

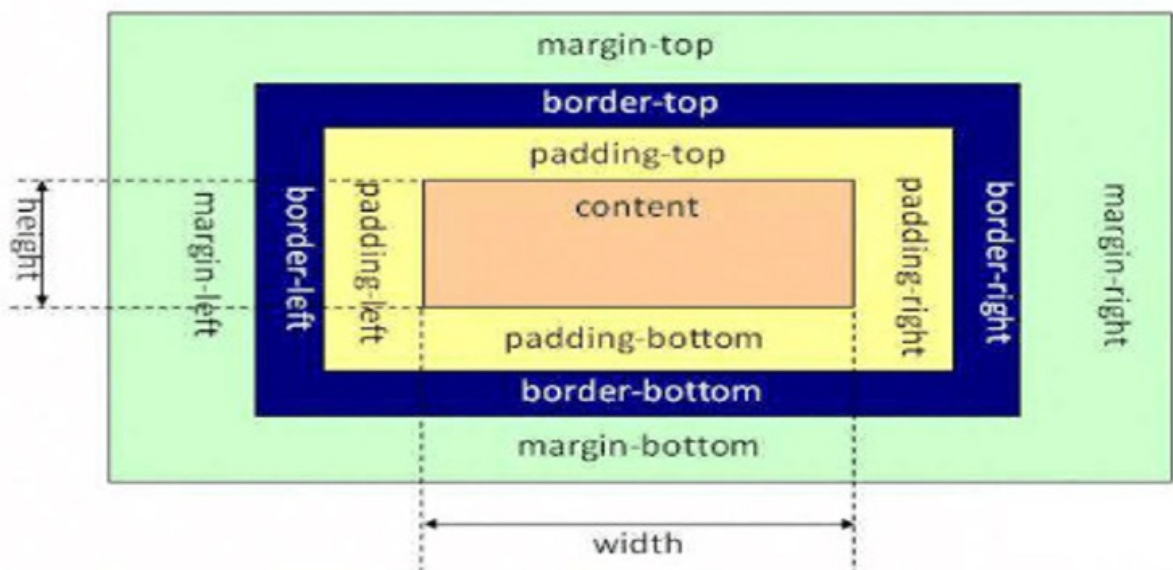
怪异盒模型

- box-sizing

1. content-box

2. 这是由 CSS2.1 规定的宽度高度行为。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。

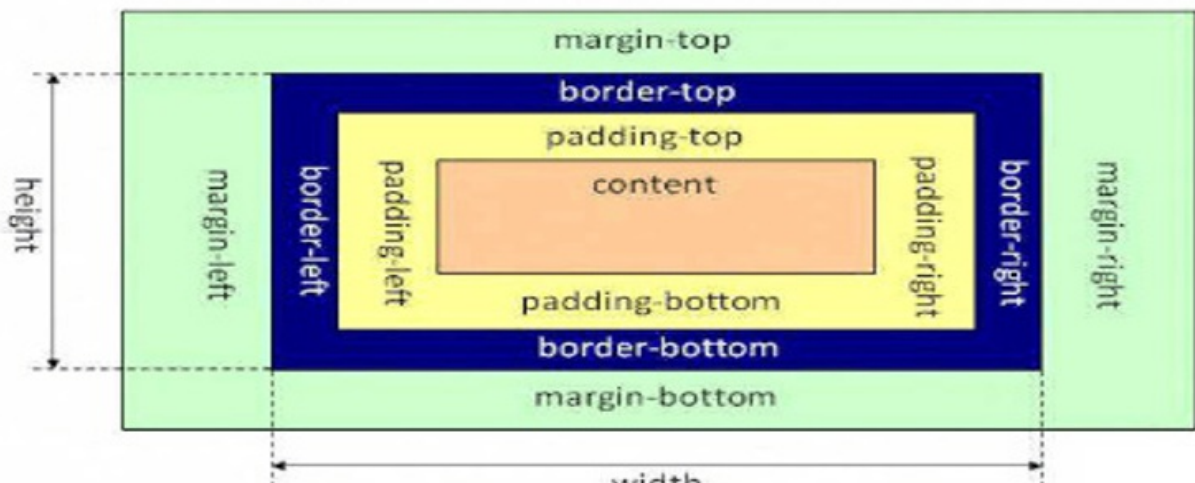
■ 标准盒子模型



1. border-box

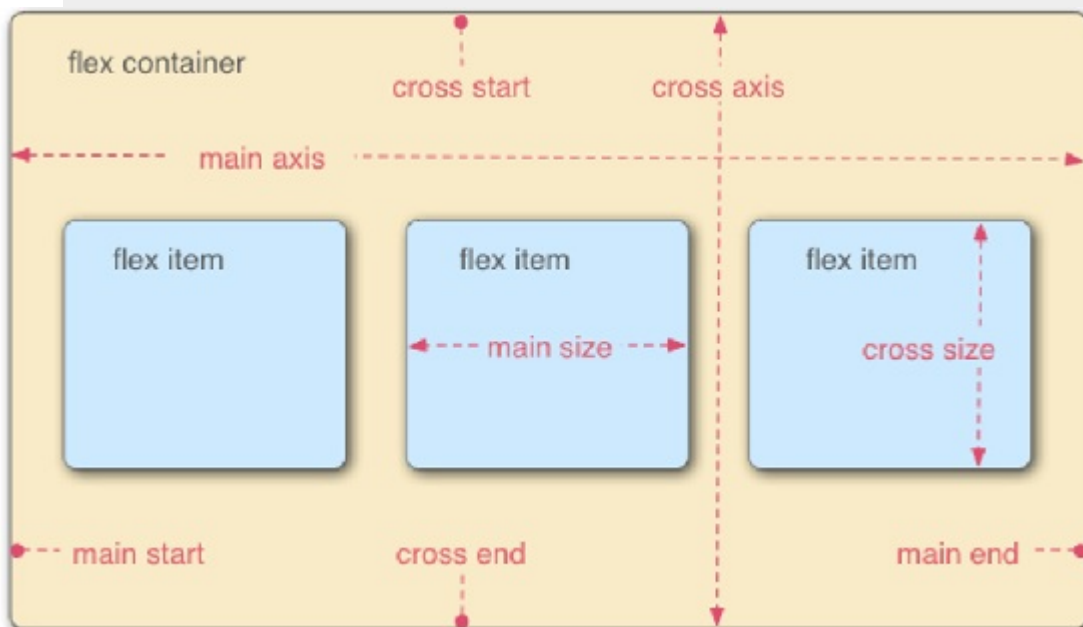
2. 为元素设定的宽度和高度决定了元素的边框盒。就是说，为元素指定的任何内边距和边框都将在已设定的宽度和高度内进行绘制。通过从已设定的宽度和高度分别减去边框和内边距才能得到内容的宽度和高度。

■ 怪异盒模型（IE盒模型）



FLEX布局

1. **Flex容器**：采用 **Flex** 布局的元素的父元素；
2. **Flex项目**：采用 **Flex** 布局的元素的父元素的子元素；
3. 容器默认存在两根轴：水平的主轴（main axis）和垂直的交叉轴（cross axis）。主轴的开始位置（与边框的交叉点）叫做main start，结束位置叫做main end；交叉轴的开始位置叫做cross start，结束位置叫做cross end。
4. 项目默认沿主轴排列。单个项目占据的主轴空间叫做main size，占据的交叉轴空间叫做cross size。



flex容器属性

1、display:flex、inline-flex

1. 注意，设为 **Flex** 布局以后，子元素的float、clear和vertical-align属性将失效。

2、flex-direction属性 决定主轴的方向（即项目的排列方向）

1. flex-direction: row | row-reverse | column | column-reverse;

3、flex-wrap属性，定义子元素是否换行显示

1. flex-wrap: nowrap | wrap | wrap-reverse;

4、flex-flow

1. flex-flow属性是flex-direction属性和flex-wrap属性的简写形式，默认值为row nowrap;

5、 justify-content属性 定义了项目在主轴（）上的对齐方式。

1. justify-content: flex-start | flex-end | center | space-between | space-around;

6、 align-items属性定义项目在交叉轴上如何对齐。

1. align-items: flex-start | flex-end | center | baseline | stretch (默认值);

