# CSS3+Canvas

# HTML简介

## 1.1什么是HTML5？

HTML5 将成为 HTML、XHTML 以及 HTML DOM 的新标准。HTML 的上一个版本诞生于 1999 年。自从那以后，Web 世界已经经历了巨变。HTML5 仍处于完善之中。然而，大部分现代浏览器已经具备了某些 HTML5 支持。

## 1.2 HTML5 建立的一些规则

* 新特性应该基于 HTML、CSS、DOM 以及 JavaScript。
* 减少对外部插件的需求（比如 Flash）
* 更优秀的错误处理
* 更多取代脚本的标记
* HTML5 应该独立于设备
* 开发进程应对公众透明

## 1.3 HTML5 中的一些有趣的新特性

* 用于绘画的 canvas 元素
* 用于媒介回放的 video 和 audio 元素
* 对本地离线存储的更好的支持
* 新的特殊内容元素，比如 article、footer、header、nav、section
* 新的表单控件，比如 calendar、date、time、email、url、search

## 1.4浏览器支持

最新版本的 Safari、Chrome、Firefox 以及 Opera 支持某些 HTML5 特性。Internet Explorer 9 将支持某些 HTML5 特性。

# 二、语义化标签

## 1.1 结构化标签

#### 1.1.1 <header></header>

定义文档头部区域，一般用在头部

#### 1.1.2 <section></ section >

定义文档中的一块区域,替代div布局

#### 1.1.3 <nav></nav>

定义导航栏

#### 1.1.4 <aside></aside>

定义侧边栏、广告、nav元素组，以及其他类似的内容部分.aside 的内容应该与 article 的内容相关.一般与正文无关的。

#### 1.1.5 <footer></footer>

定义文档底部区域内容，一般以footer结尾

#### 1.1.6 <article></article>

标签定义外部的内容。外部内容可以是来自一个外部的新闻提供者的一篇新的文章，或者是来自论坛的文本。

#### 1.1.7 <figure></figure>

定义一块独立的内容（图像，图标，代码等等），通常用来展示图片及其描述。

figcaption用来定义 figure的标题,放在figure的子元素第一个或者最后一个

<figure>  
 <figcaption></figcaption>  
 <img src=**””/**>  
</figure>

## 1.2 特殊结构标签

#### 1.2.1 <ruby></ruby>

定义注释，行内元素

<ruby>**龙**<rt>**long**</rt></ruby>

#### 1.2.2 <mark></mark>

定义带有记号的文本，用于凸显

#### 1.2.3<meter></meter>

定义已知范围或分数值内的标量测量

<meter> 标签不应用于指示进度（在进度条中）。如果标记进度条，请使用 <progress> 标签。

**行块级元素，可设置宽高**

* min 最小值
* max 最大值
* low 指定点为最低值
* optimum 指定最佳值
* high 指定点为最高值

<meter min=**”0”** max=**”10”** value=**”3”** low=**”5”** optimum=**”9”** high=**”9”** ></meter>

#### 1.2.4<progress max=“” value=“”></progress>

标签显示任务的进度或者进程一般结合js使用

当不给定值时为一种加载的状态，在谷歌里面是一直滚动的状态，火狐里面是一种闪光的状态。

支持宽高设置，但背景颜色需要结合js

#### 1.2.5 <details></ details >

用于描述文档细节部分，类似于定义列表

<details>  
 <summary>**标题**</summary>  
 <p>**对标题的描述**</p>  
</details>

#### 1.2.6 <wbr>

定义文本在何处换行，相当于单词内换行。



# 三、 HTML5兼容性

## 3.1 兼容写法

1、HTML5有部分内容兼容到IE9，IE8及以下对H5完全不兼容，后面的内容不再考虑此类浏览器。最新版本的 Safari、Chrome、Firefox 以及 Opera 支持某些HTML5 特性。

2、部分css3须加兼容前缀：

-webkit- 常用于兼容chrome浏览器，

-moz- 常用于兼容火狐，

-o- 常用于兼容opera，

-ms- 常用于兼容IE。

3、兼容性查询网站：http://caniuse.com/

## 3.2浏览器不支持标签的处理办法

在浏览器不支持标签的时候，我们有两种办法解决：

#### 1、第一种：

<script>  
 *document*.creatElement("header");  
</script>

去一个一个的在文档中创建标签

**注意：script要写在body前面，且创建的标签默认为行级**

#### 2、第二种：

引入html5shiv.js

<script src=“url”><script>

# 四、 CSS3新增选择器

## 4.1 属性选择器新增

#### 1.4.1 关联选择器

E1 ~ E2（选择E1后紧邻的E2元素）

.box1 p~p{ background: **deeppink**; }

#### 1.4.2 ^=“……”以……开头 $=“……”以……结束 \*=“……”包含

.box2 p[class^="test"]{ background: **blue**;} //选中第一个p  
.box2 p[class$="test"]{ background: **deeppink**;} //选中第二个p  
.box2 p[class\*="test"]{ background: **yellow**;} //选中第三个p

