# Flex布局专题案例开发

CSS中，我们说过有三大布局模型：流动布局，浮动布局，层布局。

今天所说的Flex布局是一种新的布局模型。

笔记参考来源：<http://www.ruanyifeng.com/blog/2015/07/flex-grammar.html>

# ：Flex布局是什么？

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。Flex是Flexible Box的缩写，意为"弹性布局"，用来为盒状模型提供最大的灵活性。



在手机淘宝的web app上就是使用flex布局：



Flex就是一个css属性，它是display的一个属性值。 display:flex。

它是C3中的一个新属性。这个属性非常的强大。

# ：基本概念

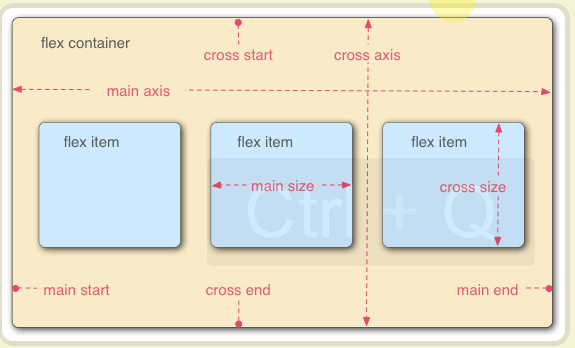
**容器和项目：**

* 容器：如果我们给一个盒子加上了一个属性display:flex，那么这个盒子就是一个容器。它里面的直接子元素就是项目。
* 项目：容器中的直接子元素，叫项目。

**主轴和交叉轴：**

* 主轴：在容器中，项目按一定的方向进行排列的，默认是从左到右进行排列，这个排列的方向，我们称为主轴。
* 交叉轴：就是和主轴重直的那个轴，称交叉轴。

容器默认存在两根轴，水平的叫主轴，主轴有一个开始位置，也有一个结束位置，交叉轴也有一个开始位置，和一个结束位置。



# ：容器的属性

在容器中有6大属性，需要灵活应用，牢固掌握：

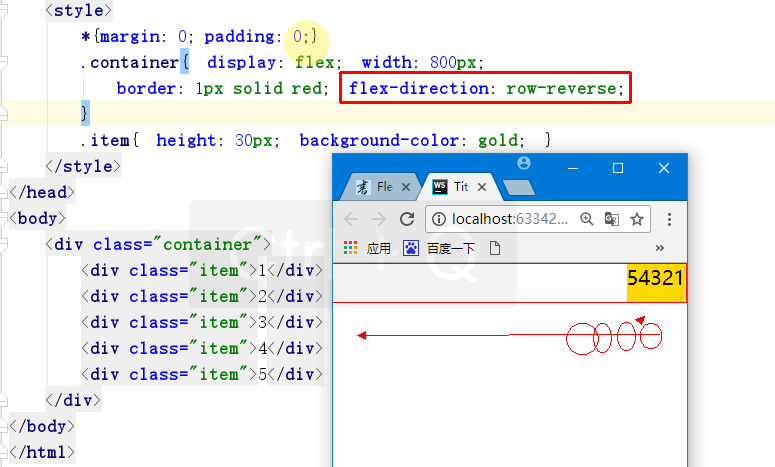
* flex-direction
* flex-wrap
* flex-flow
* justify-content
* align-items
* align-content

