

## ionic 教程



ionic 是一个强大的 **HTML5** 应用程序开发框架(**HTML5 Hybrid Mobile App Framework** )。 可以帮助您使用 **Web** 技术, 比如 **HTML**、**CSS** 和 **Javascript** 构建接近原生体验的移动应用程序。

ionic 主要关注外观和体验, 以及和你的应用程序的 **UI** 交互, 特别适合用于基于 **Hybird** 模式的 **HTML5** 移动应用程序开发。

ionic是一个轻量的手机UI库, 具有速度快, 界面现代化、美观等特点。为了解决其他一些UI库在手机上运行缓慢的问题, 它直接放弃了IOS6和Android4.1以下的版本支持, 来获取更好的使用体验。

## ionic 特点

- 1.ionic 基于**Angular**语法, 简单易学。
- 2.ionic 是一个轻量级框架。
- 3.ionic 完美的融合下一代移动框架, 支持 **Angularjs** 的特性, **MVC**, 代码易维护。
- 4.ionic 提供了漂亮的设计, 通过 **SASS** 构建应用程序, 它提供了很多 **UI** 组件来帮助开发者开发强大的应用。
- 5.ionic 专注原生, 让你看不出混合应用和原生的区别
- 6.ionic 提供了强大的命令行工具。
- 7.ionic 性能优越, 运行速度快。

## 学习本教程前你需要了解?

学习本教程前你需要了解以下基础知识:

[HTML](#)

[CSS](#)

[Javascript](#)

[Angular](#)

## ionic 相关内容

ionic 官方网站: <http://ionicframework.com/>

ionic 官方文档: <http://ionicframework.com/docs/>

Github 地址: <https://github.com/driftyco/ionic>

□ 点我分享笔记

反馈/建议

## ionic 安装

本站实例采用了ionic v1.3.2 版本，使用的 CDN 库地址：

```
<link href="https://cdn.bootcss.com/ionic/1.3.2/css/ionic.css" rel="stylesheet">

<script src="https://cdn.bootcss.com/ionic/1.3.2/js/ionic.bundle.min.js"></script>
```

ionic 最新版本下载地址：<http://ionicframework.com/docs/overview/#download>。

下载后解压压缩包，包含以下目录：

css/	=>	样式文件
fonts/	=>	字体文件
js/	=>	Javascript文件
version.json	=>	版本更新说明

你也可以在 [Github](https://github.com/driftyco/ionic) 上下载以下资源文件：<https://github.com/driftyco/ionic>（在release 目录中）。

接下来，我们只需要在项目中引入以上目录中的 `css/ionic.min.css` 和 `js/ionic.bundle.min.js` 文件即可创建 ionic 应用。

### 实例

```
<ion-header-bar class="bar-positive">
<h1 class="title">Hello World!</h1>
</ion-header-bar>
<ion-content>
<p>我的第一个 ionic 应用。</p>
</ion-content>
```

尝试一下 »

点击 "尝试一下" 按钮查看在线实例。

本教程着重讲解 ionic 框架的应用，大部分实例在浏览器中运行，移动设备上运行可以在接下来的命令行安装教程中详细了解。

注意：在移动应用如 `phonegap` 中我们只需将对应的 `js` 和 `css` 文件加入到资源库中即可。

### 命令行安装

首先您需要安装 [Node.js](#)，我们在接下来的安装中需要使用到其 `NPM` 工具，更多 `NPM` 介绍可以查看我们的[NPM 使用介绍](#)。

然后通过[命令行工具](#)安装最新版本的 `cordova` 和 `ionic`。通过参考[Android](#) 和 [iOS](#) 官方文档来安装。

Window 和 Linux 上打开命令行工具执行以下命令：

```
$ npm install -g cordova ionic
```

Mac 系统上使用以下命令：

```
sudo npm install -g cordova ionic
```

提示: *IOS*需要在*Mac Os X*. 和*Xcode*环境下面安装使用。

如果你已经安装了以上环境，可以执行以下命令来更新版本：

```
npm update -g cordova ionic
```

或

```
sudo npm update -g cordova ionic
```

## 创建应用

使用**ionic**官方提供的现成的应用程序模板，或一个空白的项目创建一个**ionic**应用：

```
$ ionic start myApp tabs
```

## 运行我们刚才创建的**ionic**项目

使用 **ionic tool** 创建，测试，运行你的**apps**(或者通过**Cordova**直接创建)。

## 创建**Android**应用

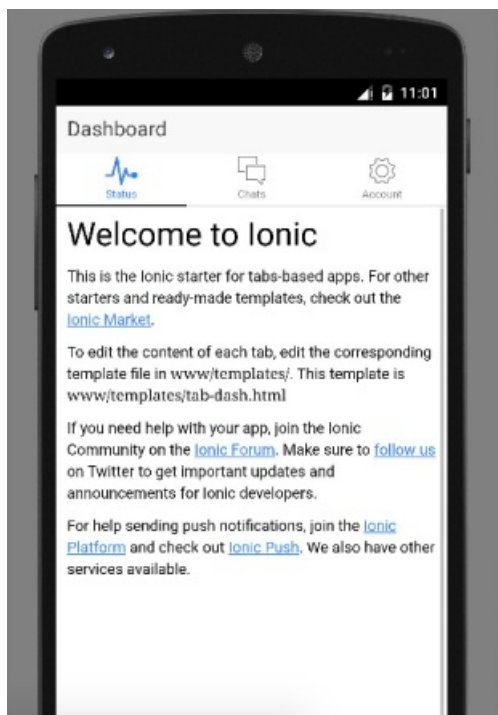
```
$ cd myApp
```

```
$ ionic cordova platform add android
```

```
$ ionic cordova build android
```

```
$ ionic cordova emulate android
```

如果一切正常会弹出模拟器，界面如下所示：



## 创建iOS应用

```
$ cd myApp

$ ionic cordova platform add ios

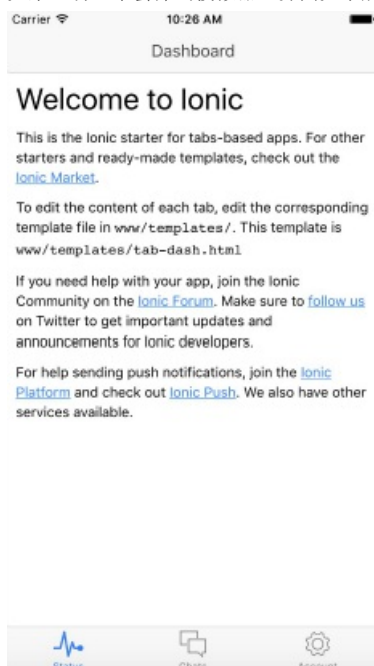
$ ionic cordova build ios

$ ionic cordova emulate ios
```

如果出现"*ios-sim was not found.*"错误，可以执行以下命令：

```
npm install -g ios-sim
```

如果一切正常会弹出模拟器，界面如下所示：



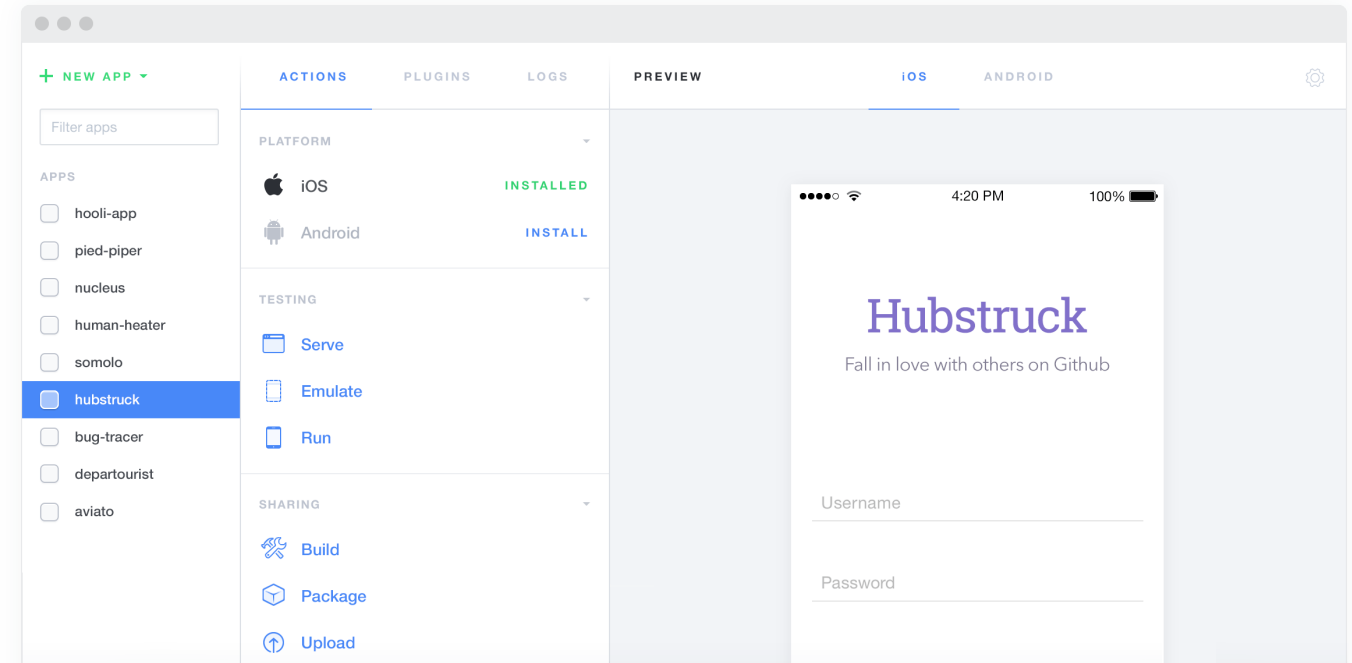
# Ionic Lab

*Ionic Lab 目前已停止更新。*

Ionic Lab 是桌面版的开发环境，如果你不喜欢使用命令行操作，Ionic Lab 将会满足你的需求。

Ionic Lab 为开发者提供了一个更简单的方法来开始，编译，运行，和模拟运行Ionic的应用程序。

Ionic Lab 支持的平台有：Window、Mac OS X、Linux，下载地址为：<http://lab.ionic.io/>，下载后直接安装即可。整个操作界面如下所示：



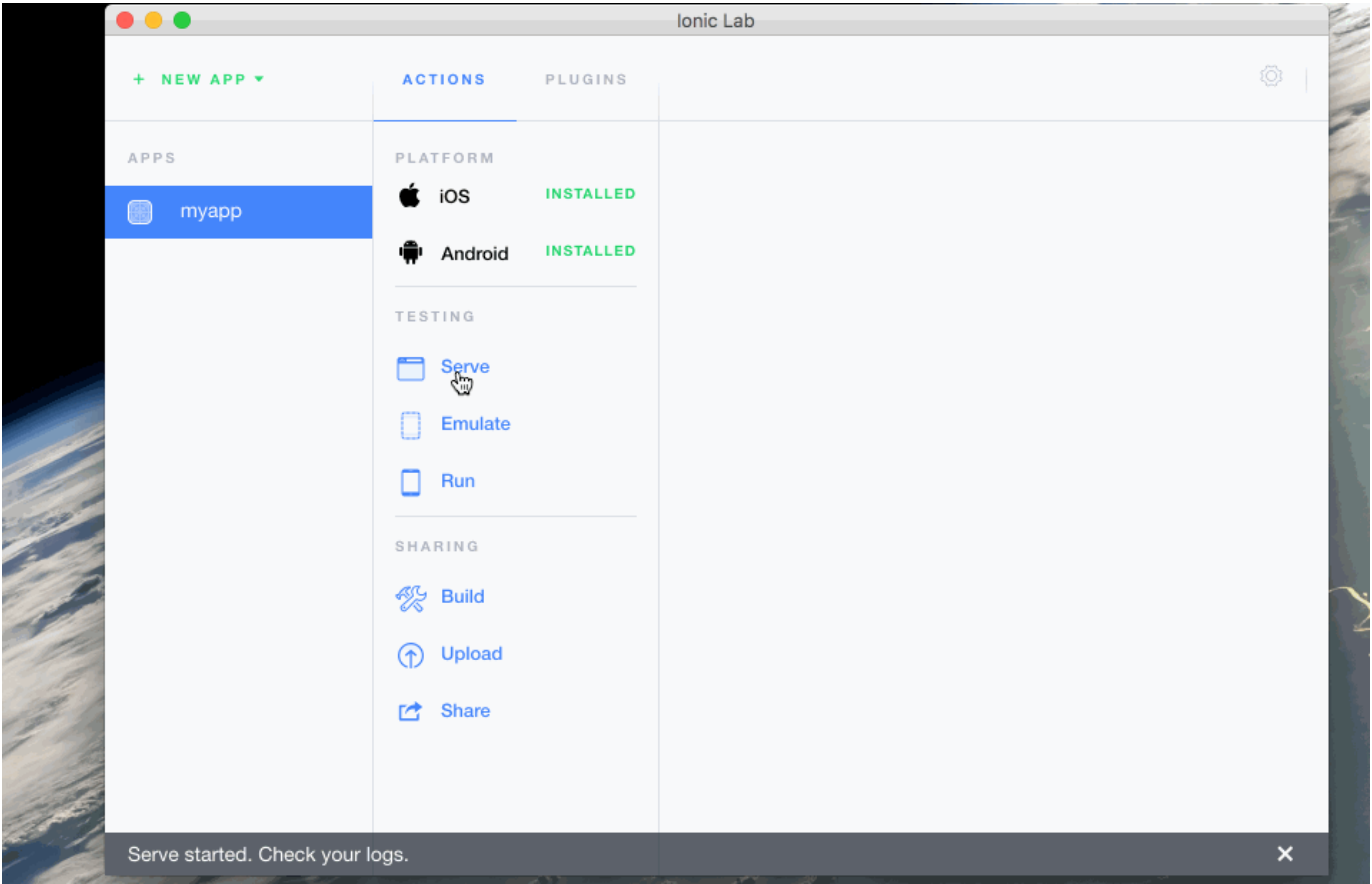
通过以上界面你可以完成以下操作：

- 创建应用
- 预览应用
- 编译应用
- 运行应用
- 上传应用
- 运行日志查看

.....

推荐使用Sublime Text作为 Ionic 项目的编辑器，我们可以通过 Ionic Lab 直接在Sublime Text 上打开项目，如下图：

## Gif 操作演示



[ionic 教程](#)

[ionic 创建 APP](#)

[点我分享笔记](#)

[反馈/建议](#)

Copyright © 2013-2018 菜鸟教程 [runoob.com](#) All Rights Reserved. 备案号: 闽ICP备15012807号-1



[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

[ionic 安装](#)

[ionic 头部与底部](#)

# ionic 创建 APP

前面的章节中我们已经学会了 **ionic** 框架如何导入到项目中。

接下来我们将为大家介绍如何创建一个 **ionic APP** 应用。

**ionic 创建 APP** 使用 **HTML**、**CSS** 和 **Javascript** 来构建，所以我们可以创建一个 **www** 目录，并在目录下创建 **index.html** 文件，代码如下：

```
<!DOCTYPE html>

<html>

  <head>
```

```
<meta charset="utf-8">

<title>Todo</title>

<meta name="viewport" content="initial-scale=1, maximum-scale=1, user-scalable=no, width=device-width">


<link href="lib/ionic/css/ionic.css" rel="stylesheet">


<script src="lib/ionic/js/ionic.bundle.js"></script>


<!-- 在使用 Cordova/PhoneGap 创建的 APP 中包含的文件，由 Cordova/PhoneGap 提供，（开发过程中显示 404） -->

<script src="cordova.js"></script>

</head>

<body>

</body>

</html>
```

以上代码中，我们引入了 Ionic CSS 文件、Ionic JS 文件及 Ionic AngularJS 扩展ionic.bundle.js（ionic.bundle.js）。

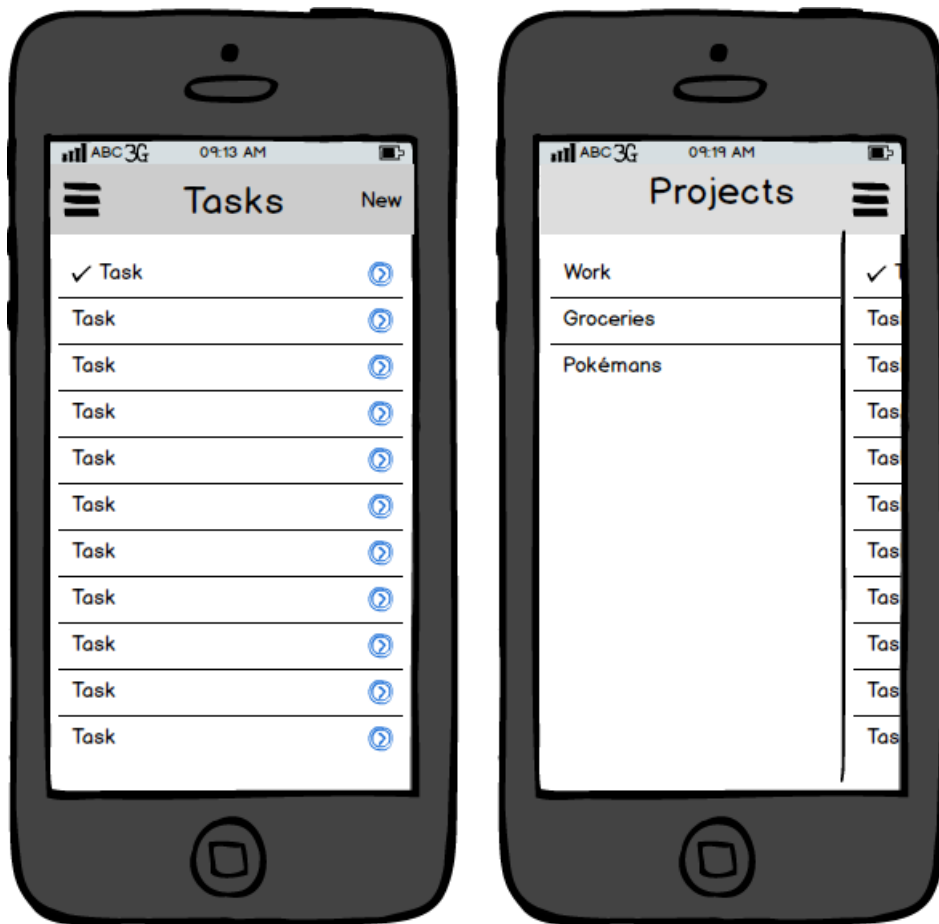
ionic.bundle.js 文件已经包含了 Ionic核心JS、AngularJS、Ionic的AngularJS扩展，如果你需要引入其他 Angular 模块，可以从 lib/js/angular 目录中调用。

cordova.js 是在使用 Cordova/PhoneGap 创建应用时生成的，不需要手动引入，你可以在 Cordova/PhoneGap 项目中找到该文件，所以在开发过程中显示 404 是正常的。

---

## 创建 APP

接下来我们来实现一个包含标题、侧边栏、列表等的应用,设计图如下：



## 创建侧边栏

侧边栏创建使用 `ion-side-menus` 控制器。

编辑我们先前创建的 `index.html` 文件，修改 `<body>` 内的内容，如下所示：

```
<body>

  <ion-side-menus>

    <ion-side-menu-content>

    </ion-side-menu-content>

    <ion-side-menu side="left">

    </ion-side-menu>

  </ion-side-menus>

</body>
```

控制器解析：

**ion-side-menus:** 是一个带有边栏菜单的容器，可以通过点击按钮或者滑动屏幕来展开菜单。

**ion-side-menu-content:** 展示主要内容的容器，可以通过滑动屏幕来展开menu。

**ion-side-menu:** 存放侧边栏的容器。

## 初始化 APP

接下来我们创建一个新的 `AngularJS` 模块，并初始化，代码位于 `www/js/app.js` 中：



```
angular.module('todo', ['ionic'])
```

之后在我们的 **body** 标签中添加 **ng-app** 属性:

```
<body ng-app="todo">
```

在 **index.html** 文件的 `<script src="cordova.js"></script>` 上面引入 **app.js** 文件:

```
<script src="js/app.js"></script>
```

修改 **index.html** 文件 **body** 标签的内容, 代码如下所示:

```
<body ng-app="todo">

  <ion-side-menus>

    <!-- 中心内容 -->

    <ion-side-menu-content>

      <ion-header-bar class="bar-dark">

        <h1 class="title">Todo</h1>

      </ion-header-bar>

      <ion-content>

      </ion-content>

    </ion-side-menu-content>

    <!-- 左侧菜单 -->

    <ion-side-menu side="left">

      <ion-header-bar class="bar-dark">

        <h1 class="title">Projects</h1>

      </ion-header-bar>

    </ion-side-menu>

  </ion-side-menus>

</body>
```

以上步骤我们已经完成了 ionic 基本 APP 的应用。

## 创建列表

首先创建一个控制器 **TodoCtrl**:

```
<body ng-app="todo" ng-controller="TodoCtrl">
```

在 **app.js** 文件中, 使用控制器定义列表数据:

```
angular.module('todo', ['ionic'])

.controller('TodoCtrl', function($scope) {

    $scope.tasks = [

        { title: '菜鸟教程' },

        { title: 'www.runoob.com' },

        { title: '菜鸟教程' },

        { title: 'www.runoob.com' }

    ];

});
```

