Exercise

Set - 1

Question 1 :

What's AJAX ?

Answer :
AJAX (Asynchronous JavaScript and XML) is a newly coined term for two powerful browser features that have been around for years, but were overlooked by many web developers until recently when applications such as Gmail, Google Suggest, and Google Maps hit the streets.

Asynchronous JavaScript and XML, or Ajax (pronounced "Aye-Jacks"), is a web development technique for creating interactive web applications using a combination of XHTML (or HTML) and CSS for marking up and styling information. (XML is commonly used, although any format will work, including preformatted HTML, plain text, JSON and even EBML).
The Document Object Model manipulated through JavaScript to dynamically display and interact with the information presented
The XMLHttpRequest object to exchange data asynchronously with the web server. In some Ajax frameworks and in some situations, an IFrame object is used instead of the XMLHttpRequest object to exchange data with the web server.
Like DHTML, LAMP, or SPA, Ajax is not a technology in itself, but a term that refers to

the use of a group of technologies together. In fact, derivative/composite technologies based substantially upon Ajax, such as AFLAX, are already appearing.
Ajax applications are mostly executed on the user's computer; they can perform a number of tasks without their performance being limited by the network. This permits the development of interactive applications, in particular reactive and rich graphic user interfaces.
Ajax applications target a well-documented platform, implemented by all major browsers on most existing platforms. While it is uncertain that this compatibility will resist the advent of the next generations of browsers (in particular, Firefox), at the moment, Ajax applications are effectively cross-platform.
While the Ajax platform is more restricted than the Java platform, current Ajax applications effectively fill part of the one-time niche of Java applets: extending the browser with portable, lightweight mini-applications.

Ajax isn't a technology. It's really several technologies, each flourishing in its own right, coming together in powerful new ways. Ajax incorporates:
* standards-based presentation using XHTML and CSS;
* dynamic display and interaction using the Document Object Model;
* data interchange and manipulation using XML and XSLT; * asynchronous data retrieval using XMLHttpRequest;
* and JavaScript binding everything together.

Question 2 :

Who's Using Ajax ?

Answer :
Google is making a huge investment in developing the Ajax approach. All of the major products Google has introduced over the last year — Orkut, Gmail, the latest beta version of Google Groups, Google Suggest, and Google Maps — are Ajax applications. (For more on the technical nuts and bolts of these Ajax implementations, check out these excellent analyses of Gmail, Google Suggest, and Google Maps.) Others are following suit: many of the features that people love in Flickr depend on Ajax, and Amazon's A9.com search engine applies similar techniques.
These projects demonstrate that Ajax is not only technically sound, but also practical for real-world applications. This isn't another technology that only works in a laboratory. And Ajax applications can be any size, from the very simple, single-function Google Suggest to the very complex and sophisticated Google Maps.

At Adaptive Path, we've been doing our own work with Ajax over the last several months, and we're realizing we've only scratched the surface of the rich interaction and responsiveness that Ajax applications can provide. Ajax is an important development for Web applications, and its importance is only going to grow. And because there are

so many developers out there who already know how to use these technologies, we expect to see many more organizations following Google's lead in reaping the competitive advantage Ajax provides.
Moving Forward

The biggest challenges in creating Ajax applications are not technical. The core Ajax technologies are mature, stable, and well understood. Instead, the challenges are for the designers of these applications: to forget what we think we know about the limitations of the Web, and begin to imagine a wider, richer range of possibilities

---

Question 3 :

Should I consider AJAX?

Answer :
AJAX definitely has the buzz right now, but it might not be the right thing for you. AJAX is limited to the latest browsers, exposes browser compatibility issues, and requires new skill-sets for many. There is a good blog entry by Alex Bosworth on AJAX Mistakes which is a good read before you jump full force into AJAX.
On the other hand you can achieve highly interactive rich web applications that are responsive and appear really fast. While it is debatable as to whether an AJAX based application is really faster, the user feels a sense of immediacy because they are given active feedback while data is exchanged in the background. If you are an early adopter and can handle the browser compatibility issues, and are willing to learn some more skills, then AJAX is for you. It may be prudent to start off AJAX-ifying a small portion or component of your application first. We all love technology, but just remember the purpose of AJAX is to enhance your user's experience and not hinder it.

