DQ1: Javascript

Top of Form

Assess the extent to which JavaScript has become more or less important than 10 years ago because of the emergence of Mobile Computing. Justify your answer.

1. Introduction
   1. What is JavaScript? (brief).
      1. As we all know from last week’s discussion, JavaScript is a web programming language used mainly to run on the client-side (browser).
   2. There are those that say that JavaScript has become less relevant due to the rise of mobile computing. Their arguments being/are:
   3. However, the evidence suggests otherwise; because of mobile computing, JavaScript actually has become more relevant.
2. Justification 1
   1. Oliver, 2012
      1. JavaScript, despite Microsoft's best efforts to make an incompatible mess out of it, has become the cross-platform language of choice for application developers and platform developers alike.
      2. we don't need any other language to write our Web or mobile applications. You can write the client in JavaScript using the jQuery library (with HTML markup, of course), transmit to and from the server in JSON, and store and query your data in JSON in the MongoDB document database -- and deploy it all to the cloud. From a management perspective, this means one set of developers, who are often cheaper than Java, Ruby, or .Net developers.
3. Justification 2
   1. Wright, 2013
      1. User experience has also become a crucial issue, due largely to Apple, who dramatically raised the bar for what is considered acceptable. Users now demand a great user experience, and the overall success and adoption of an app is highly correlated to it. In the enterprise, trends like Bring Your Own Device and Bring Your Own App are driving consumer-like behavior regarding the adoption of mobile apps. The result is that the user experience has become key to the success of any application regardless of its target and not something a CIO can ignore.
      2. JavaScript’s ability to meet the demands of rapid development, while at the same time maximizing the user experience, makes it a natural fit for mobile development platforms. It’s no coincidence that both Apple and Google have exposed JavaScript application programming interfaces to their mobile development tool kits in order to expand their developer base.
      3. It’s rare that shifts as dramatic as mobile and cloud don’t require IT organizations to re-staff or re-train around the (often complex and befuddling) New Thing. But JavaScript, owing to its power, flexibility and ubiquity, holds out just this promise. Early-adopter enterprises are already repurposing their Web developers as mobile developers, leveraging their JavaScript skills and applying them to these new domains. Similarly, server-side developers of traditional corporate applications are migrating their expertise to JavaScript and find it an easy and natural migration from more complex languages such as Java and C#.
4. Justification 3
   1. Wayner, 2014
      1. JavaScript is inescapable on the browser, which now dominates almost everything a client computer does. Now the server side is embracing it with tools like Node.js.
      2. JavaScript will assuredly become more dominant in other areas as well. Once the only way into a smartphone was to write code in the native language demanded by the manufacturer: Objective-C for the iPhone; Java for Android; C# for Microsoft. Now most mobile developers can get the performance they want out of HTML5 applications running in browser-like views. The result may not be as zippy as native code, but the JavaScript is good enough and portable to the Web, too.
      3. The browser isn't just colonizing phones; it's eating entire platforms. Chrome OS and Chromebooks are making regular operating systems obsolete. Why worry about that layer when JavaScript and the browser can do everything?
      4. The mainframe will have Cobol. Biologists will probably stick with Python. Linux will be written in C. But almost everything else is fair game as JavaScript gobbles the world.
5. Justification 4
   1. Hardawar, 2013
      1. Appcelerator found that devs now view Javascript as the most important programming language for mobile development. 47.2 percent of devs ranked Javascript their No. 1 language, while Java and Objective C trailed with 23 percent and 20 percent, respectively.
      2. You can build in Javascript and actually produce native applications,” King noted. “You can use Javascript on the server side, but also on the client side to build those apps.”
   2. Bubinas, 2014
      1. With mobile rapidly becoming the primary way many people interact with the Web, the space has seen substantial evolution in 2013 and seems poised for a shift heading into the coming year. A recent survey of mobile developers by Appcelerator and IDC found some unexpected trends in the sector, with developers highlighting Facebook's mobile emphasis, losing interest somewhat in HTML5, pinpointing Javascript as the predominant language for mobile development, witnessing a move toward larger scale app production and changing certain behaviors based on NSA surveillance revelations.
      2. "Originally developed as a browser-agnostic scripting language, JavaScript appears to becoming the lingua franca for mobile development," the report noted. "It's [sic] ability to render rich results from lightweight, simple-to-learn language has made it a natural fit for mobile's speed of development."
      3. Among developers, 47 percent ranked JavaScript as the most relevant language, while just 23 percent and 20 percent said the same of Java and Objective-C, respectively. Nearly nine out of 10 developers said it was likely or very likely that JavaScript would dominate both client-side and server-side development in the year ahead.
6. Justification 5
   1. Tee, 2013
      1. As we wrote about extensively in December of 2012, HTML5 and Java mixed with some JavaScript is proving to be a hugely beneficial standard for mobile app development. Amazon is leading the charge with its hybrid apps that use HTML5 for web view, UI and style sheets, Java for native actions like screen rotation and memory management, and JavaScript bridges serving to tie it all together. While some developers have fears that Java EE might not be up to the task of making nice with HTML5 given its strong connection to HTML4, the transition is already turning out to be pretty smooth. For example, using JavaServer Faces as the UI layer for web apps still works if you simply update the renderer associated with a given JSF component.