7、 align-content属性定义了多根轴线的对齐方式。对于单行子元素，该属性不起作用。

1. align-content: flex-start | flex-end | center | space-between | space-around | stretch;
2. align-content在侧轴上执行样式的时候，会把默认的间距给合并。对于单行子元素，该属性不起作用

flex项目属性

1、 align-self属性

1. Internet Explorer 和 Safari 浏览器不支持 align-self 属性

2. 说明：

3. align-self 属性规定灵活容器内被选中项目的对齐方式。

4. 注意：align-self 属性可重写灵活容器的 align-items 属性。

5. 属性值

6. auto 默认值。元素继承了它的父容器的 align-items 属性。如果没有父容器则为 "stretch"。

7. Stretch 元素被拉伸以适应容器。

8. Center 元素位于容器的中心。

9. flex-start 元素位于容器的开头。

10. flex-end 元素位于容器的结尾。

2、 order

1. 说明：

2. number排序优先级，数字越大越往后排，默认为0，支持负数。

3、 flex

1. 说明：

2. 复合属性。设置或检索弹性盒模型对象的子元素如何分配空间

3. 详细属性值：

4. 缩写「flex: 1」，则其计算值为「1 1 0%」

5. 缩写「flex: auto」，则其计算值为「1 1 auto」

6. flex: none」，则其计算值为「0 0 auto」

7. flex: 0 auto」或者「flex: initial」，则其计算值为「0 1 auto」，即「flex」初始值

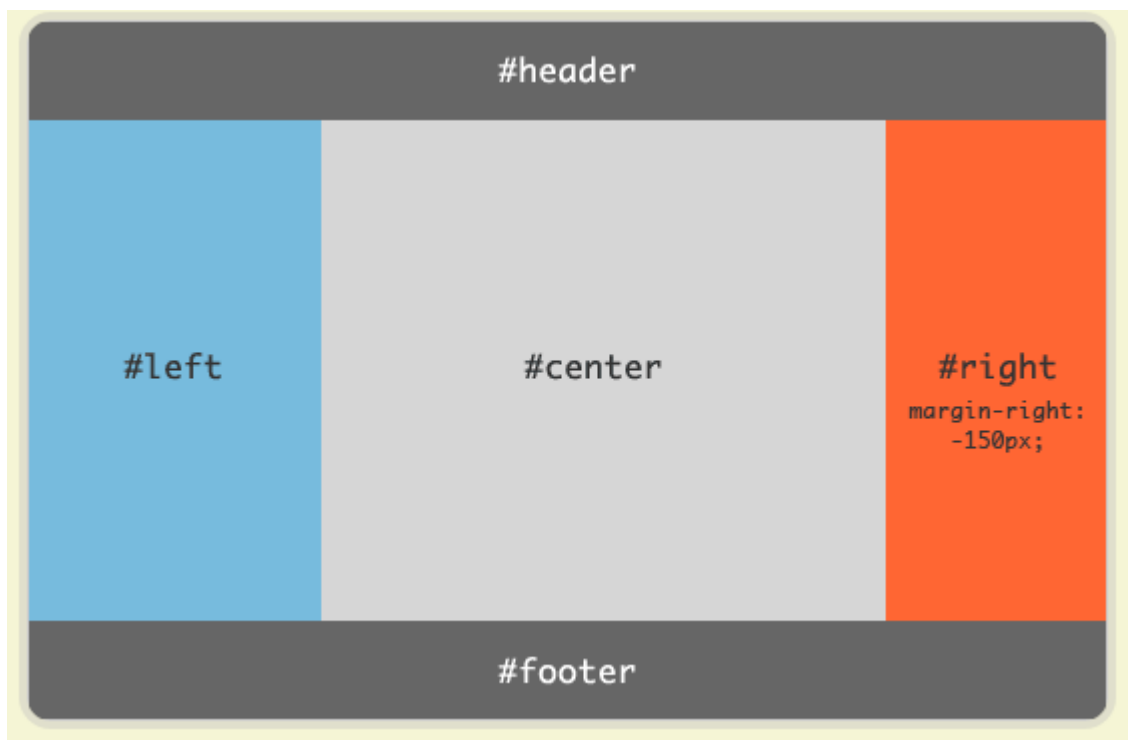
4、 flex-xxx

1. flex-grow

2. 一个数字，规定项目将相对于其他灵活的项目进行扩展的量。

3. `flex-shrink`
4. 一个数字，规定项目将相对于其他灵活的项目进行收缩的量。
5. `flex-basis`
6. 项目的长度

flex布局案例



Add-on Prepended

Amount

Q

Add-on Appended

Go

★

Appended and Prepended Add-ons

✉

Example One

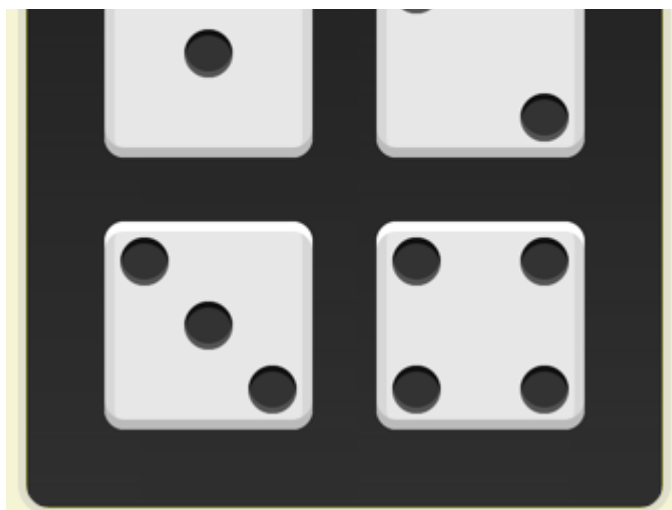
Send

🔒

Example One

Encrypt





Basic Examples



Standard Media Object