<div class=**"box2"**>  
 <p class=**"testt"**>1</p>  
 <p class=**"ttest"**>2</p>  
 <p class=**"ttestt"**>3</p>  
</div>

## 4.2 伪类新增

### 4.2.1 of-type

<div class=**"box3"**>  
 <div></div>  
 <p>**1**</p>  
 <p>**2**</p>  
 <p>**3**</p>  
 <div></div>  
</div>

#### 1、 first-of-type

**选择p，**p必须为他们各自父级的所有子元素为**p元素的第一个p**

.box3 p:first-of-type{ background: **skyblue**; }

选中p1

#### 2、p:last-of-type

**选择p，**p必须为他们各自父级的所有子元素的**p元素**的**最后一个p**

.box3 p:last-of-type{ background: **green**; }

选中p3

#### 3、p:only-of-type

**选择p，**p必须为他们父级所有子元素的**p元素的唯一p**

<div class=**"box4"**>  
 <div></div>  
 <p>1</p>  
 <div></div>  
</div>

选中p1

.box4 p:only-of-type{ background: **deeppink**; }

#### 4、p:nth-of-type（n）

**选择p**，p必须为父级所有**子元素为p的第n个p**

.box5 p:nth-of-type(1){ background: **orangered**; }

#### 5、p:nth-last-of-type（n）

**选择p**，p必须为父级所有p子元素的**倒数第n个**

.box5 p:nth-last-of-type(1){ background: **blue**; }

#### 6、p:only-child

**选择p，**p必须为他们各自父级的**唯一一个子元素**

<div class=**"box6"**>  
 <p></p>  
</div>

.box6 p:only-child{ background: **yellow**;}

#### 7、p:last-child

**选择p**，p必须为父级的**最后一个子元素**

<div class=**"box7"**>  
 <div></div>  
 <p></p>  
</div>

.box7 p:last-child{ background: **yellow**;}

#### 8、p:nth-child(n)

**选择p，**p必须为他们各自父级的**第n个子元素**

.box8 p:nth-child(1){ background: **deeppink**; }  
.box8 p:nth-child(3){ background: **blue**; }  
.box8 div:nth-child(5){ background: **red**; }

#### 9、p:nth-last-child(n)

**选择p，**这些p必须为他们父级的**倒数第n个子元素**

.box8 p:nth-last-child(3){ background: #153170; }

#### 10、p:not(.c1)

选择**除了**c1的p

<div class=**"box9"**>  
 <p class=**"c1"**>**1**</p>  
 <p>**2**</p>  
 <p class=**"c11"**>**3**</p>  
</div>

选中第二个p

.box9 p:not(.c1){ background: #153170; }

#### 11、p:empty

选择**没有内容**的p

<div class=**"box10"**>  
 <p >**1**</p>  
 <p></p>  
 <p >**3**</p>  
</div>

选中第二个P

.box10 p:empty{ background: **blue**; }

#### 12、p:target

选择当前**被锚点激活的p**

<div class=**"box11"**>  
 <a href=**"#tu"**>**点我试试1**</a>  
 <p id=**"tu"**>**1**</p>  
 <a href=**"#tuu"**>**点我试试2**</a>  
 <p id=**"tuu"**>**2**</p>  
</div>

哪个a被点击就会被激活，就会被选中。而另外的就处于没有激活的状态，可以用于建议轮播图片的切换

.box11 p:target{ background: **green**; }

#### 13、p：：selection

**被用户选中的p**

.box12 p::selection{ background: **deeppink**; }

p里面被选中的文本背景变为了粉红色

#### 14、input:disabled

选择**不能被操作**的input框

#### 15、input: enabled

选择**能被操作**的input框

#### 16、input:checked

**选择到被选中的input**

# 五、背景属性

## 1.1 background-origin背景图片起始域

### 1.1.1 background-origin:border-box;

从border区域起始



### 1.1.2 background-origin:padding-box;

默认值，从padding区域起始



### 1.1.3 background-origin:content-box;

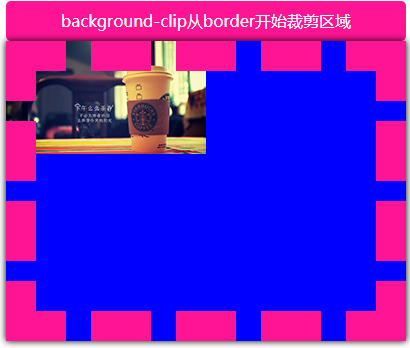
从内容部分起始



## 1.2 裁剪背景区域 background-clip

### 1.2.1 background-clip:border-box;

从border区域开始裁剪背景



### 1.2.2 background-clip:padding-box;

从padding区域开始裁剪背景



### 1.2.3 background-clip:content-box;

从内部区域开始裁剪背景



## 1.3 background-color

**ps:在使用时需要加上 –webkit-兼容前缀**

### 1.3.1 hsl（h,s,l）调色板

H：Hue(色调)。取值为：0 – 360；(120绿，240蓝，360或0红)

