1.absolute 定位

用绝对定位,它所依赖的是它的父级元素进行定位　不占文档流

relative相对定位，相对于它自己本身原来的位置进行定位　占文档流

fixed相对于浏览器窗口进行定位　不占文档流

static默认定位　　占文档流

inherit继承父元素position属性

2.background: url(../images/rectangle@2x.png) repeat bottom center/ cover;

repeat代表图片可以重复　bottom center代表图片可以放到底部居中　cover代表图片覆盖全屏。

3.font-size

设置字体大小属性

4.line-height

元素行盒（line boxes）的最小高度

5.text-align

设置 h1、h2、h3 元素的文本对齐方式：

6.text-decoration 属性

设置 h1、h2、h3、h4 元素的文本修饰

7.hover 选择器

选择鼠标指针浮动在其上的元素，并设置其样式

8.overflow

overflow属性规定当内容溢出元素框时发生的事情。

visible	默认值。内容不会被修剪，会呈现在元
hidden	内容会被修剪，并且其余内容是不可见
scroll	内容会被修剪，但是浏览器会显示滚动

auto	如果内容被修剪，则浏览器会显示滚动
inherit	规定应该从父元素继承 overflow 属性

9.box-sizing

属性用于更改用于计算元素宽度和高度的默认的 CSS 盒子模型。可以使用此属性来模拟不正确支持CSS盒子模型规范的浏览器的行为。

在CSS中，你设置一个元素的 `width` 与 `height` 只会应用到这个元素的内容区。如果这个元素有任何的 `border` 或 `padding` ，绘制到屏幕上时的盒子宽度和高度会加上设置的边框和内边距值。这意味着当你调整一个元素的宽度和高度时需要时刻注意到这个元素的边框和内边距。当我们实现响应式布局时，这个特点尤其烦人。
box-sizing 属性可以被用来调整这些表现：

- `content-box` 是默认值。如果你设置一个元素的宽为100px，那么这个元素的内容区会有100px宽，并且任何边框和内边距的宽度都会被增加到最后绘制出来的元素宽度中。
- `border-box` 告诉浏览器去理解你设置的边框和内边距的值是包含在width内的。也就是说，如果你将一个元素的width设为100px,那么这100px会包含其它的border和padding，内容区的实际宽度会是width减去border + padding的计算值。大多数情况下这使得我们更容易的去设定一个元素的宽高。

10.transform

属性向元素应用 2D 或 3D 转换。该属性允许我们对元素进行旋转、缩放、移动或倾斜。

语法

transform: none|*transform-functions*;

值	描述	测试
none	定义不进行转换。	测试
matrix(*n*,*n*,*n*,*n*,*n*,*n*)	定义 2D 转换，使用六个值的	测试

	矩阵。	
matrix3d(*n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n*)	定义 3D 转换，使用 16 个值的 4x4 矩阵。	
translate(*x,y*)	定义 2D 转换。	测试
translate3d(*x,y,z*)	定义 3D 转换。	
translateX(*x*)	定义转换，只是用 X 轴的值。	测试
translateY(*y*)	定义转换，只是用 Y 轴的值。	测试
translateZ(*z*)	定义 3D 转换，只是用 Z 轴的值。	
scale(*x,y*)	定义 2D 缩放转换。	测试
scale3d(*x,y,z*)	定义 3D 缩放转换。	
scaleX(*x*)	通过设置 X 轴的值来定义缩放转换。	测试
scaleY(*y*)	通过设置 Y 轴的值来定义缩放转换。	测试
scaleZ(*z*)	通过设置 Z 轴的值来定义 3D 缩放转换。	
rotate(*angle*)	定义 2D 旋转，在参数中规定角度。	测试
rotate3d(*x,y,z,angle*)	定义 3D 旋转。	
rotateX(*angle*)	定义沿着 X 轴的 3D 旋转。	测试
rotateY(*angle*)	定义沿着 Y 轴的 3D 旋转。	测试
rotateZ(*angle*)	定义沿着 Z 轴的 3D 旋转。	测试
skew(*x-angle,y-angle*)	定义沿着 X 和 Y 轴的 2D 倾斜转换。	测试
skewX(*angle*)	定义沿着 X 轴的 2D 倾斜转换。	测试
skewY(*angle*)	定义沿着 Y 轴的 2D 倾斜转换。	测试
perspective(*n*)	为 3D 转换元素定义透视视图。	测试

11.display: inline-block

inline 清除元素之间距离

inline-block　行内块元素

12.flex 弹性布局

flex规定了弹性元素如何伸长或缩短以适应flex容器中的可用空间。这是一个简写属性，用来设置flex-grow，flex-shrink与flex-basis。

大多数情况下，开发者需要将flex设置为auto，initial，none，或一个无单位正数。尝试调整下面的flex容器以观察这些值的作用：

auto

元素会根据自身的宽度与高度来确定尺寸，但是会自行伸长以吸收flex容器中额外的自由空间，也会缩短至自身最小尺寸以适应容器。这相当于将属性设置为 "flex: 1 1 auto".