7. Justification 6
   1. Chand, 2014
      1. Trend: JS and HTML5
         1. HTML 5 was the darling of 2011 and 2012 but 2013 has been loving JavaScript and JS related libraries. JavaScript is not only browser friendly but also very flexible and easy to use. JavaScript libraries are on the rise. There are hundreds if not thousands of good and useful libraries developed by the developers and most of them are free and open source.
         2. JavaScript libraries like jQuery, Node.js, Knockout.js, AngularJS, and Backbone.JS are becoming a part of the mainstream web development. While the mobile trends are growing, so are the needs for responsive and interactive web sites and JavaScript is a big player in it.
         3. Recently, HTML 5 has continued to become a vital part of the Web development. However, HTML 5 trends have declined compared to JavaScript.
         4. Now, these trends may provide some idea but as we all know, both HTML and JavaScript are used together. As a web developer, just one is not going to help you. You must know and use both.
      2. Trend: Mobile and Apps
         1. There are 6.587 billion mobile devices in the world, according to mobiThinking.com.
         2. Today, software development is changing rapidly. The software industry is ready to disrupt the way we think about software. Not only the businesses but the consumers and the developers think differently as a result of the way software is built, maintained and consumed.
         3. Mobiles and apps are the trends today. Every business wants to move to mobile. Every person wants to consume information and service via his/her mobile.
         4. Both the Google Play store and Apple iTunes Store have over 1 Million apps and over billons of downloads. Even Microsoft’s new baby, Windows Store has over 200,000 apps. An average US smartphone user has 41 apps downloads. That means, both iOS and Android developers are in and will be in high demand.
         5. Android app development has grown much faster in 2013 compared to iOS and Windows Phone. Google Trends in Figure 5 present the following graph trends for iOS vs Android vs Windows Phone. “We’re seeing a tremendous disruption by mobile” as huge numbers of people move from the traditional web to mobile devices, Prashant Fuloria, Chief Product Officer of Flurry said. That’s being driven by the large number of mobile devices around the world, and there’s plenty of room for more growth.
8. Justification 7
   1. Appcelerator, 2013
      1. THE FUTURE WILL BE WRITTEN IN JAVASCRIPT
         1. In ranking the relevance of development languages for mobile app development, JavaScript emerges as the clear winner: 47.2% of respondents ranked it first, more than ten percentage points ahead of the next closest language.
         2. In ranking the relevance of development languages for mobile app development, JavaScript emerges as the clear winner. 47.2% of respondents ranked it first, more than ten percentage points ahead of the next closest language, Java (35%). Objective-C was third at 32%.
         3. Originally developed as a browser-agnostic scripting language, JavaScript appears to becoming the lingua franca for mobile development. It’s ability to render rich results from lightweight, simple-to-learn language has made it a natural fit for mobile’s speed of development. There is also mobile’s partner in crime, the cloud, where JavaScript’s role is fast becoming just as central: witness its swift adoption as a backend technology platform via Node.js. More than 88% of developers found it “likely” or “very likely” that in 2014 JavaScript would increasingly dominate both client- and server-side development.
9. Justification 8
   1. Roggio, 2014
      1. The bottom line is that retailers, both online and multichannel, have three basic strategies to employ for producing and distributing a mobile app. They can build so-called native apps, use HTML5 and JavaScript with a wrapper, or, choose not to have an app and focus on the mobile web instead.
      2. Together HTML5, CSS, and JavaScript are the web stack — i.e., the markup and languages that are behind almost everything done in a web browser. And, it turns out, they can also be used to develop mobile applications that may be included in native application stores and loaded on mobile devices just like native applications.
      3. Using HTML5, CSS, and JavaScript for developing mobile applications offers two basic advantages over producing native applications.
      4. First, HTML5, CSS, and JavaScript developers are more common and, usually, less expensive. Put another way, many more folks know how to write HTML5 than know how to write Objective-C.
10. Justification 9
    1. Tucker, 2013
       1. I would not have recommended this three years ago, but JavaScript is truly becoming universal. There is not any tier that JavaScript cannot touch to some extent. Currently you can utilize JavaScript to write mobile applications, server applications, program the data tier, create a blogging engine, develop and execute a custom workflow and pretty much anything else you can think of.
       2. One of the big shifts that has been slowly happening is the ability to use JavaScript as a shared logic layer between native experiences. For example, recently in iOS 7 Apple included a true JavaScript bridge within the SDK. This means that it is possible to use JavaScript as a shared logic layer (for example between iOS and Android). As the ubiquity of web technologies grows, I expect that the ability to use JavaScript in this manner will grow to include most platform (included non-mobile ones).
11. Conclusion
    1. Tucker, 2013
       1. I would not have recommended this three years ago, but JavaScript is truly becoming universal. There is not any tier that JavaScript cannot touch to some extent. Currently you can utilize JavaScript to write mobile applications, server applications, program the data tier, create a blogging engine, develop and execute a custom workflow and pretty much anything else you can think of.

