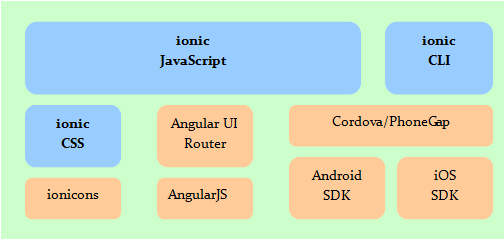
### ionic

ionic是一个强大的\_混合式/hybrid\_HTML5移动开发框架，特点是使用标准的HTML、 CSS和JavaScript，开发跨平台（目前支持：Android、iOS，计划支持：Windows Phone、Firefox OS） 的原生App应用：



ionic主要包括三个部分：

* ****CSS框架**** - 提供原生\_App质感的CSS样式模拟\_。ionic这部分的实现使用了ionicons图标样式库。
* ****JavaScript框架**** - 提供移动Web应用开发框架。ionic基于AngularJS基础框架开发，因此 自然地遵循AngularJS的框架约束；此外，ionic使用AngularJS UI Router实现前端路由。
* ****命令行/CLI**** - 命令行工具集用来简化应用的开发、构造和仿真运行。ionic命令行工具使用了 Cordova，依赖于平台SDK（Android & iOS）实现将移动web项目打包成原生app。

由于ionic使用了HTML5和CSS3的一些新规范，所以要求 iOS7+/ Android4.1+。 在低于这些版本的手机上使用ionic开发的应用，有时会发生莫名其妙的问题。

### CSS框架

如果你对AngularJS这样的东西不感兴趣，可以只使用ionic的CSS框架：直接在HTML中引入 ionic.css就可以了。

ionic的CSS框架主要提供预定义的CSS类，来帮助我们快速构建适用于手机端的UI。 ionic的预定义CSS类主要分四个方面：

* ****基本布局类****

ionic将手机页面的布局模式基本抽象为三块：头、内容、尾。基本布局类提供了 这几个区域的CSS类。

* ****颜色和图标类****

ionic定义了几个配色方案CSS类，并使用ionicons提供的字体图标类。

* ****界面组件类****

ionic定义了丰富的界面组件CSS类，让HTML元素看起来像移动平台的UI组件。

* ****栅格系统类****

和Bootstrap一样，ionic也提供了栅格系统。不过ionic的实现是基于CSS3的FlexBox 模型，更为灵活。

### 布局模式

手机App开发实践中，用户界面通常划分为几个区域 - 标题/header、内容/content和页脚/footer。 微信采用的就是典型的三段布局：



标题区总是位于屏幕顶部，页脚区总是位于屏幕底部，而内容区占据剩余的空间。 ionic使用以下CSS类声明区域性质:

* ****.bar.bar-header**** - 声明元素为标题区
* ****.bar.bar-footer**** - 声明元素为页脚区
* ****.content**** - 声明元素为内容区

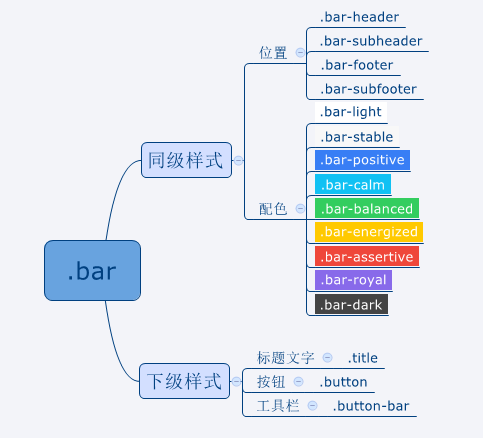
### 定高条块：.bar

样式****.bar****将元素声明为屏幕上绝对定位的块状区域，具有 固定的高度（44px）：

1. <any class="bar">...</any>

一旦元素应用了.bar样式，就可以继续选用两类预定义样式来进一步 声明元素及其内容的外观：

1. 同级样式 - 同级样式与.bar应用在同一元素上，声明元素的位置、配色等。
2. 下级样式 - 下级样式只能应用在.bar的子元素上，声明子元素的大小等特征。



### .bar : 位置

ionic使用以下样式定义条块的位置：

* ****.bar-header**** - 置顶
* ****.bar-subheader**** - header之下置顶
* ****.bar-footer**** - 置底
* ****.bar-subfooter**** - footer之上置底

