HTML5程序设计基础

第四章 地理位置定位

主要内容

01 地理位置定位的作用

02 HTML5 获取当前地理位置

03 地理位置定位示例

01 地理位置定位的作用

使用地理位置的应用

[redacted]

带有地理位置的微博

河北省,石家庄市,裕华区,南二环东路辅路 - 显示地图

40秒前 来自Weico.iPhone | 转发 | 收藏 |

使用位置定位的应用形式

- 商业：根据用户位置，提供附近店铺的打折信息
- 社交：显示附近的其他用户；将个人位置信息共享给其他好友
- 地图：定位到用户所在位置，直接显示用户附近地图
- 工具：记录个人足迹
- …

02 获取当前地理位置

地理位置

- 地理坐标系统
  - 经度与纬度的组成一个坐标系统，称为地理坐标系统，它是一种利用三度空间的球面来定义地球上的空间的球面坐标系统，能够标示地球上的任何一个位置。
- 地理坐标
  - 经度：距离英国格林威治以东或以西的数值表示
  - 纬度：距离赤道以北或以南的数值表示

地理位置的获取方式

- 地理位置由设备（笔记本电脑、手机、平板）通过 GPS 全球定位系统、Wifi、IP 地址或者蜂窝网络获取

定位功能的主要影响因素

- IP 地址地理定位数据
  - 自动查找用户的IP地址然后检索其注册的物理地址。因此如果用户的IP地址是ISP提供的，其位置往往由服务器供应商的物理地址决定！因此这个地址和用户实际的地址可能相差很大。
- GPS 地理定位数据
  - 通过搜集运行在地球周围的多个GPS卫星信号实现的。定位时间可能较长，不太适合快速响应的应用程序。而且在室内效果不是很好。

定位功能的主要影响因素

- WIFI 地理定位数据
  - 定位数据是通过三角距离计算得到的，这个三角距离是指用户当前位置已知的多个wifi接入点的距离。不同于GPS,wifi在室内也非常准确。
- 手机地理定位数据
  - 定位数据是通过用户到一些基站的三角距离确定。这种方法可提供相当准确的位置结果。这种方法通常和基于WIFI基于GPS地位结合使用。
- 用户自定义数据
  - 用户自己输入的一些地理位置信息。

位置信息来源的分类和特点

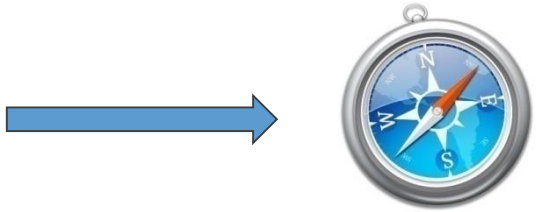
分类	优点	缺点
IP定位	任何地方都可以； 在服务器端处理	不准确，只能精确到市级
GPS定位	比较准确	定位时间长； 室内效果不好； 需要硬件设备支持
Wi-Fi定位	精确； 简单快捷； 可在室内定位	对于乡村无接入点的地区无法使用
手机定位	非常精确； 可在室内使用； 简单快捷	在没有基站的地方几乎无法使用

定位功能的主要影响因素

- 为获取用户的地理位置信息，需要使用多个资源。
  - 对于桌面浏览器，通常使用 WiFi (误差 20 米)，或者 IP 位置。
  - 对于手机设备倾向于使用测量学技术，例如 GPS（误差10米，只能在户外使用），WiFi 或者是 GSM/CDMA 的站点的 ID。
  - GPS定位由于信号比较薄弱的关系定位成功率比较低，需要花费的时间长，一般使用GPS定位的同时都会使用基站和WIFI辅助定位。

HTML5 地理位置定位

- Web程序获取定位信息



定位数据的获取

- 1. 用户打开需要获取地理位置的 web 应用。
- 2. 应用向浏览器请求地理位置，浏览器询问用户是否共享地理位置。
- 3. 假设用户允许，浏览器从设备咨询相关信息。
- 4. 浏览器将相关信息发送到一个信任的位置服务器，服务器返回具体的地理位置。

Geolocation API 概览

- HTML5地理位置的实现基于：
  - 浏览器（无需后端支持）获取用户的地理位置技术
  - 精确定位用户的地理位置(精度最高达 10m 之内，依赖设备）
  - 持续追踪用户的地理位置
  - Google Map、或者 Baidu Map 交互呈现位置信

