移动端开发：

1:mui的介绍:性能的体验的差距，一直是mobile app开发者放弃HTML5的首要原因。浏览器天生的切页白屏，不忍直视的转页动画、浮动元素的抖动、无法流畅下拉刷新等问题，这些都让HTML5开发者倍感挫败，尤其拿到Android低端机运行，摔手机的心都有；另一方面，浏览器默认控件样式又少又丑，制作一个漂亮的控件非常麻烦，也有一些制作简单的ui框架但性能低下。

Mui框架最有效的解决了这些问题，这是一个可以开发出高性能的APP框架，也是目前最接近原生App效果的框架。Mui遵守的是MIT开源协议

Mui 是封装了部分H5+功能的前端框架。

HTML5+扩展了JavaScript对象plus，使得js可以调用各种浏览器无法实现或实现不佳的系统能力，设备能力如摄像头、陀螺仪、文件系统等，业务能力如上传下载、二维码、地图、支付、语音输入、消息推送等。

Html5+plus和phoneGap，cordova的区别：

1： PhoneGap出的早，用的人比较多。phoneGap的定位是混合开发hybrid,用原生+js； Hbuilder的定位是纯js搞定。

2：5+有html5+和native.js技术，HTML5+包含常用的跨平台的几百个API，能满足常规的开发需求，而Native.js把40w原生api映射成js对象，这样js可以直接调用原生。HTML5+和native.js的组合形成了最强大的能力引擎。而phonegap需要用原生工程师写原生插件并给js开发者封装接口才能实现js调原生的能力，开发成本，对人的要求都不一样。

3：当然5+也支持原生插件，这点和phonegap类似。一个已经写好的原生SDK,无需使用Native.js，

4：H5+的直接封装的跨平台的API比较全，二维码，摇一摇，地图，微信分享，语音输入，推送这些都是常用的跨平台的，使用方便

5：phonegap做的app，在低端andriod手机上很难流畅的运行，否则h5早就火了，原生开发早就被挤压，phonegap为了避免H5的体验性不佳，常用了spa模式，但这个模式其实在低端机上也玩不转，代码复杂。

H5+的性能更高，它的动态效果是被增强引擎处理的，通过增强引擎，可以在低端机上运行各种动态效果。

Native.js是另一项创新技术。手机的原生API有四十多万，大量的API无法被HTML5使用。Native.js把几十万原生API封装成了js对象，通过js可以直接调ios和android的原生API。这部分就不再跨平台，写法分别是plus.ios和plus.android，比如调ios game center，或在android手机桌面创建快捷方式。

最后看开发便利性：

phonegap没有专业开发工具，语法提示、调试、打包都很麻烦。  
而在HBuilder里，5+的语法api提示非常完善；  
把手机通过数据线连上电脑，HBuilder可以真机运行，保存一个页面立即在手机上看到效果，Android上还可以看console.log。而用phonegap，你改完一个页面，不得不先打包，然后安装在手机上，然后发现不对，然后改下代码，然后继续打包。。。  
关于打包，phonegap由adobe提供了云打包，但需要先在本机准备资源，然后提交到国外的服务器，而HBuilder是一键打包，更加方便。当然phonegap和HBuilder都支持本地打包，那样就需要点原生开发知识了。

除了工具和runtime，还有mui框架

phonegap只是一个手机runtime，没有HBuilder工具，更没有Mui框架。  
mui是目前最接近原生App的HTML5框架，它的体验比jqm、bootstrap等框架更接近原生，它的性能远高于jqm、bootstrap、Ionic、framework7等框架。  
这种性能差别原因有2，一方面是设计思路不同，mui坚持用原生js做，不依赖jquery或angularjs，因为框架的依赖越多，App性能越差；另一方面是因为mui调用了5+的底层原生加速，这比不带原生加速的框架更快。  
mui详见：http://dcloudio.github.io/mui/

当然phonegap有一个优势，就是能支持windows phone、blackberry，这方面5+确实没有支持。