## flex-direction

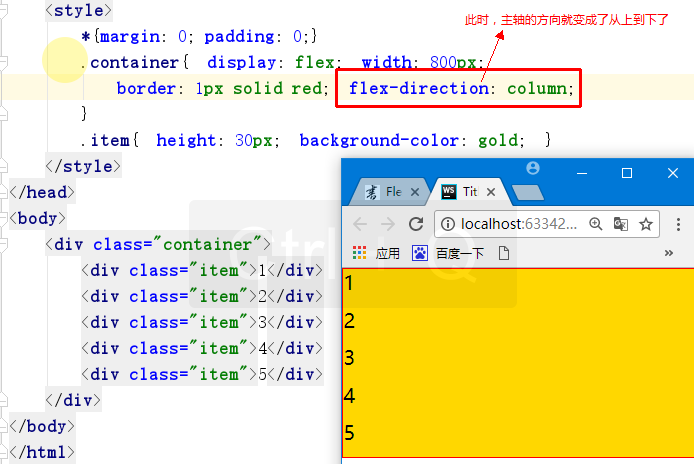
功能：用来设置主轴的方向的

取值：row | row-reverse | column | column-reverse

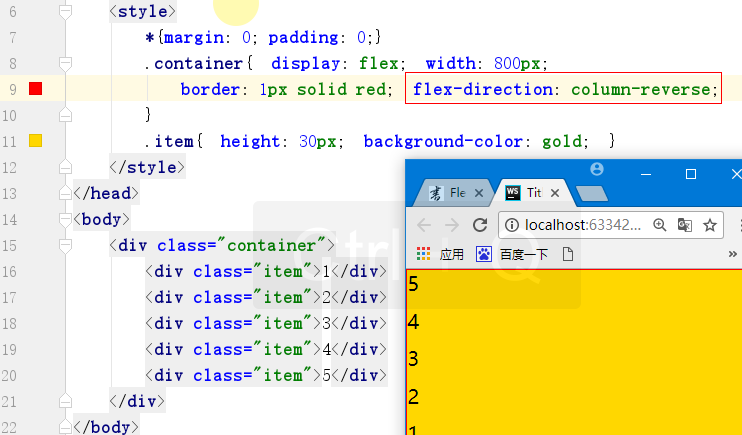
代码如下：



------------------------------------------------------------



------------------------------------------------------------



row 默认值，主轴方向为水平方向，起点在左边。

row-reverse 主轴也是水平方向，起点为右边

column 主轴是垂直的，起点在上面

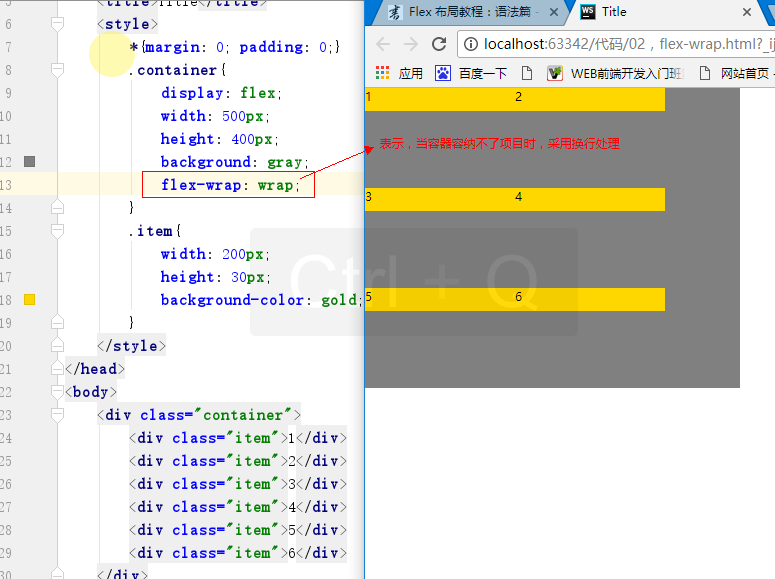
column-reverse 主轴也是垂直的，起点在下面

## flex-wrap

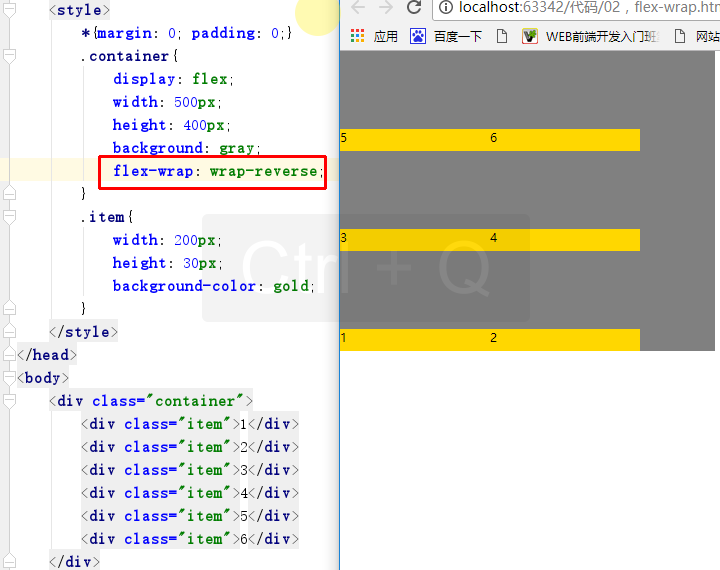
功能：用来控制容器中的项目是否换行

取值：wrap | no-wrap | wrap-reverse

代码如下：



------------------------------------------------------------



wrap 表示换行，宽度不会被挤压

nowrap 表示不换行，是默认的，如果说不换行，到时候，它的宽度会被挤压。

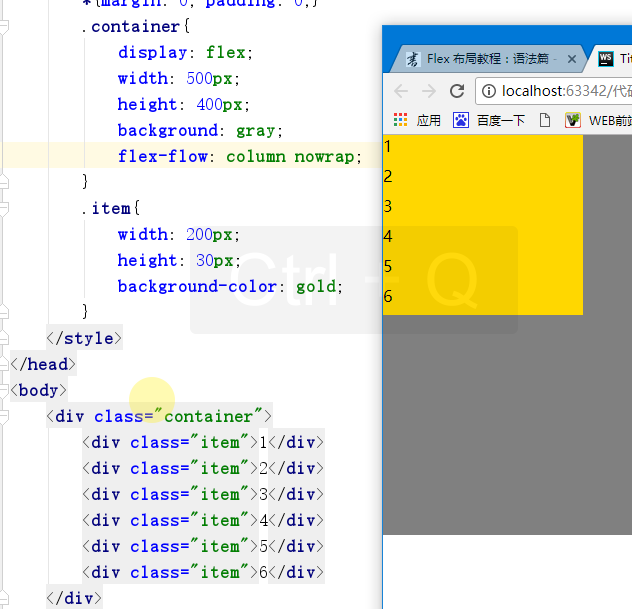
wrap-reverse 表示换行，第一行，会出现在最下面