在腾讯新闻App中，你可以看到，使用了三个条块：标题、副标题、页脚：



将副标题栏移到页脚试试！

### .bar : 嵌入子元素

在ionic中，有三种.bar子元素的样式是预定义的：

****标题文字**** - 对包含标题文字的元素应用****.title****样式，通常使用h1元素：

* 1. <any class="bar">
  2. <any class="title">...</any>
  3. </any>

****按钮**** - 对用作按钮的元素，应用****.button****样式，通常使用button 或a元素作为按钮。注意按钮将使用.bar的配色方案：

* 1. <any class="bar">
  2. <any class="button">...</any>
  3. </any>

****工具栏**** - 工具栏包含一组按钮。对用作工具栏的元素，应用****.button-bar****样式，通常 使用div元素作为工具栏：

* 1. <any class="bar">
  2. <any class="button-bar">...</any>
  3. </any>

在示例代码的标题栏中左边加入一个按钮！

### .bar : 嵌入input

一种常见的UI模式是在标题栏中嵌入搜索栏，比如大众点评：



在.bar元素中嵌入input元素，需要注意两点：

1. 在条块元素上应用****.item-input-inset****样式
2. 将input包裹在应用****.item-input-wrapper****样式的元素内

这是因为，在ionic的实现中，.bar中的.input 样式定义如下：

1. .bar.item-input-inset{
2. .item-input-wrapper{
3. .input{
4. ...
5. }
6. }
7. }

<div class="bar bar-header bar-energized item-input-inset">

<a class="button button-clear icon-right ion-android-arrow-dropdown">北京</a>

<label class="item-input-wrapper">

<input type="text" placeholder="输入商户名搜索">

</label>

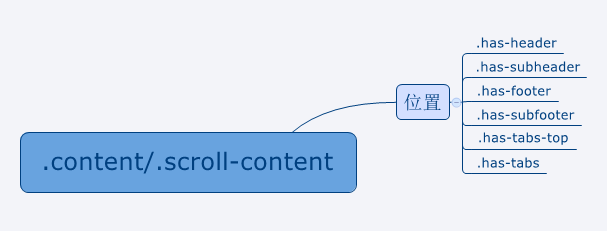
</div>

### 内容：.content和.scroll-content

ionic预定义了两个内容容器样式：

* ****.content**** - 流式定位，内容元素在文档流中按顺序定位
* ****.scroll-content**** - 绝对定位，内容元素占满整个屏幕

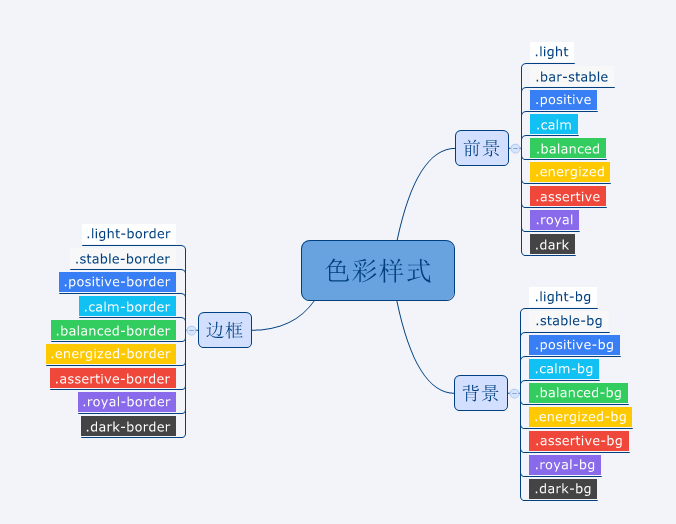
这两种样式都可以使用以下样式进一步确定位置及范围：



右边示例的内容区域，把has-header去掉，看看有什么变化？ 2. 将示例的scroll-content改成content，看看有什么变化？

### 色彩

ionic定义了九种前景/背景/边框的色彩样式，：



可以在任何元素上使用这些样式设置前景和背景颜色：

1. <any class="positive-bg energized">
2. ...
3. </any>

### 图标

ionic使用****ionicons****图标样式库。ionicons采用了TrueType 字体实现图标样式，有超过500个图标可供选择。

使用图标很简单，在元素上声明以下两个CSS类即可：

* ****.icon**** - 将元素声明为图标
* ****.ion-{icon-name}**** - 声明要使用的具体图标

通常使用i元素定义图标，例如下面声明了元素显示ion-home图标：



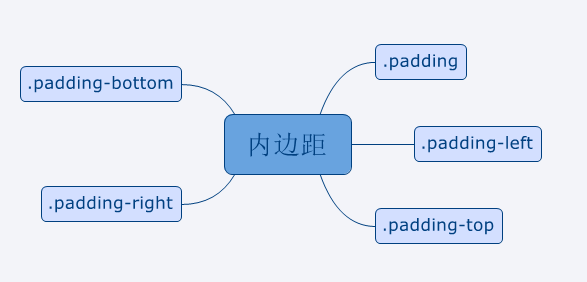
要了解有哪些图标及具体名称，需要访问[ionics.com](http://ionicons.com/)。 点击某个图标即可查看其CSS类名称。