HTML5 Geolocation API不指定设备使用哪种底层技术来定位应用程序的用户，它只是用于检索位置信息的API

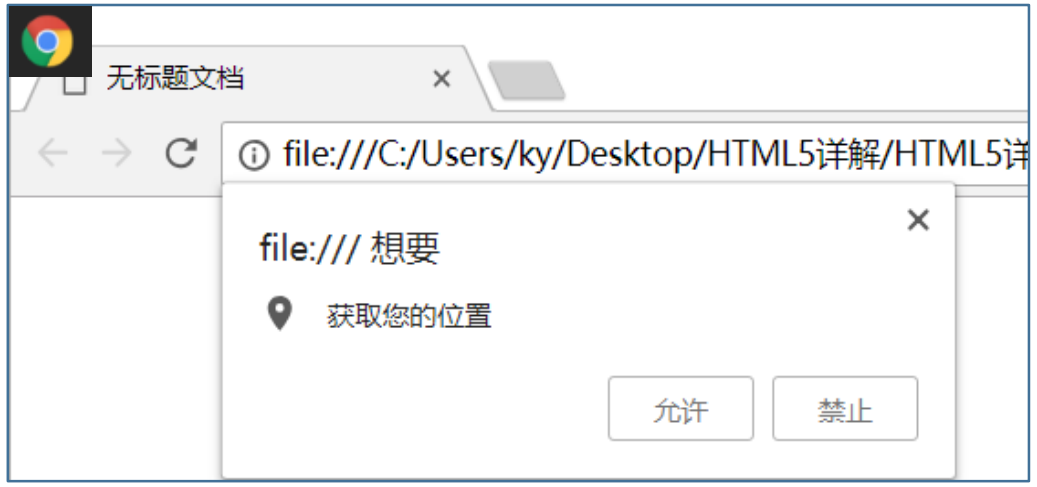
Geolocation API 概览

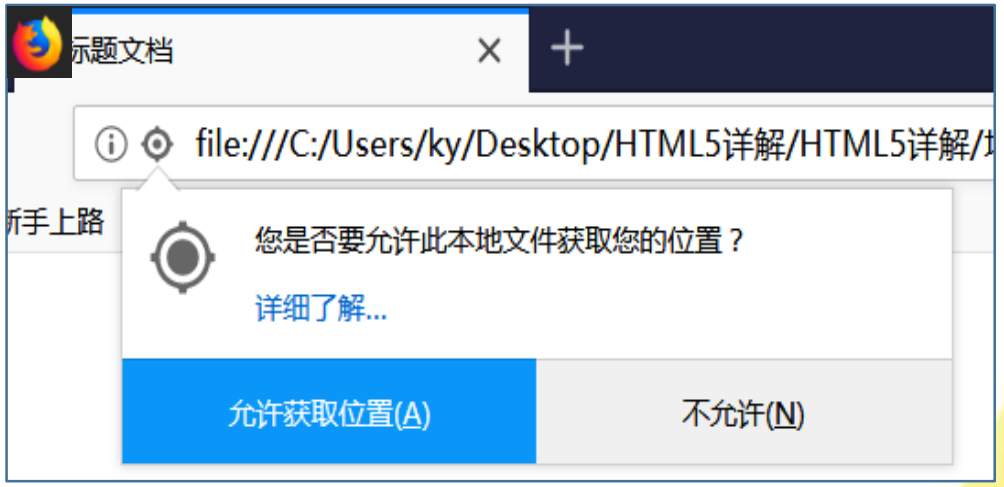
- 浏览器支持

IE 9+

- 在访问位置信息前，浏览器都会询问用户是否共享其位置信息，以保护用户隐私。

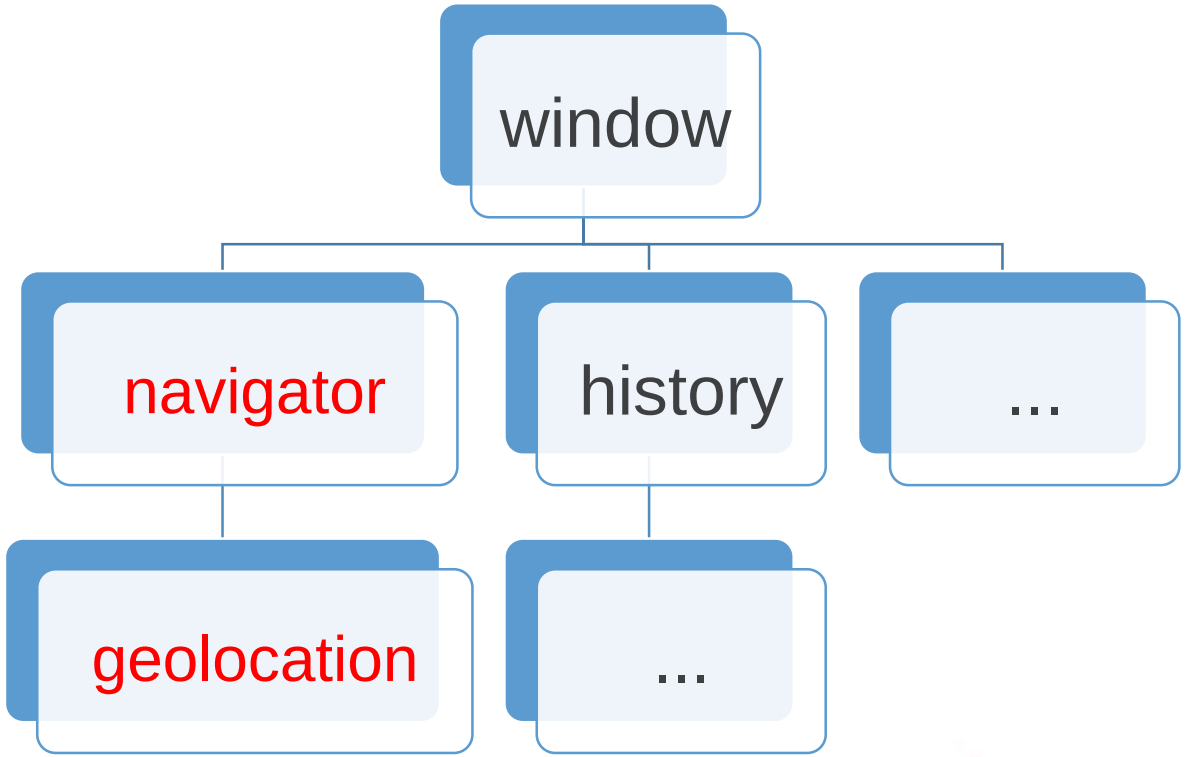
Geolocation API 概览





使用Geolocation API

- 浏览器提供 geolocation 对象，此对象为 navigator 对象的一个属性



使用Geolocation API

- Geolocation 能获得的具体数据（根据设备情况而定）
  - latitude 纬度（十进制），例如：38.0441
  - longitude 经度（十进制），例如：114.51
  - accuracy 精确度，以米为单位
  - altitude 海拔
  - heading 行驶方向
  - speed 速度
  - timestamp 获取位置的时间
  - …

距离赤道以北或以南的数值表示

距离英国格林威治以东或以西的数值表示

使用Geolocation API

- 检测浏览器兼容性
  - 通过判断 geolocation 对象是否存在判断浏览器是否支持
  - 如果不支持，则不执行获取位置的代码并给出提示信息

```
window.onload = function(){
    if(navigator.geolocation){
        //获取位置信息的代码
    }
    else{
        document.getElementById("tip").innerHTML =
        "您的浏览器版本过旧，建议使用最新版本谷歌浏览器。";
    }
}
```

使用Geolocation API

- Geolocation API 存在于navigator对象中，包含 3 个方法：
  - getCurrentPosition()　　//获取当前地理位置
  - watchPosition()　　//持续监视当前地理位置