## flex-flow

功能：简化代码，是flex-direction和flex-wrap的合写

取值：flex-flow:flex-direction，flex-wrap

代码如下：

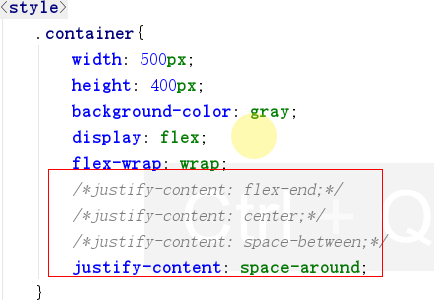


## justify-content

功能：用来处理富余空间。

取值：flex-start | flex-end | center | space-between | space-around

基本用法：



**flex-start（默认值）：左对齐**

 **flex-end：右对齐**



**center： 居中**



**space-between：两端对齐，项目之间的间隔都相等。**



**space-around：每个项目两侧的间隔相等。所以，项目之间的间隔比项目与边框的间隔大一倍。**



## align-items

功能：设置项目在交叉轴上的摆放位置

取值：flex-start | flex-end | center | baseline | stretch

基本用法：

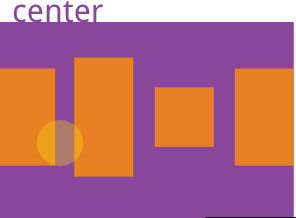
看下面。

**flex-start：交叉轴的起点对齐。**

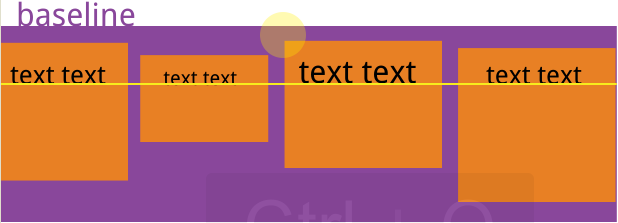
 **flex-end：交叉轴的终点对齐。**



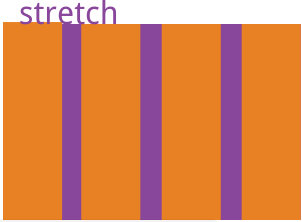
**center：交叉轴的中点对齐。**



**baseline: 项目的第一行文字的基线对齐。**



**stretch（默认值）：如果项目未设置高度或设为auto，将占满整个容器的高度。**



## align-content

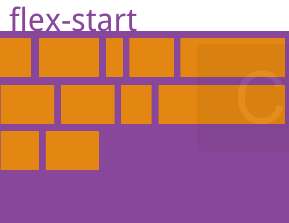