可以在任何元素中插入图标，使用元素的font-size样式指定图标的大小：

1. <any style="font-size:100px;">
3. </any>

### 内边距

ionic定义了常用的内边距样式：



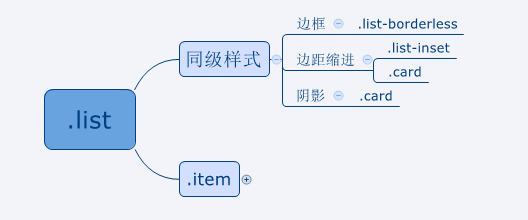
样式名很直白，边距统一为10px。可以在任何元素上应用这些样式。

### 列表 : .list

列表非常适合于手机屏幕上的信息的显示。使用****.list****定义列表容器， 使用****.item****定义列表成员：

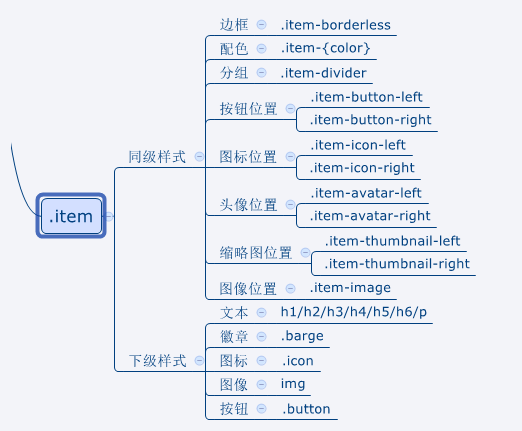
1. <any class="list">
2. <any class="item">...</any>
3. <any class="item">...</any>
4. <any class="item">...</any>
5. <any class="item">...</any>
6. </any>

对列表外观的定制化主要集中在****.item****元素上，****.list****元素仅有 少数的几个样式定义：



### 成员容器 : .item

列表的样式定制主要发生在****.item****元素上。在****.item****元素内， 可以插入文本、徽章、图标、图像（头像、缩略图或大图）、按钮等各种 样式的元素：



### .item : 嵌入文本

列表成员中经常需要显示不同规格的文本，比如今日头条的新闻列表：



****.item****元素可以使用h1~h6/p标签插入不同规格的文本。

### .item : 嵌入图标

列表成员的内容中插入图标，比单纯的文字更加生动：



要插入图标，需要满足两个条件：

1. 在.item元素上声明图标位置。图标可以位于列表的左侧或右侧， 分别使用.item-icon-left和.item-icon-right声明
2. 在.item元素内插入图标。

<div class="item item-icon-left item-icon-right">

<i class="icon ion-calendar"></i>

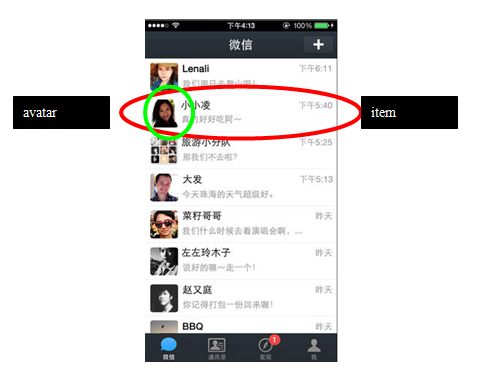
<h2>入住日期：7月27日 星期五</h2>

<i class="icon ion-ios-arrow-right"></i>

</div>

### .item : 嵌入头像

很多社交App中，一个相当固定的UI模式是包含用户头像的信息列表：



在ionic中，头像被设置为40x40固定大小。和插入图标类似，向.item 中插入头像需要满足两个条件：

1. 在.item元素上声明头像位置。头像可以位于列表的左侧或右侧， 分别使用.item-avatar-left和.item-avatar-right声明
2. 在.item元素内使用img标签插入头像。

<div class="item item-avatar">

<img src="img/a-2.jpg">

<h2>Ray</h2>

<p>真的好好吃啊...</p>

</div>

### .item : 嵌入缩略图

回到今日头条的新闻列表，我们给它加上两张新闻图片：



在ionic中，缩略图被定义为80px大小，比头像大，适合新闻图片。 和插入头像类似，向.item中插入缩略图需要满足两个条件：

1. 在.item元素上声明缩略图位置。缩略图可以位于列表的左侧或右侧， 分别使用.item-thumbnail-left和.item-thumbnail-right声明
2. 在.item元素内使用img标签插入头像。