  - clearWatch()　　//清除监视

getCurrentPosition

- 单次获取位置数据

  getCurrentPosition(onSuccess, onError, options)

  - 参数1：获取数据成功后执行的回调函数，使用 Position 对象作为唯一的参数
  - 参数2（可选）：获取数据失败时执行的回调函数，使用 PositionError 对象作为唯一的参数
  - 参数3 （可选）：可选参数列表，可通过该对象参数设定最长可接受的定位返回时间、等待请求的时间和是否获取高精度定位

getCurrentPosition

- 单次获取位置数据方法实例

```
function success(positon) {
    var coords = position.coords;
    console.log('纬度: ' + coords.latitude);
    console.log('经度: ' + coords.longitude);
};

function error(err) {
    console.log(err.code + ':' + err.message);
};

var options = {
    timeout: 5000
};

navigator.geolocation.getCurrentPosition(success, error, options);
```

成功回调函数

- 定位信息成功时返回一个 positon 对象给成功回调函数，此对象包含属性:

属性	描述
coords.latitude	十进制数的纬度
coords.longitude	十进制数的经度
coords.accuracy	位置精度
coords.altitude	海拔，海平面以上以米计
coords.altitudeAccuracy	位置的海拔精度
coords.heading	方向，从正北开始以度计
coords.speed	速度，以米/每秒计
timestamp	响应的日期/时间

demo4-1.html

失败回调函数

- 受设备情况、网络状况、用户授权等多方面影响，经常获取不到位置数据，所以错误处理函数非常必要
- 定位信息失败时返回一个 error 对象给失败回调函数。此对象包含如下属性:

