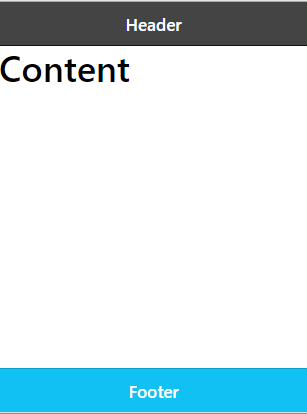
**第十三讲 ionic css布局介绍**

学习要点：

1. 基本布局
2. 色彩、图标和边距
3. 界面组件列表
4. 界面组件按钮
5. 界面组件 表单输入
6. 界面组件选项卡
7. 定制布局 Grid布局
8. **基本布局**

**1.布局模式**

基本布局：标题/header、内容/content和页脚/footer。



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类声明对应区域:

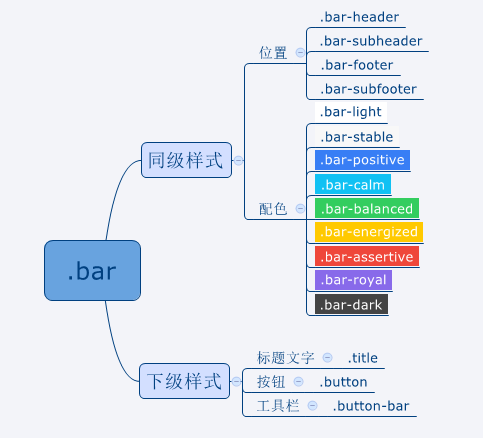
* **.bar.bar-header** - 声明元素为标题区
* **.bar.bar-footer** - 声明元素为底部
* **.content**  **. scroll-content**- 声明元素为内容区

**2. .bar位置以及颜详解**

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）：

**使用方式：**

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



**3. .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

下面例子，使用了三个条块：标题、副标题、页脚：



**4. .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：

|  |
| --- |
| <any class="bar">  <any class="title">...</any>  </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：

|  |
| --- |
| <any class="bar">  <any class="button">...</any>  </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：

|  |
| --- |
| <any class="bar">  <any class="button-bar">...</any>  </any> |

**5. bar : 嵌入input**

一种常见的UI模式是在标题栏中嵌入搜索栏，比如美团：

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

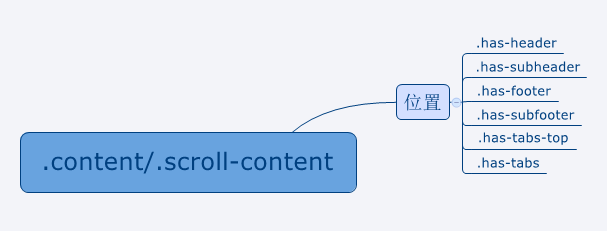
这是因为，在ionic的实现中，.bar中的.input 样式定义如下：

|  |
| --- |
| <div class="bar bar-header item-input-inset">  <label class="item-input-wrapper">  <a class="button button-clear icon-right ion-android-arrow-dropdown">北京</a>  <input type="text" placeholder="输入您要搜索的内容">  </label>  </div>  .bar .item-input-inset{  .item-input-wrapper{  .input{  ...  }  }  } |

**6. 内容：.content和.scroll-content**

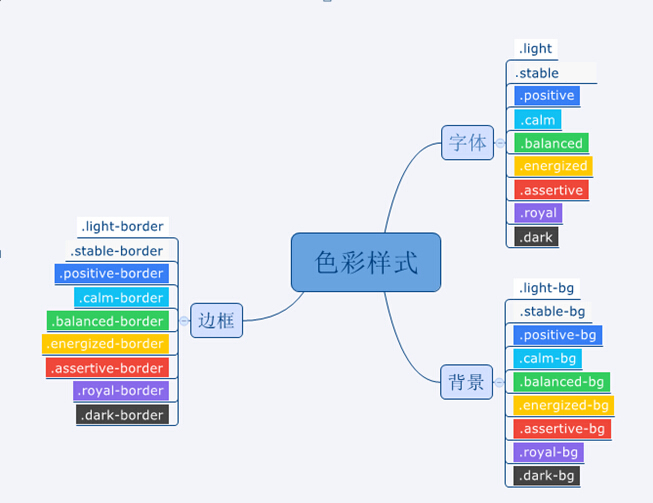
ionic预定义了两个内容容器样式：

* **.content** – 相对定位
* **.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：