在 **index.html** 页面中数据列表我们使用 **Angular ng-repeat** 来迭代数据:

```
<!-- 中心内容 -->

<ion-side-menu-content>

    <ion-header-bar class="bar-dark">

        <h1 class="title">Todo</h1>

    </ion-header-bar>

    <ion-content>

        <!-- 列表 -->

        <ion-list>

            <ion-item ng-repeat="task in tasks">

                {{task.title}}

            </ion-item>

        </ion-list>

    </ion-content>
```

```
</ion-side-menu-content>
```

修改后 index.html body 标签内代码如下：

```
<body ng-app="todo" ng-controller="TodoCtrl">

  <ion-side-menus>

    <!-- 中心内容 -->

    <ion-side-menu-content>

      <ion-header-bar class="bar-dark">

        <h1 class="title">Todo</h1>

      </ion-header-bar>

      <ion-content>

        <!-- 列表 -->

        <ion-list>

          <ion-item ng-repeat="task in tasks">

            {{task.title}}

          </ion-item>

        </ion-list>

      </ion-content>

    </ion-side-menu-content>

    <!-- 左侧菜单 -->

    <ion-side-menu side="left">

      <ion-header-bar class="bar-dark">

        <h1 class="title">Projects</h1>

      </ion-header-bar>

    </ion-side-menu>

  </ion-side-menus>

  <script src="http://www.runoob.com/static/ionic/js/app.js"></script>

  <script src="cordova.js"></script>

</body>
```

## 创建添加页面

修改 `index.html` 在 `</ion-side-menus>` 后添加如下代码:

```
<script id="new-task.html" type="text/ng-template">

  <div class="modal">

    <!-- Modal header bar -->

    <ion-header-bar class="bar-secondary">

      <h1 class="title">New Task</h1>

      <button class="button button-clear button-positive" ng-click="closeNewTask()">Cancel</button>

    </ion-header-bar>

    <!-- Modal content area -->

    <ion-content>

      <form ng-submit="createTask(task)">

        <div class="list">

          <label class="item item-input">

            <input type="text" placeholder="What do you need to do?" ng-model="task.title">

          </label>

        </div>

        <div class="padding">

          <button type="submit" class="button button-block button-positive">Create Task</button>

        </div>

      </form>

    </ion-content>

  </div>

</script>
```

以上代码中我们通过 `<script id="new-task.html" type="text/ng-template">` 定义了新的模板页面。

表单提交时触发 `createTask(task)` 函数。

`ng-model="task.title"` 会将表单的输入数据赋值给 `task` 对象的 `title` 属性。

修改 `<ion-side-menu-content>` 内的内容，代码如下：

```
<!-- 中心内容 -->

<ion-side-menu-content>

<ion-header-bar class="bar-dark">

  <h1 class="title">Todo</h1>

  <!-- 新增按钮 -->

  <button class="button button-icon" ng-click="newTask()">

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

  </button>

</ion-header-bar>

<ion-content>

  <!-- 列表 -->

  <ion-list>

    <ion-item ng-repeat="task in tasks">

      {{task.title}}

    </ion-item>

  </ion-list>

</ion-content>

</ion-side-menu-content>
```

`app.js` 文件中，控制器代码如下：

```
angular.module('todo', ['ionic'])

.controller('TodoCtrl', function($scope, $ionicModal) {

  $scope.tasks = [

    { title: '菜鸟教程' },

    { title: 'www.runoob.com' },

    { title: '菜鸟教程' },

    { title: 'www.runoob.com' }
```

```

];

// 创建并载入模型

$ionicModal.fromTemplateUrl('new-task.html', function(modal) {

    $scope.taskModal = modal;

}, {

    scope: $scope,

    animation: 'slide-in-up'

});

// 表单提交时调用

$scope.createTask = function(task) {

    $scope.tasks.push({

        title: task.title

    });

    $scope.taskModal.hide();

    task.title = "";

};

// 打开新增的模型

$scope.newTask = function() {

    $scope.taskModal.show();

};

// 关闭新增的模型

$scope.closeNewTask = function() {

    $scope.taskModal.hide();

};

});

```

## 创建侧边栏

通过以上步骤我们已经实现了新增功能，现在我们为 **app** 添加侧边栏功能。  
修改 **<ion-side-menu-content>** 内的内容，代码如下：

```

<!-- 中心内容 -->

<ion-side-menu-content>

  <ion-header-bar class="bar-dark">

    <button class="button button-icon" ng-click="toggleProjects()">

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

    </button>

    <h1 class="title">{{activeProject.title}}</h1>

    <!-- 新增按钮 -->

    <button class="button button-icon" ng-click="newTask()">

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

    </button>

  </ion-header-bar>

  <ion-content scroll="false">

    <ion-list>

      <ion-item ng-repeat="task in activeProject.tasks">

        {{task.title}}

      </ion-item>

    </ion-list>

  </ion-content>

</ion-side-menu-content>

```

添加侧边栏:

```

<!-- 左边栏 -->

<ion-side-menu side="left">

  <ion-header-bar class="bar-dark">

    <h1 class="title">Projects</h1>

    <button class="button button-icon ion-plus" ng-click="newProject()">

    </button>

  </ion-header-bar>

  <ion-content scroll="false">

    <ion-list>

      <ion-item ng-repeat="project in projects" ng-click="selectProject(project, $index)" ng-class="{active: activeProj

```

```

ect == project}">

    {{project.title}}

</ion-item>

</ion-list>

</ion-content>

</ion-side-menu>

```

`<ion-item>` 中的 `ng-class` 指令设置菜单激活样式。

这里我们创建一个新的js 文件 `app2.js`(为了不与前面的混淆)，代码如下：

```

angular.module('todo', ['ionic'])

/**
 * The Projects factory handles saving and loading projects
 * from local storage, and also lets us save and load the
 * last active project index.
 */
.factory('Projects', function() {

    return {

        all: function() {

            var projectString = window.localStorage['projects'];

            if(projectString) {

                return angular.fromJson(projectString);

            }

            return [];

        },

        save: function(projects) {

            window.localStorage['projects'] = angular.toJson(projects);

        },

        newProject: function(projectTitle) {

            // Add a new project

            return {

                title: projectTitle,

                tasks: []

            };

        }

    };

}

```



```

    },

    getLastActiveIndex: function() {

        return parseInt(window.localStorage['lastActiveProject']) || 0;

    },

    setLastActiveIndex: function(index) {

        window.localStorage['lastActiveProject'] = index;

    }

}

}))

.controller('TodoCtrl', function($scope, $timeout, $ionicModal, Projects, $ionicSideMenuDelegate) {

    // A utility function for creating a new project

    // with the given projectTitle

    var createProject = function(projectTitle) {

        var newProject = Projects.newProject(projectTitle);

        $scope.projects.push(newProject);

        Projects.save($scope.projects);

        $scope.selectProject(newProject, $scope.projects.length-1);

    }

    // Load or initialize projects

    $scope.projects = Projects.all();

    // Grab the last active, or the first project

    $scope.activeProject = $scope.projects[Projects.getLastActiveIndex()];

    // Called to create a new project

    $scope.newProject = function() {

        var projectTitle = prompt('Project name');

        if(projectTitle) {

```

```
        createProject(projectTitle);

    }

};

// Called to select the given project

$scope.selectProject = function(project, index) {

    $scope.activeProject = project;

    Projects.setLastActiveIndex(index);

    $ionicSideMenuDelegate.toggleLeft(false);

};

// Create our modal

$ionicModal.fromTemplateUrl('new-task.html', function(modal) {

    $scope.taskModal = modal;

}, {

    scope: $scope

});

$scope.createTask = function(task) {

    if(!$scope.activeProject || !task) {

        return;

    }

    $scope.activeProject.tasks.push({

        title: task.title

    });

    $scope.taskModal.hide();

    // Inefficient, but save all the projects

    Projects.save($scope.projects);

    task.title = "";

};
```

```

$scope.newTask = function() {

    $scope.taskModal.show();

};

$scope.closeNewTask = function() {

    $scope.taskModal.hide();

}

$scope.toggleProjects = function() {

    $ionicSideMenuDelegate.toggleLeft();

};

// Try to create the first project, make sure to defer
// this by using $timeout so everything is initialized
// properly

$timeout(function() {

    if($scope.projects.length == 0) {

        while(true) {

            var projectTitle = prompt('Your first project title:');

            if(projectTitle) {

                createProject(projectTitle);

                break;

            }

        }

    }

});

});

```

body 中 ion-side-menus 代码如下：

```
<ion-side-menus>
```

```
<!-- 中心内容 -->
```

```
<ion-side-menu-content>
```

```
  <ion-header-bar class="bar-dark">
```

```
    <button class="button button-icon" ng-click="toggleProjects()">
```

```
      <i class="icon ion-navicon"></i>
```

```
    </button>
```

```
    <h1 class="title">{{activeProject.title}}</h1>
```

```
    <!-- 新增按钮 -->
```

```
    <button class="button button-icon" ng-click="newTask()">
```

```
      <i class="icon ion-compose"></i>
```

```
    </button>
```

```
  </ion-header-bar>
```

```
<ion-content scroll="false">
```

```
  <ion-list>
```

```
    <ion-item ng-repeat="task in activeProject.tasks">
```

```
      {{task.title}}
```

```
    </ion-item>
```

```
  </ion-list>
```

```
</ion-content>
```

```
</ion-side-menu-content>
```

```
<!-- 左边栏 -->
```

```
<ion-side-menu side="left">
```

```
  <ion-header-bar class="bar-dark">
```

```
    <h1 class="title">Projects</h1>
```

```
    <button class="button button-icon ion-plus" ng-click="newProject()">
```

```
  </button>
```

```
  </ion-header-bar>
```

```
<ion-content scroll="false">
```

```
  <ion-list>
```

```
    <ion-item ng-repeat="project in projects" ng-click="selectProject(project, $index)" ng-class="{active: activeProject == project}">
```

```
        {{project.title}}

    </ion-item>

</ion-list>

</ion-content>

</ion-side-menu>

</ion-side-menus>
```

尝试一下 »

[ionic 安装](#)

[ionic 头部与底部](#)

[点我分享笔记](#)

[反馈/建议](#)

Copyright © 2013-2018 菜鸟教程 [runoob.com](#) All Rights Reserved. 备案号: 闽ICP备15012807号-1



[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

[ionic 创建 APP](#)

[ionic 按钮](#)

## ionic 头部与底部

### Header(头部)

Header是固定在屏幕顶部的组件,可以包如标题和左右的功能按钮。

ionic 默认提供了许多种颜色样式, 你可以调用不同的样式名, 当然也可以自定义一个。

#### bar-light

```
<div class="bar bar-header bar-light">

    <h1 class="title">bar-light</h1>

</div>
```

尝试一下 »

#### bar-stable

```
<div class="bar bar-header bar-stable">

  <h1 class="title">bar-stable</h1>

</div>
```

尝试一下 »

## bar-positive

```
<div class="bar bar-header bar-positive">

  <h1 class="title">bar-positive</h1>

</div>
```

尝试一下 »

## bar-calm

```
<div class="bar bar-header bar-calm">

  <h1 class="title">bar-calm</h1>

</div>
```

尝试一下 »

## bar-balanced

```
<div class="bar bar-header bar-balanced">

  <h1 class="title">bar-balanced</h1>

</div>
```

尝试一下 »

## bar-energized

```
<div class="bar bar-header bar-energized">

  <h1 class="title">bar-energized</h1>

</div>
```

尝试一下 »

## bar-assertive

```
<div class="bar bar-header bar-assertive">

  <h1 class="title">bar-assertive</h1>

</div>
```

尝试一下 »

## bar-royal

```
<div class="bar bar-header bar-royal">

  <h1 class="title">bar-royal</h1>

</div>
```

尝试一下 »

## bar-dark

```
<div class="bar bar-header bar-dark">

  <h1 class="title">bar-dark</h1>

</div>
```

尝试一下 »

## Sub Header（副标题）

Sub Header同样是固定在顶部，只是在Header的下面，就算没有写Header这个，Sub Header这个样式也会距离顶部有一个Header的距离。颜色样式同Header。

```
<div class="bar bar-header">

  <h1 class="title">Header</h1>

</div>

<div class="bar bar-subheader">

  <h2 class="title">Sub Header</h2>

</div>
```

尝试一下 »

## Footer(底部)

Footer是在屏幕的最下方，可以包含多种内容类型。

```
<div class="bar bar-footer bar-balanced">
```

```
<div class="title">Footer</div>

</div>
```

尝试一下 »

Footer 同上面的 Header，只是把样式名 bar-header 换做 bar-footer。

```
<div class="bar bar-footer">

  <button class="button button-clear">Left</button>

  <div class="title">Title</div>

  <button class="button button-clear">Right</button>

</div>
```

尝试一下 »

此外，如果底部没有标题，但是又需要右边的按钮，你需要在右侧按钮添加 pull-right如：

```
<div class="bar bar-footer">

  <button class="button button-clear pull-right">Right</button>

</div>
```

尝试一下 »

## ionic 按钮

按钮是移动app不可或缺的一部分，不同风格的app，需要的不同按钮的样式。

默认情况下，按钮显示样式为：**display: inline-block。**



```
<button class="button">
```

```
    Default
```

```
</button>
```

```
<button class="button button-light">
```

```
    button-light
```

```
</button>
```

```
<button class="button button-stable">
```

```
    button-stable
```

```
</button>
```

```
<button class="button button-positive">
```

```
    button-positive
```

```
</button>
```

```
<button class="button button-calm">
```

```
    button-calm
```

```
</button>
```

```
<button class="button button-balanced">
```

```
    button-balanced
```

```
</button>
```

```
<button class="button button-energized">
```

```
    button-energized
```

```
</button>
```

```
<button class="button button-assertive">
```

```
    button-assertive
```

```
</button>
```

```
<button class="button button-royal">
```

```
    button-royal
```

```
</button>
```

```
<button class="button button-dark">
```

```
    button-dark
```

```
</button>
```

尝试一下 »

`button-block` 样式按钮显示为: **display: block**, 它将完全填充父元素的宽度, 包含了内边距属性`padding`。

```
<button class="button button-block button-positive">
```

```
    Block Button
```

```
</button>
```

尝试一下 »

使用 `button-full` 类, 可以让按钮显示完全宽度, 且不包含内边距`padding`。

```
<button class="button button-full button-positive">
```

```
    Full Width Block Button
```

```
</button>
```

尝试一下 »

## 不同大小的按钮

`button-large` 设置为大按钮, `button-small` 设置为小按钮。

```
<button class="button button-small button-assertive">
```

```
    Small Button
```

```
</button>
```

```
<button class="button button-large button-positive">
```

```
    Large Button
```

```
</button>
```

尝试一下 »

## 无背景按钮

`button-outline` 设置背景为透明。

```
<button class="button button-outline button-positive">

  Outlined Button

</button>
```

尝试一下 »

## 无背景与边框按钮

`button-clear` 设置按钮背景为透明，且无边框。

```
<button class="button button-clear button-positive">

  Clear Button

</button>
```

尝试一下 »

## 图标按钮

我们可以在按钮上添加图标。

```
<button class="button icon-left ion-home">Home</button>

<button class="button icon-left ion-star button-positive">Favorites</button>

<a class="button icon-right ion-chevron-right button-calm">Learn More</a>

<a class="button icon-left ion-chevron-left button-clear button-dark">Back</a>

<button class="button icon ion-gear-a"></button>

<a class="button button-icon icon ion-settings"></a>

<a class="button button-outline icon-right ion-navicon button-balanced">Reorder</a>
```

尝试一下 »

## 头部/底部添加按钮

头部/底部可以添加按钮，按钮的样式根据头部/底部来设定，所以你不需要为按钮添加额外的样式。

```
<div class="bar bar-header">

  <button class="button icon ion-navicon"></button>

  <h1 class="title">Header Buttons</h1>

  <button class="button">Edit</button>

</div>
```

尝试一下 »

**button-clear** 类可以设置无背景和边框的头部/底部按钮。

```
<div class="bar bar-header">

  <button class="button button-icon icon ion-navicon"></button>

  <div class="h1 title">Header Buttons</div>

  <button class="button button-clear button-positive">Edit</button>

</div>
```

尝试一下 »

## 按钮栏

我们可以使用 **button-bar** 类来设置按钮栏。以下实例中，我们在头部和内容中添加了按钮栏。

```
<div class="button-bar">

  <a class="button">First</a>

  <a class="button">Second</a>

  <a class="button">Third</a>

</div>
```

尝试一下 »

[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)[☐ ionic 按钮](#)[ionic 卡片 ☐](#)

## ionic 列表

列表是一个应用广泛的界面元素，在所有移动app中几乎都会使用到。

列表可以是基本文字、按钮，开关，图标和缩略图等。

列表项可以是任何的HTML元素。容器元素需要list类，每个列表项需要使用item类。

ionList和ionItem可以很容易的支持各种交互方式，比如，滑动编辑，拖动排序，以及删除项。

### 基本用法:

```
<ul class="list">

  <li class="item">

    ...

  </li>

</ul>
```

[尝试一下 »](#)

### 列表分隔符

我们可以使用 **item-divider** 类来为列表创建分隔符，默认情况下，列表项以不同的背景颜色和字体加粗来区分，但你也可以很容易的定制他。

```
<div class="list">

  <div class="item item-divider">

    Candy Bars

  </div>

  <a class="item" href="#">

    Butterfinger

  </a>

  ...

</div>
```

</div>

尝试一下 »

## 带图标列表

我们可以在列表项的左侧或右侧指定图标。

使用 `item-icon-left` 图标在左侧，`item-icon-right` 设置图标在右侧。如果你需要在两边都有图标，则两个类都添加上即可。

以下实例中，我们在列表项中使用了 `<a>` 标签，使得每个列表项可点击。

列表项在使用 `<a>`或`<button>` 元素时，如果右侧未添加图标，则会自动添加上箭头号。

实例中，第一项只有左侧图标，第二项左右均有图标，第三项有右侧图标（还有注释 `item-note`），第四项有`badge`（标记）元素。

```
<div class="list">

  <a class="item item-icon-left" href="#">

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

    Check mail

  </a>

  <a class="item item-icon-left item-icon-right" href="#">

    <i class="icon ion-chatbubble-working"></i>

    Call Ma

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

  </a>

  <a class="item item-icon-left" href="#">

    <i class="icon ion-mic-a"></i>

    Record album

    <span class="item-note">

      Grammy

    </span>

  </a>

  <a class="item item-icon-left" href="#">

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

    Friends
```

```
<span class="badge badge-assertive">0</span>

</a>

</div>
```

尝试一下 »

## 按钮列表

使用 `item-button-right` 或 `item-button-left` 类将按钮放在列表项中。

```
<div class="list">

  <div class="item item-button-right">

    Call Ma

    <button class="button button-positive">

      <i class="icon ion-ios-telephone"></i>

    </button>

  </div>

  ...

</div>
```

尝试一下 »

## 带头像列表

使用 `item-avatar` 来创建一个带头像的列表：

```
<div class="list">

  <a class="item item-avatar" href="#">

    <h2>Venkman</h2>

    <p>Back off, man. I'm a scientist.</p>

  </a>
```

...

</div>

尝试一下 »

## 缩略图列表

`item-thumbnail-left` 类用于添加左侧对齐的缩略图， `item-thumbnail-right` 类用于添加右侧对齐的缩略图。

```
<div class="list">

  <a class="item item-thumbnail-left" href="#">

    <h2>Pretty Hate Machine</h2>

    <p>Nine Inch Nails</p>

  </a>

  ...

</div>
```

尝试一下 »

## 内嵌列表(inset list)

我们可以在容器当中内嵌列表，列表不会显示完整的宽度。

内嵌列表的样式为：`list list-inset`，与常规列表区别是，它设置了外边距（`margin`）,类似于选项卡。

内嵌列表是没有阴影效果的，滚动时效果会更好。

```
<div class="list list-inset">

  <div class="item">

    Raiders of the Lost Ark

  </div>

  ...

</div>
```



</div>

尝试一下 »

❏ ionic 按钮

ionic 卡片 ❏

❏ 点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 [runoob.com](http://runoob.com) All Rights Reserved. 备案号：闽ICP备15012807号-1

❏

[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

❏ ionic 列表

ionic 表单和输入框 ❏

## ionic 卡片

近年来卡片(card)的应用越来越流行，卡片提供了一个更好组织信息展示的工具。

针对移动端的应用，卡片会根据屏幕大小自适应大小。

我们可以很灵活的控制卡片的显示效果，甚至实现动画效果。

卡片一般放在页面顶部，当然也可以实现左右切换的功能。

```
<div class="card">

  <div class="item item-text-wrap">

    基本卡片，包含了文本信息。

  </div>

</div>
```

尝试一下 »

卡片(card)默认样式带有box-shadow(阴影)，由于性能的原因，和他类似的元素像 list list-inset 并没有阴影。

如果你有很多的卡片，每个卡片都有很多子元素，建议使用内嵌列表（inset list）。

### 卡片的头部与底部

我们可以通过添加 item-divider 类为卡片添加头部与底部：

```
<div class="card">
```

```
<div class="item item-divider">

    卡片头部!

</div>

<div class="item item-text-wrap">

    基本卡片，包含了文本信息。

</div>

<div class="item item-divider">

    卡片底部!

</div>

</div>
```

尝试一下 »

## 卡片列表

使用 `list card` 类来设置卡片列表：

```
<div class="list card">

    <a href="#" class="item item-icon-left">

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

        Enter home address

    </a>

    <a href="#" class="item item-icon-left">

        <i class="icon ion-ios-telephone"></i>

        Enter phone number

    </a>

    <a href="#" class="item item-icon-left">

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

        Enter wireless password

    </a>

    <a href="#" class="item item-icon-left">
```

```
<i class="icon ion-card"></i>

Enter card information

</a>

</div>
```

尝试一下 »

## 带图片卡片

卡片中使用图片，效果会更好，实例如下：

```
<div class="list card">

  <div class="item item-avatar">

    <h2>Pretty Hate Machine</h2>

    <p>Nine Inch Nails</p>

  </div>

  <div class="item item-image">

  </div>

  <a class="item item-icon-left assertive" href="#">

    <i class="icon ion-music-note"></i>

    Start listening

  </a>

</div>
```

尝试一下 »

运行效果如下：

# Error response

Error code 404.