注意：将img标签放到.item内容的开头！

<li class="item item-thumbnail-right">

<img src="img/g-1.jpg">

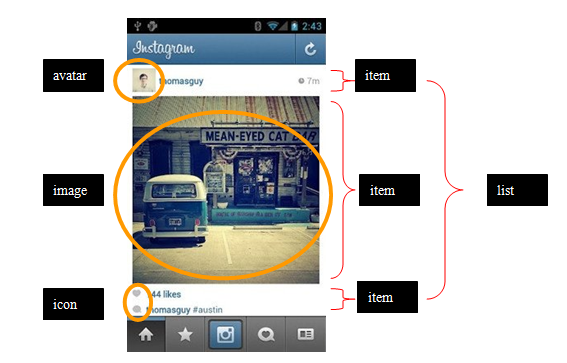
<h2>大黄鸭在北京两个月创收超两亿元 今日离京</h2>

<p>中国经营报</p>

</li>

### .item : 嵌入大图

大图非常有冲击力，在图片类App中很常见：



<div class="item item-image">

<img src="img/road.png">

</div>

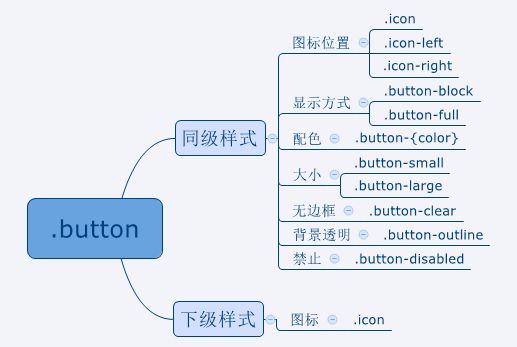
### 按钮 : .button

ionic使用.button样式定义按钮元素：

1. <any class="button">...</any>

一旦元素应用了.button样式，就可以继续选用两类预定义样式来进一步 声明元素及其内容的外观：

1. 同级样式 - 同级样式与.button应用在同一元素上，声明元素的位置、配色等。
2. 下级样式 - 下级样式只能应用在.button的子元素上，声明子元素的大小等特征。



### .button : 嵌入图标

使用内置的****Ionicons****样式，图标可以很容易地加入到.button中：

1. <a class="button">
2. Home
3. </a>

但是更简洁的办法是：直接在.button上设置样式，这样可以有效减少元素的数目：

1. <a class="button icon-left ion-home">...</a>

* ****.icon-left**** - 将图标置于按钮左侧
* ****.icon-right**** - 将图标置于按钮右侧

### 在列表中使用按钮

和插入图标类似，向.item中插入按钮需要满足两个条件：

1. 在.item元素上声明按钮位置。按钮可以位于列表的左侧或右侧， 分别使用.item-button-left和.item-button-right声明
2. 在.item元素内插入按钮。

### 输入组件容器 : .item-input

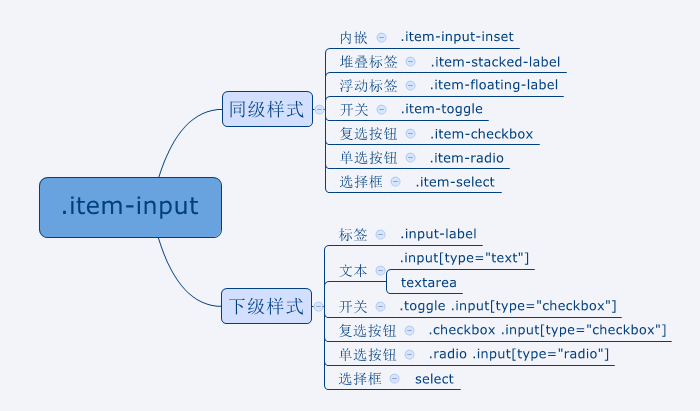
在ionic中，各种输入组件被定义成不同的HTML模板，以便将描述标签和输入元素 打包在一起。在这个模板的根元素上，需要声明****.item-input****样式：

1. <any class="item-input">...</any>

不同的输入元素有不同的模板定义。比如，文本输入通常包含一个描述标签：

1. <div class="item-input">
2. <label class="input-label">用户名</label>
3. <input type="text" placeholder="请输入你的用户账号">
4. </div>

ionic预定义了文本输入、开关、复选按钮、单选按钮、选择框、滑动条等常用 输入组件的样式：

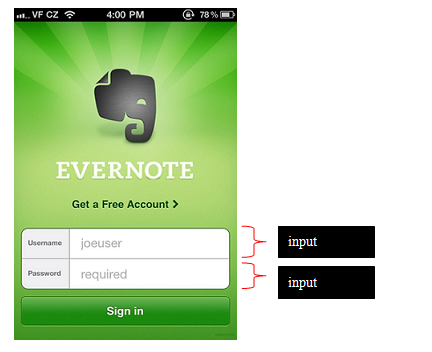


对于多个输入组件，可以将它们放到一个列表中，让表单更加整齐：

1. <any class="list">
2. <any class="item item-input">...</any>
3. <any class="item item-input">...</any>
4. ...
5. </any>

