# css3 基础

[1. css3 基础 1](#_Toc9737)

[1.1. 伸缩布局携程网首页案例 2](#_Toc205)

[1.2. Css3 兼容性查询 2](#_Toc24122)

[1.3. Web 字体 2](#_Toc15793)

[1.3.1. 字体格式介绍 2](#_Toc13636)

[1.3.2. 字体图标 2](#_Toc21782)

[1.3.3. 字体图标的使用 2](#_Toc18608)

[1.4. 360 官网宣传案例 2](#_Toc156)

[1.4.1. jQuery fullpage 插件的介绍 2](#_Toc15581)

[1.4.2. 案例代码逻辑的编写 2](#_Toc18301)

[1.5. 移动开发的概念 2](#_Toc26403)

[1.5.1. webApp的概念 2](#_Toc4243)

[1.5.2. 混合app的概念 2](#_Toc26191)

[1.5.3. 原生app 的概念 2](#_Toc28003)

## Css3 兼容性查询

通过http://caniuse.com/ 可查询CSS3各特性的支持程度，一般兼容性处理的常见方法是为属性添加私有前缀，如不能解决，应避免使用，无需刻意去处理CSS3的兼容性问题。

## Web 字体

开发人员可以为自已的网页指定特殊的字体，无需考虑用户电脑上是否安装了此特殊字体，从此把特殊字体处理成图片的时代便成为了过去。

支持程度比较好，甚至IE低版本浏览器也能支持。

### 字体格式介绍

不同浏览器所支持的字体格式是不一样的，我们有必要了解一下有关字体格式的知识。

1、TureTpe(.ttf)格式

.ttf字体是Windows和Mac的最常见的字体，是一种RAW格式，支持这种字体的浏览器有IE9+、Firefox3.5+、Chrome4+、Safari3+、Opera10+、iOS Mobile、Safari4.2+；

2、OpenType(.otf)格式

.otf字体被认为是一种原始的字体格式，其内置在TureType的基础上，支持这种字体的浏览器有Firefox3.5+、Chrome4.0+、Safari3.1+、Opera10.0+、iOS Mobile、Safari4.2+；

3、Web Open Font Format(.woff)格式

woff字体是Web字体中最佳格式，他是一个开放的TrueType/OpenType的压缩版本，同时也支持元数据包的分离，支持这种字体的浏览器有IE9+、Firefox3.5+、Chrome6+、Safari3.6+、Opera11.1+；

4、Embedded Open Type(.eot)格式

.eot字体是IE专用字体，可以从TrueType创建此格式字体，支持这种字体的浏览器有IE4+；

5、SVG(.svg)格式

.svg字体是基于SVG字体渲染的一种格式，支持这种字体的浏览器有Chrome4+、Safari3.1+、Opera10.0+、iOS Mobile Safari3.2+；

了解了上面的知识后，我们就需要为不同的浏览器准备不同格式的字体，通常我们会通过字体生成工具帮我们生成各种格式的字体，因此无需过于在意字体格式间的区别差异。

阿里妈妈web 在线字体

<http://iconfont.cn/webfont#!/webfont/index>

### 字体图标

其实我们可以把文字理解成是一种特殊形状的图片，反之我们是不是也可以把图片制作成字体呢？

答案是肯定的。

常见的是把网页常用的一些小的图标，借助工具帮我们生成一个字体包，然后就可以像使用文字一样使用图标了。

优点：

1、将所有图标打包成字体库，减少请求；

2、具有矢量性，可保证清晰度；

3、使用灵活，便于维护；

**我们可以通过该网站来完成在线图标打包成字体包,**

https://icomoon.io/

### 字体图标的使用

1: 引入字体文件

2：使用字体样式

## 360 官网宣传案例

### jQuery fullpage 插件的介绍

fullPage.js 是一个基于 jQuery 的插件，它能够很方便、很轻松的制作出全屏网站，

主要功能有：

1. 支持鼠标滚动
2. 支持前进后退和键盘控制
3. 多个回调函数
4. 支持手机、平板触摸事件
5. 支持 CSS3 动画
6. 支持窗口缩放
7. 窗口缩放时自动调整
8. 可设置滚动宽度、背景颜色、滚动速度、循环选项、回调、文本对齐方式等