Message: File not found.

Error code explanation: 404 = Nothing matches the given URI.

---

## 卡片展现

以下实例中使用几种不同的选项的卡片展现方式。 开始使用了 `list card` 元素，并使用了 `item-avatar` , `item-body` 元素用于展示图片和文本信息，底部使用 `item-divider` 类。

```
<div class="list card">

  <div class="item item-avatar">

    <h2>Marty McFly</h2>

    <p>November 05, 1955</p>

  </div>

  <div class="item item-body">

    <p>

      菜鸟教程 -- 学的不仅是技术，更是梦想！<br>

      菜鸟教程 -- 学的不仅是技术，更是梦想！<br>

      菜鸟教程 -- 学的不仅是技术，更是梦想！<br>

      菜鸟教程 -- 学的不仅是技术，更是梦想！

    </p>

  </div>

</div>
```

```
<p>

  <a href="#" class="subdued">1 喜欢</a>

  <a href="#" class="subdued">5 评论</a>

</p>

</div>

<div class="item tabs tabs-secondary tabs-icon-left">

  <a class="tab-item" href="#">

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

    喜欢

  </a>

  <a class="tab-item" href="#">

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

    评论

  </a>

  <a class="tab-item" href="#">

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

    分享

  </a>

</div>

</div>
```

尝试一下 »

运行效果如下:

# Error response

Error code 404.

Message: File not found.

Error code explanation: 404 = Nothing matches the given URI.

[ionic 列表](#)

[ionic 表单和输入框](#)

[点我分享笔记](#)

[反馈/建议](#)

Copyright © 2013-2018 菜鸟教程 [runoob.com](#) All Rights Reserved. 备案号：闽ICP备15012807号-1



[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

[ionic 卡片](#)

[ionic toggle\(切换开关\)](#)

## ionic 表单和输入框

`list` 类同样可以用于 `input` 元素。`item-input` 和 `item` 类指定了文本框及其标签。

### 输入框属性：placeholder

以下实例中，默认为100%宽度（左右两侧没有边框），并使用 `placeholder` 属性设置输入字段预期值的提示信息。

```
<div class="list">

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

```
<input type="text" placeholder="First Name">

</label>

<label class="item item-input">

  <input type="text" placeholder="Last Name">

</label>

<label class="item item-input">

  <textarea placeholder="Comments"></textarea>

</label>

</div>
```

尝试一下 »

## 输入框属性：input-label

使用 `input-label` 将标签放置于输入框 `input` 的左侧。

```
<div class="list">

  <label class="item item-input">

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

    <input type="text">

  </label>

  <label class="item item-input">

    <span class="input-label">密码: </span>

    <input type="password">

  </label>

</div>
```

尝试一下 »

## 堆叠标签

堆叠标签通常位于输入框的头部。每个选项使用 `item-stacked-label` 类指定。每个输入框需要指定 `input-label`。以下实例也使用了 `placeholder` 属性来设置信息输入提示。

```
<div class="list">

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

    <span class="input-label">First Name</span>
```

```
<input type="text" placeholder="John">

</label>

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

  <span class="input-label">Last Name</span>

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

</label>

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

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

  <input type="text" placeholder="john@suhr.com">

</label>

</div>
```

尝试一下 »

## 浮动标签

浮动标签类似于堆叠标签，但浮动标签有一个动画的效果，每个选项需要指定 `item-floating-label` 类，输入标签需要指定 `input-label`。

```
<div class="list">

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

    <span class="input-label">First Name</span>

    <input type="text" placeholder="First Name">

  </label>

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

    <span class="input-label">Last Name</span>

    <input type="text" placeholder="Last Name">

  </label>

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

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

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

  </label>

</div>
```

尝试一下 »

## 内嵌表单



默认情况下每个输入域宽度都是100%，但我们可以使用 `list list-inset` 或 `card` 类设置表单的内边距(padding)， `card` 类带有阴影。

```
<div class="list list-inset">

  <label class="item item-input">

    <input type="text" placeholder="First Name">

  </label>

  <label class="item item-input">

    <input type="text" placeholder="Last Name">

  </label>

</div>
```

尝试一下 »

## 内嵌输入域

使用 `list-inset` 设置内嵌实体列表。 使用 `item-input-inset` 样式可以内嵌一个按钮。

```
<div class="list">

  <div class="item item-input-inset">

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

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

    </label>

    <button class="button button-small">

      Submit

    </button>

  </div>

</div>
```

尝试一下 »

## 带图标的输入框

`item-input` 输入框可以很简单的添加图标。 图标可以在 `<input>` 前添加。

```
<div class="list list-inset">

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

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

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

</label>

</div>
```

尝试一下 »

## 头部输入框

输入框可放置在头部，并可添加提交或取消按钮。

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

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

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

    <input type="search" placeholder="搜索">

  </label>

  <button class="button button-clear">

    取消

  </button>

</div>
```

尝试一下 »

☐ ionic 卡片

ionic toggle(切换开关) ☐

☐ 点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 [runoob.com](http://runoob.com) All Rights Reserved. 备案号：闽ICP备15012807号-1

□

[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

☐ ionic 表单和输入框

ionic checkbox ☐

## ionic Toggle(切换开关)

切换开关类似与 HTML 的 `checkbox` 标签，但它更易于在移动设备上使用。

切换开关可以使用 `toggle-assertive` 来指定颜色。

```
<label class="toggle">

  <input type="checkbox">

  <div class="track">

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

  </div>

</label>
```

该实例有是多个切换开关列表。注意，每个选项的 `item` 类后需要添加 `item-toggle` 类。

```
<ul class="list">

  <li class="item item-toggle">

    HTML5

    <label class="toggle toggle-assertive">

      <input type="checkbox">

      <div class="track">

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

      </div>

    </label>

  </li>

  ...

</ul>
```

尝试一下 »

运行效果如下：

# Error response

Error code 404.

Message: File not found.

Error code explanation: 404 = Nothing matches the given URI.

☐ ionic 表单和输入框

ionic checkbox ☐

☐ 点我分享笔记

反馈/建议



☐ ionic toggle(切换开关)

ionic 单选框 ☐

## ionic checkbox（复选框）

ionic 里面的 **Checkbox** 和普通的 **Checkbox** 效果上其实相差不大，只是样式上有所不同。

以下实例演示了多个复选框的列表。

注意，每个选项的 **item** 类后需要添加 **item-checkbox** 类。

复选框可以使用 **checkbox-assertive** 来指定颜色。

```
<ul class="list">

  <li class="item item-checkbox">

    <label class="checkbox">
```

```
<input type="checkbox">

</label>

Flux Capacitor

</li>

...

</ul>
```

尝试一下 »

运行效果如下：

## Error response

Error code 404.

Message: File not found.

Error code explanation: 404 = Nothing matches the given URI.

☐ ionic toggle(切换开关)

ionic 单选框 ☐

☐ 点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 [runoob.com](#) All Rights Reserved. 备案号：闽ICP备15012807号-1



[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

☐ ionic checkbox

ionic range ☐

# ionic 单选框

ionic 单选按钮与标准 `type="radio"` 的 `input` 元素类似。以下展示了现代app类型的单选按钮。

每个 `item-radio` 中的 `type="radio"` 的 `input` 元素的 `name` 属性都相同。`radio-icon` 类用于是否显示图标。

ionic 在单选项中使用了 `<label>` 元素，使其更易点击。

## 实例

```
<div class="list">
<label class="item item-radio">
<input type="radio" name="group" value="go" checked="checked">
<div class="radio-content">
<div class="item-content">
Go
</div>
<i class="radio-icon ion-checkmark"></i>
</div>
</label>
<label class="item item-radio">
<input type="radio" name="group" value="python">
<div class="radio-content">
<div class="item-content">
Python
</div>
<i class="radio-icon ion-checkmark"></i>
</div>
</label>
<label class="item item-radio">
<input type="radio" name="group" value="ruby">
<div class="radio-content">
<div class="item-content">
Ruby
</div>
<i class="radio-icon ion-checkmark"></i>
</div>
</label>
<label class="item item-radio">
<input type="radio" name="group" value=".net">
<div class="radio-content">
<div class="item-content">
.Net
</div>
<i class="radio-icon ion-checkmark"></i>
</div>
</label>
<label class="item item-radio">
<input type="radio" name="group" value="java">
<div class="radio-content">
<div class="item-content">
Java
</div>
<i class="radio-icon ion-checkmark"></i>
</div>
</label>
<label class="item item-radio">
<input type="radio" name="group" value="php">
<div class="radio-content">
<div class="item-content">
PHP
</div>
<i class="radio-icon ion-checkmark"></i>
</div>
</label>
</div>
```

尝试一下 »

运行效果如下：

# Error response

Error code 404.

Message: File not found.

Error code explanation: 404 = Nothing matches the given URI.

☐ ionic checkbox

ionic range ☐

☐ 点我分享笔记

反馈/建议



☐ ionic 单选框

ionic select ☐

## ionic Range

ionic Range 是一个滑块控件，ionic 为 Range 提供了很多种默认的风格。而且你可以在许多种元素里使用它比如列表或者 Card 。

### 实例

```
<div class="range">

  <i class="icon ion-volume-low"></i>

  <input type="range" name="volume">

  <i class="icon ion-volume-high"></i>

</div>

<div class="list" style="margin-top: 13px">
```

```
<div class="item item-divider">

  Ranges In A List

</div>

<div class="item range range-positive">

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

  <input type="range" name="volume" min="0" max="100" value="12">

  <i class="icon ion-ios-sunny"></i>

</div>

<div class="item range range-calm">

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

  <input type="range" name="volume" min="0" max="100" value="25">

  <i class="icon ion-ios-lightbulb"></i>

</div>

<div class="item range range-balanced">

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

  <input type="range" name="volume" min="0" max="100" value="38">

  <i class="icon ion-ios-bolt"></i>

</div>

<div class="item range range-energized">

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

  <input type="range" name="volume" min="0" max="100" value="50">

  <i class="icon ion-ios-moon"></i>

</div>

<div class="item range range-assertive">

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

  <input type="range" name="volume" min="0" max="100" value="63">

  <i class="icon ion-ios-partlysunny"></i>

</div>

<div class="item range range-royal">

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

  <input type="range" name="volume" min="0" max="100" value="75">

  <i class="icon ion-ios-rainy"></i>

</div>
```



```
<div class="item range range-dark">

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

  <input type="range" name="volume" min="0" max="100" value="88">

  <i class="icon ion-ios-lightbulb"></i>

</div>

</div>
```

尝试一下 »

运行效果如下:

## Error response

Error code 404.

Message: File not found.

Error code explanation: 404 = Nothing matches the given URI.

□ ionic 单选框

ionic select □

□ 点我分享笔记

反馈/建议

## ionic select

ionic select 的 select 相比原生的要更加美观些。但是弹出的可选选项样式是浏览器默认的。

每个平台上的可选项样式都是不一样的，在PC电脑的浏览器上，你会看到传统的下拉界面，Android 上会弹出单选按钮选项，iOS 有个滚轮操作界面。

### 实例

```
<div class="list">

  <div class="item item-input item-select">

    <div class="input-label">

      Lightsaber

    </div>

    <select>

      <option>Blue</option>

      <option selected="">Green</option>

      <option>Red</option>

    </select>

  </div>

  <div class="item item-input item-select">

    <div class="input-label">

      Fighter

    </div>

    <select>

      <option>ARC-170</option>

      <option>A-wing</option>

      <option>Delta-7</option>

      <option>Naboo N-1</option>

      <option>TIE</option>

      <option selected="">X-wing</option>

      <option>Y-wing</option>

    </select>

  </div>

</div>
```

```
</select>
```

```
</div>
```

```
<div class="item item-input item-select">
```

```
<div class="input-label">
```

```
Droid
```

```
</div>
```

```
<select>
```

```
<option>2-1B</option>
```

```
<option>B1</option>
```

```
<option>C-3P0</option>
```

```
<option>IG-88</option>
```

```
<option>IT-O</option>
```

```
<option selected="">R2-D2</option>
```

```
</select>
```

```
</div>
```

```
<div class="item item-input item-select">
```

```
<div class="input-label">
```

```
Planet
```

```
</div>
```

```
<select>
```

```
<option>Alderaan</option>
```

```
<option selected="">Dagobah</option>
```

```
<option>Felucia</option>
```

```
<option>Geonosis </option>
```

```
<option>Hoth</option>
```

```
<option>Kamino</option>
```

```
<option>Mygeeto</option>
```

```
<option>Naboo</option>
```

```
<option>Tataouine</option>
```

```
<option>Utapau</option>
```

```
<option>Yavin</option>
```

```
</select>
```

```
</div>
```

```
</div>
```

尝试一下 »

运行效果如下：

## Error response

Error code 404.

Message: File not found.

Error code explanation: 404 = Nothing matches the given URI.

☐ ionic range

ionic tab(选项卡) ☐

☐ 点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 [runoob.com](http://runoob.com) All Rights Reserved. 备案号：闽ICP备15012807号-1



[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

## ionic tab(选项卡)

**ionic tab**(选项卡) 是水平排列的按钮或者链接，用以页面间导航的切换。它可以包含文字和图标的组合，是一种移动设备上流行的导航方法。

以下选项卡容器使用了 **tabs** 类，每个选项卡使用 **tab-item** 类。默认情况下，选项卡是文本的，并没有图标。

### 实例

```
<div class="tabs">

  <a class="tab-item">

    主页

  </a>

  <a class="tab-item">

    收藏

  </a>

  <a class="tab-item">

    设置

  </a>

</div>
```

[尝试一下 »](#)

默认情况，选项卡颜色为默认，你可以设置以下不同颜色样式：**tabs-default**，**tabs-light**，**tabs-stable**，**tabs-positive**，**tabs-calm**，**tabs-balanced**，**tabs-energized**，**tabs-assertive**，**tabs-royal**，**tabs-dark**。

要隐藏选项卡栏，可使用 **tabs-item-hide** 类。

### 图标选项卡

在 **tabs** 类后添加 **tabs-icon-only** 类可设置只显示图标选项卡。

```
<div class="tabs tabs-icon-only">

  <a class="tab-item">

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

  </a>

  <a class="tab-item">

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

  </a>

  <a class="tab-item">

    <i class="icon ion-gear-a"></i>

  </a>

</div>
```

```
</div>
```

[尝试一下 »](#)

## 顶部图标+文字选项卡

在 `tabs` 类后添加 `tabs-icon-top` 类可设置顶部图标+文字选项卡。

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

  <a class="tab-item" href="#">

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

    主页

  </a>

  <a class="tab-item" href="#">

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

    收藏

  </a>

  <a class="tab-item" href="#">

    <i class="icon ion-gear-a"></i>

    设置

  </a>

</div>
```

[尝试一下 »](#)

## 左侧图标+文字选项卡

在 `tabs` 类后添加 `tabs-icon-left` 类可设置左侧图标+文字选项卡。

```
<div class="tabs tabs-icon-left">

  <a class="tab-item">

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

    主页

  </a>

  <a class="tab-item">

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

    收藏

  </a>
```

```
<a class="tab-item">

  <i class="icon ion-gear-a"></i>

  设置

</a>

</div>
```

尝试一下 »

## 条纹样式选项卡

可以在带有 **tabs** 的样式名的元素上添加 **tabs-striped** 来实现像 Android 风格的 **tabs**。也可以添加 **tabs-top** 来实现选项卡在页面顶部。

条纹选项卡颜色可通过 **tabs-background-{color}** 和 **tabs-color-{color}** 来控制，**{color}** 值可以是：default, light, stable, positive, calm, balanced, energized, assertive, royal, 或 dark。

注意：如果要再选项卡上设置头部标题，需要使用 **has-tabs-top** 类。

```
<div class="tabs-striped tabs-top tabs-background-positive tabs-color-light">

  <div class="tabs">

    <a class="tab-item active" href="#">

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

      Test

    </a>

    <a class="tab-item" href="#">

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

      Favorites

    </a>

    <a class="tab-item" href="#">

      <i class="icon ion-gear-a"></i>

      Settings

    </a>

  </div>

</div>

<div class="tabs-striped tabs-color-assertive">

  <div class="tabs">

    <a class="tab-item active" href="#">

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

      Test

    </a>
```

```
<a class="tab-item" href="#">

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

  Favorites

</a>

<a class="tab-item" href="#">

  <i class="icon ion-gear-a"></i>

  Settings

</a>

</div>

</div>
```

尝试一下 »

运行效果如下:

## Error response

Error code 404.

Message: File not found.

Error code explanation: 404 = Nothing matches the given URI.

☐ ionic select

ionic 网格(Grid) ☐

☐ 点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 [runoob.com](http://runoob.com) All Rights Reserved. 备案号: 闽ICP备15012807号-1





## ionic 网格(Grid)

ionic 的网格(Grid)和其他大部分框架有所不同，它采用了弹性盒子模型(Flexible Box Model)。而且在移动端，基本上的手机都支持。**row** 样式指定行，**col** 样式指定列。

### 同等大小网格

在带有 **row** 的样式的元素里如果包含了 **col** 的样式，**col** 就会设置为同等大小。

以下实例中 **row** 的样式包含了 5 个 **col** 样式，每个 **col** 的宽度为 20%。

```
<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>
```

[尝试一下 »](#)

### 指定列宽

你可以设定一行中各个列的大小不一样。默认情况下，列都会被划分为同等大小。但你也可以按百分比来设置列的宽度（一行为 12 个网格）。

```
<div class="row">

  <div class="col col-50">.col.col-50</div>

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

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

</div>
```

```
<div class="row">

  <div class="col col-75">.col.col-75</div>

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

</div>
```

```
<div class="row">

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

  <div class="col col-75">.col.col-75</div>
```

```
</div>

<div class="row">

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

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

</div>
```

尝试一下 »

注意：实例中，每个 `col` 样式会自动添加上边框和灰色背景。

下面列出了指定列宽的一些百分比的样式名：

.col-10	10%
.col-20	20%
.col-25	25%
.col-33	33.3333%
.col-50	50%
.col-67	66.6666%
.col-75	75%
.col-80	80%
.col-90	90%

### 有偏移量的网格

列可以设置左侧偏移量，实例如下：

```
<div class="row">

  <div class="col col-33 col-offset-33">.col</div>

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

</div>

<div class="row">

  <div class="col col-33">.col</div>

  <div class="col col-33 col-offset-33">.col</div>

</div>

<div class="row">

  <div class="col col-33 col-offset-67">.col</div>
```

</div>

尝试一下 »

下面是一些百分比的偏移量样式名：

.col-offset-10	10%
.col-offset-20	20%
.col-offset-25	25%
.col-offset-33	33.3333%
.col-offset-50	50%
.col-offset-67	66.6666%
.col-offset-75	75%
.col-offset-80	80%
.col-offset-90	90%

### 纵向对齐网格

弹性盒子模型可以很容易设置列纵向对齐。纵向对齐包含顶部，中间部分，底部，可以应用到每一行的列，或者指定的某列。

实例中，最后一列设置了最高的内容用于更好的演示纵向对齐网格。

```
<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>
```

尝试一下 »

## 响应式网格

手持设备屏幕在切换时，例如横屏，竖屏等。就需要设置每行的网格可以实现根据不同宽度自适应大小。

不同设备响应式类的样式如下：

响应式类	描述
.responsive-sm	小于手机横屏
.responsive-md	小于平板竖屏
.responsive-lg	小于平板横屏

```
<div class="row responsive-sm">

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

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

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

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

</div>

尝试一下 »

[ionic tab\(选项卡\)](#)

[ionic 颜色](#)

[点我分享笔记](#)

[反馈/建议](#)

Copyright © 2013-2018 菜鸟教程 [runoob.com](#) All Rights Reserved. 备案号: 闽ICP备15012807号-1



[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

[ionic 网格\(Grid\)](#)

[ionic icon\(图标\)](#)

# ionic 颜色

ionic 提供了很多颜色的配置，当然你可以根据自己的需要自定义颜色。

## 实例

```
<ul class="list color-list-demo">
<li class="item dark">
light
<span class="color-demo light-bg light-border"></span>
</li>
<li class="item stable-dark">
stable
<span class="color-demo stable-bg stable-border"></span>
</li>
<li class="item positive">
positive
<span class="color-demo positive-bg positive-border"></span>
</li>
<li class="item calm">
calm
<span class="color-demo calm-bg calm-border"></span>
</li>
<li class="item balanced">
balanced
<span class="color-demo balanced-bg balanced-border"></span>
</li>
<li class="item energized">
energized
<span class="color-demo energized-bg energized-border"></span>
</li>
<li class="item assertive">
assertive
<span class="color-demo assertive-bg assertive-border"></span>
</li>
<li class="item royal">
royal
<span class="color-demo royal-bg royal-border"></span>
</li>
<li class="item dark">
```

```
dark
<span class="color-demo dark-bg dark-border"></span>
</li>
</ul>
```

尝试一下 »

实例运行结果:

## Error response

Error code 404.

Message: File not found.

Error code explanation: 404 = Nothing matches the given URI.