### 文本输入 : input[type="text"]

文本输入通常包含一个文本input元素和一个描述型标签：



在ionic中，使用以下HTML模板建立一个带有文本标签的输入框：

1. <any class="item-input">
2. <span class="input-label">...</span>
3. <input type="text" placeholder="...">
4. </any>

### 文本输入：使用占位符做标签

有时可以将描述标签省略，这样可以获得一种简洁的效果：



这是input元素原生的功能，设置其placeholder属性 即可实现：

1. <label class="item item-input">
2. <input type="text" placeholder="label-text">
3. </label>

### 添加小图标可以替换input-label

### 文本输入：堆叠式标签

堆叠式标签意味着将描述性标签占据单独的一行：



使用****.item-stacked-label****样式声明堆叠式标签：

1. <any class="item-input item-stacked-label">
2. <any class="input-label">...</any>
3. <input type="text" placeholder="...">
4. </any>

### 开关 : .toggle input[type="checkbox"]

开关通常用来设置两种状态 - 开启和关闭：



正如上图中所见，开关的可视部件包括两部分：滑轨（.track）和手柄（.handle）。 ionic使用如下HTML模板创建开关组件：

1. <any class="toggle">
2. <input type="checkbox">
3. <any class="track">
4. <any class="handle"></any>
5. </any>
6. </any>

开关也有配色方案样式：****.toggle-{color}****，用来改变滑轨的背景色。

<li class="item item-toggle">

iMessage

<label class="toggle toggle-balanced">

<input type="checkbox">

<div class="track">

<div class="handle"></div>

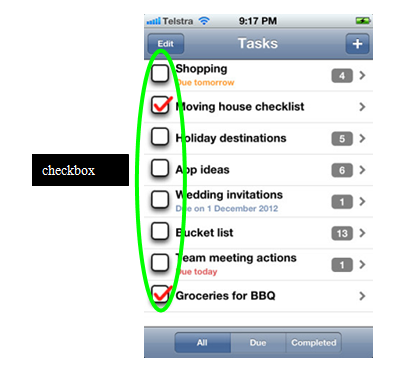
</div>

</label>

</li>

### 复选按钮

复选框通常用来在一组列表中选中部分成员：



和开关一样，复选按钮也是基于HTML的checkbox input实现的。使用如下的HTML模板声明复选按钮：

1. <any class="item-input">
2. <any class="checkbox">
3. <input type="checkbox">
4. </any>
5. </any>

你也许已经猜到，复选按钮的配色方案样式为：****.checkbox-{color}****。

<ul class="list">

<li class="item item-checkbox item-button-right">

<label class="checkbox">

<input type="checkbox">

</label>

<h2>Shopping</h2>

<p>Due tommorow</p>

<a class="button button-clear icon ion-ios-arrow-right"></a>

</li>

<li class="item item-checkbox item-button-right">

<label class="checkbox">

<input type="checkbox" checked>

</label>

<h2>Moving house checklist</h2>

<a class="button button-clear icon ion-ios-arrow-right"></a>

</li>

<li class="item item-checkbox item-button-right">

<label class="checkbox">

<input type="checkbox">

</label>

<h2>Holiday destinations</h2>

<a class="button button-clear icon ion-ios-arrow-right"></a>

</li>

<li class="item item-checkbox item-button-right">

<label class="checkbox">

<input type="checkbox">

</label>

<h2>App ideas</h2>

<a class="button button-clear icon ion-ios-arrow-right"></a>

</li>

</ul>

### 单选按钮 : .item-radio input[type="radio"]

单选按钮用来从一组选择中作出仅仅一个选择：



正如上图所见，单选按钮的可视部件包括两部分：选中图标（.radio-icon） 和描述内容（.item-content），你可以在.item-content随便添加 内容。

单选按钮基于HTML的radio input元素实现。使用如下的HTML模板 声明单选按钮：

1. <any class="item-radio">
2. <input type="radio" name="{group-name}">
3. <any class="item-content">...</any>
4. <any class="radio-icon ion-checkmark"></any>
5. </any>

单选按钮通常不单独使用，将他们放入一个列表中：

1. <any class="list">
2. <any class="item item-radio">...</any>
3. <any class="item item-radio">...</any>
4. ...
5. </any>