功能：当容器中有多行项目（前提是允许换行），每一行项目就会产生一根主轴，多行项目就有多根主轴。这个属性决定这多根主轴之间的位置关系。

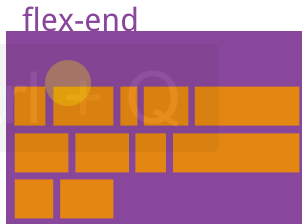
取值：flex-start | flex-end | center | space-between | stretch | space-around

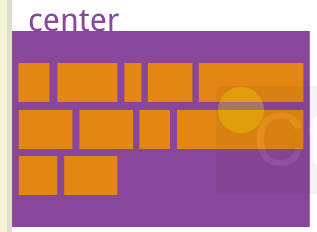
基本用法：

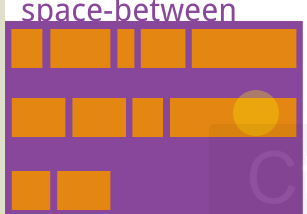
看代码！

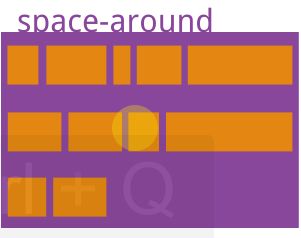
**flex-start：与交叉轴的起点对齐。**

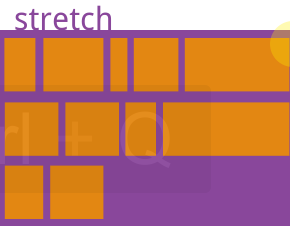
 **flex-end：与交叉轴的终点对齐。**

 **center：与交叉轴的中点对齐。**

 **space-between：与交叉轴两端对齐，轴线之间的间隔平均分布。**

 **space-around：每根轴线两侧的间隔都相等。所以，轴线之间的间隔比轴线与边框的间隔大一倍。**

 **stretch（默认值）：轴线占满整个交叉轴。**



# ：项目的属性

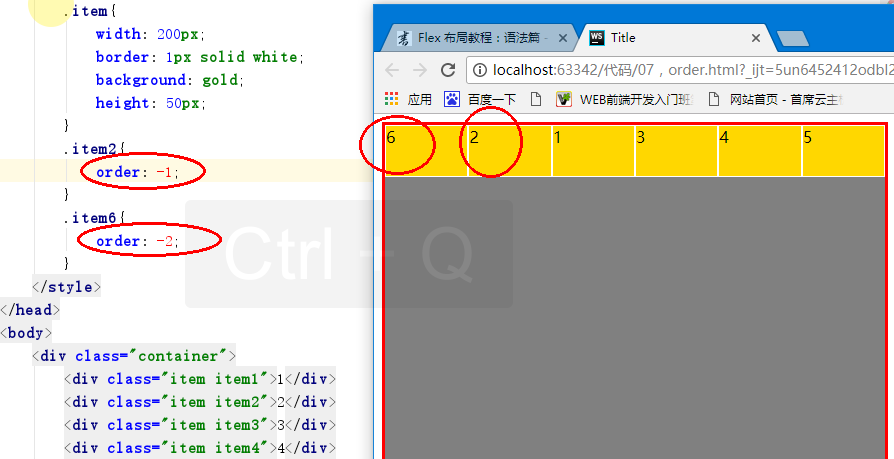
* order
* flex-grow
* flex-shrink
* flex-basis
* flex
* align-self