[ionic 网格\(Grid\)](#)

[ionic icon\(图标\)](#)

[点我分享笔记](#)

反馈/建议

Copyright © 2013-2018 菜鸟教程 [runoob.com](#) All Rights Reserved. 备案号: 闽ICP备15012807号-1



[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

[ionic 颜色](#)

[ionic 上拉菜单\(ActionSheet\)](#)

## ionic icon(图标)

ionic 也默认提供了许多的图标, 大概有 700 多个, 针对 Android 和 iOS 有不同的样式。

Ionic icons 官网: <http://ionicons.com/>  
 国内图标样式 CDN 地址: <https://cdn.bootcss.com/ionicons/2.0.1/css/ionicons.min.css>。  
 用法也非常的简单，由一个 icon 类及指定图标类组成，如：

实例

```
<i class="icon ion-star"></i>
<i class = "icon ion-happy-outline"></i>
<i class = "icon ion-ios-analytics"></i>
<i class = "icon ion-ios-eye"></i>
```

尝试一下 »

我们可以通过 font-size 来设置图片的大小：

实例

```
.icon {
font-size: 50px;
}
```

尝试一下 »

默认图标按钮：

代码	图标
<i class="icon ion-ionic"></i>	
<i class="icon ion-arrow-up-a"></i>	
<i class="icon ion-arrow-right-a"></i>	
<i class="icon ion-arrow-down-a"></i>	
<i class="icon ion-arrow-left-a"></i>	
<i class="icon ion-arrow-up-b"></i>	
<i class="icon ion-arrow-right-b"></i>	
<i class="icon ion-arrow-down-b"></i>	
<i class="icon ion-arrow-left-b"></i>	
<i class="icon ion-arrow-up-c"></i>	
<i class="icon ion-arrow-right-c"></i>	
<i class="icon ion-arrow-down-c"></i>	
<i class="icon ion-arrow-left-c"></i>	
<i class="icon ion-arrow-return-right"></i>	
<i class="icon ion-arrow-return-left"></i>	
<i class="icon ion-arrow-swap"></i>	
<i class="icon ion-arrow-shrink"></i>	
<i class="icon ion-arrow-expand"></i>	
<i class="icon ion-arrow-move"></i>	

<i class="icon ion-arrow-resize"></i>	
<i class="icon ion-chevron-up"></i>	
<i class="icon ion-chevron-right"></i>	
<i class="icon ion-chevron-down"></i>	
<i class="icon ion-chevron-left"></i>	
<i class="icon ion-navicon-round"></i>	
<i class="icon ion-navicon"></i>	
<i class="icon ion-drag"></i>	
<i class="icon ion-log-in"></i>	
<i class="icon ion-log-out"></i>	
<i class="icon ion-checkmark-round"></i>	
<i class="icon ion-checkmark"></i>	
<i class="icon ion-checkmark-circled"></i>	
<i class="icon ion-close-round"></i>	
<i class="icon ion-close"></i>	
<i class="icon ion-close-circled"></i>	
<i class="icon ion-plus-round"></i>	
<i class="icon ion-plus"></i>	
<i class="icon ion-plus-circled"></i>	
<i class="icon ion-minus-round"></i>	
<i class="icon ion-minus"></i>	
<i class="icon ion-minus-circled"></i>	
<i class="icon ion-information"></i>	
<i class="icon ion-informaticon ion-circled"></i>	
<i class="icon ion-help"></i>	
<i class="icon ion-help-circled"></i>	
<i class="icon ion-backspace-outline"></i>	
<i class="icon ion-backspace"></i>	
<i class="icon ion-help-buoy"></i>	
<i class="icon ion-asterisk"></i>	
<i class="icon ion-alert"></i>	



<i class="icon ion-alert-circled"></i>	
<i class="icon ion-refresh"></i>	
<i class="icon ion-loop"></i>	
<i class="icon ion-shuffle"></i>	
<i class="icon ion-home"></i>	
<i class="icon ion-search"></i>	
<i class="icon ion-flag"></i>	
<i class="icon ion-star"></i>	
<i class="icon ion-heart"></i>	
<i class="icon ion-heart-broken"></i>	
<i class="icon ion-gear-a"></i>	
<i class="icon ion-gear-b"></i>	
<i class="icon ion-toggle-filled"></i>	
<i class="icon ion-toggle"></i>	
<i class="icon ion-settings"></i>	
<i class="icon ion-wrench"></i>	
<i class="icon ion-hammer"></i>	
<i class="icon ion-edit"></i>	
<i class="icon ion-trash-a"></i>	
<i class="icon ion-trash-b"></i>	
<i class="icon ion-document"></i>	
<i class="icon ion-document-text"></i>	
<i class="icon ion-clipboard"></i>	
<i class="icon ion-scissors"></i>	
<i class="icon ion-funnel"></i>	
<i class="icon ion-bookmark"></i>	
<i class="icon ion-email"></i>	
<i class="icon ion-email-unread"></i>	
<i class="icon ion-folder"></i>	
<i class="icon ion-filing"></i>	
<i class="icon ion-archive"></i>	

<i class="icon ion-reply"></i>	
<i class="icon ion-reply-all"></i>	
<i class="icon ion-forward"></i>	
<i class="icon ion-share"></i>	
<i class="icon ion-paper-airplane"></i>	
<i class="icon ion-link"></i>	
<i class="icon ion-paperclip"></i>	
<i class="icon ion-compose"></i>	
<i class="icon ion-briefcase"></i>	
<i class="icon ion-medkit"></i>	
<i class="icon ion-at"></i>	
<i class="icon ion-pound"></i>	
<i class="icon ion-quote"></i>	
<i class="icon ion-cloud"></i>	
<i class="icon ion-upload"></i>	
<i class="icon ion-more"></i>	
<i class="icon ion-grid"></i>	
<i class="icon ion-calendar"></i>	
<i class="icon ion-clock"></i>	
<i class="icon ion-compass"></i>	
<i class="icon ion-pinpoint"></i>	
<i class="icon ion-pin"></i>	
<i class="icon ion-navigate"></i>	
<i class="icon ion-location"></i>	
<i class="icon ion-map"></i>	
<i class="icon ion-lock-combination"></i>	
<i class="icon ion-locked"></i>	
<i class="icon ion-unlocked"></i>	
<i class="icon ion-key"></i>	
<i class="icon ion-arrow-graph-up-right"></i>	
<i class="icon ion-arrow-graph-down-right"></i>	

<i class="icon ion-arrow-graph-up-left"></i>	
<i class="icon ion-arrow-graph-down-left"></i>	
<i class="icon ion-stats-bars"></i>	
<i class="icon ion-connecticon ion-bars"></i>	
<i class="icon ion-pie-graph"></i>	
<i class="icon ion-chatbubble"></i>	
<i class="icon ion-chatbubble-working"></i>	
<i class="icon ion-chatbubbles"></i>	
<i class="icon ion-chatbox"></i>	
<i class="icon ion-chatbox-working"></i>	
<i class="icon ion-chatboxes"></i>	
<i class="icon ion-person"></i>	
<i class="icon ion-person-add"></i>	
<i class="icon ion-person-stalker"></i>	
<i class="icon ion-woman"></i>	
<i class="icon ion-man"></i>	
<i class="icon ion-female"></i>	
<i class="icon ion-male"></i>	
<i class="icon ion-transgender"></i>	
<i class="icon ion-fork"></i>	
<i class="icon ion-knife"></i>	
<i class="icon ion-spoon"></i>	
<i class="icon ion-soup-can-outline"></i>	
<i class="icon ion-soup-can"></i>	
<i class="icon ion-beer"></i>	
<i class="icon ion-wineglass"></i>	
<i class="icon ion-coffee"></i>	
<i class="icon ion-icecream"></i>	
<i class="icon ion-pizza"></i>	
<i class="icon ion-power"></i>	
<i class="icon ion-mouse"></i>	

<i class="icon ion-battery-full"></i>	
<i class="icon ion-battery-half"></i>	
<i class="icon ion-battery-low"></i>	
<i class="icon ion-battery-empty"></i>	
<i class="icon ion-battery-charging"></i>	
<i class="icon ion-wifi"></i>	
<i class="icon ion-bluetooth"></i>	
<i class="icon ion-calculator"></i>	
<i class="icon ion-camera"></i>	
<i class="icon ion-eye"></i>	
<i class="icon ion-eye-disabled"></i>	
<i class="icon ion-flash"></i>	
<i class="icon ion-flash-off"></i>	
<i class="icon ion-qr-scanner"></i>	
<i class="icon ion-image"></i>	
<i class="icon ion-images"></i>	
<i class="icon ion-wand"></i>	
<i class="icon ion-contrast"></i>	
<i class="icon ion-aperture"></i>	
<i class="icon ion-crop"></i>	
<i class="icon ion-easel"></i>	
<i class="icon ion-paintbrush"></i>	
<i class="icon ion-paintbucket"></i>	
<i class="icon ion-monitor"></i>	
<i class="icon ion-laptop"></i>	
<i class="icon ion-ipad"></i>	
<i class="icon ion-iphone"></i>	
<i class="icon ion-ipod"></i>	
<i class="icon ion-printer"></i>	
<i class="icon ion-usb"></i>	
<i class="icon ion-outlet"></i>	

<i class="icon ion-bug"></i>	
<i class="icon ion-code"></i>	
<i class="icon ion-code-working"></i>	
<i class="icon ion-code-download"></i>	
<i class="icon ion-fork-repo"></i>	
<i class="icon ion-network"></i>	
<i class="icon ion-pull-request"></i>	
<i class="icon ion-merge"></i>	
<i class="icon ion-xbox"></i>	
<i class="icon ion-playstation"></i>	
<i class="icon ion-steam"></i>	
<i class="icon ion-closed-captioning"></i>	
<i class="icon ion-videocamera"></i>	
<i class="icon ion-film-marker"></i>	
<i class="icon ion-disc"></i>	
<i class="icon ion-headphone"></i>	
<i class="icon ion-music-note"></i>	
<i class="icon ion-radio-waves"></i>	
<i class="icon ion-speakerphone"></i>	
<i class="icon ion-mic-a"></i>	
<i class="icon ion-mic-b"></i>	
<i class="icon ion-mic-c"></i>	
<i class="icon ion-volume-high"></i>	
<i class="icon ion-volume-medium"></i>	
<i class="icon ion-volume-low"></i>	
<i class="icon ion-volume-mute"></i>	
<i class="icon ion-levels"></i>	
<i class="icon ion-play"></i>	
<i class="icon ion-pause"></i>	
<i class="icon ion-stop"></i>	
<i class="icon ion-record"></i>	

<i class="icon ion-skip-forward"></i>	
<i class="icon ion-skip-backward"></i>	
<i class="icon ion-eject"></i>	
<i class="icon ion-bag"></i>	
<i class="icon ion-card"></i>	
<i class="icon ion-cash"></i>	
<i class="icon ion-pricetag"></i>	
<i class="icon ion-pricetags"></i>	
<i class="icon ion-thumbsup"></i>	
<i class="icon ion-thumbsdown"></i>	
<i class="icon ion-happy-outline"></i>	
<i class="icon ion-happy"></i>	
<i class="icon ion-sad-outline"></i>	
<i class="icon ion-sad"></i>	
<i class="icon ion-bowtie"></i>	
<i class="icon ion-tshirt-outline"></i>	
<i class="icon ion-tshirt"></i>	
<i class="icon ion-trophy"></i>	
<i class="icon ion-podium"></i>	
<i class="icon ion-ribbon-a"></i>	
<i class="icon ion-ribbon-b"></i>	
<i class="icon ion-university"></i>	
<i class="icon ion-magnet"></i>	
<i class="icon ion-beaker"></i>	
<i class="icon ion-erlenmeyer-flask"></i>	
<i class="icon ion-egg"></i>	
<i class="icon ion-earth"></i>	
<i class="icon ion-planet"></i>	
<i class="icon ion-lightbulb"></i>	
<i class="icon ion-cube"></i>	
<i class="icon ion-leaf"></i>	

<i class="icon ion-waterdrop"></i>	
<i class="icon ion-flame"></i>	
<i class="icon ion-fireball"></i>	
<i class="icon ion-bonfire"></i>	
<i class="icon ion-umbrella"></i>	
<i class="icon ion-nuclear"></i>	
<i class="icon ion-no-smoking"></i>	
<i class="icon ion-thermometer"></i>	
<i class="icon ion-speedometer"></i>	
<i class="icon ion-model-s"></i>	
<i class="icon ion-plane"></i>	
<i class="icon ion-jet"></i>	
<i class="icon ion-load-a"></i>	
<i class="icon ion-load-b"></i>	
<i class="icon ion-load-c"></i>	
<i class="icon ion-load-d"></i>	

## iOS 风格图标

代码	图标
<i class="icon ion-ios-ionic-outline"></i>	
<i class="icon ion-ios-arrow-back"></i>	
<i class="icon ion-ios-arrow-forward"></i>	
<i class="icon ion-ios-arrow-up"></i>	
<i class="icon ion-ios-arrow-right"></i>	
<i class="icon ion-ios-arrow-down"></i>	
<i class="icon ion-ios-arrow-left"></i>	
<i class="icon ion-ios-arrow-thin-up"></i>	
<i class="icon ion-ios-arrow-thin-right"></i>	
<i class="icon ion-ios-arrow-thin-down"></i>	
<i class="icon ion-ios-arrow-thin-left"></i>	
<i class="icon ion-ios-circle-filled"></i>	
<i class="icon ion-ios-circle-outline"></i>	

<i class="icon ion-ios-checkmark-empty"></i>	
<i class="icon ion-ios-checkmark-outline"></i>	
<i class="icon ion-ios-checkmark"></i>	
<i class="icon ion-ios-plus-empty"></i>	
<i class="icon ion-ios-plus-outline"></i>	
<i class="icon ion-ios-plus"></i>	
<i class="icon ion-ios-close-empty"></i>	
<i class="icon ion-ios-close-outline"></i>	
<i class="icon ion-ios-close"></i>	
<i class="icon ion-ios-minus-empty"></i>	
<i class="icon ion-ios-minus-outline"></i>	
<i class="icon ion-ios-minus"></i>	
<i class="icon ion-ios-informaticon ion-empty"></i>	
<i class="icon ion-ios-informaticon ion-outline"></i>	
<i class="icon ion-ios-information"></i>	
<i class="icon ion-ios-help-empty"></i>	
<i class="icon ion-ios-help-outline"></i>	
<i class="icon ion-ios-help"></i>	
<i class="icon ion-ios-search"></i>	
<i class="icon ion-ios-search-strong"></i>	
<i class="icon ion-ios-star"></i>	
<i class="icon ion-ios-star-half"></i>	
<i class="icon ion-ios-star-outline"></i>	
<i class="icon ion-ios-heart"></i>	
<i class="icon ion-ios-heart-outline"></i>	
<i class="icon ion-ios-more"></i>	
<i class="icon ion-ios-more-outline"></i>	
<i class="icon ion-ios-home"></i>	
<i class="icon ion-ios-home-outline"></i>	
<i class="icon ion-ios-cloud"></i>	
<i class="icon ion-ios-cloud-outline"></i>	



<i class="icon ion-ios-cloud-upload"></i>	
<i class="icon ion-ios-cloud-upload-outline"></i>	
<i class="icon ion-ios-cloud-download"></i>	
<i class="icon ion-ios-cloud-download-outline"></i>	
<i class="icon ion-ios-upload"></i>	
<i class="icon ion-ios-upload-outline"></i>	
<i class="icon ion-ios-download"></i>	
<i class="icon ion-ios-download-outline"></i>	
<i class="icon ion-ios-refresh"></i>	
<i class="icon ion-ios-refresh-outline"></i>	
<i class="icon ion-ios-refresh-empty"></i>	
<i class="icon ion-ios-reload"></i>	
<i class="icon ion-ios-loop-strong"></i>	
<i class="icon ion-ios-loop"></i>	
<i class="icon ion-ios-bookmarks"></i>	
<i class="icon ion-ios-bookmarks-outline"></i>	
<i class="icon ion-ios-book"></i>	
<i class="icon ion-ios-book-outline"></i>	
<i class="icon ion-ios-flag"></i>	
<i class="icon ion-ios-flag-outline"></i>	
<i class="icon ion-ios-glasses"></i>	
<i class="icon ion-ios-glasses-outline"></i>	
<i class="icon ion-ios-browsers"></i>	
<i class="icon ion-ios-browsers-outline"></i>	
<i class="icon ion-ios-at"></i>	
<i class="icon ion-ios-at-outline"></i>	
<i class="icon ion-ios-pricetag"></i>	
<i class="icon ion-ios-pricetag-outline"></i>	
<i class="icon ion-ios-pricetags"></i>	
<i class="icon ion-ios-pricetags-outline"></i>	
<i class="icon ion-ios-cart"></i>	

<i class="icon ion-ios-cart-outline"></i>	
<i class="icon ion-ios-chatboxes"></i>	
<i class="icon ion-ios-chatboxes-outline"></i>	
<i class="icon ion-ios-chatbubble"></i>	
<i class="icon ion-ios-chatbubble-outline"></i>	
<i class="icon ion-ios-cog"></i>	
<i class="icon ion-ios-cog-outline"></i>	
<i class="icon ion-ios-gear"></i>	
<i class="icon ion-ios-gear-outline"></i>	
<i class="icon ion-ios-settings"></i>	
<i class="icon ion-ios-settings-strong"></i>	
<i class="icon ion-ios-toggle"></i>	
<i class="icon ion-ios-toggle-outline"></i>	
<i class="icon ion-ios-analytics"></i>	
<i class="icon ion-ios-analytics-outline"></i>	
<i class="icon ion-ios-pie"></i>	
<i class="icon ion-ios-pie-outline"></i>	
<i class="icon ion-ios-pulse"></i>	
<i class="icon ion-ios-pulse-strong"></i>	
<i class="icon ion-ios-filing"></i>	
<i class="icon ion-ios-filing-outline"></i>	
<i class="icon ion-ios-box"></i>	
<i class="icon ion-ios-box-outline"></i>	
<i class="icon ion-ios-compose"></i>	
<i class="icon ion-ios-compose-outline"></i>	
<i class="icon ion-ios-trash"></i>	
<i class="icon ion-ios-trash-outline"></i>	
<i class="icon ion-ios-copy"></i>	
<i class="icon ion-ios-copy-outline"></i>	
<i class="icon ion-ios-email"></i>	
<i class="icon ion-ios-email-outline"></i>	

<i class="icon ion-ios-undo"></i>	
<i class="icon ion-ios-undo-outline"></i>	
<i class="icon ion-ios-redo"></i>	
<i class="icon ion-ios-redo-outline"></i>	
<i class="icon ion-ios-paperplane"></i>	
<i class="icon ion-ios-paperplane-outline"></i>	
<i class="icon ion-ios-folder"></i>	
<i class="icon ion-ios-folder-outline"></i>	
<i class="icon ion-ios-paper"></i>	
<i class="icon ion-ios-paper-outline"></i>	
<i class="icon ion-ios-list"></i>	
<i class="icon ion-ios-list-outline"></i>	
<i class="icon ion-ios-world"></i>	
<i class="icon ion-ios-world-outline"></i>	
<i class="icon ion-ios-alarm"></i>	
<i class="icon ion-ios-alarm-outline"></i>	
<i class="icon ion-ios-speedometer"></i>	
<i class="icon ion-ios-speedometer-outline"></i>	
<i class="icon ion-ios-stopwatch"></i>	
<i class="icon ion-ios-stopwatch-outline"></i>	
<i class="icon ion-ios-timer"></i>	
<i class="icon ion-ios-timer-outline"></i>	
<i class="icon ion-ios-clock"></i>	
<i class="icon ion-ios-clock-outline"></i>	
<i class="icon ion-ios-time"></i>	
<i class="icon ion-ios-time-outline"></i>	
<i class="icon ion-ios-calendar"></i>	
<i class="icon ion-ios-calendar-outline"></i>	
<i class="icon ion-ios-photos"></i>	
<i class="icon ion-ios-photos-outline"></i>	
<i class="icon ion-ios-albums"></i>	

<i class="icon ion-ios-albums-outline"></i>	
<i class="icon ion-ios-camera"></i>	
<i class="icon ion-ios-camera-outline"></i>	
<i class="icon ion-ios-reverse-camera"></i>	
<i class="icon ion-ios-reverse-camera-outline"></i>	
<i class="icon ion-ios-eye"></i>	
<i class="icon ion-ios-eye-outline"></i>	
<i class="icon ion-ios-bolt"></i>	
<i class="icon ion-ios-bolt-outline"></i>	
<i class="icon ion-ios-color-wand"></i>	
<i class="icon ion-ios-color-wand-outline"></i>	
<i class="icon ion-ios-color-filter"></i>	
<i class="icon ion-ios-color-filter-outline"></i>	
<i class="icon ion-ios-grid-view"></i>	
<i class="icon ion-ios-grid-view-outline"></i>	
<i class="icon ion-ios-crop-strong"></i>	
<i class="icon ion-ios-crop"></i>	
<i class="icon ion-ios-barcode"></i>	
<i class="icon ion-ios-barcode-outline"></i>	
<i class="icon ion-ios-briefcase"></i>	
<i class="icon ion-ios-briefcase-outline"></i>	
<i class="icon ion-ios-medkit"></i>	
<i class="icon ion-ios-medkit-outline"></i>	
<i class="icon ion-ios-medical"></i>	
<i class="icon ion-ios-medical-outline"></i>	
<i class="icon ion-ios-infinite"></i>	
<i class="icon ion-ios-infinite-outline"></i>	
<i class="icon ion-ios-calculator"></i>	
<i class="icon ion-ios-calculator-outline"></i>	
<i class="icon ion-ios-keypad"></i>	
<i class="icon ion-ios-keypad-outline"></i>	