S：Saturation(饱和度)。取值为：0.0% - 100.0%；

L：Lightness(亮度)。取值为：0.0% - 100.0%。

### 1.3.2 线性渐变 linear-gradient()

渐变方向：left、right、top、bottom可单独，也可两两组合，注意，没有center

.main div:nth-of-type(15){ background: -webkit-linear-gradient(**left**,**deeppink**,**yellow**,#153170);}

.main div:nth-of-type(16){ background: -webkit-linear-gradient(**left**,**deeppink** 10%,**yellow** 60%,#153170);}

使用百分比的时候表示：从10%以前都是粉红色的没有渐变，而10%-60%就会和黄色搭配产生渐变，60%以后就是黄色域蓝色搭配产生渐变。

### 1.3.3径向渐变 radial-gradient()

渐变方向：right、left、top、bottom两两组合，可center，但必须单独使用。

.main div:nth-of-type(17){ background: -webkit-radial-gradient(**center**,**deeppink**,**yellow**,#153170);}

# 六、文本属性

## 1.1文本从右向左显示

两属性必须搭配使用

.main p:nth-child(1){ direction: **rtl**; unicode-bidi:**bidi-override**; }

## 1.2多行文本超出显示省略号

.main p:nth-child(2){  
 display: **-webkit-box**; /\*继承block的属性\*/  
 -webkit-box-orient:**vertical**; /\*元素垂直显示\*/  
 -webkit-line-clamp:2; /\*设置文本显示的行数\*/  
 overflow: **hidden**; /\*（不能使用padding）\*/  
}

## 1.3 text-shadow 字体阴影

参数： x轴偏移量 Y轴偏移量 模糊度 颜色

### 1.3.1 浮雕

.main div:nth-of-type(1){ text-shadow: -5**px** -5**px** 5**px** #000; color: **deeppink**; }



### 1.3.2 立体字

.main div:nth-of-type(2){  
 color:#fff;  
 text-shadow: 0 1**px** 0 #ccc,0 2**px** 0 #9c9c9c,0 3**px** 0 #bbb, 0 4**px** 0 #9b9b9b,0 5**px** 0 #aaa;  
}



### 1.3.3 模糊字

.main div:nth-of-type(3){ text-shadow: 0 0 10**px** #000; }



### 1.3.4 描边

.main div:nth-of-type(4){ -webkit-text-stroke: 1**px deeppink**; color:#fff; }



### 1.3.5 字体填充

.main div:nth-of-type(5){  
 -webkit-text-fill-color: **blue**;  
 -webkit-text-stroke: 1**px deeppink**;

}



### 1.3.6 霓虹灯

.main div:nth-of-type(6){  
 text-shadow:0 0 10**px** #f3c,0 0 20**px** #f39,0 0 40**px** #f33,0 0 60**px** #f30;  
 color: #fff;  
}



## 1.4 resize

**Resize盒子拖动，同时也可设置文本域防止拖动。**

resize: none; 不可以拖动  
resize: both; 水平和垂直都可以拖动  
resize: vertical; 垂直可以拖动  
resize: horizontal; 水平可以拖动

**PS：不可拖动比原来的宽高小，必须配合overflow: auto;使用**

## 1.5 user-select

**user-select定义文本是否可被选中**

text 默认值，可被选中  
none 不可被选中  
all成标签域复制 给那个标签，点击这个标签就可以选中标签内所有的内容

## 1.6 column 列

**columns规定列的宽度和列数**

column：每列的最小宽度 列数  
column-gap： 列之间的间隔  
column-rule: 粗细 样式 颜色（列之间的边框）

.box4 p{ columns: 100**px** 4; column-gap: 50**px**; column-rule: 5**px solid deeppink**; }

# 七、动画相关

## 1.1 transition过渡

### 1.1.1 单个属性

* transition-property：要过渡的属性名称（如width、height）
* transition-duration：过渡效果持续时间
* transition-delay：延迟过渡时间（可不写）
* transition-timing-function: 过渡效果运行曲线

##### Transition-timing-function：运动曲线

* linear: 匀速
* ease：慢快慢
* ease-in：匀加速
* ease-out：匀减速

### 1.1.2 复合属性

transition： color 2s 3s linear;

## 1.2 animate动画

### 1.2.1 动画的定义

@-webkit-keyframes animationname{

keyframes-selector{css-style}

}

* animationname：自定义动画名称

##### keyframes-selector：动画时长百分比

* 0%-100%（在之间需要执行什么事件）

@-webkit-keyframes move{  
 0%{ left: 0; top: 0;} /\*动画开始执行时的状态\*/  
 25%{ left: 300**px**; top: 0;} /\*0-25%这个阶段动画状态\*/  
 50%{ left: 300**px**; top: 300**px**;}  
 75%{ left: 0; top: 300**px**;}  
 100%{ left: 0; top: 0;}  
}