## Order

功能：表示项目在容器中的一个排列顺序。 默认是0， 数值超小，越靠前，可以是负值。

取值：数值。

基本用法：

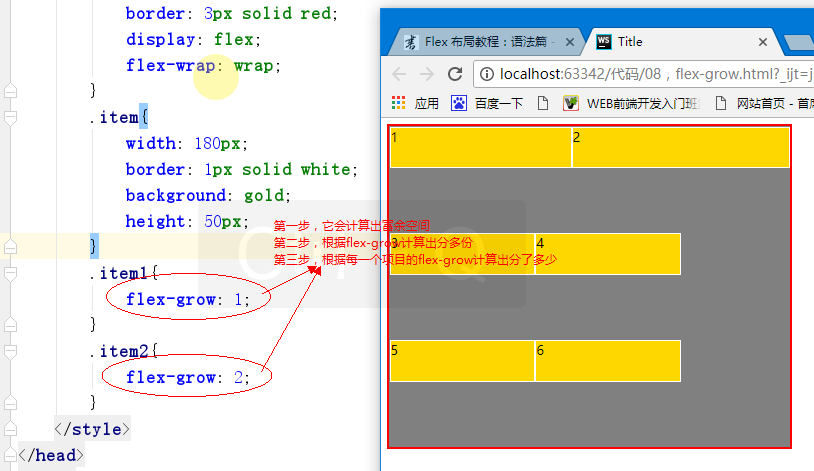


## flex-grow

功能：放大因子。 使用前提：主轴方向上有多余的富余空间只可以让项目去伸展。

取值：数值 默认值0， 表示不去放大项目。

基本使用：



计算步骤：

1，统计多余的空间：M = 容器的宽度 - 所有项目的宽度之和

2，确定均分的份数：N = 项目的flex-grow的值之和

3，计算单位空间：P = M/N

4，项目放大后的宽度：R = 项目之前的宽度 + P \* 当前的放大因子

结论：

1，如果一个项目的flex-grow为0。 则表示它的宽度不会放大。

2，项目的flex-grow越大，说明它在放大后，会得到更多的空间去伸展。

3，项目的flex-grow越小，说明它在放大后，会得到更少的空间去伸展。

4，如果所有的项目的flex-grow一样，那么它们会均分多余的空间。

## flex-shrink

功能：缩小因子。使用前提：在主轴方向上空间不够时，项目被压缩。默认值为1. 负值是没有效的。

说明：当容器主轴方向不能容纳所有项目时，并且也不允许换行，由于flex-shrink为1.所有项目都会被压缩，我们可以改变这个值，来改变压缩比例。

计算步骤：

1，总差值，及需要被压缩的大小： M = 容器的宽度 - 项目的宽度之和

2，加权和： N = flex-shrink \* 项目宽度值之和

3，缩小后的宽度 = 项目值 - 项目值\*flex-shrink / N \* M

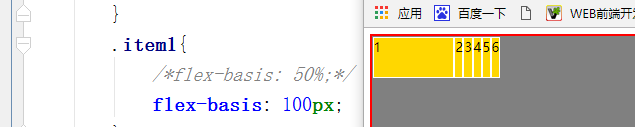
## flex-basis

**功能：**

flex-basis属性定义了在分配多余空间之前，项目占据的主轴空间（main size）。浏览器根据这个属性，计算主轴是否有多余空间。它的默认值为auto，即项目的本来大小。它可以设为跟width或height属性一样的值（比如350px），则项目将占据固定空间。