<i class="icon ion-ios-telephone"></i>	
<i class="icon ion-ios-telephone-outline"></i>	
<i class="icon ion-ios-drag"></i>	
<i class="icon ion-ios-location"></i>	
<i class="icon ion-ios-locaticon ion-outline"></i>	
<i class="icon ion-ios-navigate"></i>	
<i class="icon ion-ios-navigate-outline"></i>	
<i class="icon ion-ios-locked"></i>	
<i class="icon ion-ios-locked-outline"></i>	
<i class="icon ion-ios-unlocked"></i>	
<i class="icon ion-ios-unlocked-outline"></i>	
<i class="icon ion-ios-monitor"></i>	
<i class="icon ion-ios-monitor-outline"></i>	
<i class="icon ion-ios-printer"></i>	
<i class="icon ion-ios-printer-outline"></i>	
<i class="icon ion-ios-game-controller-a"></i>	
<i class="icon ion-ios-game-controller-a-outline"></i>	
<i class="icon ion-ios-game-controller-b"></i>	
<i class="icon ion-ios-game-controller-b-outline"></i>	
<i class="icon ion-ios-americanfootball"></i>	
<i class="icon ion-ios-americanfootball-outline"></i>	
<i class="icon ion-ios-baseball"></i>	
<i class="icon ion-ios-baseball-outline"></i>	
<i class="icon ion-ios-basketball"></i>	
<i class="icon ion-ios-basketball-outline"></i>>	
<i class="icon ion-ios-tennisball"></i>	
<i class="icon ion-ios-tennisball-outline"></i>	
<i class="icon ion-ios-football"></i>	
<i class="icon ion-ios-football-outline"></i>	
<i class="icon ion-ios-body"></i>	
<i class="icon ion-ios-body-outline"></i>	

<i class="icon ion-ios-person"></i>	
<i class="icon ion-ios-person-outline"></i>	
<i class="icon ion-ios-personadd"></i>	
<i class="icon ion-ios-personadd-outline"></i>	
<i class="icon ion-ios-people"></i>	
<i class="icon ion-ios-people-outline"></i>	
<i class="icon ion-ios-musical-notes"></i>	
<i class="icon ion-ios-musical-note"></i>	
<i class="icon ion-ios-bell"></i>	
<i class="icon ion-ios-bell-outline"></i>	
<i class="icon ion-ios-mic"></i>	
<i class="icon ion-ios-mic-outline"></i>	
<i class="icon ion-ios-mic-off"></i>	
<i class="icon ion-ios-volume-high"></i>	
<i class="icon ion-ios-volume-low"></i>	
<i class="icon ion-ios-play"></i>	
<i class="icon ion-ios-play-outline"></i>	
<i class="icon ion-ios-pause"></i>	
<i class="icon ion-ios-pause-outline"></i>	
<i class="icon ion-ios-recording"></i>	
<i class="icon ion-ios-recording-outline"></i>	
<i class="icon ion-ios-fastforward"></i>	
<i class="icon ion-ios-fastforward-outline"></i>	
<i class="icon ion-ios-rewind"></i>	
<i class="icon ion-ios-rewind-outline"></i>	
<i class="icon ion-ios-skipbackward"></i>	
<i class="icon ion-ios-skipbackward-outline"></i>	
<i class="icon ion-ios-skipforward"></i>	
<i class="icon ion-ios-skipforward-outline"></i>	
<i class="icon ion-ios-shuffle-strong"></i>	
<i class="icon ion-ios-shuffle"></i>	

<i class="icon ion-ios-videocam"></i>	
<i class="icon ion-ios-videocam-outline"></i>	
<i class="icon ion-ios-film"></i>	
<i class="icon ion-ios-film-outline"></i>	
<i class="icon ion-ios-flask"></i>	
<i class="icon ion-ios-flask-outline"></i>	
<i class="icon ion-ios-lightbulb"></i>	
<i class="icon ion-ios-lightbulb-outline"></i>	
<i class="icon ion-ios-wineglass"></i>	
<i class="icon ion-ios-wineglass-outline"></i>	
<i class="icon ion-ios-pint"></i>	
<i class="icon ion-ios-pint-outline"></i>	
<i class="icon ion-ios-nutrition"></i>	
<i class="icon ion-ios-nutriticon ion-outline"></i>	
<i class="icon ion-ios-flower"></i>	
<i class="icon ion-ios-flower-outline"></i>	
<i class="icon ion-ios-rose"></i>	
<i class="icon ion-ios-rose-outline"></i>	
<i class="icon ion-ios-paw"></i>	
<i class="icon ion-ios-paw-outline"></i>	
<i class="icon ion-ios-flame"></i>	
<i class="icon ion-ios-flame-outline"></i>	
<i class="icon ion-ios-sunny"></i>	
<i class="icon ion-ios-sunny-outline"></i>	
<i class="icon ion-ios-partlysunny"></i>	
<i class="icon ion-ios-partlysunny-outline"></i>	
<i class="icon ion-ios-cloudy"></i>	
<i class="icon ion-ios-cloudy-outline"></i>	
<i class="icon ion-ios-rainy"></i>	
<i class="icon ion-ios-rainy-outline"></i>	
<i class="icon ion-ios-thunderstorm"></i>	

<i class="icon ion-ios-thunderstorm-outline"></i>	
<i class="icon ion-ios-snowy"></i>	
<i class="icon ion-ios-moon"></i>	
<i class="icon ion-ios-moon-outline"></i>	
<i class="icon ion-ios-cloudy-night"></i>	
<i class="icon ion-ios-cloudy-night-outline"></i>	

## Android 风格图标

Code	Result
<i class="icon ion-android-arrow-up"></i>	
<i class="icon ion-android-arrow-forward"></i>	
<i class="icon ion-android-arrow-down"></i>	
<i class="icon ion-android-arrow-back"></i>	
<i class="icon ion-android-arrow-dropup"></i>	
<i class="icon ion-android-arrow-dropup-circle"></i>	
<i class="icon ion-android-arrow-dropright"></i>	
<i class="icon ion-android-arrow-dropright-circle"></i>	
<i class="icon ion-android-arrow-dropdown"></i>	
<i class="icon ion-android-arrow-dropdown-circle"></i>	
<i class="icon ion-android-arrow-dropleft"></i>	
<i class="icon ion-android-arrow-dropleft-circle"></i>	
<i class="icon ion-android-add"></i>	
<i class="icon ion-android-add-circle"></i>	
<i class="icon ion-android-remove"></i>	
<i class="icon ion-android-remove-circle"></i>	
<i class="icon ion-android-close"></i>	
<i class="icon ion-android-cancel"></i>	
<i class="icon ion-android-radio-button-off"></i>	
<i class="icon ion-android-radio-button-on"></i>	
<i class="icon ion-android-checkmark-circle"></i>	
<i class="icon ion-android-checkbox-outline-blank"></i>	
<i class="icon ion-android-checkbox-outline"></i>	
<i class="icon ion-android-checkbox-blank"></i>	



<i class="icon ion-android-checkbox"></i>	
<i class="icon ion-android-done"></i>	
<i class="icon ion-android-done-all"></i>	
<i class="icon ion-android-menu"></i>	
<i class="icon ion-android-more-horizontal"></i>	
<i class="icon ion-android-more-vertical"></i>	
<i class="icon ion-android-refresh"></i>	
<i class="icon ion-android-sync"></i>	
<i class="icon ion-android-wifi"></i>	
<i class="icon ion-android-call"></i>	
<i class="icon ion-android-apps"></i>	
<i class="icon ion-android-settings"></i>	
<i class="icon ion-android-options"></i>	
<i class="icon ion-android-funnel"></i>	
<i class="icon ion-android-search"></i>	
<i class="icon ion-android-home"></i>	
<i class="icon ion-android-cloud-outline"></i>	
<i class="icon ion-android-cloud"></i>	
<i class="icon ion-android-download"></i>	
<i class="icon ion-android-upload"></i>	
<i class="icon ion-android-cloud-done"></i>	
<i class="icon ion-android-cloud-circle"></i>	
<i class="icon ion-android-favorite-outline"></i>	
<i class="icon ion-android-favorite"></i>	
<i class="icon ion-android-star-outline"></i>	
<i class="icon ion-android-star-half"></i>	
<i class="icon ion-android-star"></i>	
<i class="icon ion-android-calendar"></i>	
<i class="icon ion-android-alarm-clock"></i>	
<i class="icon ion-android-time"></i>	
<i class="icon ion-android-stopwatch"></i>	

<i class="icon ion-android-watch"></i>	
<i class="icon ion-android-locate"></i>	
<i class="icon ion-android-navigate"></i>	
<i class="icon ion-android-pin"></i>	
<i class="icon ion-android-compass"></i>	
<i class="icon ion-android-map"></i>	
<i class="icon ion-android-walk"></i>	
<i class="icon ion-android-bicycle"></i>	
<i class="icon ion-android-car"></i>	
<i class="icon ion-android-bus"></i>	
<i class="icon ion-android-subway"></i>	
<i class="icon ion-android-train"></i>	
<i class="icon ion-android-boat"></i>	
<i class="icon ion-android-plane"></i>	
<i class="icon ion-android-restaurant"></i>	
<i class="icon ion-android-bar"></i>	
<i class="icon ion-android-cart"></i>	
<i class="icon ion-android-camera"></i>	
<i class="icon ion-android-image"></i>	
<i class="icon ion-android-film"></i>	
<i class="icon ion-android-color-palette"></i>	
<i class="icon ion-android-create"></i>	
<i class="icon ion-android-mail"></i>	
<i class="icon ion-android-drafts"></i>	
<i class="icon ion-android-send"></i>	
<i class="icon ion-android-archive"></i>	
<i class="icon ion-android-delete"></i>	
<i class="icon ion-android-attach"></i>	
<i class="icon ion-android-share"></i>	
<i class="icon ion-android-share-alt"></i>	
<i class="icon ion-android-bookmark"></i>	

<i class="icon ion-android-document"></i>	
<i class="icon ion-android-clipboard"></i>	
<i class="icon ion-android-list"></i>	
<i class="icon ion-android-folder-open"></i>	
<i class="icon ion-android-folder"></i>	
<i class="icon ion-android-print"></i>	
<i class="icon ion-android-open"></i>	
<i class="icon ion-android-exit"></i>	
<i class="icon ion-android-contract"></i>	
<i class="icon ion-android-expand"></i>	
<i class="icon ion-android-globe"></i>	
<i class="icon ion-android-chat"></i>	
<i class="icon ion-android-textsms"></i>	
<i class="icon ion-android-hangout"></i>	
<i class="icon ion-android-happy"></i>	
<i class="icon ion-android-sad"></i>	
<i class="icon ion-android-person"></i>	
<i class="icon ion-android-people"></i>	
<i class="icon ion-android-person-add"></i>	
<i class="icon ion-android-contact"></i>	
<i class="icon ion-android-contacts"></i>	
<i class="icon ion-android-playstore"></i>	
<i class="icon ion-android-lock"></i>	
<i class="icon ion-android-unlock"></i>	
<i class="icon ion-android-microphone"></i>	
<i class="icon ion-android-microphone-off"></i>	
<i class="icon ion-android-notifications-none"></i>	
<i class="icon ion-android-notifications"></i>	
<i class="icon ion-android-notifications-off"></i>	
<i class="icon ion-android-volume-mute"></i>	
<i class="icon ion-android-volume-down"></i>	

<i class="icon ion-android-volume-up"></i>	
<i class="icon ion-android-volume-off"></i>	
<i class="icon ion-android-hand"></i>	
<i class="icon ion-android-desktop"></i>	
<i class="icon ion-android-laptop"></i>	
<i class="icon ion-android-phone-portrait"></i>	
<i class="icon ion-android-phone-landscape"></i>	
<i class="icon ion-android-bulb"></i>	
<i class="icon ion-android-sunny"></i>	
<i class="icon ion-android-alert"></i>	
<i class="icon ion-android-warning"></i>	

☐ ionic 颜色

ionic 上拉菜单(ActionSheet) ☐

☐ 点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 [runoob.com](http://runoob.com) All Rights Reserved. 备案号：闽ICP备15012807号-1



[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

☐ ionic icon(图标)

ionic 背景层 ☐

## ionic 上拉菜单(ActionSheet)

上拉菜单(ActionSheet)通过往上弹出的框，来让用户选择选项。

非常危险的选项会以高亮的红色来让人第一时间识别。你可以通过点击取消按钮或者点击空白的地方来让它消失。

### 实例

#### HTML 代码

```
<body ng-app="starter" ng-controller="actionsheetCtl" >

  <ion-pane>

    <ion-content >
```

```
        <h2 ng-click="show()">Action Sheet</h2>

    </ion-content>

</ion-pane>

</body>
```

## JavaScript 代码

在代码中触发上拉菜单，需要在你的 **angular** 控制器中使用 **\$ionicActionSheet** 服务：

```
angular.module('starter', ['ionic'])

.run(function($ionicPlatform) {

    $ionicPlatform.ready(function() {

        // Hide the accessory bar by default (remove this to show the accessory bar above the keyboard
        // for form inputs)

        if(window.cordova && window.cordova.plugins.Keyboard) {

            cordova.plugins.Keyboard.hideKeyboardAccessoryBar(true);

        }

        if(window.StatusBar) {

            StatusBar.styleDefault();

        }

    });

})

.controller( 'actionsheetCtl',['$scope','$ionicActionSheet','$timeout' ,function($scope,$ionicActionSheet,$timeout){

    $scope.show = function() {

        var hideSheet = $ionicActionSheet.show({

            buttons: [

                { text: '<b>Share</b> This' },

                { text: 'Move' }

            ],

            destructiveText: 'Delete',

            titleText: 'Modify your album',
```

```
cancelText: 'Cancel',

cancel: function() {

    // add cancel code..

},

buttonClicked: function(index) {

    return true;

}

});

$timeout(function() {

    hideSheet();

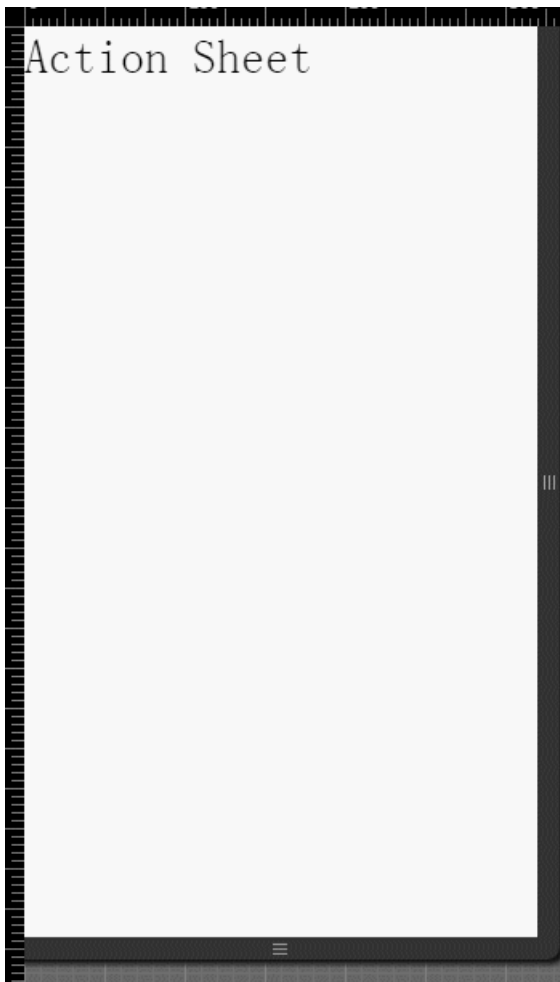
}, 2000);

};

}))
```

尝试一下 »

运行效果如下图：



# ionic 背景层

我们经常需要在 UI 上，例如在弹出框、加载框、其他弹出层中显示或隐藏背景层。

在组件中可以使用\$ionicBackdrop.retain()来显示背景层，使用\$ionicBackdrop.release()隐藏背景层。

每次调用retain后，背景会一直显示，直到调用release消除背景层。

## 实例

### HTML 代码

```
<body ng-app="starter" ng-controller="actionsheetCtl" >

  <ion-pane>

    <ion-content >

      <h2 ng-click="action()">$ionicBackdrop</h2>

    </ion-content>

  </ion-pane>

</body>
```

### JavaScript 代码

```
angular.module('starter', ['ionic'])

.run(function($ionicPlatform) {

  $ionicPlatform.ready(function() {

    // Hide the accessory bar by default (remove this to show the accessory bar above the keyboard
    // for form inputs)
```

```
if(window.cordova && window.cordova.plugins.Keyboard) {

    cordova.plugins.Keyboard.hideKeyboardAccessoryBar(true);

}

if(window.StatusBar) {

    StatusBar.styleDefault();

}

});

})

.controller( 'actionsheetCtl',['$scope','$timeout' , '$ionicBackdrop',function($scope,$timeout,$ionicBackdrop){

    $scope.action = function() {

        $ionicBackdrop.retain();

        $timeout(function() {    //默认让它1秒后消失

            $ionicBackdrop.release();

        }, 1000);

    };

}])
```

尝试一下 »

显示效果如下图所示:





[ionic 上拉菜单\(ActionSheet\)](#)

[ionic 下拉刷新](#)

[点我分享笔记](#)

[反馈/建议](#)

Copyright © 2013-2018 菜鸟教程 [runoob.com](#) All Rights Reserved. 备案号: 闽ICP备15012807号-1



[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

[ionic 背景层](#)

[ionic 复选框](#)

## ionic 下拉刷新

在加载新数据的时候，我们需要实现下拉刷新效果，代码如下：

### 实例

#### HTML 代码

```
<body ng-app="starter" ng-controller="actionsheetCtl" >
```

```

<ion-pane>

  <ion-content >

    <ion-refresher pulling-text="下拉刷新" on-refresh="doRefresh()"></ion-refresher>

    <ion-list>

      <ion-item ng-repeat="item in items" ng-bind="item.name"></ion-item>

    </ion-list>

  </ion-content>

</ion-pane>

</body>

```

## JavaScript 代码

```

angular.module('starter', ['ionic'])

.run(function($ionicPlatform) {

  $ionicPlatform.ready(function() {

    // Hide the accessory bar by default (remove this to show the accessory bar above the keyboard
    // for form inputs)

    if(window.cordova && window.cordova.plugins.Keyboard) {

      cordova.plugins.Keyboard.hideKeyboardAccessoryBar(true);

    }

    if(window.StatusBar) {

      StatusBar.styleDefault();

    }

  });

})

.controller( 'actionsheetCtl',['$scope','$timeout' , '$http',function($scope,$timeout,$http){

  $scope.items=[

    {

      "name": "HTML5"

    },

    {

```

```

        "name": "JavaScript"
    },
    {
        "name": "Css3"
    }
];

$scope.doRefresh = function() {

    $http.get('http://www.runoob.com/try/demo_source/item.json') //注意改为自己本站的地址，不然会有跨域问题

        .success(function(newItems) {

            $scope.items = newItems;

        })

        .finally(function() {

            $scope.$broadcast('scroll.refreshComplete');

        });

};

})

```

item.json 文件数据:

```

[
    {
        "name": "菜鸟教程"
    },
    {
        "name": "www.runoob.com"
    }
]

```

尝试一下 »

效果如下所示:

# Error response

Error code 404.

Message: File not found.

Error code explanation: 404 = Nothing matches the given URI.

[ionic 背景层](#)

[ionic 复选框](#)

[点我分享笔记](#)

反馈/建议



[ionic 下拉刷新](#)

[ionic 单选框操作](#)

## ionic 复选框

ionic 复选框（checkbox）与普通的 HTML 复选框没什么区别，以下实例演示了 ionic 复选框 ion-checkbox 的应用。

```
<ion-checkbox ng-model="isChecked">复选框标签</ion-checkbox>
```

### 实例

实例中，会根据复选框是否选中，修改 checked 值，true 为选中，false 为未选中。

### HTML 代码

```
<ion-header-bar class="bar-positive">

  <h1 class="title">复选框</h1>

</ion-header-bar>
```

```
<ion-content>
```

```
<div class="list">
```

```
<ion-checkbox ng-repeat="item in devList"
```

```
    ng-model="item.checked"
```

```
    ng-checked="item.checked">
```

```
    {{ item.text }}
```

```
</ion-checkbox>
```

```
<div class="item">
```

```
    <div ng-bind="devList | json"></div>
```

```
</div>
```

```
<div class="item item-divider">
```

```
    Notifications
```

```
</div>
```

```
<ion-checkbox ng-model="pushNotification.checked"
```

```
    ng-change="pushNotificationChange()">
```

```
    Push Notifications
```

```
</ion-checkbox>
```

```
<div class="item">
```

```
    <div ng-bind="pushNotification | json"></div>
```

```
</div>
```

```
<ion-checkbox ng-model="emailNotification"
```

```
    ng-true-value="'Subscribed'"
```

```
    ng-false-value="'Unsubscribed'">
```

```
    Newsletter
```

```

</ion-checkbox>

<div class="item">

  <div ng-bind="emailNotification | json"></div>

</div>

</div>

</ion-content>

```

## JavaScript 代码

```

angular.module('starter', ['ionic'])

.run(function($ionicPlatform) {

  $ionicPlatform.ready(function() {

    // Hide the accessory bar by default (remove this to show the accessory bar above the keyboard
    // for form inputs)

    if(window.cordova && window.cordova.plugins.Keyboard) {

      cordova.plugins.Keyboard.hideKeyboardAccessoryBar(true);

    }

    if(window.StatusBar) {

      StatusBar.styleDefault();

    }

  });

})

.controller( 'actionsheetCtl',['$scope',function($scope){

  $scope.devList = [

    { text: "HTML5", checked: true },

    { text: "CSS3", checked: false },

    { text: "JavaScript", checked: false }

  ];

```

```
$scope.pushNotificationChange = function() {  
  
    console.log('Push Notification Change', $scope.pushNotification.checked);  
  
};  
  
$scope.pushNotification = { checked: true };  
  
$scope.emailNotification = 'Subscribed';  
  
}})
```

CSS 代码:

```
body {  
  
    cursor: url('http://www.runoob.com/try/demo_source/finger.png'), auto;  
  
}
```

尝试一下 »

效果如下所示:

## Error response

Error code 404.

