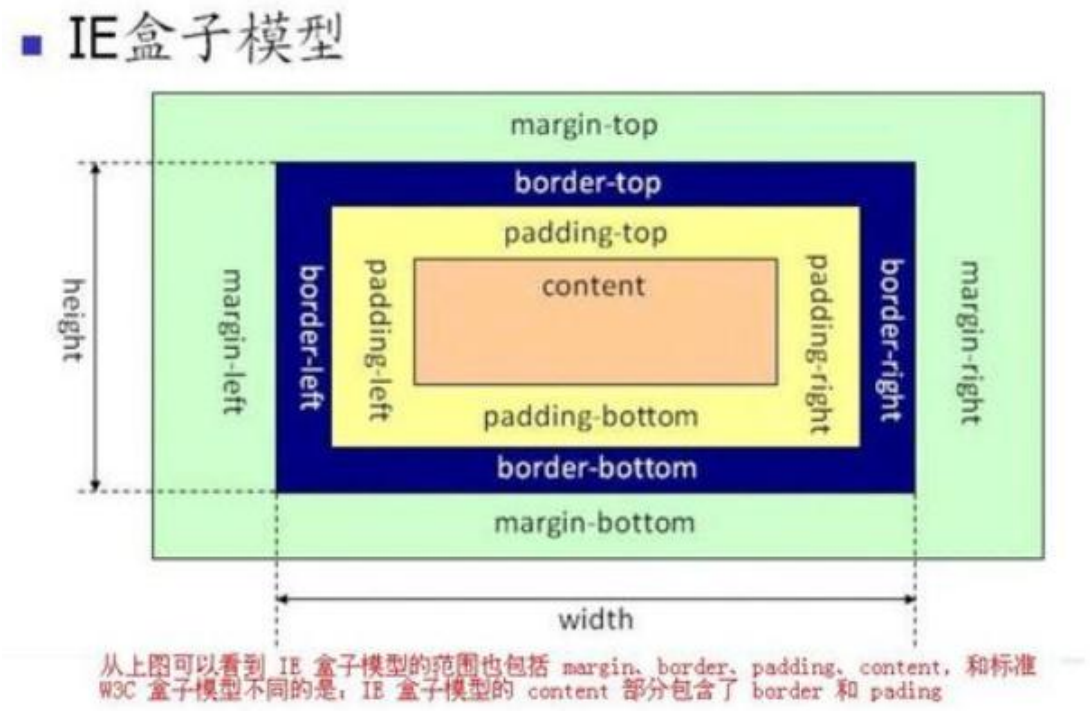
```
<body>

<div  class="box">

<div></div>

<div></div>

<div></div>

<div></div>

<div></div>

<div></div>

<div></div>

<div></div>

</div>

</body>
```

```
<style>
.box{
width: 1240px;
margin: 0 auto;
}
.box>div{
width: 303px;
height: 375px;
background-color: red;
float: left;
margin-bottom: 10px;
margin-right: 9px;
}
.box>div:nth-child(4n){
margin-right: 0;
}
</style>
```

三、怪异盒模型

1、概念

IE 盒子模型(怪异盒模型)



在该模式下，浏览器的 width 属性不是内容的宽度，而是内容、内边距和边框的宽度的总和；即在怪异模式下的盒模型，盒子的(content)宽度+内边距 padding+边框 border 宽度=我们设置的 width(height 也是如此)，盒子总宽度/高度=width/height + margin 。

2、Box-sizing

CSS3 指定盒子模型种类

box-sizing 属性允许您以特定的方式定义匹配某个区域的特定元素。

box-sizing: content-box;//宽度和高度分别应用到元素的内容框。在宽度和高度之外绘制元素的内边距和边框。

box-sizing: border-box;// 为元素设定的宽度和高度决定了元素的边框盒。就是说，为元素指定的任何内边距和边框都将在已设定的宽度和高度内进行绘制。通过从已设定的宽度和高度分别减去边框和内边距才能得到内容的宽度和高度。