需要注意的是，group-name决定了单选按钮的分组，所以对于互斥的选择项，必须 将它们的group-name设置为相同的名称。

### 滑动条 : .range input[type="range"]

range是HTML5新引入的元素，常用来进行连续值的调节：



从上图可以看到，滑动条的可视部件包括三部分：左右图标和中间的滑动条。 图标是可选的，当不用图标时，滑动条将占据整个宽度。

在ionic中，使用如下HTML模板声明滑动条：

1. <any class="range">
2. <any class="icon {left-icon-name}"></any>
3. <input type="range" name="{range-name}">
4. <any class="icon {right-icon-name}"></any>
5. </any>

使用****.range-{color}****样式声明滑动条的配色方案，这主要影响滑轨的颜色。

### 选择框 : .item-select select

在每个平台上，选择框的表现形式都不一样，例如，在PC上，是一个传统的下拉框，在Android 是一个单选弹出窗，而iOS上是一个覆盖半个窗体的定制滚动器：



通常总是将选择框与文字一起放入列表条目中，使用如下HTML模板创建选择框条目：

1. <label class="item-input item-select">
2. <any class="input-label"></any>
3. <select>
4. <option>...</option>
5. <option>...</option>
6. ...
7. </select>
8. </label>

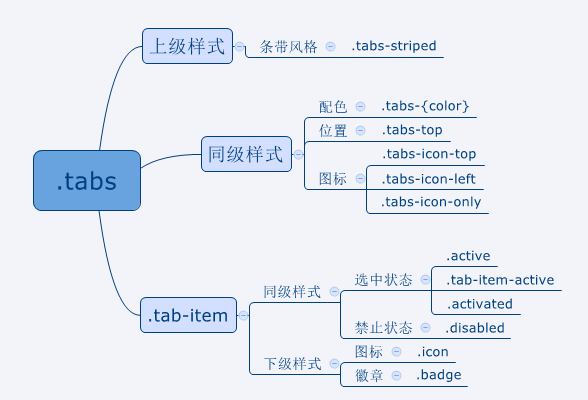
### 选项卡 : .tabs

选项卡是一个可以包含多个按钮或链接的容器，通常用于提供一致的导航体验。 ionic中使用****.tabs****样式声明选项卡，使用****.tab-item****样式声明选项卡 成员

1. <any class="tabs">
2. <any class="tab-item">...</any>
3. <any class="tab-item">...</any>
4. ...
5. </any>

选项卡默认地位于屏幕底部。一旦元素应用了.tabs样式，就可以继续 选用三类预定义样式来进一步声明元素及其内容的外观：

1. 同级样式 - 同级样式与.tabs应用在同一元素上，声明选项卡的位置、配色等。
2. 下级样式 - 下级样式只能应用在.tabs的子元素上，声明子元素的大小等特征。
3. 上级样式 - 上级样式应用于.tabs的父元素中，声明选项卡的平台特征。



### .tab-item : 使用图标

选项卡的选项默认的只包含文字，使用图标会使其更加生动：比如 微博（weibo）和微信（wechat）：



要在某个选项元素中插入图标，需要注意两点：

1. 使用i标签在****.tab-item****中插入图标
2. 在.tabs容器上使用****.tabs-icon-{position}****声明图标位置。

有三种方式定义图标位置：

* ****.tabs-icon-top**** - 将图标置于文字之上
* ****.tabs-icon-left**** - 将图标置于文字左侧
* ****.tabs-icon-only**** - 只使用图标，不显示文字

<div class="tabs tabs-stable tabs-icon-top">

<a class="tab-item">

<i class="icon ion-ios-home-outline"></i>首页

</a>

<a class="tab-item">

<i class="icon ion-ios-email-outline"></i>信息

</a>

<a class="tab-item energized-bg light">

<i class="icon ion-ios-plus-empty"></i>发微博

</a>

<a class="tab-item">

<i class="icon ion-ios-eye-outline"></i>发现

</a>

<a class="tab-item">

<i class="icon ion-ios-person-outline"></i>我

</a>

</div>

### .tab-item : 使用徽章

微信（wechat）的选项卡中第三项包含了一个红色的信息元素，被称为徽章：



要在****.tab-item****内包含徽章，需要注意两点：

1. 使用****.badge****样式插入徽章元素
2. 在****.tab-item****同级声明****.has-badge****样式
   1. <any class="tabs">
   2. <any class="tab-item has-badge">
   3. <any class="badge">...</any>
   4. </any>
   5. ...
   6. </any>