##### from to

* from(0%) to(100%)
* Css-style：一个或多个合法的css属性

这种定义方式就是有开始和结尾两种状态

@-webkit-keyframes play{  
 from{left: 0; top:0;} 0%  
 to{left: 500px; top:0;} 100%  
}

### 1.2.2 执行属性

#### 1. 单个属性

.main p{

animation-name: **move**; /\*动画名称\*/  
 animation-duration: 4**s**;/\*动画执行时间\*/  
 animation-timing-function: **linear**;/\*动画运行曲线\*/  
 animation-iteration-count: **infinite**;/\*播放次数 infinite无限循环\*/  
 animation-direction: **alternate**;/\*是否应该轮流反向播放动画 alternate反向轮流\*/  
}

.main:nth-child(1):hover p{  
 animation-play-state: **running**;/\*停止状态转换为运动状态\*/  
}  
.main:nth-child(2):hover p{  
 animation-play-state: **paused**;/\*运动状态转换为停止状态\*/  
}

#### 2. 复合属性

animation: move 4s linear infinite alternate;

animation动画属性：函数名 运动时间 运动曲线 播放次数 是否反向播放动画;

## 1.3 transform变换

### 1.3.1 transform：rotate（180deg） （旋转）

##### 单位：

* deg 角度（180°）
* turn 圈

##### 写法：

* transform： rotate(180deg);（默认值，绕Z轴转）
* transform： rotateX(180deg);（绕X轴顺时针转）
* transform： rotateY(180deg);（绕Y轴逆时针转）
* transform： rotateZ(180deg);（绕Z轴顺时针转）

### 1.3.2 transform：scale（0.5） （缩放）

* 大于1表示放大，小于1表示缩小
* 一个值 既代表X轴，也代表Y轴
* 两个值 第一个代表X轴，第二个代表Y轴，逗号隔开

### 1.3.3 transform：translate（X，Y） （位移）

位移后元素初始位置依然占据， 且移动时不会影响其他元素的布局

##### 一个值

* transform：translate（100px）; 代表X轴方向
* transform：translateX（100px）; 代表X轴方向
* transform：translateY（100px）; 代表Y轴方向
* transform：translateZ（100px）; 代表Z轴方向，需要3D环境

##### 两个值

第一个值代表X轴方向，第二个值代表Y轴方向

transform：translate（100px,100px）;

### 1.3.4 transform：skew（X，Y） （倾斜）

##### 一个值

* transform：skew（30deg）; 代表X轴方向
* transform： skewX （30deg）; 代表X轴方向
* transform： skewY （30deg）; 代表Y轴方向

##### 两个值

第一个值代表X轴方向，第二个值代表Y轴方向（逗号隔开）

transform： skew （30px，30px）;

### 1.3.5 变换基点transform-origin

变换的基点（即参考点），默认的基点：绝对中心点，该属性提供两个参数值：

* 如果设置两个，第一个为X轴，第二个为Y轴（两值以空格隔开）
* 如果设置一个，该值为X轴，第二个默认Y轴50%（给定值时）

#### 取值：

##### 百分比

* 百分比指定坐标值，可以为负值
* transform-origin：50% 50%;

##### px

* 长度指定坐标值，可以为负值。
* transform-origin：50px 50px;

##### left、right、center、top、bottom两两组合

* left、right、center X轴
* top、center、bottom Y轴
* 只给一个值，另一个值默认center

## 1.4 3D变换

### 1.4.1 perspective 井深

指定观察者与「z=0」平面的距离，使具有三维位置变换的元素产生透视效果。值越大看起来距自己越近，元素越大。值越小，越远，元素越小。

给需要进行3D动画的盒子的父元素

值：一般使用 看起来最适应800px 或者 1200px

测试网址：http://www.css88.com/book/css/properties/transform/perspective.htm

### 1.4.2 transform-style 3D环境

定义： 指定子元素定位在三维空间内， 当该属性值为「preserve-3d」时，元素将会创建局部堆叠上下文。

给需要进行3D动画的盒子的父元素或者，设置给需要进行3D的盒子

值：transform-style: preserve-3d 创建3D环境

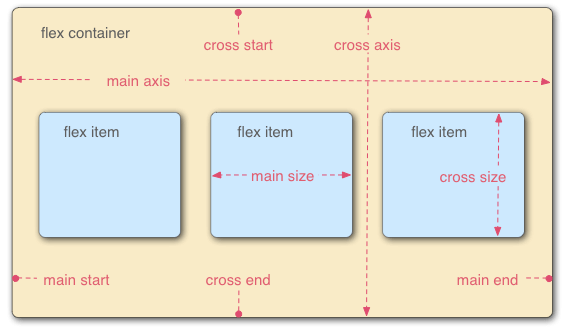
# 八、弹性盒子模型

## 1.1 概念

### 1.1.1 弹性盒子模型概念

布局的传统解决方案，基于盒状模型，依赖 display属性 + position属性 + float属性。它对于那些特殊布局非常不方便，比如，垂直居中就不容易实现。2009年，W3C提出了一种新的方案—-Flex布局，可以简便、完整、响应式地实现各种页面布局。目前，它已经得到了所有浏览器的支持，这意味着，现在就能很安全地使用这项功能。**采用Flex布局的元素，称为Flex容器**（flex container），简称”容器”。它的**所有子元素自动成为容器成员，称为Flex项目**（flex item），简称”项目”。