Message: File not found.

Error code explanation: 404 = Nothing matches the given URI.

[☐ ionic 下拉刷新](#)

[ionic 单选框操作 ☐](#)

[☐ 点我分享笔记](#)

[反馈/建议](#)



# ionic 单选框操作

实例中，根据选中的不同选项，显示不同的值。

## HTML 代码

```
<ion-header-bar class="bar-positive">

  <h1 class="title">当选按钮</h1>

</ion-header-bar>

<ion-content>

  <div class="list">

    <div class="item item-divider">

      选取的值为: {{ data.clientSide }}

    </div>

    <ion-radio ng-repeat="item in clientSideList"

      ng-value="item.value"

      ng-model="data.clientSide">

      {{ item.text }}

    </ion-radio>

    <div class="item item-divider">

      Serverside, Selected Value: {{ data.serverSide }}

    </div>

    <ion-radio ng-repeat="item in serverSideList"

      ng-value="item.value"
```



```
        ng-model="data.serverSide"

        ng-change="serverSideChange(item)"

        name="server-side">

        {{ item.text }}

    </ion-radio>

</div>

</ion-content>
```

## JavaScript 代码

```
angular.module('ionicApp', ['ionic'])

.controller('MainCtrl', function($scope) {

    $scope.clientSideList = [

        { text: "Backbone", value: "bb" },

        { text: "Angular", value: "ng" },

        { text: "Ember", value: "em" },

        { text: "Knockout", value: "ko" }

    ];

    $scope.serverSideList = [

        { text: "Go", value: "go" },

        { text: "Python", value: "py" },

        { text: "Ruby", value: "rb" },

        { text: "Java", value: "jv" }

    ];

    $scope.data = {

        clientSide: 'ng'

    };

});
```

```
$scope.serverSideChange = function(item) {  
  
    console.log("Selected Serverside, text:", item.text, "value:", item.value);  
  
};  
  
});
```

CSS 代码:

```
body {  
  
    cursor: url('http://www.runoob.com/try/demo_source/finger.png'), auto;  
  
}
```

尝试一下 »

效果如下所示:

## Error response

Error code 404.

Message: File not found.

Error code explanation: 404 = Nothing matches the given URI.

☐ ionic 复选框

ionic 切换开关操作 ☐

☐ 点我分享笔记

反馈/建议

## ionic 切换开关操作

以下实例中，通过切换不同开关 **checked** 显示不同的值，**true** 为打开，**false** 为关闭。

### HTML 代码

```
<ion-header-bar class="bar-positive">

  <h1 class="title">开关切换</h1>

</ion-header-bar>


<ion-content>


  <div class="list">


    <div class="item item-divider">

      Settings

    </div>


    <ion-toggle ng-repeat="item in settingsList"

      ng-model="item.checked"

      ng-checked="item.checked">

      {{ item.text }}

    </ion-toggle>


    <div class="item">

      <!-- 使用 pre 标签展示效果更美观 -->

      <div ng-bind="settingsList | json"></div>

    </div>


    <div class="item item-divider">

      Notifications

    </div>
```

```

<ion-toggle ng-model="pushNotification.checked"

      ng-change="pushNotificationChange()">

      Push Notifications

</ion-toggle>


<div class="item">

  <!-- 使用 pre 标签展示效果更美观 -->

  <div ng-bind="pushNotification | json"></div>

</div>


<ion-toggle toggle-class="toggle-assertive"

      ng-model="emailNotification"

      ng-true-value="'Subscribed'"

      ng-false-value="'Unsubscribed'">

      Newsletter

</ion-toggle>


<div class="item">

  <!-- 使用 pre 标签展示效果更美观 -->

  <div ng-bind="emailNotification | json"></div>

</div>


</div>


</ion-content>

```

由于pre标签冲突，实例中的 pre 已替换为 **div** 标签，具体可以在["尝试一下"](#)中查看。

## JavaScript 代码

```

angular.module('ionicApp', ['ionic'])

.controller('MainCtrl', function($scope) {

```

```
$scope.settingsList = [

    { text: "Wireless", checked: true },

    { text: "GPS", checked: false },

    { text: "Bluetooth", checked: false }

];


$scope.pushNotificationChange = function() {

    console.log('Push Notification Change', $scope.pushNotification.checked);

};


$scope.pushNotification = { checked: true };

$scope.emailNotification = 'Subscribed';


});
```

CSS 代码:

```
body {

    cursor: url('http://www.runoob.com/try/demo_source/finger.png'), auto;

}
```

尝试一下 »

效果如下所示:

## Error response

Error code 404.

Message: File not found.

Error code explanation: 404 = Nothing matches the given URI.

# ionic 手势事件

事件	描述	用法	实例
on-hold	长按的时间是500毫秒。	<pre>&lt;button    on-hold="onHold()"    class="button"&gt;    Test  &lt;/button&gt;</pre>	<div>尝试一下 »</div>
on-tap	这个是手势轻击事件，如果长按时间超过250毫秒，那就不是轻击了。。	<pre>&lt;button    on-tap="onTap()"    class="button"&gt;    Test  &lt;/button&gt;</pre>	<div>尝试一下 »</div>

事件	描述	用法	实例
on-double-tap	手双击屏幕事件	<pre>&lt;button      on-double-tap="onDoubleTap()"      class="button"&gt;      Test  &lt;/button&gt;</pre>	<div>尝试一下 »</div>
on-touch	这个和 <b>on-tap</b> 还是有区别的，这个是立即执行，而且是用户点击立马执行。不用等待 <b>touchend/mouseup</b> 。	<pre>&lt;button on-touch="onTouch()"      class="button"&gt;      Test  &lt;/button&gt;</pre>	<div>尝试一下 »</div>
on-release	当用户结束触摸事件时触发。	<pre>&lt;button      on-release="onRelease()"      class="button"&gt;      Test  &lt;/button&gt;</pre>	<div>尝试一下 »</div>
on-drag	这个有点类似于PC端的拖拽。当你一直点击某个物体，并且手开始移动，都会触发 <b>on-drag</b> 。	<pre>&lt;button      on-drag="onDrag()"      class="button"&gt;      Test  &lt;/button&gt;</pre>	<div>尝试一下 »</div>

事件	描述	用法	实例
on-drag-up	向上拖拽。	<pre>&lt;button    on-drag-up="onDragUp()"    class="button"&gt;    Test  &lt;/button&gt;</pre>	<div>尝试一下 »</div>
on-drag-right	向右拖拽。	<pre>&lt;button    on-drag-right="onDragRight()"    class="button"&gt;    Test  &lt;/button&gt;</pre>	<div>尝试一下 »</div>
on-drag-down	向下拖拽。	<pre>&lt;button    on-drag-down="onDragDown()"    class="button"&gt;    Test  &lt;/button&gt;</pre>	<div>尝试一下 »</div>
on-drag-left	向左边拖拽。	<pre>&lt;button    on-drag-left="onDragLeft()"    class="button"&gt;    Test  &lt;/button&gt;</pre>	<div>尝试一下 »</div>



事件	描述	用法	实例
on-swipe	指手指滑动效果，可以是任何方向上的。而且也和 <b>on-drag</b> 类似，都有四个方向上单独的事件。	<pre>&lt;button      on-swipe="onSwipe()"      class="button"&gt;      Test  &lt;/button&gt;</pre>	<div>尝试一下 »</div>
on-swipe-up	向上的手指滑动效果。	<pre>&lt;button      on-swipe-up="onSwipeUp()"      class="button"&gt;      Test  &lt;/button&gt;</pre>	<div>尝试一下 »</div>
on-swipe-right	向右的手指滑动效果。	<pre>&lt;button      on-swipe-right="onSwipeRight()"      class="button"&gt;      Test  &lt;/button&gt;</pre>	<div>尝试一下 »</div>
on-swipe-down	向下的手指滑动效果。	<pre>&lt;button      on-swipe-down="onSwipeDown()"      class="button"&gt;      Test  &lt;/button&gt;</pre>	<div>尝试一下 »</div>

事件	描述	用法	实例
on-swipe-left	向左的手指滑动效果。	<pre>&lt;button      on-swipe-left="onSwipeLeft()"      class="button"&gt;      Test  &lt;/button&gt;</pre>	<div>尝试一下 »</div>

## \$ionicGesture

一个angular服务展示ionicionic.EventController手势。

### 方法

<code>on(eventType, callback, \$element)</code>
---

在一个元素上添加一个事件监听器。

参数	类型	详情
eventType	string	监听的手势事件。
callback	function(e)	当手势事件发生时触发的事件。
\$element	element	angular元素监听的事件。
options	object	对象。

<code>off(eventType, callback, \$element)</code>
--

在一个元素上移除一个手势事件监听器。

参数	类型	详情
eventType	string	移除监听的手势事件。
callback	function(e)	移除监听器。
\$element	element	被监听事件的angular元素。

<div>☐ ionic 切换开关操作</div>	ionic 头部和底部 ☐
<div>☐ 点我分享笔记</div>	

反馈/建议

# ionic 头部和底部

## ion-header-bar

这个是固定在屏幕顶部的一个头部标题栏。如果给它加上'bar-subheader' 这个样式，它就是副标题。

### 用法

```
<ion-header-bar align-title="left" class="bar-positive">

  <div class="buttons">

    <button class="button" ng-click="doSomething()">Left Button</button>

  </div>

  <h1 class="title">Title!</h1>

  <div class="buttons">

    <button class="button">Right Button</button>

  </div>

</ion-header-bar>

<ion-content>

  Some content!

</ion-content>
```

### API

属性	类型	描述
align-title (optional)	string	这个是对齐 title 的。如果没有设置，它将会按照手机的默认排版(los的默认是居中，Android默认是居左)。它的值可以是 'left','center','right'。
no-tap-scroll (optional)	boolean	默认情况下，头部标题栏在点击屏幕时内容会滚动到头部，可以将 no-tap-scroll 设置为 true 来禁止该动作。。它的值是布尔值（true/false）。

## ion-footer-bar

知道了 ion-header-bar ，理解ion-footer-bar就轻松多啦！只是 ion-footer-bar 是在屏幕的底部。

### 用法

```
<ion-content>

  Some content!
```

```
</ion-content>

<ion-footer-bar align-title="left" class="bar-assertive">

  <div class="buttons">

    <button class="button">Left Button</button>

  </div>

  <h1 class="title">Title!</h1>

  <div class="buttons" ng-click="doSomething()">

    <button class="button">Right Button</button>

  </div>

</ion-footer-bar>
```

API

与 ion-header-bar 不同的是，ion-footer-bar 只有 align-title 这个 API。

属性	类型	描述
align-title (optional)	string	这个是对齐 title 的。如果没有设置，它将会按照手机的默认排版(ios的默认是居中，Android默认是居左)。它的值可以是 'left', 'center', 'right'。

[☐ ionic 手势事件](#)

[ionic 列表操作](#) ☐

[☐ 点我分享笔记](#)

反馈/建议



[☐ ionic 头部和底部](#)

[ionic 加载动作](#) ☐

ionic 列表操作

列表是一个应用广泛在几乎所有移动app中的界面元素。ionList 和 ionItem 这两个指令还支持多种多样的交互模式，比如移除其中的某一项，拖动重新排序，滑动编辑等等。

用法

```
<ion-list>

  <ion-item ng-repeat="item in items">
```

```
    Hello, {{item}}!  
  
  </ion-item>  
  
</ion-list>
```

## 高级用法: 缩略图, 删除按钮, 重新排序, 滑动

```
<ion-list ng-controller="MyCtrl"  
  
  show-delete="shouldShowDelete"  
  
  show-reorder="shouldShowReorder"  
  
  can-swipe="listCanSwipe">  
  
  <ion-item ng-repeat="item in items"  
  
    class="item-thumbnail-left">  
  
  
      
  
    <h2>{{item.title}}</h2>  
  
    <p>{{item.description}}</p>  
  
    <ion-option-button class="button-positive"  
  
      ng-click="share(item)">  
  
      分享  
  
    </ion-option-button>  
  
    <ion-option-button class="button-info"  
  
      ng-click="edit(item)">  
  
      编辑  
  
    </ion-option-button>  
  
    <ion-delete-button class="ion-minus-circled"  
  
      ng-click="items.splice($index, 1)">  
  
    </ion-delete-button>  
  
    <ion-reorder-button class="ion-navicon"  
  
      on-reorder="reorderItem(item, $fromIndex, $toIndex)">  
  
    </ion-reorder-button>  
  
  </ion-item>  
  
</ion-list>
```

API

属性	类型	详情
delegate-handle (可选)	字符串	该句柄定义带有\$ionicListDelegate的列表。
show-delete (可选)	布尔值	列表项的删除按钮当前是显示还是隐藏。
show-reorder (可选)	布尔值	列表项的排序按钮当前是显示还是隐藏。
can-swipe (可选)	布尔值	列表项是否被允许滑动显示选项按钮。默认：true。

实例

HTML 代码：

```
<html ng-app="ionicApp">

  <head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">

    <title>Ionic List Directive</title>

    <link href="http://www.runoob.com/static/ionic/css/ionic.min.css" rel="stylesheet">

    <script src="http://www.runoob.com/static/ionic/js/ionic.bundle.min.js"></script>

  </head>

  <body ng-controller="MyCtrl">

    <ion-header-bar class="bar-positive">

      <div class="buttons">

        <button class="button button-icon icon ion-ios-minus-outline"

          ng-click="data.showDelete = !data.showDelete; data.showReorder = false"></button>

      </div>

      <h1 class="title">Ionic Delete/Option Buttons</h1>

      <div class="buttons">

        <button class="button" ng-click="data.showDelete = false; data.showReorder = !data.showReorder">

          Reorder

        </button>

      </div>

    </ion-header-bar>

  </body>

</html>
```

```
</div>
```

```
</ion-header-bar>
```

```
<ion-content>
```

```
<!-- The list directive is great, but be sure to also checkout the collection repeat directive when scrolling through large lists -->
```

```
<ion-list show-delete="data.showDelete" show-reorder="data.showReorder">
```

```
<ion-item ng-repeat="item in items"
```

```
  item="item"
```

```
  href="#/item/{{item.id}}" class="item-remove-animate">
```

```
  Item {{ item.id }}
```

```
<ion-delete-button class="ion-minus-circled"
```

```
  ng-click="onItemDelete(item)">
```

```
</ion-delete-button>
```

```
<ion-option-button class="button-assertive"
```

```
  ng-click="edit(item)">
```

```
  Edit
```

```
</ion-option-button>
```

```
<ion-option-button class="button-calm"
```

```
  ng-click="share(item)">
```

```
  Share
```

```
</ion-option-button>
```

```
<ion-reorder-button class="ion-navicon" on-reorder="moveItem(item, $fromIndex, $toIndex)"></ion-reorder-button>
```

```
</ion-item>
```

```
</ion-list>
```

```
</ion-content>
```

```
</body>
```

```
</html>
```

## CSS 代码

```
body {  
  
    cursor: url('http://www.runoob.com/try/demo_source/finger.png'), auto;  
  
}
```

## JavaScript 代码

```
angular.module('ionicApp', ['ionic'])  
  
.controller('MyCtrl', function($scope) {  
  
    $scope.data = {  
  
        showDelete: false  
  
    };  
  
    $scope.edit = function(item) {  
  
        alert('Edit Item: ' + item.id);  
  
    };  
  
    $scope.share = function(item) {  
  
        alert('Share Item: ' + item.id);  
  
    };  
  
    $scope.moveItem = function(item, fromIndex, toIndex) {  
  
        $scope.items.splice(fromIndex, 1);  
  
        $scope.items.splice(toIndex, 0, item);  
  
    };  
  
    $scope.onItemDelete = function(item) {  
  
        $scope.items.splice($scope.items.indexOf(item), 1);  
  
    };  
  
});
```



```
$scope.items = [  
  
  { id: 0 },  
  
  { id: 1 },  
  
  { id: 2 },  
  
  { id: 3 },  
  
  { id: 4 },  
  
  { id: 5 },  
  
  { id: 6 },  
  
  { id: 7 },  
  
  { id: 8 },  
  
  { id: 9 },  
  
  { id: 10 },  
  
  { id: 11 },  
  
  { id: 12 },  
  
  { id: 13 },  
  
  { id: 14 },  
  
  { id: 15 },  
  
  { id: 16 },  
  
  { id: 17 },  
  
  { id: 18 },  
  
  { id: 19 },  
  
  { id: 20 },  
  
  { id: 21 },  
  
  { id: 22 },  
  
  { id: 23 },  
  
  { id: 24 },  
  
  { id: 25 },  
  
  { id: 26 },  
  
  { id: 27 },  
  
  { id: 28 },  
  
  { id: 29 },  
  
  { id: 30 },
```

```
{ id: 31 },

{ id: 32 },

{ id: 33 },

{ id: 34 },

{ id: 35 },

{ id: 36 },

{ id: 37 },

{ id: 38 },

{ id: 39 },

{ id: 40 },

{ id: 41 },

{ id: 42 },

{ id: 43 },

{ id: 44 },

{ id: 45 },

{ id: 46 },

{ id: 47 },

{ id: 48 },

{ id: 49 },

{ id: 50 }

];

});
```

[尝试一下 »](#)

[☐ ionic 头部和底部](#)

[ionic 加载动作 ☐](#)

[☐ 点我分享笔记](#)

[反馈/建议](#)

# ionic 加载动作

`$ionicLoading` 是 `ionic` 默认的一个加载交互效果。里面的内容也是可以在模板里面修改。

## 用法

```
angular.module('LoadingApp', ['ionic'])

.controller('LoadingCtrl', function($scope, $ionicLoading) {

  $scope.show = function() {

    $ionicLoading.show({

      template: 'Loading...'

    });

  };

  $scope.hide = function(){

    $ionicLoading.hide();

  };

});
```

## 方法

显示一个加载效果。

```
show(opts)
```

参数	类型	详情
opts	object	<p><b>loading</b>指示器的选项。可用属性：</p> <p><code>{string=} template</code> 指示器的<b>html</b>内容。</p> <p><code>{string=} templateUrl</code> 一个加载<b>html</b>模板的<b>url</b>作为指示器的内容。</p> <p><code>{boolean=} noBackdrop</code> 是否隐藏背景。默认情况下它会显示。</p> <p><code>{number=} delay</code> 指示器延迟多少毫秒显示。默认为不延迟。</p> <p><code>{number=} duration</code> 等待多少毫秒后自动隐藏指示器。默认情况下，指示器会一直显示，直到触发<b><code>.hide()</code></b>。</p>

隐藏一个加载效果。

```
hide()
```

## API

属性	类型	详情
<code>delegate-handle</code> (可选)	字符串	该句柄定义带有\$ionicListDelegate的列表。
<code>show-delete</code> (可选)	布尔值	列表项的删除按钮当前是显示还是隐藏。
<code>show-reorder</code> (可选)	布尔值	列表项的排序按钮当前是显示还是隐藏。
<code>can-swipe</code> (可选)	布尔值	列表项是否被允许滑动显示选项按钮。默认： <code>true</code> 。

## 实例

### HTML 代码：

```
<html ng-app="ionicApp">

  <head>

    <meta charset="utf-8">

    <meta name="viewport" content="initial-scale=1, maximum-scale=1, user-scalable=no, width=device-width">

    <title>Ionic Modal</title>

    <link href="http://www.runoob.com/static/ionic/css/ionic.min.css" rel="stylesheet">

    <script src="http://www.runoob.com/static/ionic/js/ionic.bundle.min.js"></script>

  </head>

  <body ng-controller="AppCtrl">

    <ion-view title="Home">

      <ion-header-bar>

        <h1 class="title">The Stooges</h1>

      </ion-header-bar>

      <ion-content has-header="true">

        <ion-list>

          <ion-item ng-repeat="stooge in stooges" href="#">{{stooge.name}}</ion-item>

        </ion-list>

      </ion-content>

    </ion-view>

  </body>

</html>
```

```
</ion-view>

</body>

</html>
```

## JavaScript 代码

```
angular.module('ionicApp', ['ionic'])

.controller('AppCtrl', function($scope, $timeout, $ionicLoading) {

    // Setup the loader

    $ionicLoading.show({

        content: 'Loading',

        animation: 'fade-in',

        showBackdrop: true,

        maxWidth: 200,

        showDelay: 0

    });

    // Set a timeout to clear loader, however you would actually call the $ionicLoading.hide(); method whenever everyth
    ing is ready or loaded.

    $timeout(function () {

        $ionicLoading.hide();

        $scope.stooges = [{name: 'Moe'}, {name: 'Larry'}, {name: 'Curly'}];

    }, 2000);

});
```

尝试一下 »

## \$ionicLoadingConfig

设置加载的默认选项:

用法:

```
var app = angular.module('myApp', ['ionic'])

app.constant('$ionicLoadingConfig', {

  template: '默认加载模板.....'

});

app.controller('AppCtrl', function($scope, $ionicLoading) {

  $scope.showLoading = function() {

    $ionicLoading.show(); //配置选项在 $ionicLoadingConfig 设置

  };

});
```