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur ac nisl quis massa vulputate adipiscing. Vivamus sit amet risus ligula. Nunc eu pulvinar augue.



Standard Media Object

Donec imperdiet sem leo, id rutrum risus aliquam vitae. Cras tincidunt porta mauris, vel feugiat mauris accumsan eget.

Media Object Reversed

Phasellus vel felis purus. Aliquam consequat pellentesque dui, non mollis erat dictum sit amet. Curabitur non quam dictum, consectetur arcu in, vehicula justo. Donec tortor massa, eleifend nec viverra in, aliquet at eros. Mauris laoreet condimentum mauris, non tempor massa fermentum ut. Integer gravida pharetra cursus. Nunc in suscipit nunc.



Non-images



Using Icons

Donec imperdiet sem leo, id rutrum risus aliquam vitae. Vestibulum ac turpis non lacus dignissim dignissim eu sed dui.



Vertically Centering the Figure

Nunc nec fermentum dolor. Duis at iaculis turpis. Sed rutrum elit ac egestas dapibus. Duis nec consequat enim.

css3-多列

1. 多列布局类似报纸或杂志中的排版方式，主要用以控制大篇幅文本。

□

多列属性

1、column-count

1. 属性规定元素应该被分隔的列数
2. 适用于：除table外的非替换块级元素，table cells, inline-block元素

2、column-gap

1. 属性规定列之间的间隔大小

3、column-rule

1. 设置或检索对象的列与列之间的边框。复合属性。
2. column-rule-color规定列之间规则的颜色。
3. column-rule-style规定列之间规则的样式。
4. column-rule-width规定列之间规则的宽度。