* 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。
* 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。



## 1.2 容器属性（给父元素）

**首要条件：display: flex;**

以下6个属性设置在父元素容器上

* flex-direction
* flex-wrap
* flex-flow
* justify-content
* align-items
* align-content

### 1.2.1 flex-direction（给容器）

##### 定义： 决定主轴的方向

##### 取值：

* row：从左到右排列（abc）。
* row-reverse: 从左到右排列，但元素顺序与row相反（cba）
* column: 主轴为垂直方向，起点在上沿。
* column-reverse: 主轴为垂直方向，起点在下沿。

### 1.2.2 flex-wrap（给容器）

##### 定义： 该属性控制flex容器是单行或者多行

##### 值：

* nowrap： 不换行。该情况下flex子项可能会溢出容器（默认值）
* wrap：flex容器为多行。该情况下flex子项溢出的部分会被放置到新行，换行。
* wrap-reverse: 反转 wrap 排列。（注意：不是倒序）

### 1.2.3 flex-flow（给容器）

##### 定义：

flex-direction和flex-wrap的复合简写形式，默认为row nowrap

##### 值：

flex-direction

flex-wrap

##### 写法：

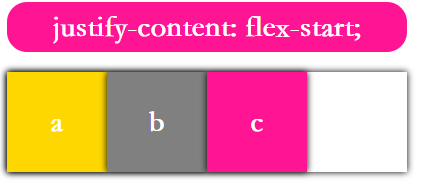
flex-flow: flex-direction flex-wrap

### 1.2.4 Justify-content

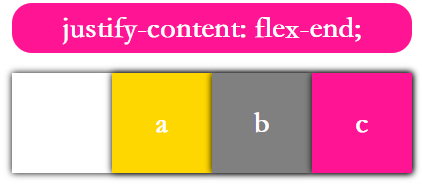
##### 定义：

**定义了项目在主轴（通常是x轴）上的对其方式**

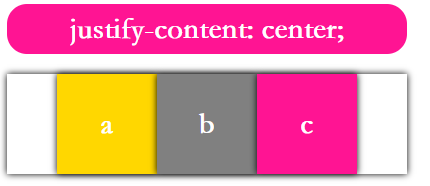
#### 1、justify-content：flex-start;（左对齐）



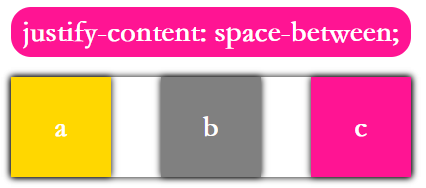
#### 2、justify-content: flex-end；（右对齐）



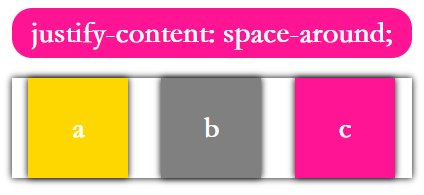
#### 3、justify-content: center；（居中对齐）



#### 4、justify-content: space-between；（两端对齐，项目之间间隔相等）



#### 5、justify-content: space-around；（每个项目两侧间隔相等）



#### 1.2.4.2 最后一行不between

可以在所有项的后面添加隐藏的li元素，一行本来多少项就添加多少项隐藏元素

<ul class="work">

<li></li>

<li></li>

<li></li>

<li></li>

<li></li>

<li></li>

<li style="height: 0px;visibility: hidden;">div.box</li>

<li style="height: 0px;visibility: hidden;">div.box</li>

<li style="height: 0px;visibility: hidden;">div.box</li>

<li style="height: 0px;visibility: hidden;">div.box</li>

</ul>

.work{

display: flex;

   flex-wrap: wrap;

    justify-content:space-between;

}

#### 1.2.4.3 js动态添加

<ul class="work">

<li class="default" @click="add">点击创建项目</li>

<li style="height: 0px;visibility: hidden;">div.box</li>

<li style="height: 0px;visibility: hidden;">div.box</li>

<li style="height: 0px;visibility: hidden;">div.box</li>

<li style="height: 0px;visibility: hidden;">div.box</li>

</ul>

let a = prompt("请输入项目名称：")

let li = $("<li>"+a+"</li>")

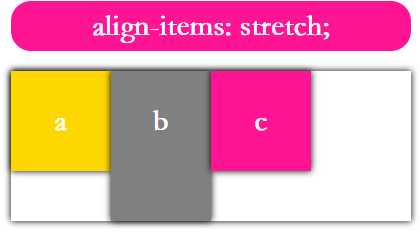
*if*(a) $(".default").after(li)