1. **Ionic 色彩、图标和边距**

**1.ionic定义了九种前景/背景/边框的色彩样式**



**2.ionic 中的图标**

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：

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

要了解有哪些图标及具体名称，需要访问[ionics.com](http://ionicons.com/" \t "_blank)。 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：

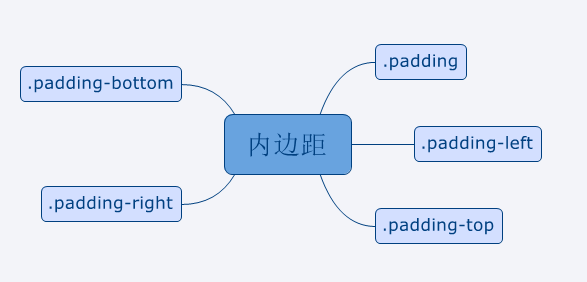
<any style="font-size:100px;">

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

</any>

**3.Ionic 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。

1. **ionic界面组件列表**
2. **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：

<any class="list">

<any class="item">...</any>

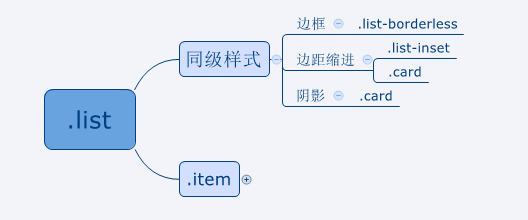
<any class="item">...</any>

<any class="item">...</any>

<any class="item">...</any>

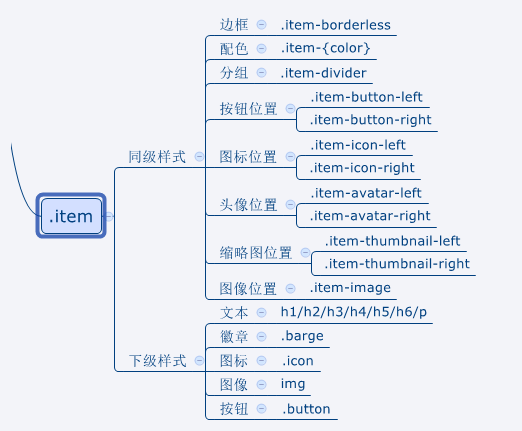
</any>

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：



1. **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：



1. **.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。

1. **.item : 嵌入图标**

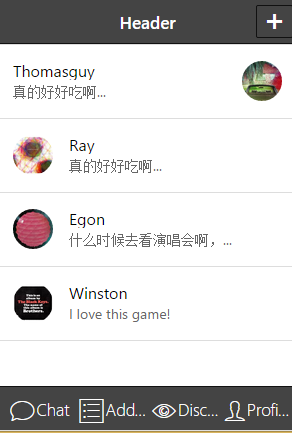
列表成员的内容中插入图标，比单纯的文字更加生动：



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 list-inset。
3. **.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。
3. **.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：

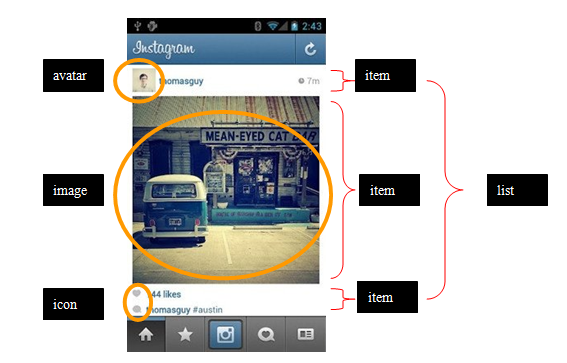
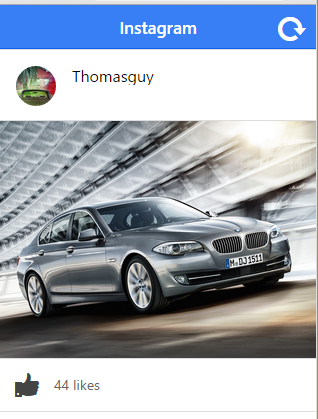


在ionic中，缩略图被定义为80px大小，比头像大，适合新闻图片。 和插入头像类似，向*.item*中插入缩略图需要满足两个条件：

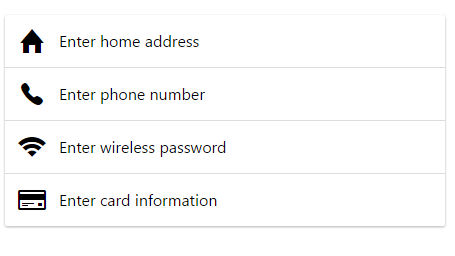
1. 在*.item*元素上声明缩略图位置。缩略图可以位于列表的左侧或右侧， 分别使用*.item-thumbnail-left*和*.item-thumbnail-right*声明
2. 在*.item*元素内使用*img*标签插入头像。