initial

属性默认值， 元素会根据自身宽高设置尺寸。它会缩短自身以适应容器，但不会伸长并吸收flex容器中的额外自由空间来适应容器 。相当于将属性设置为"flex: 0 1 auto"。

none

元素会根据自身宽高来设置尺寸。它是完全非弹性的：既不会缩短，也不会伸长来适应flex容器。相当于将属性设置为"flex: 0 0 auto"。

<positive-number>

元素会被赋予一个容器中自由空间的指定占比。这相当于设置属性为"flex: <positive-number> 1 0"。

默认情况下，元素不会缩短至小于内容框尺寸，若想改变这一状况，请设置元素的min-width与 min-height属性。

初始值	as each of the properties of the shorthand: flex-grow: 0 flex-shrink: 1 flex-basis: auto
适用元素	flex items, including in-flow pseudo-elements
是否是继承属性	否
适用媒体	visual
计算值	as each of the properties of the shorthand: flex-grow: as specified

	flex-shrink: as specified flex-basis: as specified, but with relative lengt converted into absolute lengths
Animation type	as each of the properties of the shorthand: flex-grow: a number flex-shrink: a number flex-basis: a length, percentage or calc();
正规顺序	order of appearance in the formal grammar o the values

更多属性和定义请参见 使用 CSS 弹性盒子

链接到章节语法

```
/* Basic values */
flex: auto;
flex: initial;
flex: none;
flex: 2;

/* One value, unitless number: flex-grow */
flex: 2;

/* One value, width/height: flex-basis */
flex: 10em;
flex: 30px;

/* Two values: flex-grow | flex-basis */
flex: 1 30px;

/* Two values: flex-grow | flex-shrink */
flex: 2 2;

/* Three values: flex-grow | flex-shrink | flex-basis */
flex: 2 2 10%;

/* Global values */
flex: inherit;
flex: initial;
flex: unset;
```

`flex` 属性可以指定1个，2个或3个值。

单值语法: 值必须为以下其中之一:

- 一个无单位**数(**`<number>`**)**: 它会被当作`<flex-grow>`的值。
- 一个有效的**宽度(**`width`**)**值: 它会被当作 `<flex-basis>`的值。
- 关键字`none`, `auto`,或`initial`.

双值语法：第一个值必须为一个无单位数，并且它会被当作<flex-grow>的值。第二个值必须为以下之一：

- 一个无单位数：它会被当作<flex-shrink>的值。
- 一个有效的宽度值：它会被当作<flex-basis>的值。

三值语法:

- 第一个值必须为一个无单位数，并且它会被当作<flex-grow>的值。
- 第二个值必须为一个无单位数，并且它会被当作 <flex-shrink>的值。
- 第三个值必须为一个有效的宽度值， 并且它会被当作<flex-basis>的值。

链接到章节取值

<'flex-grow'>

定义 flex 元素的 flex-grow 属性，详见 <number>。默认值为 1，负值无效。

<'flex-shrink'>

定义 flex 元素的 flex-shrink 属性，详见 <number>。默认值为 1，负值无效。

<'flex-basis'>

定义 flex 元素的 flex-basis 属性。任何可用于 width 和 height 的值都可接受。若值为0，则必须加上单位，以免被视作伸缩性。 默认值为0%。

none

该关键字等于 0 0 auto.

当使用一个或两个无单位数时，flex-basis会从auto变为0. 可以参考Flexible Box Layout Module草案来了解更多信息。

链接到章节正式语法

none | [<'flex-grow'> <'flex-shrink'>? || <'flex-basis'>]

链接到章节示例

```
#flex-container {
  display: flex;
  flex-direction: row;
}

#flex-container > .flex-item {
  flex: auto;
}
```

```
#flex-container > .raw-item {
  width: 5rem;
}
<div id="flex-container">
  <div class="flex-item" id="flex">Flex box (click to toggle raw box)</div>
  <div class="raw-item" id="raw">Raw box</div>
</div>
```

链接到章节Result

链接到章节规范

规范	状态
CSS Flexible Box Layout Module flex	Candidate Recommendation

链接到章节浏览器兼容性

We're converting our compatibility data into a machine-readable JSON format. This compatibility table still uses the old format, because we haven't yet converted the data it contains. **Find out how you can help!**

- **Desktop**
- Mobile

特性	Firefox (Gecko)	Chrome	Edge	Internet Ex
基础支持	18.0 (18.0)[1] 20.0 (20.0) 28.0 (28.0)[2]	21.0-webkit 29.0	(Yes)-webkit (Yes)	10.0-ms[3] 11.0[3]

[1] 在 Gecko 18.0 (Firefox 18.0 / Thunderbird 18.0 / SeaMonkey 2.15) and 19.0 (Firefox 19.0 / Thunderbird 19.0 / SeaMonkey 2.16) 中， 弹性框可通过修改about:config 中的偏好设置 layout.css.flexbox.enabled 来被支持, 默认值为 false.