References:

Appcelerator (2013) Q4 2013 MOBILE TRENDS REPORT [Online]. Available from: <http://www.appcelerator.com.s3.amazonaws.com/pdf/q4-2013-devsurvey.pdf> (Accessed: 7 June 2014)

Bubinas , C. (2014) ‘Mobile developer trends show interest in JavaScript, emphasis on scale’, Klockwork [Online]. Available from: <http://www.klocwork.com/blog/static-analysis/mobile-developer-trends-show-interest-in-javascript-emphasis-on-scale/> (Accessed: 7 June 2014)

Chand, M. (2014) ‘Top 5 Developer Technology Trends For 2014’, *C#Corner* [Online]. Available from: <http://www.c-sharpcorner.com/UploadFile/mahesh/top-5-developer-technology-trends-for-2014/> (Accessed: 7 June 2014).

Hardawar, D. (2013) ‘Latest mobile dev trends: Facebook’s mobile API & Javascript are hot; HTML5 & old web architecture are not’, *VentureBeat* [Online]. Available from: <http://venturebeat.com/2013/12/19/latest-mobile-dev-trends-facebooks-mobile-api-javascript-are-hot-html5-old-web-architecture-are-not/> (Accessed: 7 June 2014)

Oliver, A. (2012) ‘Will the future be written entirely in JavaScript?’, *InfoWorld* [Online]. Available from: <http://www.infoworld.com/d/application-development/will-the-future-be-written-entirely-in-javascript-205095> (Accessed: 7 June 2014)

Tee, J. (2013) ‘The JavaScript and HTML5 trend continues for mobile development’, *TheServerSide.com* [Online]. Available from: <http://www.theserverside.com/feature/The-JavaScript-and-HTML5-trend-continues-for-mobile-development> (Accessed: 7 June 2014)

Tucker, D. (2013) ‘10 Skills Developers Should Invest In for 2014’, *Medium* [Online]. Available from: <https://medium.com/web-design-and-development/10-skills-developers-should-invest-in-for-2014-64eb28781489> (Accessed: 7 June 2014)

Wayner, P. (2014) ‘12 predictions for the future of programming’, *JavaWorld* [Online]. Available from: <http://www.javaworld.com/article/2093747/java-ios-developer/12-predictions-for-the-future-of-programming.html> (Accessed: 7 June 2014)

From Slashdot:

“By what you said, I guess I would recommend you to start with HTML 5 and Javascript. Both are growing, well stabishled, not very hard to learn and very usefull (for desktop, mobile and even server-side using node.js).”

Learn Javascript. It's everywhere:

\* client-side in browsers as little helper fragments attached to buttons, etc

\* used for building rich client applications within browsers

\* can be used server-side with things like Node.js

\* is embedded as a scripting language into various tools

And it's actually not a bad language. The underlying concepts are simple and elegant, although some features for large-scale programming are lacking (no module system, etc).

It's also fun to learn, because you can achieve interesting/amusing results within a browser with just a few lines of javascript.

With some solid javascript knowledge for both client and server-side style code, a lot of job opportunities will open. Learning something like Python is an alternative, but even then employers who look for Python programmers for web-based apps will usually also want javascript skills...

References:

<http://ravibail.wordpress.com/2013/02/16/growing-importance-of-javascript/>Bottom of Form

<http://www.quora.com/What-are-the-reasons-for-the-decline-of-JavaScripts-TIOBE-score-during-2010>

<http://redmonk.com/sogrady/2014/01/22/language-rankings-1-14/>

<http://techcrunch.com/2013/03/15/the-future-of-javascript/>

<http://ocpsoft.org/opensource/javascript-is-the-new-perl/>

<http://sendgrid.com/blog/what-is-the-best-language-to-learn-programming/>

Fankhauser, D. (2012) *Is JavaScript the Future of Programming?* [Online]. Available from: <http://mashable.com/2012/11/12/javascript/> (Accessed: 7 June 2014)