### 1.2.5 align-items

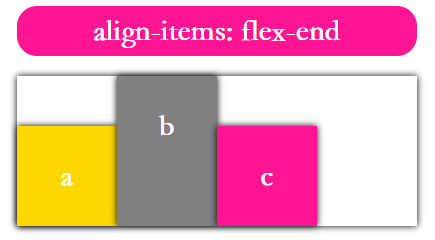
##### 定义：

**弹性盒子元素在单个交叉轴对齐方式。**

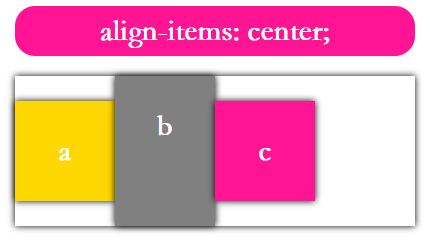
#### 1、align-items：stretch;（默认值）



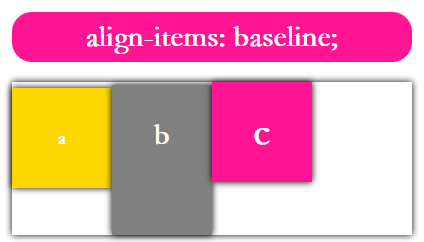
#### 2、align-items：flex-end;（交叉轴的终点对齐）



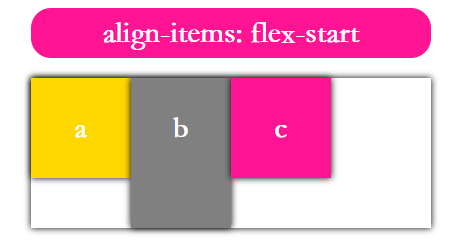
#### 3、align-items：flex-center;（交叉轴的中点对齐）



#### 4、align-items：baseline;（项目的第一行文字的基线对齐）



#### 5、align-items：flex-start;（交叉轴的起点对齐）

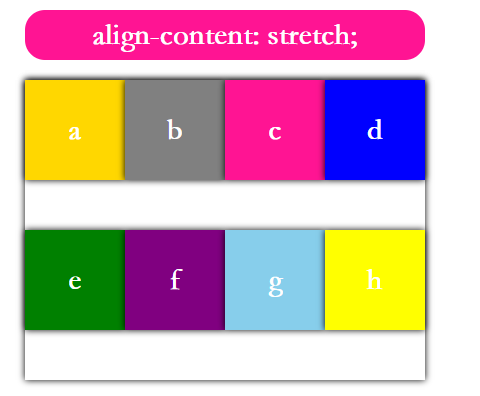


### 1.2.6 align-content

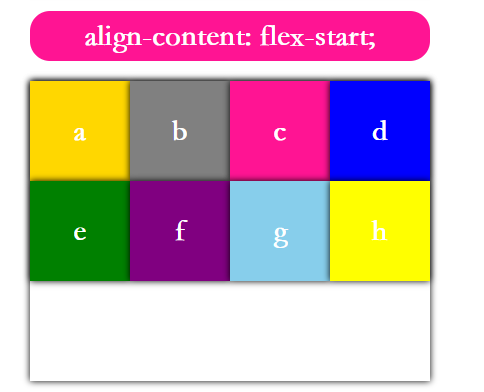
##### 定义：

**定义了多根主轴对齐方式，如果项目只有一根轴线，则不起作用**

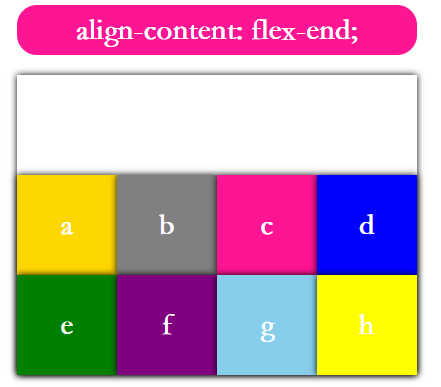
#### 1、align-content: stretch（默认值，轴线占满整个交叉轴）



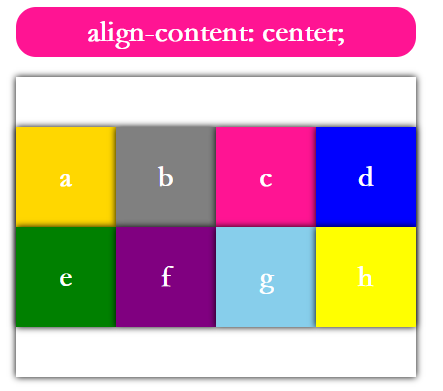
#### 2、align-content: flex-star（与交叉轴上沿对齐）



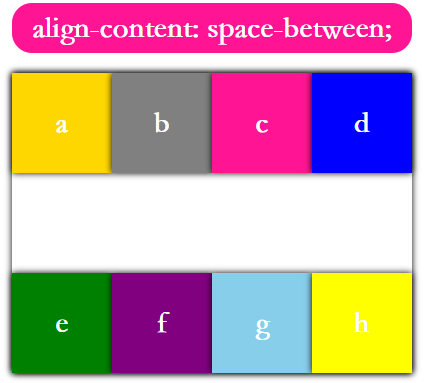
#### 3、align-content: flex-end（与交叉轴下沿对齐）