4、column-fill

1. 设置或检索对象所有列的高度是否统一
2. auto：列高度自适应内容
3. balance：所有列的高度以其中最高的一列统一

5、column-span

1. 设置或检索对象元素是否横跨所有列。
2. none：不跨列
3. all：横跨所有列

6、column-width

1. 设置或检索对象每列的宽度

7、columns

1. 设置或检索对象的列数和每列的宽度。复合属性
2. <' column-width '> || <' column-count '>

注：Internet Explorer 10 和 Opera 支持多列属性。

Firefox 需要前缀 -moz-。

Chrome 和 Safari 需要前缀 -webkit-。

多列案例（瀑布流）

□

常见的布局方案

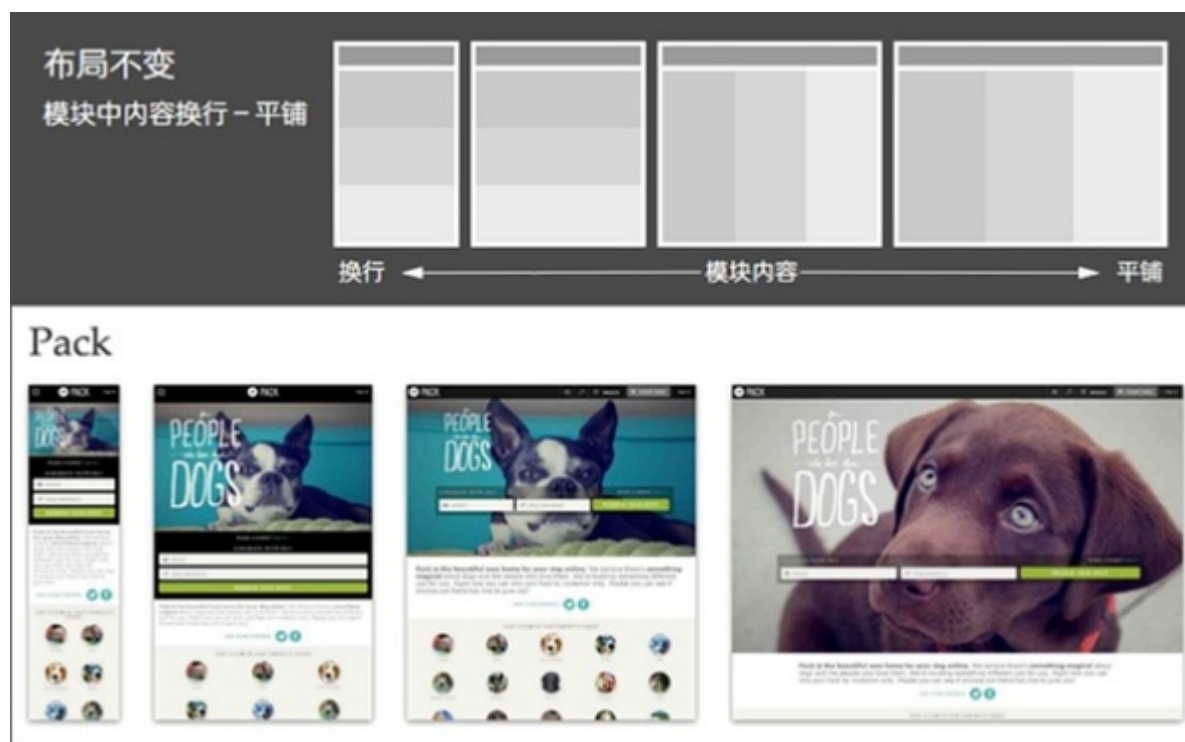
1. 固定布局：以像素作为页面的基本单位，不管设备屏幕及浏览器宽度，只设计一套尺寸；
- 2.
3. 可切换的固定布局：同样以像素作为页面单位，参考主流设备尺寸，设计几套不同宽度的布局。通过识别的屏幕尺寸或浏览器宽度，选择最合适的那套宽度布局；
- 4.
5. 弹性布局：以百分比作为页面的基本单位，可以适应一定范围内所有尺寸的设备屏幕及浏览器宽度，并能完美利用有效空间展现最佳效果；
- 6.
7. 混合布局：同弹性布局类似，可以适应一定范围内所有尺寸的设备屏幕及浏览器宽度，并能完美利用有效空间展现最佳效果；只是混合像素、和百分比两种单位作为页面单位。
- 8.
- 9.
10. 布局响应：对页面进行响应式的设计实现，需要对相同内容进行不同宽度的布局设计，有两种方式：pc优先（从pc端开始向下设计）；移动优先（从移动端向上设计）；无论基于那种模式的设计，要兼容所有设备，布局响应时不可避免地需要对模块布局做一些变化（发生布局改变的临界点称之为断点），

常见响应式布局方案

- （1）模块中内容：挤压一拉（布局不变）



- (2) 模块中内容：换行—平铺（布局不变）



- (3) 模块中内容：删减—增加（布局不变）



- (4) 模块位置变换（布局改变）



- (5) 模块展示方式改变：隐藏—展开（布局改变）



- (6) 模块数量改变：删减—增加（布局改变）





响应式开发的特点

1. 设计特点：
2. 面对不同分辨率设备灵活性强
3. 能够快速解决多设备显示适应问题
1. 缺点：
2. 兼容各种设备工作量大，效率低下
3. 代码累赘，会出现隐藏无用的元素，加载时间加长
4. 其实这是一种折中性质的设计解决方案，多方面因素影响而达不到最佳效果
5. 一定程度上改变了网站原有的布局结构，会出现用户混淆的情况