即 box-sizing 属性可以指定盒子模型种类，content-box 指定盒子模型为 W3C（标准盒模型），border-box 为 IE 盒子模型（怪异盒模型）。

四、弹性盒模型（flex box）

1、定义

弹性盒子是 CSS3 的一种新的布局模式。

CSS3 弹性盒是一种当页面需要适应不同的屏幕大小以及设备类型时确保元素拥有恰当的行为的布局方式。

引入弹性盒布局模型的目的是提供一种更加有效的方式来对一个容器中的子元素进行排列、对齐和分配空白空间。

2、CSS3 弹性盒内容

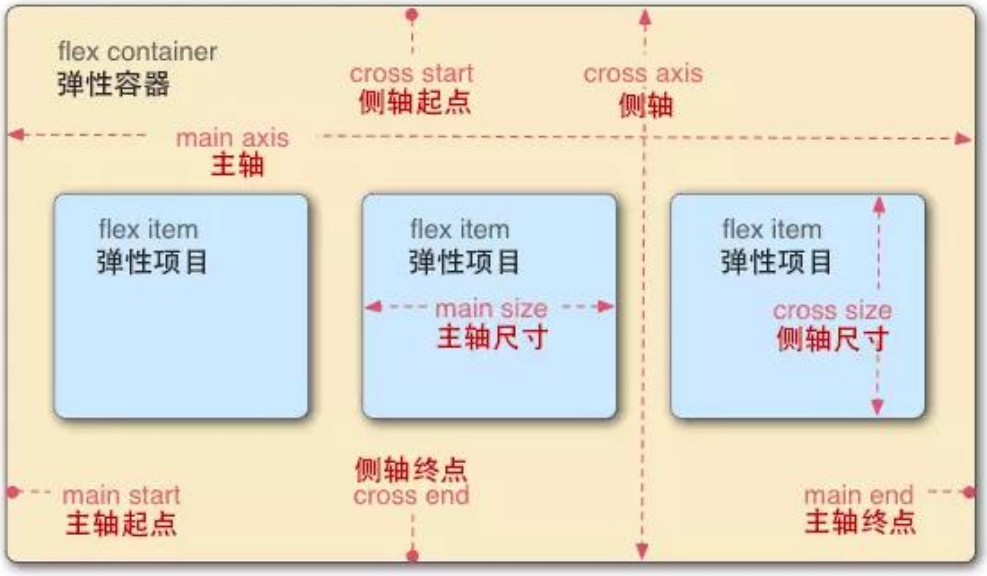
弹性盒子由弹性容器(Flex container)和弹性子元素(Flex item)组成。

弹性容器通过设置 display 属性的值为 flex 将其定义为弹性容器。

弹性容器内包含了一个或多个弹性子元素。

注意：弹性容器外及弹性子元素内是正常渲染的。弹性盒子只定义了弹性子元素如何在弹性容器内布局。

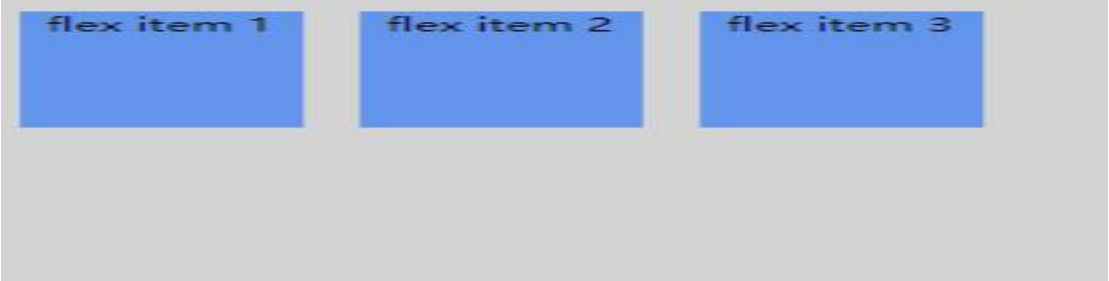
弹性子元素通常在弹性盒子内一行显示。默认情况每个容器只有一行。



注意：主轴与侧轴的概念

- 主轴就是弹性盒子子元素沿着排列的轴；与主轴垂直的轴称为侧轴。
- 如果你有 row 或者默认值,则主轴是水平方向，侧轴是垂直方向。
- 如果你有 column,则主轴是垂直方向，侧轴是水平方向。