注意：将*img*标签放到*.item*内容的开头！

1. **.item : 嵌入大图 item-image**

1. **.item card**

****

要插入图标，需要满足两个条件：

1. *.* *card 可以让list和左右有一些边距*
2. 在*.* *item-divider*元素定义list的头和尾

<div class="list card">

</div>

|  |
| --- |
| <div class="card">  <div class="item item-divider">  I'm a Header in a Card!  </div>  <div class="item item-text-wrap">  This is a basic Card with some text.  </div>  <div class="item item-divider">  I'm a Footer in a Card!  </div>  </div> |

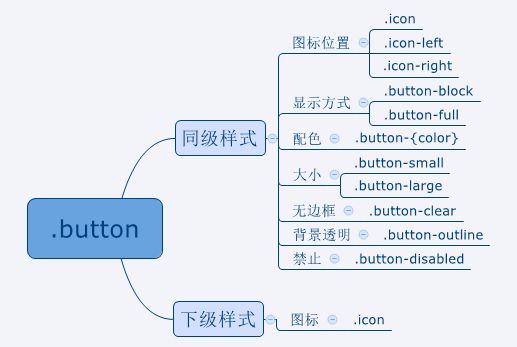
1. **ionic界面组件按钮**
2. **按钮 : .button**

ionic使用*.button*样式定义按钮元素：

<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。



**2. .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：

<a class="button">

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

</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：

<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

**3. 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。

**4. button-block**

button显示成块元素

<button class="button button-calm button-block">button-light</button>

1. **ionic界面组件 表单输入**

**1.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：

<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：

<div class="item-input">

<label class="input-label">用户名</label>

<input placeholder="请输入你的用户账号" type="text">

</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：

**2. 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：



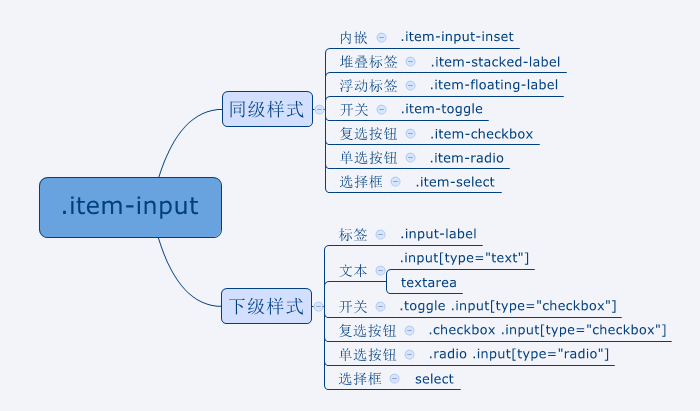
在ionic中，使用以下HTML模板建立一个带有文本标签的输入框：

<any class="item-input">

<span class="input-label">...</span>

<input placeholder="..." type="text">

</any>



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：

<any class="list">

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

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

...

</any>

**3. 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：

<label class="item item-input">

<i class="icon ion-ios-unlocked-outline positive"></i>

<input type="password" placeholder="请输入您的密码">

</label>

**4. 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：

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

<span class="input-label">Email</span>

<input type="text" placeholder="me@vucek.com">

</label>

**5.浮动标签 .** **item-floating-label**

****

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

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

<span class="input-label">Email</span>

<input type="text" placeholder="me@vucek.com">

</label>

**6.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：

<any class="toggle">

<input type="checkbox">

<any class="track">

<any class="handle"></any>

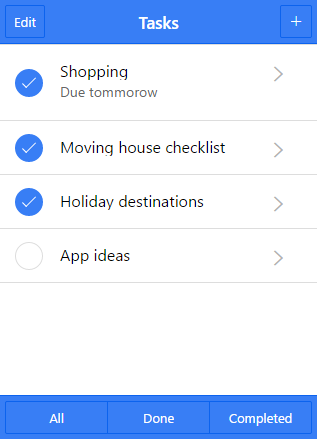
</any>

</any>

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

**7.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：

<any class="item-input">

<any class="checkbox">

<input type="checkbox">

</any>

</any>

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

**8.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：

<any class="item-radio">

<input name="{group-name}" type="radio">

<any class="item-content">...</any>

<any class="radio-icon ion-checkmark"></any>

</any>

单选按钮通常不单独使用，将他们放入一个列表中：

<any class="list">

<any class="item item-radio">...</any>

<any class="item item-radio">...</any>

...