属性	描述
code	错误编码： UNKNOWN_ERROR（错误编号0）- 未知错误 PERMISSION_DENIED（错误编号1） - 用户不允许地理定位 POSITION_UNAVAILABLE （错误编号2） - 无法获取当前位置 TIMEOUT （错误编号3） - 操作超时
message	错误信息

demo4-2.html

options可选参数列表

- options 数据格式为 JSON，有三个可选的属性：
  - enableHighAcuracy — 布尔值，表示是否启用高精确度模式(GPS)启用此模式，浏览器在获取位置信息时需耗费更多的时间。
  - timeout — 整数，表示等待响应的最大时间，否则触发errorCallback，默认为 0 毫秒，表示无穷时间。
  - maximumAge — 整数/常量，表示是否使用最近缓存的位置数据。默认为 0 毫秒，表示必须在每次请求时查找一个新位置。

demo4-2.html

思考

- 为什么使用回调函数的方式处理数据获取？
  - 获取数据的时间可能很长，使用回调函数的方式可以避免程序一直处于等待状态

getCurrentPosition

- 持续性获取位置数据方法

  watchPosition(onSuccess, onError, options)

  - 参数同 getCurrentPosition 相同
  - 监测用户位置，位置发生改变时即调用成功回调函数数
  - 清除监控：该方法会返回一个 ID，如要取消监听可以通过 clearWatch(watchId) 传入该 ID 实现取消的目的。

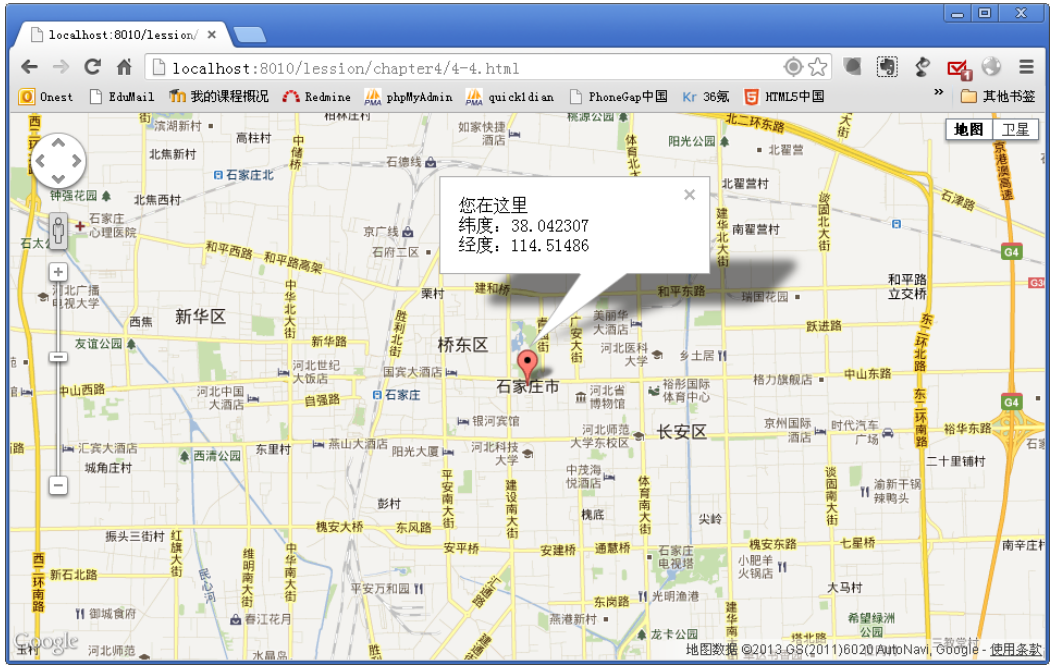
- 只针对移动设备

demo4-3.html

03 地理位置定位示例

地理位置定位示例

- 百度地图API：
  - http://lbsyun.baidu.com/index.php?title=jspopular



demo4_4.html

地理位置定位示例

- 在页面中导入 Baidu Map API 的脚本文件：
  - http://api.map.baidu.com/api?v=2.0&ak=PpjWkkFdn3WBbuuBMxnBjhX2eQ9B6DQZ
- 在项目中引入地理位置信息转换类：
  - http://developer.baidu.com/map/jsdemo/demo/convertor.js

地理位置定位示例

- 百度地图 API 和 HTML5 原生的定位 API 搭配使用，会导致精准误差问题。国内的地图产品，其地理位置大多数都进行了 GCJ-02 加密，即加入随机的偏差。而 HTML5 原生的定位 API 获取到的地理位置，是未经加密的。因此，为了保证 HTML5 原生的定位 API 获取到的地理位置在百度地图上较为准确的解析，就需要用官方提供的转换类。

demo4_5.html

THANKYOU