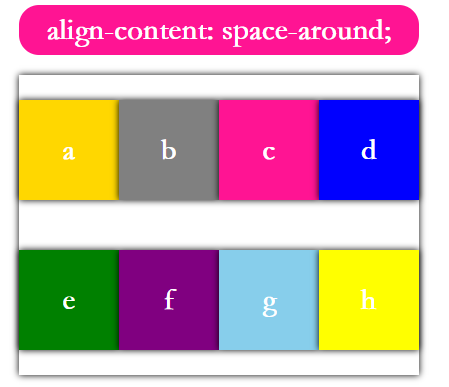
#### 4、align-content: center（与交叉轴中部对齐）



#### 5、align-content: space-between（与交叉轴两端对齐，中间主轴宽度平均分配）



#### 6、align-content: space-around（交叉轴两侧间隔相等）



## 1.3 项目属性

**以下6个属性设置在项目上：**

* flex-grow
* flex-shrink
* flex-basis
* flex
* align-self
* order

### 1.3.1 flex-grow

##### 定义：

定义项目放大比例，默认为0（即如果存在剩余空间，也不放大。 ）

.flex {  
 display: **flex**;  
 width: 400**px**;  
 line-height: 100**px**;  
 box-shadow: 0 0 10**px** 0 #000;  
}

#### 1、没有宽度，直接按照比例分配

.box1 .flex li:nth-child(1){ flex-grow:1; }  
.box1 .flex li:nth-child(2){flex-grow:1; }  
.box1 .flex li:nth-child(3){flex-grow:2; }

**最终结果： 100 100 200**

#### 2、有宽度，给的值不一样的时候

实际宽度： 100+100\*1/5=120 100+100\*1/5=120 100+100\*3/5=160

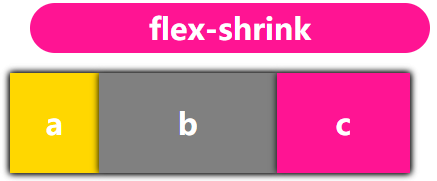
.box2 .flex li:nth-child(1){ width: 100**px**; flex-grow: 1; }  
.box2 .flex li:nth-child(2){ width: 100**px**; flex-grow:1;}  
.box2 .flex li:nth-child(3){ width: 100**px**; flex-grow:3;}

### 1.3.2 flex-shrink

flex-shrink:定义项目缩小比例，默认为1，即如果空间不足，该项目将缩小

容器宽度=400  
项目宽度=500 超出=500-400=100  
加权综合：盒子宽度\*shrink值 =100\*1+200\*1+200\*3=900  
收缩值：（盒子宽度\*shrink/加权）=(100\*1/900)\*100= 100/9  
 (200\*1/900)\*100= 200/9  
 (200\*3/900)\*100= 200/3  
  
最终值：盒子宽度-收缩值 =100-100/9= 88  
 200-200/9= 177  
 200-200/3= 133

.box3 .flex li:nth-child(1){ flex-shrink:1; width: 100**px**;}  
.box3 .flex li:nth-child(2){ flex-shrink:1; width: 200**px**;}  
.box3 .flex li:nth-child(3){ flex-shrink:3; width: 200**px**;}



### 1.3.3 flex-basis

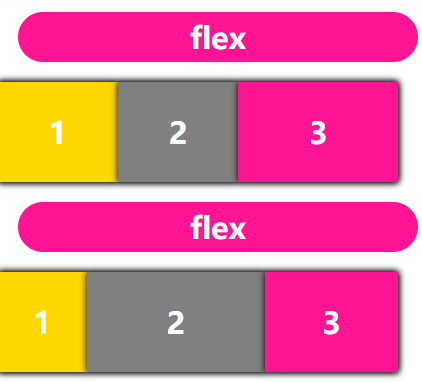
flex-basis实际上和width一样，只不过浏览器分配空间要根据这个值来算，没有这个值得时候默认为auto（以width或者内容撑开的宽度计算）

.box4 .flex li:nth-child(1){ flex-basis: 150**px**;}  
.box4 .flex li:nth-child(2){ flex-basis: 100**px**;}  
.box4 .flex li:nth-child(3){ flex-basis: 50**px**;}

### 1.3.4 flex

flex： flex-grow，flex-shrink，flex-basis的复合属性  
flex：0 1 auto；（默认,auto就相当于没有给basis就以自身宽度或者内容宽度计算）  
当父级宽度改变时，会自动的根据给定的比例进行伸缩计算

.box5 .flex li:nth-child(1){ flex: 1 1 100**px**;}  
.box5 .flex li:nth-child(2){ flex: 1 1 100**px**;}  
.box5 .flex li:nth-child(3){ flex: 3 3 100**px**;}  
  