[❏ ionic 列表操作](#)

[ionic 模态窗口](#) ❏

[❏ 点我分享笔记](#)

[反馈/建议](#)

Copyright © 2013-2018 菜鸟教程 [runoob.com](#) All Rights Reserved. 备案号：闽ICP备15012807号-1



[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

[❏ ionic 加载动作](#)

[ionic 导航](#) ❏

# ionic 模态窗口

## \$ionicModal

**\$ionicModal** 可以遮住用户主界面的内容框。

你可以在你的 **index** 文件或者是其他文件内嵌入以下代码(里面的代码可以根据你自己的业务场景相应的改变)。

```
<script id="my-modal.html" type="text/ng-template">

  <ion-modal-view>

    <ion-header-bar>

      <h1 class="title">My Modal title</h1>

    </ion-header-bar>

    <ion-content>

      Hello!
```

```
</ion-content>

</ion-modal-view>

</script>
```

然后你就可以在你的 **Controller** 里面的注入 **\$ionicModal** 。然后调用你刚刚写入的模板，进行初始化操作。就像下面的代码：

```
angular.module('testApp', ['ionic'])

.controller('MyController', function($scope, $ionicModal) {

  $ionicModal.fromTemplateUrl('my-modal.html', {

    scope: $scope,

    animation: 'slide-in-up'

  }).then(function(modal) {

    $scope.modal = modal;

  });

  $scope.openModal = function() {

    $scope.modal.show();

  };

  $scope.closeModal = function() {

    $scope.modal.hide();

  };

  //Cleanup the modal when we're done with it!

  $scope.$on('$destroy', function() {

    $scope.modal.remove();

  });

  // Execute action on hide modal

  $scope.$on('modal.hidden', function() {

    // Execute action

  });

  // Execute action on remove modal

  $scope.$on('modal.removed', function() {

    // Execute action

  });

});
```

## 方法

`fromTemplate(templateString, options)`

参数	类型	详情
templateString	字符串	模板的字符串作为模态窗口的内容。
options	对象	options 会传递到 <code>ionicModal#initialize</code> 方法中。

返回: 对象, 一个ionicModal控制器的实例。

`fromTemplateUrl(templateUrl, options)`

参数	类型	详情
templateUrl	字符串	载入模板的url。
options	对象	通过ionicModal#initialize方法传递对象。

返回: `promise`对象。`Promises`对象是CommonJS工作组提出的一种规范，目的是为异步编程提供统一接口。

## ionicModal

由`$ionicModal`服务实例化。

提示: 当你完成每个模块清除时，确保调用`remove()`方法，以避免内存泄漏。

注意: 一个模块从它的初始范围广播出 `'modal.shown'` 和 `'modal.hidden'`，把自身作为一个参数来传递。

## 方法

`initialize(可选)`

创建一个新的模态窗口控制器示例。

参数	类型	详情
options	对象	一个选项对象具有一下属性：  <code>{object=}</code> 范围 子类的范围。默认：创建一个 <code>\$rootScope</code> 子类。  <code>{string=}</code> 动画 带有显示或隐藏的动画。默认: <code>'slide-in-up'</code>  <code>{boolean=}</code> 第一个输入框获取焦点 当显示时，模态窗口的第一个输入元素是否自动获取焦点。默认: <code>false</code> 。  <code>{boolean=}</code> <code>backdropClickToClose`</code> 点击背景时是否关闭模态窗口。默认: <code>true</code> 。

`show()`

显示模态窗口实例

返回值: `promise` `promise`对象,在模态窗口完成动画后得到解析

`hide()`

隐藏模态窗口。

返回值: `promise` `promise`对象,在模态窗口完成动画后得到解析



```
remove()
```

从 DOM 中移除模态窗口实例并清理。

返回值: promise **promise**对象,在模态窗口完成动画后得到解析

```
isShown()
```

返回: 布尔值, 用于判断模态窗口是否显示。

## 实例

### HTML 代码

```
<html ng-app="ionicApp">

<head>

  <meta charset="utf-8">

  <meta name="viewport" content="initial-scale=1, maximum-scale=1, user-scalable=no, width=device-width">

  <title>菜鸟教程(runoob.com)</title>

  <link href="http://www.runoob.com/static/ionic/css/ionic.min.css" rel="stylesheet">

  <script src="http://www.runoob.com/static/ionic/js/ionic.bundle.min.js"></script>

</head>

<body ng-controller="AppCtrl">

  <ion-header-bar class="bar-positive">

    <h1 class="title">Contacts</h1>

    <div class="buttons">

      <button class="button button-icon ion-compose" ng-click="modal.show()">

    </button>

    </div>

  </ion-header-bar>

  <ion-content>

    <ion-list>

      <ion-item ng-repeat="contact in contacts">

        {{contact.name}}

      </ion-item>

    </ion-list>

  </ion-content>

</body>

</html>
```

```
</ion-content>
```

```
<script id="templates/modal.html" type="text/ng-template">
```

```
<ion-modal-view>
```

```
<ion-header-bar class="bar bar-header bar-positive">
```

```
<h1 class="title">New Contact</h1>
```

```
<button class="button button-clear button-primary" ng-click="modal.hide()">Cancel</button>
```

```
</ion-header-bar>
```

```
<ion-content class="padding">
```

```
<div class="list">
```

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

```
<span class="input-label">First Name</span>
```

```
<input ng-model="newUser.firstName" type="text">
```

```
</label>
```

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

```
<span class="input-label">Last Name</span>
```

```
<input ng-model="newUser.lastName" type="text">
```

```
</label>
```

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

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

```
<input ng-model="newUser.email" type="text">
```

```
</label>
```

```
<button class="button button-full button-positive" ng-click="createContact(newUser)">Create</button>
```

```
</div>
```

```
</ion-content>
```

```
</ion-modal-view>
```

```
</script>
```

```
</body>
```

```
</html>
```

```
body {  
  
  cursor: url('http://www.runoob.com/try/demo_source/finger.png'), auto;  
  
}
```

## JavaScript 代码

```
angular.module('ionicApp', ['ionic'])  
  
.controller('AppCtrl', function($scope, $ionicModal) {  
  
  $scope.contacts = [  
  
    { name: 'Gordon Freeman' },  
  
    { name: 'Barney Calhoun' },  
  
    { name: 'Lamarr the Headcrab' },  
  
  ];  
  
  $ionicModal.fromTemplateUrl('templates/modal.html', {  
  
    scope: $scope  
  
  }).then(function(modal) {  
  
    $scope.modal = modal;  
  
  });  
  
  $scope.createContact = function(u) {  
  
    $scope.contacts.push({ name: u.firstName + ' ' + u.lastName });  
  
    $scope.modal.hide();  
  
  };  
  
});
```

尝试一下 »

[☐ ionic 加载动作](#)

[ionic 导航 ☐](#)

[☐ 点我分享笔记](#)

## ionic 导航

### ion-nav-view

当用户在你的app中浏览时，ionic能够检测到浏览历史。通过检测浏览历史，实现向左或向右滑动时可以正确转换视图。

采用AngularUI路由器模块等应用程序接口可以分为不同的`$state`(状态)。Angular的核心为路由服务，URLs可以用来控制视图。

AngularUI路由提供了一个更强大的状态管理，即状态可以被命名，嵌套，以及合并视图，允许一个以上模板呈现在同一个页面。

此外，每个状态无需绑定到一个URL，并且数据可以更灵活地推送到每个状态。

以下实例中，我们将创建一个应用程序中包含不同状态的导航视图。

我们的标记中选择`ionNavView`作为顶层指令。显示一个页眉栏我们用 `ionNavBar` 指令通过导航更新。

接下来，我们需要设置我们的将渲染的状态值。

```
var app = angular.module('myApp', ['ionic']);

app.config(function($stateProvider) {

  $stateProvider

    .state('index', {

      url: '/',

      templateUrl: 'home.html'

    })

    .state('music', {

      url: '/music',

      templateUrl: 'music.html'

    });

});
```

再打开应用，`$stateProvider` 会查询url, 看是否匹配 `index` 状态值, 再加载`index.html`到`<ion-nav-view>`。

页面加载都是通过URLs配置的。在Angular中创建模板最简单的方式就是直接将他放到html模板文件中并且用`<script type="text/ng-template">` 包含。

所以这也是一种方式将`Home.html`加入到我们的APP中来:

```
<script id="home" type="text/ng-template">
```

```
<!-- ion-view 标题将显示在 navbar 中 -->

<ion-view view-title="Home">

  <ion-content ng-controller="HomeCtrl">

    <!-- The content of the page -->

    <a href="#/music">Go to music page!</a>

  </ion-content>

</ion-view>

</script>
```

尝试一下 »

这是一个很好的方法，因为模板会很快的加载并被缓存就不同通过网络再去加载。

## 缓存

通常情况下，视图都被缓存了能提升性能。当跳出视图时，他的元素被保留在Dom中，并且它的作用域也从 **\$watch** 中移除。

当我们跳到一个已经被缓存了的视图，视图会被激活，它的作用域被重新连接上，Dom中也保存了他的元素。这也允许保持以前的视图滚动位置。

缓存也可以通过很多方式来开启和关闭的。默认**ionic**将最大缓存页面数为**10**个，并且这并不是唯一可以定制的，应用程序可以显式状态来设置视图应不应该被缓存。

注意，因为我们是缓存这些视图，我们没有破坏作用域。相反，它的作用域也从 **\$watch** 中移除。

因为接下来的观看作用域并没有被摧毁和重建，控制器也没被再次加载。如果**app/controller**需要知道什么时候进入或离开一个视图，再视图事件从 **ion View** 作用内发出，例如 **\$ionicView.enter**，可能是有用的。

## 全局禁用缓存

**\$ionicConfigProvider** 可以用于设置最大允许缓存的视图数量，通过设置为**0**来禁用所有缓存。

```
$ionicConfigProvider.views.maxCache(0);
```

## 在 STATE PROVIDER中禁用缓存

```
$stateProvider.state('myState', {

  cache: false,

  url : '/myUrl',

  templateUrl : 'my-template.html'

})
```

## 通过属性禁用缓存

```
<ion-view cache-view="false" view-title="My Title!">

  ...

</ion-view>
```

AngularUI 路由

请访问 [AngularUI Router's docs](#) 了解更多。

API

属性	类型	详情
name (可选)	字符串	一个视图的名字。这个名字应该是在相同的状态下其他视图中唯一的。你可以在不同的状态中有相同名称的视图。欲了解详细信息，查看ui-router的 <a href="#">ui-view 文档</a> 。

ion-view

隶属于ionNavController。 一个内容的容器，用于展示视图或导航栏信息。

下面是一个带有"我的页面"标题的导航栏载入页面的例子。

```
<ion-nav-bar></ion-nav-bar>

<ion-nav-view class="slide-left-right">

  <ion-view title="我的页面">

    <ion-content>

      你好!

    </ion-content>

  </ion-view>

</ion-nav-view>
```

API

属性	类型	详情
title (可选)	字符串	显示在父ionNavController的标题。
hide-back-button (可选)	布尔值	默认情况下，是否在父ionNavController隐藏后退按钮。
hide-nav-bar (可选)	布尔值	默认情况下，是否隐藏父ionNavController。

ion-nav-bar

创建一个顶部工具栏，当程序状态改变时更新。

用法

```
<body ng-app="starter">

  <!-- 当我们浏览时，导航栏会随之更新。 -->

  <ion-nav-bar class="bar-positive nav-title-slide-ios7">

    </ion-nav-bar>
```

```
<!-- 初始化时渲染视图模板 -->

<ion-nav-view></ion-nav-view>

</body>
```

API

属性	类型	详情
<code>delegate-handle</code> (可选)	字符串	该句柄用\$ionicNavBarDelegate标识此导航栏。
<code>align-title</code> (可选)	字符串	导航栏标题对齐的位置。可用: 'left', 'right', 'center'。默认为 'center'。

ion-nav-buttons

隶属于ionNaView  
在ionView内的ionNavBar上用ionNavButtons设置按钮。  
你设置的任何按钮都将被放置在导航栏的相应位置，当用户离开父视图时会被销毁。

用法

```
<ion-nav-bar>

</ion-nav-bar>

<ion-nav-view>

  <ion-view>

    <ion-nav-buttons side="left">

      <button class="button" ng-click="doSomething()">

        我是一个在导航栏左侧的按钮！

      </button>

    </ion-nav-buttons>

    <ion-content>

      这里是一些内容！

    </ion-content>

  </ion-view>

</ion-nav-view>
```

API

属性	类型	详情
<code>side</code>	字符串	在父ionNavBar中按钮放置的位置。 可用: 'left' 或 'right'。

ion-nav-back-button

在一个ionNavBar中创建一个按钮。

当用户在当前导航能够后退时，将显示后退按钮。

## 用法

默认按钮动作：

```
<ion-nav-bar>

  <ion-nav-back-button class="button-clear">

    <i class="ion-arrow-left-c"></i> 后退

  </ion-nav-back-button>

</ion-nav-bar>
```

自定义点击动作，用 `$ionicNavBarDelegate`：

```
<ion-nav-bar ng-controller="MyCtrl">

  <ion-nav-back-button class="button-clear"

    ng-click="canGoBack && goBack()">

    <i class="ion-arrow-left-c"></i> 后退

  </ion-nav-back-button>

</ion-nav-bar>
```

```
function MyCtrl($scope, $ionicNavBarDelegate) {

  $scope.goBack = function() {

    $ionicNavBarDelegate.back();

  };

}
```

在后退按钮上显示上一个标题，也用 `$ionicNavBarDelegate`。

```
<ion-nav-bar ng-controller="MyCtrl">

  <ion-nav-back-button class="button-icon">

    <i class="icon ion-arrow-left-c"></i>{{getPreviousTitle() || 'Back'}}

  </ion-nav-back-button>

</ion-nav-bar>
```

```
function MyCtrl($scope, $ionicNavBarDelegate) {
```



```
$scope.getPreviousTitle = function() {  
  
    return $ionicNavBarDelegate.getPreviousTitle();  
  
};  
  
}
```

## nav-clear

**nav-clear** 一个当点击视图上的元素时用到的属性指令，比如一个 `<a href>` 或者一个 `<button ui-sref>`。

当点击时，**nav-clear** 将会导致给定的元素，禁止下一个视图的转换。这个指令很有用，比如，侧栏菜单内的链接。

### 用法

下面是一个侧栏菜单内添加了 **nav-clear** 指令的一个链接。点击该链接将禁用视图间的任何动画。

```
<a nav-clear menu-close href="#/home" class="item">首页</a>
```

## ion-nav-title

**ion-nav-title** 用于设置 **ion-view** 模板中的标题。

### 用法

```
<ion-nav-bar>  
  
</ion-nav-bar>  
  
<ion-nav-view>  
  
  <ion-view>  
  
    <ion-nav-title>  
  
        
  
    </ion-nav-title>  
  
    <ion-content>  
  
      Some super content here!  
  
    </ion-content>  
  
  </ion-view>  
  
</ion-nav-view>
```

## nav-transition

设置使用的过渡类型，可以是：**ios**, **android**, 和 **none**。

### 用法

```
<a nav-transition="none" href="#/home">Home</a>
```

## nav-direction

设置导航视图中过渡动画的方向，可以是forward, back, enter, exit, swap。

### 用法

```
<a nav-direction="forward" href="#/home">Home</a>
```

## \$ionicNavBarDelegate

授权控制 ion-nav-bar 指令。

### 用法

```
<body ng-controller="MyCtrl">

  <ion-nav-bar>

    <button ng-click="setNavTitle('banana')">

      Set title to banana!

    </button>

  </ion-nav-bar>

</body>
```

```
function MyCtrl($scope, $ionicNavBarDelegate) {

  $scope.setNavTitle = function(title) {

    $ionicNavBarDelegate.title(title);

  }

}
```

### 方法

```
align([direction])
```

在浏览历史中后退。

参数	类型	详情
event (可选)	DOMEvent	事件对象（如来自点击事件）

```
align([direction])
```

带有按钮的标题对齐到指定的方向。

参数	类型	详情
<code>direction</code> (可选)	字符串	标题文字对齐的方向。可用: 'left', 'right', 'center'。默认: 'center'。

返回值: 布尔值， 后退按钮是否显示。

```
showBar(show)
```

设置或获取 `ion-nav-bar` 是否显示。

参数	类型	详情
<code>show</code>	布尔值	导航栏是否显示。

返回值: 布尔值， 导航栏是否显示。

```
showBackButton([show])
```

设置或获取 `ion-nav-back-button` 是否显示。

参数	类型	详情
<code>show</code> (可选)	布尔值	是否显示后退按钮

```
title(title)
```

为`ion-nav-bar`设置标题。

参数	类型	详情
<code>title</code>	字符串	显示新标题。

## \$ionicHistory

\$ionicHistory 用于跟踪用户在 `app` 内的浏览记录。

### 方法

```
viewHistory()
```

用于查看历史记录。

```
currentView()
```

`app` 的当前视图。

```
currentHistoryId()
```

历史堆栈的 `ID`，是当前视图的父类容器。

currentTitle([val])

获取或设置当前视图的标题。

backView()

返回上次浏览的视图。

backTitle()

获取上次浏览的视图的标题。

forwardView()

返回历史堆栈中当前视图的上一个视图。

currentStateName()

返回当前状态名。

goBack([backCount])

app 回退视图，如果回退的视图存在。

[ionic 模态窗口](#)

[ionic 平台](#)

[点我分享笔记](#)

[反馈/建议](#)

Copyright © 2013-2018 菜鸟教程 [runoob.com](#) All Rights Reserved. 备案号：闽ICP备15012807号-1



[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

[ionic 导航](#)

[ionic 浮动框](#)

ionic 平台

\$ionicPlatform

\$ionicPlatform 用来检测当前的平台，以及诸如在PhoneGap/Cordova中覆盖Android后退按钮。

方法

```
onHardwareBackButton(callback)
```

有硬件的后退按钮的平台，可以用这种方法绑定到它。

参数	类型	详情
callback	function	当该事件发生时，触发回调函数。

```
offHardwareBackButton(callback)
```

移除后退按钮的监听事件。

参数	类型	详情
callback	function	最初绑定的监视器函数。

```
registerBackButtonAction(callback, priority, [actionId])
```

注册硬件后退按钮动作。当点击按钮时，只有一个动作会执行，因此该方法决定了注册的后退按钮动作具有最高的优先级。

例如，如果一个上拉菜单已经显示，后退按钮应该关闭上拉菜单，而不是返回一个页面视图或关闭一个打开的模型。

参数	类型	详情
callback	function	当点击返回按钮时触发，如果该监视器具有最高的优先级。
priority	number	仅最高优先级的会执行。
actionId (可选)	*	该id指定这个动作。默认：一个随机且唯一的id。

返回值: 函数， 一个被触发的函数，将会注销 backButtonAction。

```
ready([callback])
```

设备准备就绪，则触发一个回调函数。

参数	类型	详情
callback (可选)	function=	触发的函数。

返回: promise对象, 对象被构造 成功后得到解析。

ionic 导航

ionic 浮动框

点我分享笔记

反馈/建议

## ionic 浮动框

### \$ionicPopover

\$ionicPopover 是一个可以浮在app内容上的一个视图框。

可以实现以下功能点：

在当前页面显示更多信息。

选择一些工具或配置。

在页面提供一个操作列表。

### 方法

```
fromTemplate(templateString, options)
```

或

```
fromTemplateUrl(templateUrl, options)
```

参数说明：

templateString: 模板字符串。

templateUrl: 载入的模板 URL。

options: 初始化选项。

### 实例

#### HTML 代码部分

```
<p>
<button ng-click="openPopover($event)">打开浮动框</button>
</p>
<script id="my-popover.html" type="text/ng-template">
<ion-popover-view>
<ion-header-bar>
<h1 class="title">我的浮动框标题</h1>
</ion-header-bar>
<ion-content>
Hello!
</ion-content>
</ion-popover-view>
</script>
```

#### fromTemplateUrl 方法

```
angular.module('ionicApp', ['ionic'])
.controller(' AppCtrl', ['$scope', '$ionicPopover', '$timeout', function($scope, $ionicPopover, $timeout){
  $scope.popover = $ionicPopover.fromTemplateUrl('my-popover.html', {
    scope: $scope
  });
  // .fromTemplateUrl() 方法
  $ionicPopover.fromTemplateUrl('my-popover.html', {
    scope: $scope
  }).then(function(popover) {
    $scope.popover = popover;
```

```

});
$scope.openPopover = function($event) {
  $scope.popover.show($event);
};
$scope.closePopover = function() {
  $scope.popover.hide();
};
// 清除浮动框
$scope.$on('$destroy', function() {
  $scope.popover.remove();
});
// 在隐藏浮动框后执行
$scope.$on('popover.hidden', function() {
  // 执行代码
});
// 移除浮动框后执行
$scope.$on('popover.removed', function() {
  // 执行代码
});
}]]

```

尝试一下 »