【实例 4.1】




```
<div class="flex-container">
<div class="flex-item">flex item 1</div>
<div class="flex-item">flex item 2</div>
<div class="flex-item">flex item 3</div>
</div>
<style>
.flex-container {
display: flex;
width: 400px;
height: 250px;
background-color: lightgrey;
}
.flex-item {
background-color: cornflowerblue;
width: 100px;
height: 100px;
margin: 10px;}
</style>
```

3、父元素上的属性

3.1display 属性

display:flex;开启弹性盒

display:flex;属性设置后子元素默认水平排列

3.2flex-direction 属性

（1）定义

flex-direction 属性指定了弹性子元素在父容器中的位置。

（2）语法

```
flex-direction: row | row-reverse | column | column-reverse
```

flex-direction 的值有:

row：横向从左到右排列（左对齐），默认的排列方式。

row-reverse：反转横向排列（右对齐，从后往前排，最后一项排在最前面。

column：纵向排列。

column-reverse：反转纵向排列，从后往前排，最后一项排在最上面。

（3）实例

```
.flex-container {

display: flex;

flex-direction: column;

width: 400px;

height: 250px;

background-color: lightgrey;}
```

3.3justify-content 属性

（1）定义

内容对齐（justify-content）属性应用在弹性容器上，把弹性项沿着弹性容器的主轴线（main axis）对齐

（2）语法

```
justify-content: flex-start | flex-end | center | space-between | space-around
```

各个值解析:

①flex-start：

弹性项目向行头紧挨着填充。这个是默认值。第一个弹性项的 main-start 外边距边线被放置在该行的 main-start 边线，而后续弹性项依次平齐摆放。

②flex-end：

弹性项目向行尾紧挨着填充。第一个弹性项的 main-end 外边距边线被放置在该行的 main-end 边线，而后续弹性项依次平齐摆放。

③center：

弹性项目居中紧挨着填充。（如果剩余的自由空间是负的，则弹性项目将在两个方向上同时溢出）。

④space-between：

弹性项目平均分布在该行上。如果剩余空间为负或者只有一个弹性项，则该值等同于 flex-start。否则，第 1 个弹性项的外边距和行的 main-start 边线对齐，而最后 1 个弹性项的外边距和行的 main-end 边线对齐，然后剩余的弹性项分布在该行上，相邻项目的间隔相等。

⑤space-around：

弹性项目平均分布在该行上，两边留有一半的间隔空间。如果剩余空间为负或者只有一个弹性项，则该值等同于 center。否则，弹性项目沿该行分布，且彼此间隔相等（比如是 20px），同时首尾两边和弹性容器之间留有一半的间隔（1/2*20px=10px）。

效果图展示：

（3）实例

```
.flex-container {
display: flex;
justify-content: center;
width: 400px;
height: 250px;
background-color: lightgrey;
}
```

3.4align-items 属性

（1）定义

align-items 设置或检索弹性盒子元素在侧轴（纵轴）方向上的对齐方式。

（2）语法

```
align-items: flex-start | flex-end | center
```

各个值解析:

①flex-start：弹性盒子元素的侧轴（纵轴）起始位置的边界紧靠住该行的侧轴起始边界。

②flex-end :弹性盒子元素的侧轴(纵轴)起始位置的边界紧靠住该行的侧轴结束边界。

③center：弹性盒子元素在该行的侧轴（纵轴）上居中放置。（如果该行的尺寸小于弹性盒子元素的尺寸，则会向两个方向溢出相同的长度）。

（3）实例

```
.flex-container {
    display: flex;
    align-items: center;
    width: 400px;
    height: 250px;
    background-color: lightgrey;
}
```

4、子元素上的属性

4.1flex-grow

flex-grow 根据弹性盒子元素所设置的扩展因子作为比率来分配剩余空间。

- 默认为 0，即如果存在剩余空间，也不放大。

  如果只有一个子元素设置，那么按扩展因子转化的百分比对其分配剩余空间。0.1 即 10%，1 即 100%，超出按 100%

【实例 4-2】


```
<div class="flex-container">
<div class="flex-item1">flex item 1</div>
<div class="flex-item2">flex item 2</div>
<div class="flex-item3">flex item 3</div>
</div>
<style>
.flex-container {
display: flex;
width: 400px;
height: 250px;
background-color: gold;
}
.flex-item1 {
height: 150px;
background-color: red;
flex-grow: 1;
}
```

```
.flex-item2 {
height: 150px;
background-color: green;
flex-grow: 2;
}
.flex-item3 {
height: 150px;
background-color: blue;
flex-grow: 1;
}
</style>
```

五、浏览器内核

浏览器最重要或者说核心的部分是“Rendering Engine”，可大概译为“渲染引擎”，不过我们一般习惯将之称为“浏览器内核”。负责对网页语法的解释（如标准通用标记语言下的一个应用 HTML、JavaScript）并渲染（显示）网页。 所以，通常所谓的浏览器内核也就是浏览器所采用的渲染引擎，渲染引擎决定了浏览器如何显示网页的内容以及页面的格式信息。不同的浏览器内核对网页编写语法的解释也有不同，因此同一网页在不同的内核的浏览器里的渲染（显示）效果也可能不同，这也是网页编写者需要在不同内核的浏览器中测试网页显示效果的原因。

六、厂商前缀

css 标准中各个属性也要经历从草案(WD)到推荐(REC)的过程，css 中的属性进展都不一样。浏览器厂商在标准尚未明确情况下提前支持会有风险，同时也会出现有的浏览器厂商支持的好，有的支持的不好，所以就用厂商前缀加以区分。

CSS3 中一些新功能也是目前导致各大浏览器不兼容的一个原因，这些新功能的出现，浏览器厂商们变便开始尝试融合、试验，所以就在这些功能前加上自己的特定前缀来执行自己的特定解决方法，为了让这些功能能在完全确认下来前使用；

下面就是我们经常用到的前缀及其兼容浏览器：

-webkit-

Apple Webkit 团队，兼容 Android, Safari, Chrome 等；

-moz-

Mozilla，兼容 Firefox 等;

-ms-

Microsoft 基金会，兼容 IE;

-o-

兼容 Opera

浏览器->浏览器内核：
1、IE浏览器内核：Trident内核，也是俗称的IE内核；
2、Chrome浏览器内核：统称为Chromium内核或Chrome内核，以前是Webkit内核，现在是Blink内核；
3、Firefox浏览器内核：Gecko内核，俗称Firefox内核；
4、Safari浏览器内核：Webkit内核；
5、Opera浏览器内核：最初是自己的Presto内核，后来是Webkit，现在是Blink内核；
6、360浏览器、猎豹浏览器内核：IE+Chrome双内核；
7、搜狗、遨游、QQ浏览器内核：Trident（兼容模式）+Webkit（高速模式）；
8、百度浏览器、世界之窗内核：IE内核；
9、2345浏览器内核：以前是IE内核，现在也是IE+Chrome双内核；

因此对于一些较新的 css3 特性，需要添加以上前缀兼容每个浏览器，例如实现线性渐变，标准写法是 linear-gradient()，但是一下浏览器还未完全确定这一特性，就在前面添加一个前缀来进行试验执行，如 -webkit-linear-gradient；