取值：数值

代码：



## Flex

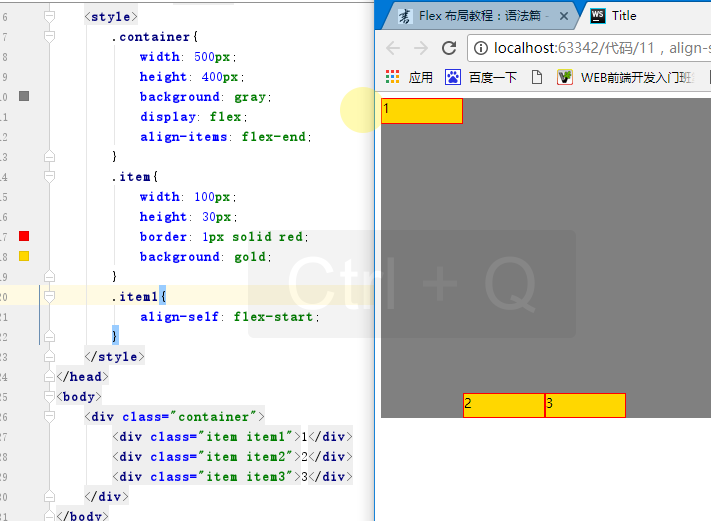
功能：flex属性是flex-grow, flex-shrink 和 flex-basis的简写，默认值为0 1 auto。后两个属性可选。