<div class="tabs tabs-dark tabs-icon-top">

<a class="tab-item active">

<i class="icon ion-ios-chatbubble-outline"></i>微信

</a>

<a class="tab-item">

<i class="icon ion-ios-paper-outline"></i>通信录

</a>

<a class="tab-item has-badge">

<i class="icon ion-ios-eye-outline"></i>发现

<i class="badge">12</i>

</a>

<a class="tab-item">

<i class="icon ion-ios-person-outline"></i>我

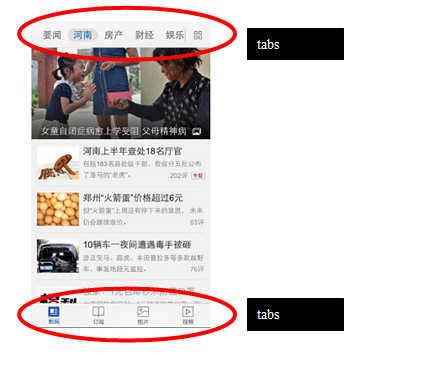
</a>

</div>

### .tabs : 顶部选项卡

默认情况下，选项卡位于屏幕底部，可以使用****.tabs-top****样式将选项卡置于顶部。

顶部选项卡应用也很广泛：



<div class="tabs tabs-top tabs-stable">

<a class="tab-item">要闻</a>

<a class="tab-item active">河南</a>

<a class="tab-item">房产</a>

<a class="tab-item">财经</a>

<a class="tab-item">娱乐</a>

</div>

<div class="content stable-bg">

</div>

<div class="tabs tabs-stable tabs-icon-top">

<a class="tab-item active">

<i class="icon ion-ios-paper-outline"></i>新闻

</a>

<a class="tab-item">

<i class="icon ion-ios-book-outline"></i>订阅

</a>

<a class="tab-item">

<i class="icon ion-images"></i>图片

</a>

<a class="tab-item">

<i class="icon ion-ios-videocam-outline"></i>视频

</a>

</div>

### .tab-striped .tabs: 条带风格选项卡

条带风格的选项卡起源于Android平台，它使用一个细长的条带表示选项的选中状态：



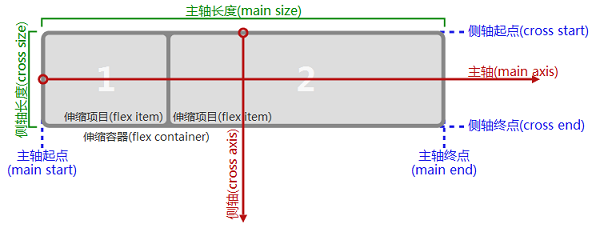
在ionic中，使用****.tabs-striped****样式声明条带风格选项卡：

1. <any class="tabs-striped">
2. <any class="tabs">
3. <any class="tab-item">...</any>
4. <any class="tab-item">...</any>
5. ...
6. </any>
7. </any>

### 栅格系统

ionic的栅格系统采用了CSS3的弹性盒（Flexible Box）模型，这使得它与大多数 栅格系统都有所区别。

Flex布局主要思想是让容器有能力改变其子元素的宽度、高度甚至先后顺序， 从而以最佳方式填充可用空间。CSS3引入Flex容器的主要目的是为了适应所 有类型的显示设备和屏幕大小：



在ionic中使用栅格系统主要使用两个类：

* ****.row**** - 在容器元素上使用.row类，表示将其设置为弹性容器，即Flexible Box。
* ****.col**** - 在子元素上使用.col类，其扩展系数和收缩系数都被设置为1。这意味着 所有的子元素将平分容器的宽度。

<div class="bar bar-header">

<h1 class="title">等宽列</h1>

</div>

<div class="scroll-content has-header">

<div class="row">

<div class="col">.col</div>

<div class="col">.col</div>

<div class="col">.col</div>

<div class="col">.col</div>

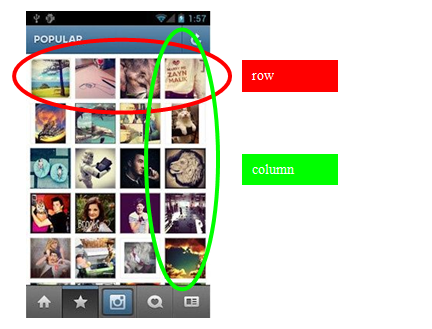
<div class="col">.col</div>

</div>

</div>

### .col : 默认的定宽列

在ionic的栅格中，每一行的各列默认是等宽的，这使得实现一个图片 浏览界面非常简单。下图是instagram app的截图：



<div class="row">

<div class="col"><img src="img/g-1.jpg"></div>

<div class="col"><img src="img/g-2.jpg"></div>

<div class="col"><img src="img/g-3.jpg"></div>

<div class="col"><img src="img/g-4.jpg"></div>

</div>

### .col-{width-percent} : 指定列宽

当然，我们也可以显式地指定某些列的宽度。ionic提供了一些预置的CSS类， 供我们快速指定列宽：

* ****.col-10**** - 占据容器10%宽度
* ****.col-20**** - 占据容器20%宽度
* ****.col-25**** - 占据容器25%宽度
* ****.col-33**** - 占据容器33%宽度
* ****.col-50**** - 占据容器50%宽度
* ****.col-67**** - 占据容器67%宽度
* ****.col-75**** - 占据容器75%宽度
* ****.col-80**** - 占据容器80%宽度
* ****.col-90**** - 占据容器90%宽度