我们也可以把模板当作一个字符串，使用 `.fromTemplate()` 方法来实现弹框：

## fromTemplate 方法

```

angular.module('ionicApp', ['ionic'])
.controller( 'AppCtrl', ['$scope', '$ionicPopover', '$timeout', function($scope, $ionicPopover, $timeout){
  $scope.popover = $ionicPopover.fromTemplateUrl('my-popover.html', {
    scope: $scope
  });
  // .fromTemplate() 方法
  var template = '<ion-popover-view><ion-header-bar> <h1 class="title">我的浮动框标题</h1> </ion-header-bar> <ion-con
tent> Hello! </ion-content></ion-popover-view>';
  $scope.popover = $ionicPopover.fromTemplate(template, {
    scope: $scope
  });
  $scope.openPopover = function($event) {
    $scope.popover.show($event);
  };
  $scope.closePopover = function() {
    $scope.popover.hide();
  };
  // 清除浮动框
  $scope.$on('$destroy', function() {
    $scope.popover.remove();
  });
  // 在隐藏浮动框后执行
  $scope.$on('popover.hidden', function() {
    // 执行代码
  });
  // 移除浮动框后执行
  $scope.$on('popover.removed', function() {
    // 执行代码
  });
}]]

```

尝试一下 »

□ ionic 平台

ionic 对话框 □

□ 点我分享笔记

反馈/建议

[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)[☐ ionic 浮动框](#)[ionic 滚动条 ☐](#)

## ionic 对话框

### \$ionicPopup

ionic 对话框服务允许程序创建、显示弹出窗口。

\$ionicPopup 提供了3个方法: `alert()`, `prompt()`, 以及 `confirm()`。

### 实例

#### HTML 代码

```
<body class="padding" ng-controller="PopupCtrl">

  <button class="button button-dark" ng-click="showPopup()">

    弹窗显示

  </button>

  <button class="button button-primary" ng-click="showConfirm()">

    确认对话框

  </button>

  <button class="button button-positive" ng-click="showAlert()">

    警告框

  </button>

  <script id="popup-template.html" type="text/ng-template">

    <input ng-model="data.wifi" type="text" placeholder="Password">

  </script>

</body>
```

#### JavaScript 代码

```
angular.module('mySuperApp', ['ionic'])

.controller('PopupCtrl',function($scope, $ionicPopup, $timeout) {
```



```
// Triggered on a button click, or some other target

$scope.showPopup = function() {

    $scope.data = {}

    // 自定义弹窗

    var myPopup = $ionicPopup.show({

        template: '<input type="password" ng-model="data.wifi">',

        title: 'Enter Wi-Fi Password',

        subTitle: 'Please use normal things',

        scope: $scope,

        buttons: [

            { text: 'Cancel' },

            {

                text: '<b>Save</b>',

                type: 'button-positive',

                onTap: function(e) {

                    if (!$scope.data.wifi) {

                        // 不允许用户关闭，除非输入 wifi 密码

                        e.preventDefault();

                    } else {

                        return $scope.data.wifi;

                    }

                }

            }

        ],

    });

    myPopup.then(function(res) {

        console.log('Tapped!', res);

    });

    $timeout(function() {

        myPopup.close(); // 3秒后关闭弹窗

    }, 3000);

}
```

```
};

// confirm 对话框

$scope.showConfirm = function() {

  var confirmPopup = $ionicPopup.confirm({

    title: 'Consume Ice Cream',

    template: 'Are you sure you want to eat this ice cream?'

  });

  confirmPopup.then(function(res) {

    if(res) {

      console.log('You are sure');

    } else {

      console.log('You are not sure');

    }

  });

};

// alert（警告）对话框

$scope.showAlert = function() {

  var alertPopup = $ionicPopup.alert({

    title: 'Don\'t eat that!',

    template: 'It might taste good'

  });

  alertPopup.then(function(res) {

    console.log('Thank you for not eating my delicious ice cream cone');

  });

};

});
```

尝试一下 »

[☐ ionic 浮动框](#)

[ionic 滚动条 ☐](#)

[☐ 点我分享笔记](#)



# ionic 滚动条

## ion-scroll

ion-scroll 用于创建一个可滚动的容器。

### 用法

```
<ion-scroll

  [delegate-handle=""]

  [direction=""]

  [paging=""]

  [on-refresh=""]

  [on-scroll=""]

  [scrollbar-x=""]

  [scrollbar-y=""]

  [zooming=""]

  [min-zoom=""]

  [max-zoom=""]>

  ...

</ion-scroll>
```

### API

属性	类型	详情
delegate-handle (可选)	字符串	该句柄利用\$ionicScrollDelegate指定滚动视图。
direction (可选)	字符串	滚动的方向。 'x' 或 'y'。 默认 'y'。
paging (可选)	布尔值	分页是否滚动。

属性	类型	详情
<code>on-refresh</code> (可选)	表达式	调用下拉刷新， 由ionRefresher触发。
<code>on-scroll</code> (可选)	表达式	当用户滚动时触发。
<code>scrollbar-x</code> (可选)	布尔值	是否显示水平滚动条。默认为false。
<code>scrollbar-y</code> (可选)	布尔值	是否显示垂直滚动条。默认为true。
<code>zooming</code> (可选)	布尔值	是否支持双指缩放。
<code>min-zoom</code> (可选)	整数	允许的最小缩放量（默认为0.5）
<code>max-zoom</code> (可选)	整数	允许的最大缩放量（默认为3）

### 实例

#### HTML 代码

```
<ion-scroll zooming="true" direction="xy" style="width: 500px; height: 500px">

  <div style="width: 5000px; height: 5000px; background: url('http://www.runoob.com/try/demo_source/Europe_geologic
al_map-en.jpg') repeat"></div>

</ion-scroll>
```

#### CSS 代码

```
body {

  cursor: url('http://www.runoob.com/try/demo_source/finger.png'), auto;

}
```

#### JavaScript 代码

```
angular.module('ionicApp', ['ionic']);
```

尝试一下 »

### ion-infinite-scroll

当用户到达页脚或页脚附近时，ionInfiniteScroll指令允许你调用一个函数。

当用户滚动的距离超出底部的内容时，就会触发你指定的on-infinite。

### 用法

```
<ion-content ng-controller="MyController">

  <ion-infinite-scroll

    on-infinite="loadMore()"

    distance="1%">

  </ion-infinite-scroll>

</ion-content>
```

```
function MyController($scope, $http) {

  $scope.items = [];

  $scope.loadMore = function() {

    $http.get('/more-items').success(function(items) {

      useItems(items);

      $scope.$broadcast('scroll.infiniteScrollComplete');

    });

  };

  $scope.$on('stateChangeSuccess', function() {

    $scope.loadMore();

  });

}
```

当没有更多数据加载时，就可以用一个简单的方法阻止无限滚动，那就是angular的ng-if 指令：

```
<ion-infinite-scroll

  ng-if="moreDataCanBeLoaded()"

  icon="ion-loading-c"

  on-infinite="loadMoreData()">

</ion-infinite-scroll>
```

API

属性	类型	详情
on-infinite	表达式	当滚动到底部时触发的事件。

属性	类型	详情
distance (可选)	字符串	从底部滚动到触发on-infinite表达式的距离。默认: 1%。
icon (可选)	字符串	当加载时显示的图标。默认: 'ion-loading-d'。

## \$ionicScrollDelegate

授权控制滚动视图（通过ion-content 和 ion-scroll指令创建）。

该方法直接被\$ionicScrollDelegate服务触发，来控制所有滚动视图。用 \$getByHandle方法控制特定的滚动视图。

### 用法

```
<body ng-controller="MainCtrl">

  <ion-content>

    <button ng-click="scrollTop()">滚动到顶部!</button>

  </ion-content>

</body>
```

```
function MainCtrl($scope, $ionicScrollDelegate) {

  $scope.scrollTop = function() {

    $ionicScrollDelegate.scrollTop();

  };

}
```

### 方法

```
resize()
```

告诉滚动视图重新计算它的容器大小。

```
scrollTop([shouldAnimate])
```

参数	类型	详情
shouldAnimate (可选)	布尔值	是否应用滚动动画。

```
scrollBottom([shouldAnimate])
```

参数	类型	详情
----	----	----

参数	类型	详情
shouldAnimate (可选)	布尔值	是否应用滚动动画。

☐ ionic 对话框

ionic 侧栏菜单 ☐

☐ 点我分享笔记

反馈/建议

☐

首页 HTML CSS JS 本地书签

☐ ionic 滚动条

ionic 滑动框 ☐

# ionic 侧栏菜单

一个容器元素包含侧边菜单和主要内容。通过把主要内容区域从一边拖动到另一边，来让左侧或右侧的侧栏菜单进行切换。

效果图如下所示：

☰ Slide me

Content

## 用法

要使用侧栏菜单，添加一个父元素<ion-side-menus>，一个中间内容 <ion-side-menu-content>， 和一个或更多 <ion-side-menu> 指令。

<ion-side-menus>

```
<!-- 中间内容 -->

<ion-side-menu-content ng-controller="ContentController">

</ion-side-menu-content>


<!-- 左侧菜单 -->

<ion-side-menu side="left">

</ion-side-menu>


<!-- 右侧菜单 -->

<ion-side-menu side="right">

</ion-side-menu>

</ion-side-menus>
```

```
function ContentController($scope, $ionicSideMenuDelegate) {

    $scope.toggleLeft = function() {

        $ionicSideMenuDelegate.toggleLeft();

    };

}
```

尝试一下 »

API

属性	类型	详情
enable-menu-with-back-views (可选)	布尔值	在返回按钮显示时，确认是否启用侧边栏菜单。
delegate-handle	字符串	该句柄用于标识带有\$ionicScrollDelegate的滚动视图。

ion-side-menu-content

一个可见主体内容的容器，同级的一个或多个ionSideMenu 指令。

用法

```
<ion-side-menu-content

    drag-content="true">

</ion-side-menu-content>
```

API



属性	类型	详情
drag-content (可选)	布尔值	内容是否可被拖动。默认为true。

## ion-side-menu

一个侧栏菜单的容器，同级的一个ion-side-menu-content 指令。

### 用法

```
<ion-side-menu

  side="left"

  width="myWidthValue + 20"

  is-enabled="shouldLeftSideMenuBeEnabled()">

</ion-side-menu>
```

### API

属性	类型	详情
side	字符串	侧栏菜单当前在哪一边。可选的值有: 'left' 或 'right'。
is-enabled (可选)	布尔值	该侧栏菜单是否可用。
width (可选)	数值	侧栏菜单应该有多少像素的宽度。默认为275。

## menu-toggle

在一个指定的侧栏中切换菜单。

### 用法

下面是一个在导航栏内链接的例子。点击此链接会自动打开指定的侧栏菜单。

```
<ion-view>

  <ion-nav-buttons side="left">

    <button menu-toggle="left" class="button button-icon icon ion-navicon"></button>

  </ion-nav-buttons>

  ...

</ion-view>
```

## menu-close

关闭当前打开的侧栏菜单。

### 用法

下面是一个在导航栏内链接的例子。点击此链接会自动打开指定的侧栏菜单。

```
<a menu-close href="#/home" class="item">首页</a>
```

## \$ionicSideMenuDelegate

该方法直接触发\$ionicSideMenuDelegate服务，来控制所有侧栏菜单。用\$getByHandle方法控制特定情况下的ionSideMenus。

### 用法

```
<body ng-controller="MainCtrl">

  <ion-side-menus>

    <ion-side-menu-content>

      内容!

      <button ng-click="toggleLeftSideMenu()">

        切换左侧侧栏菜单

      </button>

    </ion-side-menu-content>

    <ion-side-menu side="left">

      左侧菜单!

      <ion-side-menu>

      </ion-side-menu>

    </ion-side-menu>

  </ion-side-menus>

</body>
```

```
function MainCtrl($scope, $ionicSideMenuDelegate) {

  $scope.toggleLeftSideMenu = function() {

    $ionicSideMenuDelegate.toggleLeft();

  };

}
```

### 方法

```
toggleLeft([isOpen])
```

切换左侧侧栏菜单（如果存在）。

参数	类型	详情
isOpen (可选)	布尔值	是否打开或关闭菜单。默认：切换菜单。

```
toggleRight([isOpen])
```

切换右侧侧栏菜单（如果存在）。

参数	类型	详情
isOpen (可选)	布尔值	是否打开或关闭菜单。默认：切换菜单。

```
getOpenRatio()
```

获取打开菜单内容超出菜单宽度的比例。比如，一个宽度为100px的菜单被宽度为50px以50%的比例打开，将会返回一个比例值为0.5。

返回值：浮点 0 表示没被打开，如果左侧菜单处于已打开或正在打开为0 到 1，如果右侧菜单处于已打开或正在打开为0 到-1。

```
isOpen()
```

返回值：布尔值，判断左侧或右侧菜单是否已经打开。

```
isOpenLeft()
```

返回值：布尔值左侧菜单是否已经打开。

```
isOpenRight()
```

返回值：布尔值右侧菜单是否已经打开。

```
canDragContent([canDrag])
```

参数	类型	详情
canDrag (可选)	布尔值	设置是否可以拖动内容打开侧栏菜单。

返回值：布尔值，是否可以拖动内容打开侧栏菜单。

```
$getByHandle(handle)
```

参数	类型	详情
handle	字符串	

例如:

```
$ionicSideMenuDelegate.$getByHandle('my-handle').toggleLeft();
```



# ionic 滑动框

## ion-slide-box

滑动框是一个包含多页容器的组件，每页滑动或拖动切换：

效果图如下：



### 用法

```
<ion-slide-box on-slide-changed="slideHasChanged($index)">

  <ion-slide>

    <div class="box blue"><h1>BLUE</h1></div>

  </ion-slide>

  <ion-slide>

    <div class="box yellow"><h1>YELLOW</h1></div>

  </ion-slide>

  <ion-slide>

    <div class="box pink"><h1>PINK</h1></div>

  </ion-slide>

</ion-slide-box>
```

## API

属性	类型	详情
<code>delegate-handle</code> (可选)	字符串	该句柄用\$ionicSlideBoxDelegate来标识这个滑动框。
<code>does-continue</code> (可选)	布尔值	滑动框是否开启循环滚动。

属性	类型	详情
<b>auto-play</b> (可选)	boolean	设置滑动框是否循环播放，如果 <b>does-continue</b> 为 <b>true</b> ，默认也为 <b>true</b> 。
<b>slide-interval</b> (可选)	数字	等待多少毫秒开始滑动（如果继续则为 <b>true</b> ）。默认为 <b>4000</b> 。
<b>show-pager</b> (可选)	布尔值	滑动框的页面是否显示。
<b>pager-click</b> (可选)	表达式	当点击页面时，触发该表达式（如果 <b>show-pager</b> 为 <b>true</b> ）。传递一个'索引'变量。
<b>on-slide-changed</b> (可选)	表达式	当滑动时，触发该表达式。传递一个'索引'变量。
<b>active-slide</b> (可选)	表达式	将模型绑定到当前滑动框。

## 实例

### HTML 代码



```
<ion-slide-box active-slide="myActiveSlide">

  <ion-slide>

    <div class="box blue"><h1>BLUE</h1></div>

  </ion-slide>

  <ion-slide>

    <div class="box yellow"><h1>YELLOW</h1></div>

  </ion-slide>

  <ion-slide>

    <div class="box pink"><h1>PINK</h1></div>

  </ion-slide>

</ion-slide-box>
```

### CSS 代码

```
.slider {

  height: 100%;

}

.slider-slide {

  color: #000;

  background-color: #fff; text-align: center;
```

```
font-family: "HelveticaNeue-Light", "Helvetica Neue Light", "Helvetica Neue", Helvetica, Arial, "Lucida Grande", sans-serif; font-weight: 300; }

.blue {

    background-color: blue;

}

.yellow {

    background-color: yellow;

}

.pink {

    background-color: pink;

}

.box{

    height:100%;

}

.box h1{

    position:relative; top:50%; transform:translateY(-50%);

}
```

## JavaScript 代码

```
angular.module('ionicApp', ['ionic'])

.controller('SlideController', function($scope) {

    $scope.myActiveSlide = 1;

})
```

尝试一下 »

☐ ionic 侧栏菜单

ionic 加载动画 ☐

☐ 点我分享笔记

[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)[☐ ionic 滑动框](#)[ionic 选项卡栏操作 ☐](#)

## ionic 加载动画

### ion-spinner

`ionSpinner` 提供了许多种旋转加载的动画图标。当你的界面加载时，你就可以呈现给用户相应的加载图标。  
该图标采用的是SVG。

#### 用法

```
<ion-spinner icon="spiral"></ion-spinner>    //默认用法
```

像大部分其他的**ionic**组件一样，**spinner**也可以使用**ionic**的标准颜色命名规则，就像下面这样：

```
<ion-spinner class="spinner-energized"></ion-spinner>
```

### 实例

#### HTML 代码

```
<ion-content scroll="false" class="has-header">

  <p>

    <ion-spinner icon="android"></ion-spinner>

    <ion-spinner icon="ios"></ion-spinner>

    <ion-spinner icon="ios-small"></ion-spinner>

    <ion-spinner icon="bubbles" class="spinner-balanced"></ion-spinner>

    <ion-spinner icon="circles" class="spinner-energized"></ion-spinner>

  </p>

  <p>
```

```
<ion-spinner icon="crescent" class="spinner-royal"></ion-spinner>

<ion-spinner icon="dots" class="spinner-dark"></ion-spinner>

<ion-spinner icon="lines" class="spinner-calm"></ion-spinner>

<ion-spinner icon="ripple" class="spinner-assertive"></ion-spinner>

<ion-spinner icon="spiral"></ion-spinner>

</p>

</ion-content>
```

## CSS 代码

```
body {

  cursor: url('http://www.runob.com/try/demo_source/finger.png'), auto;

}

p {

  text-align: center;

  margin-bottom: 40px !important;

}

.spinner svg {

  width: 19% !important;

  height: 85px !important;

}
```

## JavaScript 代码

```
angular.module('ionicApp', ['ionic'])

.controller('MyCtrl', function($scope) {

});
```

尝试一下 »

效果如下所示:



# Error response

Error code 404.

Message: File not found.

Error code explanation: 404 = Nothing matches the given URI.

[ionic 滑动框](#)

[ionic 选项卡栏操作](#)

[点我分享笔记](#)

反馈/建议



[ionic 加载动画](#)

## ionic 选项卡栏操作

### ion-tabs

**ion-tabs** 是有一组页面选项卡组成的选项卡栏。可以通过点击选项来切换页面。  
对于 iOS，它会出现 在屏幕的底部，Android 会出现在屏幕的顶部(导航栏下面)。

#### 用法

#### 实例

```
<ion-tabs class="tabs-positive tabs-icon-only">
<ion-tab title="首页" icon-on="ion-ios-filing" icon-off="ion-ios-filing-outline">
<!-- 标签 1 内容 -->
</ion-tab>
<ion-tab title="关于" icon-on="ion-ios-clock" icon-off="ion-ios-clock-outline">
<!-- 标签 2 内容 -->
</ion-tab>
<ion-tab title="设置" icon-on="ion-ios-gear" icon-off="ion-ios-gear-outline">
<!-- 标签 3 内容 -->
</ion-tab>
</ion-tabs>
```

尝试一下 »

效果如下所示：

# Error response

Error code 404.

Message: File not found.

Error code explanation: 404 = Nothing matches the given URI.

## API

属性	类型	详情
delegate-handle (可选)	字符串	该句柄用 <i>SionicTabsDelegate</i> 来标识这些选项卡。

## ion-tab

隶属于ionTabs

包含一个选项卡内容。该内容仅存在于被选中的给定选项卡中。

每个ionTab都有自己的浏览历史。

## 用法

```
<ion-tab

    title="Tab!"
```

```
    icon="my-icon"

    href="#/tab/tab-link"

    on-select="onTabSelected()"

    on-deselect="onTabDeselected()">

</ion-tab>
```

API

属性	类型	详情
title	字符串	选项卡的标题。
href (可选)	字符串	但触碰的时候，该选项卡将会跳转的链接。
icon (可选)	字符串	选项卡的图标。如果给定值，它将成为ion-on和ion-off的默认值。
icon-on (可选)	字符串	被选中标签的图标。
icon-off (可选)	字符串	没被选中标签的图标。
badge (可选)	表达式	选项卡上的徽章（通常是一个数字）。
badge-style (可选)	表达式	选项卡上徽章的样式（例，tabs-positive）。
on-select (可选)	表达式	选项卡被选中时触发。
on-deselect (可选)	表达式	选项卡取消选中时触发。
ng-click (可选)	表达式	通常，点击时选项卡会被选中。如果设置了 ng-Click，它将不会被选中。 你可以用\$ionicTabsDelegate.select()来指定切换标签。

\$ionicTabsDelegate

授权控制ionTabs指令。

该方法直接调用\$ionicTabsDelegate服务，控制所有ionTabs指令。用\$getByHandle方法控制具体的ionTabs实例。

用法

```
<body ng-controller="MyCtrl">

  <ion-tabs>
```

```
<ion-tab title="Tab 1">

  你好，标签1!

  <button ng-click="selectTabWithIndex(1)">选择标签2</button>

</ion-tab>

<ion-tab title="Tab 2">你好标签2! </ion-tab>

</ion-tabs>

</body>
```

```
function MyCtrl($scope, $ionicTabsDelegate) {

  $scope.selectTabWithIndex = function(index) {

    $ionicTabsDelegate.select(index);

  }

}
```

方法

```
select(index, [shouldChangeHistory])
```

选择标签来匹配给定的索引。

参数	类型	详情
index	数值	选择标签的索引。
shouldChangeHistory (可选)	布尔值	此选项是否应该加载这个标签的浏览历史（如果存在），并使用，或仅加载默认页面。默认为false。提示： 如果一个ion-nav-view在选项卡里，你可能需要设置它为true。

```
selectedIndex()
```

返回值: 数值, 被选中标签的索引，如 -1。

```
$getByHandle(handle)
```

参数	类型	详情
handle	字符串	

例如:

```
$ionicTabsDelegate.$getByHandle('my-handle').select(0);
```

☐ ionic 加载动画

☐ 点我分享笔记

反馈/建议