下面是开发中常用的兼容写法：

```
body {

background: linear-gradient(0, green, blue);
```

```
background: -webkit-linear-gradient(0, green, blue);
background: -moz-linear-gradient(0, green, blue);
background: -o-linear-gradient(0, green, blue);
background: -ms-linear-gradient(0, green, blue);
}
```

七、Css Hack

由于不同厂商的浏览器，或者是同一厂商的浏览器的不同版本，如 IE6 和 IE7，对 CSS 的解析认识不一样，因此会导致生成的页面效果不同，得不到我们所需要的页面效果。 这个时候我们就需要针对不同的浏览器去写不同的 CSS，让它能够同时兼容不同的浏览器，能在不同的浏览器中也能得到我们想要的页面效果。

css hack 并不是什么技术，只是针对于在不同浏览器上的 bug 的解决方案。一般都是利用各浏览器的支持 CSS 的能力和 BUG 来进行的。

hack 这个词，看翻译也知道它属于是一种“暴力”的作法，尽量找到通用方法而减少对 CSS Hack 的使用 ,大规模使用 CSS Hack 会带来维护成本的提高以及浏览器版本变化而带来类似 Hack 失效等系列问题。(对当前浏览器版本起作用的 hack，也许升级一下，界面就又乱了)

下面我们学习常用的几个 hack，主要仅针对 IE 浏览器，IE6 以下不再考虑。

1.条件 hack

【示例 1】

```
<!--[if IE 9]>
    <style>
        body{ background-color:orange; }  //在 IE9 下，背景颜色为橘色
    </style>
<![endif]-->
```

【示例 2】

```
<!--[if IE]>
    <p>如果你使用 IE 浏览器将能看到我</p>    //在 IE 浏览器显示段落 p
<![endif]-->
```

【示例 3】

```
<!--[if gt IE 6]>
    <style>
    .test{color:red;}
    </style>
<![endif]-->
//大于 ie6 版本的.test 类字体为红色
```

通过上面三个例子我们可以发现：
条件注释它的格式为

```
<!--[ if 条件 ]>
        代码块
<![ endif ]-->
```

大于 gt
大于或等于 gte
小于 lt
小于或等于 lte
非指定版本 !
这些条件帮我们更精确的筛选浏览器版本

2、属性级 Hack

同一段文字在 IE6,7,8 显示为不同颜色

```
.test{
    _color:#ff0;          /* IE6 及以下*/
    *color:#f00;          /* IE7 及以下 */
    color:#090\9;         /* IE6 及以上 */
    }
<p class=" test" >this is paragraph!!</p>
```

hack	写法	实例	IE6 (S)	IE6 (Q)	IE7 (S)	IE7 (Q)	IE8 (S)	IE8 (Q)	IE9 (S)	IE9 (Q)	IE10 (S)	IE10 (Q)
*	*color	青色	Y	Y	Y	Y	N	Y	N	Y	N	Y
+	+color	绿色	Y	Y	Y	Y	N	Y	N	Y	N	Y
-	-color	黄色	Y	Y	N	N	N	N	N	N	N	N
_	_color	蓝色	Y	Y	N	Y	N	Y	N	Y	N	N
#	#color	紫色	Y	Y	Y	Y	N	Y	N	Y	N	Y
\0	color:red\0	红色	N	N	N	N	Y	N	Y	N	Y	N
\9\0	color:red\9\0	粉色	N	N	N	N	N	N	Y	N	Y	N
!important	color:blue !important;color:green;	棕色	N	N	Y	N	Y	N	Y	N	Y	Y

- “-”减号是IE6专有的hack
- “\9” IE6/IE7/IE8/IE9/IE10都生效
- “\0” IE8/IE9/IE10都生效，是IE8/9/10的hack
- “\9\0” 只对IE9/IE10生效，是IE9/10的hack

八、作业

https://www.meizu.com/