.box6 .flex li:nth-child(1){ flex: 1 1 100**px**;}  
.box6 .flex li:nth-child(2){ flex: 1 1 200**px**;}  
.box6 .flex li:nth-child(3){ flex: 3 3 200**px**;}



### 1.3.5 align-self

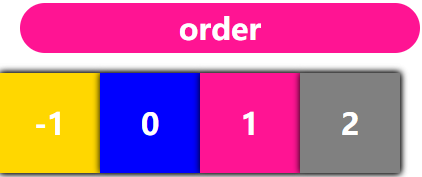
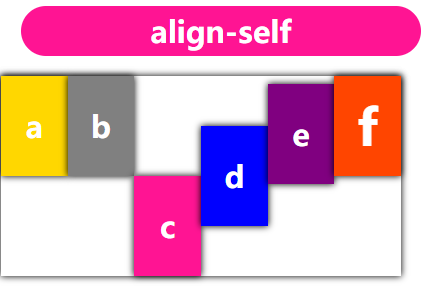
align-self和我们之前的容器属性类似，不过这一个对齐是给自己，而我们刚才的是给所有的项目属性，那这里我们也可以看出，容器属性是让全部的项目怎样做，而项目属性是让项目自己怎么做.  
align-self是自己在交叉轴的对齐方式，可覆盖容器属性align-items

.box7 .flex{height: 200**px**; }  
.box7 .flex li{ width: 100**px**;}  
.box7 .flex li:nth-child(1){ align-self: **stretch**;}  
.box7 .flex li:nth-child(2){ align-self: **flex-start**;}  
.box7 .flex li:nth-child(3){ align-self: **flex-end**;}  
.box7 .flex li:nth-child(4){ align-self: **center**;}   
.box7 .flex li:nth-child(5){ align-self: **baseline**; font-size: 30**px**;}  
.box7 .flex li:nth-child(6){ align-self: **baseline**; font-size: 50**px**;}

### 1.3.6 order

order排序，值越大越靠后，允许负值

.box8 .flex li{width:100**px**; height: 100**px**; }  
.box8 .flex li:nth-child(1){ order: -1;}  
.box8 .flex li:nth-child(2){ order: 2;}  
.box8 .flex li:nth-child(3){ order: 1;}  
.box8 .flex li:nth-child(4){ order: 0;}



## 1.4 怪异盒子模型

怪异盒子模型  
 实际宽度= 原来的宽度-pdding-左右border（注意：这里指的宽度是内容的宽度，也就是给定的宽度width:100px）

神奇的是，我们给了盒子padding和border以后不会使我们原来的盒子变大，好像是向内压缩，减小了我们内容的宽度，这就解决了我们平时给padding使我们盒子变大的弊端。

## 1.5 倒影

倒影 box-reflect: direction offset mask-box-image  
direction: above：指定倒影在对象的上边  
 below：指定倒影在对象的下边  
 left：指定倒影在对象的左边  
 right：指定倒影在对象的右边  
offset:（对象与倒影的间隔） 单位：px/百分比 甚至还可以拉到负的距离  
mask-box-img（遮罩图片）: 可以用渐变 也可以用图片（图片要是png的图片）

# 九、HTML5智能表单

#### 1、email

验证是否为正确的email格式，如果不，点击提交按钮会显示提示。

<input type=**"email"** class=**"in1"**>

#### 2、tel

规定表单内容是数字，跳转到数字键盘（只有在移动端才会显示）

input type=**"tel"** class=**"in1"**>

#### 3、url

规定表单内容是网址，如果不，则会显示提示。

<input type=**"url"** class=**"in1"**>

#### 4、range

拖动条，step属性可以设置每次变换大小，支持value，max，min等属性

<input type=**"range"** class=**"in1"** step=**"2"** max=**"10"** min=**"0"** value=**"5"**>

#### 5、search

在表单内输入内容之后会显示一个关闭符号，可以删除输入的内容

<input type=**"search"** class=**"in1"**>

#### 6、number

输入数字可以点击加减

<input type=**"number"** class=**"in1"**>

#### 7、color

点击弹出调色板

<input type=**"color"** class=**"in1"**>

#### 8、datetime-local

显示完整日期，不含时区

<input type=**"datetime-local"** class=**"in1"**>

#### 9、time

显示时间，不含时区

<input type=**"time"** class=**"in1"**>

#### 10、date

显示日期

<input type=**"date"** class=**"in1"**>

#### 11、week

显示周

<input type=**"week"** class=**"in1"**>

#### 12、month

显示月

<input type=**"month"** class=**"in1"**>

#### 13、required

是属性 输入的内容不能为空，如为空会有相应的提示

<input type=**"text"** class=**"in1"** required>

#### 14、datalist

输入关键字显示备用选项 list属性对应datalist的id值

**请填写你的性别：**<input type=**"text"** class=**"in1"** list=**"lis"**>  
<datalist id=**"lis"**>  
 <option value=**"男"**></option>  
 <option value=**"女"**></option>  
 <option value=**"人妖"**></option>  
</datalist>