## align-self

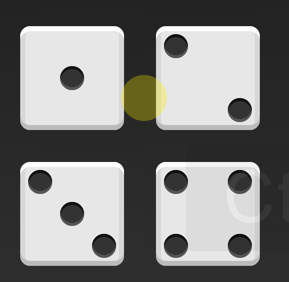
功能：用来设置单个项目的一个对齐方式

取值：flex-start | flex-end | center | baseline | stretch | auto

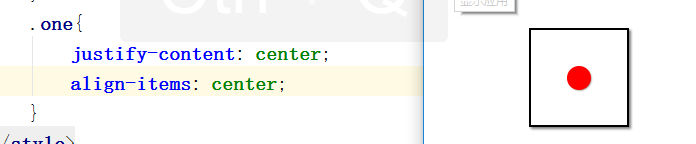
代码如下：



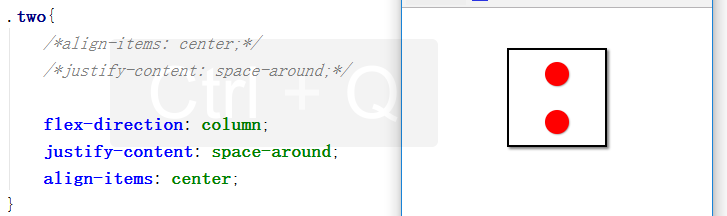
# ：色子案例练习



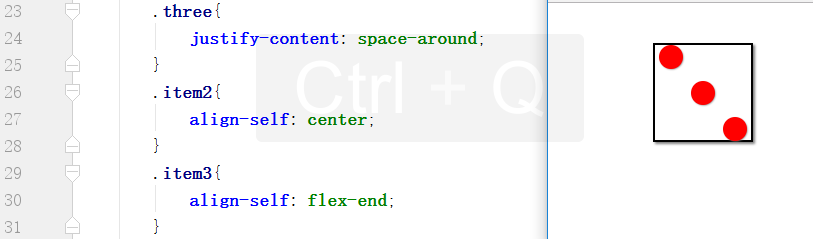
**一点**



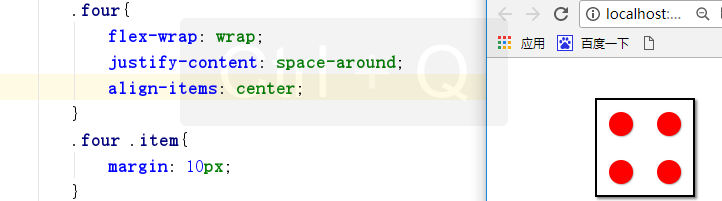
**二点**



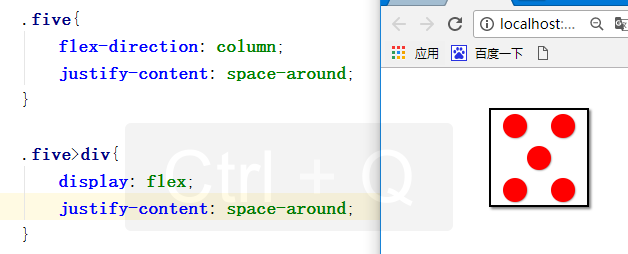
**三点**



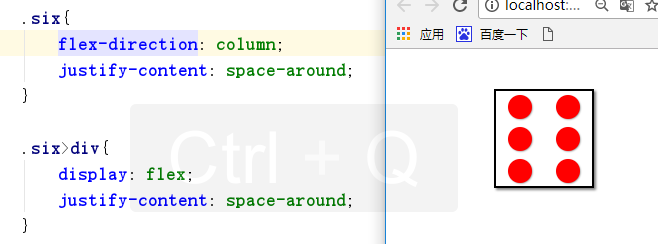
**四点**



**五点**



**六点**

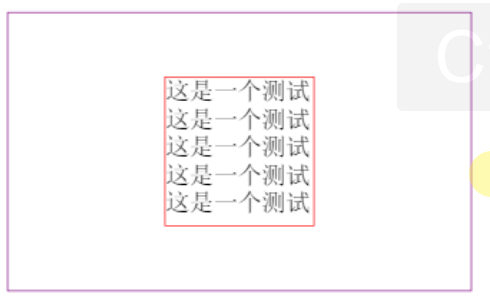


# ：flex的其它布局方式

笔记来源：<http://www.cnblogs.com/lynnmn/p/6262941.html>

## 元素居中

效果：



代码：

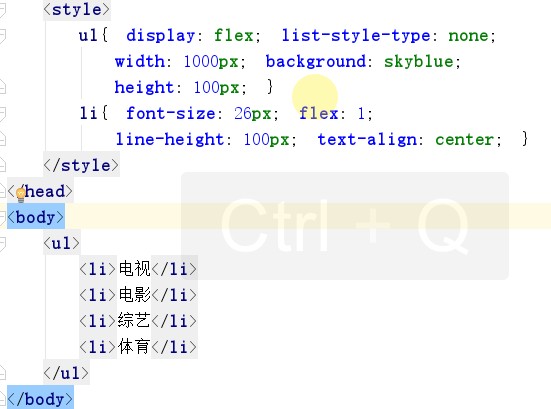


## 导航栏

效果：



代码：



## 左图右文

效果：



代码：

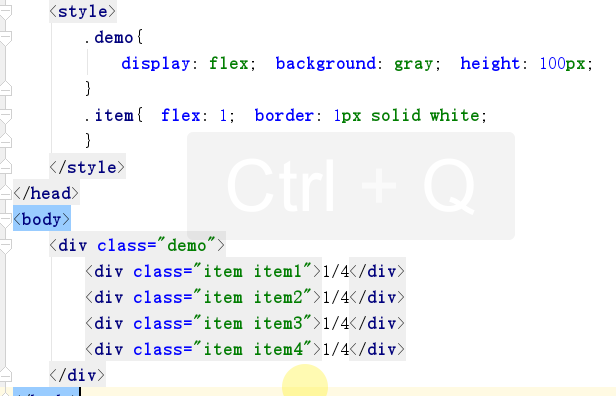


## 百分比布局

效果：

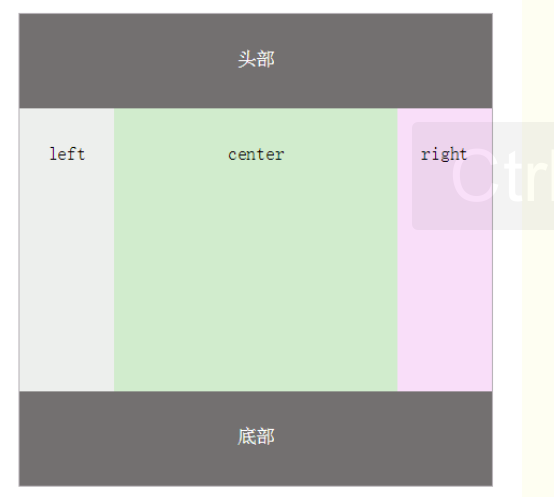


代码：



看代码！

## 圣杯布局

效果：  


代码：



## （6）输入框布局

## （7）底部固定

略

# ：百度糯米实战

**效果：**