### 案例代码逻辑的编写

使用步骤

1：引入jQuery fullpage 插件

2：编写html 结构

<div id="dowebok">

    <div class="section">

        <h3>第一屏</h3>

    </div>

    <div class="section">

        <h3>第二屏</h3>

    </div>

    <div class="section">

        <h3>第三屏</h3>

    </div>

    <div class="section">

        <h3>第四屏</h3>

    </div>

</div>

每个 section 代表一屏，默认显示“第一屏”，如果要指定加载页面时显示的“屏幕”，可以在对应的 section 加上 class=”active”

Javascript 代码

$(function(){

    $('#fullpage').fullpage({

// 设置背景色

sectionsColor : ['#0da5d6', '#2AB561', '#DE8910', '#16BA9D', '#0DA5D6'],

afterLoad: function (anchorLink, index) {

$('.section').removeClass('current');

// 延时100毫秒执行

setTimeout(function () {

$('.section').eq(index - 1).addClass('current');

}, 100);

}

});});

## 移动开发的概念

移动开发也称为[手机开发](http://baike.baidu.com/item/%E6%89%8B%E6%9C%BA%E5%BC%80%E5%8F%91/222789)，或叫做移动互联网开发。是指以手机、PDA、UMPC等便携终端为基础，进行相应的开发工作，由于这些随身设备基本都采用无线上网的方式，因此，业内也称作为无线开发。

而我们针对移动端的开发，一般都是开发app

### webApp的概念

移动WebApp是指基于Web的[系统](http://baike.baidu.com/item/%E7%B3%BB%E7%BB%9F)和[应用](http://baike.baidu.com/item/%E5%BA%94%E7%94%A8)，其作用是向广大的最终用户发布一组复杂的内容和功能。我们一般指的是移动端的网页，Web App就是运行于网络和标准浏览器上，基于网页技术开发实现特定功能的应用。它使用的技术无非就是HTML或HTML5、CSS3、JavaScript，webApp不能挂在应用商店供用户去下载. 这种类型的app 通过网址去进行访问，关闭浏览器之后，不能像客户端push 数据.

### 混合app的概念 hybrid app

Hybrid App（混合模式移动应用）是指介于web-app、native-app这两者之间的app，兼具“Native App良好用户交互体验的优势”和“Web App跨平台开发的优势”。它虽然看上去是一个Native App，但只有一个UI WebView，里面访问的是一个Web App，比如街旁网最开始的应用就是包了个客户端的壳，其实里面是HTML5的网页，后来才推出真正的原生应用。再彻底一点的，如掌上百度和淘宝客户端Android版，走的也是Hybrid App的路线，不过掌上百度里面封装的不是WebView，而是自己的浏览内核，所以体验上更像客户端，更高效。“Hybrid App同时使用网页语言与程序语言开发，通过应用商店区分移动操作系统分发，用户需要安装使用的移动应用”。总体特性更接近Native App但是和Web App区别较大。只是因为同时使用了网页语言编码，所以开发成本和难度比Native App要小很多。因此说，Hybrid App兼具了Native App的所有优势，也兼具了Web App使用HTML5跨平台开发低成本的优势

一般我们是通过HTML5,css3，JavaScript 技术开发一套移动端的页面，然后基于工具把webApp 打包成不同平台的app，然后挂在到应用商店中

### 原生app 的概念

Native App是一种基于智能手机本地操作系统如iOS、Android、WP并使用原生程式编写运行的第三方应用程序,也叫本地app。一般使用的开发语言为JAVA、C++、Objective-C。

自[iOS](http://baike.baidu.com/item/iOS/45705)和[Android](http://baike.baidu.com/item/Android/60243)这两个的[手机操作系统](http://baike.baidu.com/item/%E6%89%8B%E6%9C%BA%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F)发布以来，在互联网界从此就多了一个新的名词：[App](http://baike.baidu.com/item/App)(意为运行在智能的移动终端设备第三方应用程序)。

开发者们都知道在高端智能手机系统中有两种应用程序：一种是基于本地（操作系统）运行的APP；一种是基于高端机的浏览器运行的[Web App](http://baike.baidu.com/item/Web%20App)。

Native App因为位于平台层上方，向下访问和兼容的能力会比较好一些，可以支持在线或离线，消息推送或本地资源访问，摄像拨号功能的调取。但是由于设备碎片化，App的开发成本要高很多，维持多个版本的更新升级比较麻烦，用户的安装门槛也比较高。但是比较乐观的是，[App](http://baike.baidu.com/item/App)Store培养了一种比较好的用户付费模式，所以在Apple的生态圈里，开发者的盈利模式是一种明朗状态，其他market也在往这条路上靠拢。