</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。

**9.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：

<any class="range">

<any class="icon {left-icon-name}"></any>

<input name="{range-name}" type="range">

<any class="icon {right-icon-name}"></any>

</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。

**10.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：

<label class="item-input **item-select**">

<any class="input-label"></any>

<select>

<option>...</option>

<option>...</option>

...

</select>

</label>

1. **界面组件选项卡**
2. **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

<any class="tabs">

<any class="tab-item">...</any>

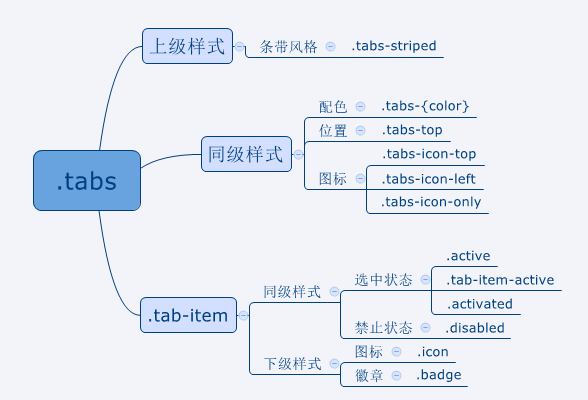
<any class="tab-item">...</any>

...

</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。



**tabs-icon-top 图标显示在上面**

**tabs-assertive 改变背景条的颜色**

***tabs-color-assertive***  *用tabs 控制他子元素 图标和文字的颜色*

***tabs-item-hide*** *隐藏tab*

**2.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

1. **.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：

使用.badge样式插入徽章元素

在.tab-item同级声明.has-badge样式

<any class="tabs">

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

<any class="badge">...</any>

</any>

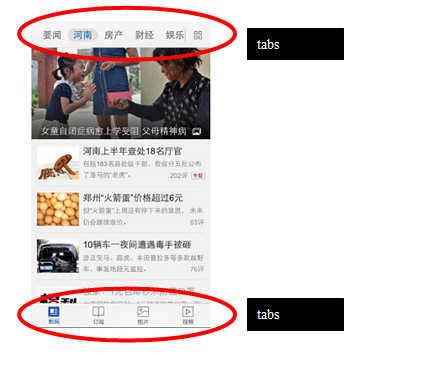
...

</any>

**4. .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：



**5. .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：

<any class="tabs-striped">

<any class="tabs">

<any class="tab-item">...</any>

<any class="tab-item">...</any>

...

</any>

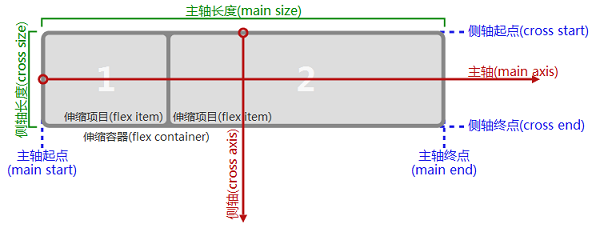
</any>

1. **定制布局 Grid布局**

### 1. Grid布局

ionic的Grid布局采用了CSS3的弹性盒（Flexible Box）模型，这使得它与大多数 Grid布局都有所区别。

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：

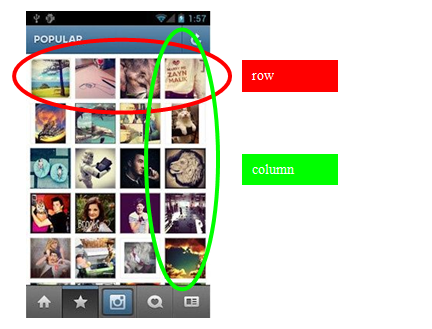


在ionic中使用Grid布局主要使用两个类：

* **.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。

### .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：



### .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：

.col-78{

-webkit-box-flex: 0;

-webkit-flex: 0 0 78%;

-moz-box-flex: 0;

-moz-flex: 0 0 %78;

-ms-flex: 0 0 78%;

flex: 0 0 78%;

}

### .col-{width-percent} : 指定列宽

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。

### .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：

.col-offset-78{

margin-left:78%;

}

### .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。