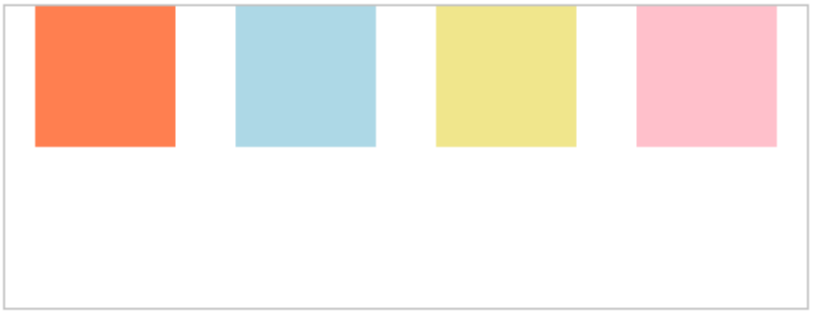
[2] 多行弹性框从 Gecko 28.0 (Firefox 28.0 / Thunderbird 28.0 / SeaMonkey 2.25) 开始支持。

除了无前缀版本的支持之外, Gecko 48.0 (Firefox 48.0 / Thunderbird 48.0 / SeaMonkey 2.45) 出于web兼容性的原因添加了对 -webkit前缀版本的支持，可通过修改偏好设置中的 layout.css.prefixes.webkit 开启, 默认值

为 `false`。 从 Gecko 49.0 (Firefox 49.0 / Thunderbird 49.0 / SeaMonkey 2.46) 开始该参数默认值变为 `true`。
[3] 在Internet Explorer 10-11 (不包括 12+) 中，`flex` 语句的 flex-basis 部分在使用`calc()` 时会被忽略。 可以通过使用 longhand 写法而不是 shorthand 写法来解决。更多信息，请参阅 Flexbug #8。 并且在一个 `flex` 表达式中，使用无单位数的 `flex-basis` 部分在这些版本中会被识别为无效并会因此被忽略。一种解决方法是在使用 shorthand 写法时，让 `flex` 的 `flex-basis` 部分始终带上单位。更多信息，请参阅 Flexbug #4。

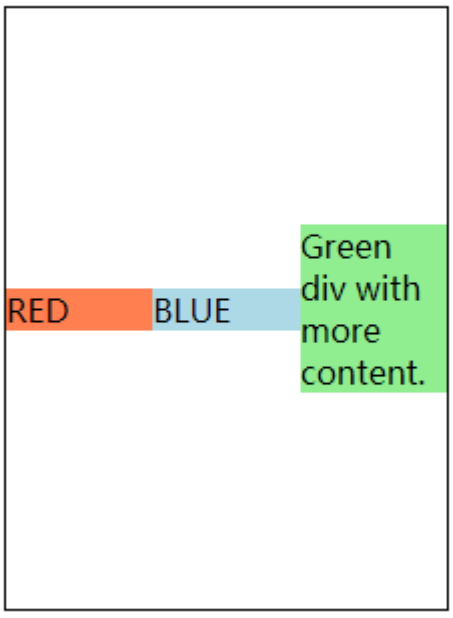
13. justify-content 属性

在弹性盒对象的 元素中的各项周围留有空白：



14.align-items 属性

居中对齐弹性盒的各项 元素：



15.background-size: contain; 属性和background-size: cover;属性

contain 值指定可以不用考虑容器的大小，把图像扩展至最大尺寸，以使其宽度和高度完全适应内容区域。

cover 属性指定背景图可以被调整到任意大小，以使背景图完全覆盖背景区域。

16.cursor 属性

cursor 属性规定要显示的光标的类型（形状）。

值	描述
url	需使用的自定义光标的 URL。 注释：请在此列表的末端始终定义一种 URL 定义的可用光标。
default	默认光标（通常是一个箭头）
auto	默认。浏览器设置的光标。
crosshair	光标呈现为十字线。
pointer	光标呈现为指示链接的指针（一只手）
move	此光标指示某对象可被移动。
e-resize	此光标指示矩形框的边缘可被向右（东
ne-resize	此光标指示矩形框的边缘可被向上及向
nw-resize	此光标指示矩形框的边缘可被向上及向
n-resize	此光标指示矩形框的边缘可被向上（北
se-resize	此光标指示矩形框的边缘可被向下及向
sw-resize	此光标指示矩形框的边缘可被向下及向
s-resize	此光标指示矩形框的边缘可被向下移动
w-resize	此光标指示矩形框的边缘可被向左移动
text	此光标指示文本。

wait	此光标指示程序正忙（通常是一只表或
help	此光标指示可用的帮助（通常是一个问

17.appearance 属性

appearance 属性允许您使元素看上去像标准的用户界面元素。

18.vertical-align 属性

vertical-align 属性设置一个元素的垂直对齐方式