如果你有特殊的需求，比如，非要指定78%的宽度，可以这样：

1. .col-78{
2. -webkit-box-flex: 0;
3. -webkit-flex: 0 0 78%;
4. -moz-box-flex: 0;
5. -moz-flex: 0 0 %78;
6. -ms-flex: 0 0 78%;
7. flex: 0 0 78%;
8. }

现在清楚了，这个比例值对应的就是元素的flex-basis样式。

### col-{width-percent} : metro风格

windows phone的metro风格在信息展示方面有很大的优势，虽然携程的APP 没有突出这一点:-( ：



通过定制栅格的列宽，我们可以简单地实现类似的界面。

<div class="scroll-content">

<div class="row">

<div class="col col-50 positive-bg light">

<div>机票</div>

<i class="icon ion-plane"></i>

</div>

<div class="col col-25 assertive-bg light">

<div>火车票</div>

<i class="icon ion-android-train"></i>

</div>

<div class="col col-25 calm-bg light">

<div>用车</div>

<i class="icon ion-model-s"></i>

</div>

</div>

<div class="row">

<div class="col calm-bg light">

<div>酒店</div>

<i class="icon ion-home"></i>

</div>

<div class="col stable-bg energized" >

<i class="icon ion-person"></i>

<div>我的携程</div>

</div>

</div>

<div class="row">

<div class="col balanced-bg light">

<div>旅游</div>

<i class="icon ion-image"></i>

</div>

<div class="col energized-bg light">

<div>景点门票</div>

<i class="icon ion-pricetag"></i>

</div>

<div class="col royal-bg light">

<div>攻略社区</div>

<i class="icon ion-person-stalker"></i>

</div>

</div>

</div>

### .col-offset-{width-percent} : 指定列偏移

列可以从默认位置偏移，ionic预置了一些CSS类供我们快速设置列偏移：

* ****.col-offset-10**** - 偏移默认位置10%容器宽度
* ****.col-offset-20**** - 偏移默认位置20%容器宽度
* ****.col-offset-25**** - 偏移默认位置25%容器宽度
* ****.col-offset-33**** - 偏移默认位置33%容器宽度
* ****.col-offset-50**** - 偏移默认位置50%容器宽度
* ****.col-offset-67**** - 偏移默认位置67%容器宽度
* ****.col-offset-75**** - 偏移默认位置75%容器宽度
* ****.col-offset-80**** - 偏移默认位置80%容器宽度
* ****.col-offset-90**** - 偏移默认位置90%容器宽度

这实际上是通过设置子元素的margin-left样式实现的，所以如果你需要让它 偏移正常位置78%，可以这样：

1. .col-offset-78{
2. margin-left:78%;
3. }

### .col-{align} : 列纵向对齐

如果一行中各元素的高度不一样，那么比较矮的那些元素将自动被拉伸以适应 整行的高度。

有时你不想这样，希望那些元素保持自身的高度。ionic提供了一些预置的CSS类用来 指定这些元素纵向的对齐方式：

* ****.col-top**** - 让元素纵向顶对齐
* ****.col-center**** - 让元素居中对齐
* ****.col-bottom**** - 让元素向底对齐

这是通过设置元素的CSS3样式align-self来实现的。

<div class="bar bar-header">

<h1 class="title">列纵向对齐</h1>

</div>

<div class="scroll-content has-header">

<div class="row">

<div class="col">.col</div>

<div class="col">.col</div>

<div class="col">.col</div>

<div class="col">1<br>2<br>3<br>4</div>

</div>

<div class="row">

<div class="col col-top">.col</div>

<div class="col col-center">.col</div>

<div class="col col-bottom">.col</div>

<div class="col">1<br>2<br>3<br>4</div>

</div>

<div class="row row-top">

<div class="col">.col</div>

<div class="col">.col</div>

<div class="col">.col</div>

<div class="col">1<br>2<br>3<br>4</div>

</div>

<div class="row row-center">

<div class="col">.col</div>

<div class="col">.col</div>

<div class="col">.col</div>

<div class="col">1<br>2<br>3<br>4</div>

</div>

<div class="row row-bottom">

<div class="col">.col</div>

<div class="col">.col</div>

<div class="col">.col</div>

<div class="col">1<br>2<br>3<br>4</div>

</div>

</div>