Question 4 :

Does AJAX work with Java?

Answer :
Absolutely. Java is a great fit for AJAX! You can use Java Enterprise Edition servers to generate AJAX client pages and to serve incoming AJAX requests, manage server side state for AJAX clients, and connect AJAX clients to your enterprise resources. The JavaServer Faces component model is a great fit for defining and using AJAX components.

Question 5 :

Won't my server-side framework provide me with AJAX?

Answer :
You may be benefiting from AJAX already. Many existing Java based frameworks already have some level of AJAX interactions and new frameworks and component libraries are being developed to provide better AJAX support. I won't list all the Java frameworks that use AJAX here, out of fear of missing someone, but you can find a good list at www.ajaxpatterns.org/Java_Ajax_Frameworks.
If you have not chosen a framework yet it is recommended you consider using JavaServer Faces or a JavaServer Faces based framework. JavaServer Faces components can be created and used to abstract many of the details of generating JavaScript, AJAX interactions, and DHTML processing and thus enable simple AJAX used by JSF application developer and as plug-ins in JSF compatible IDE's, such as Sun Java Studio Creator.

Question 6 :

Where should I start?

Answer :
Assuming the framework you are using does not suffice your use cases and you would like to develop your own AJAX components or functionality I suggest you start with the article Asynchronous JavaScript Technology and XML (AJAX) With Java 2 Platform, Enterprise Edition.
If you would like to see a very basic example that includes source code you can check out the tech tip Using AJAX with Java Technology. For a more complete list of AJAX resources the Blueprints AJAX home page.
Next, I would recommend spending some time investigating AJAX libraries and frameworks. If you choose to write your own AJAX clients-side script you are much better off not re-inventing the wheel.
AJAX in Action by Dave Crane and Eric Pascarello with Darren James is good resource. This book is helpful for the Java developer in that in contains an appendix for learning JavaScript for the Java developer.

Question 7 :

Did Adaptive Path invent Ajax? Did Google? Did Adaptive Path help build Google's Ajax applications?

Answer :
Neither Adaptive Path nor Google invented Ajax. Google's recent products are simply the highest-profile examples of Ajax applications. Adaptive Path was not involved in the development of Google's Ajax applications, but we have been doing Ajax work for some of our other clients.

Question 8 :

Is it possible to set session variables from javascript?

Answer :
It's not possible to set any session variables directly from javascript as it is purely a client side technology. You can use AJAX though to asyncronously...

Cannot parse XML generated by JSP I am generating an XML using JSP, when i run the JSP in IE it shows the XML as per DOM, but when i try to parse it using Javascript , the command xmldoc.documentElement...

This is working code I am using, it might help you. if (!isIE) xmldoc = req.responseXML; else { //IE does not take the responseXML as...

Question 9 :

What do I need to know to create my own AJAX functionality?

Answer :
If you plan not to reuse and existing AJAX component here are some of the things you will need to know.
Plan to learn Dynamic HTML (DHTML), the technology that is the foundation for AJAX. DHTML enables browser-base real time interaction between a user and a web page. DHTML is the combination of JavaScript, the Document Object Model (DOM) and Cascading Style Sheets (CSS).
* JavaScript - JavaScript is a loosely typed object based scripting language supported by all major browsers and essential for AJAX interactions. JavaScript in a page is called when an event in a page occurs such as a page load, a mouse click, or a key press in a

form element.
* DOM - An API for accessing and manipulating structured documents. In most cases DOM represent the structure of XML and HTML documents.
* CSS - Allows you to define the presentation of a page such as fonts, colors, sizes, and positioning. CSS allow for a clear separation of the presentation from the content and may be changed programmatically by JavaScript.
Understanding the basic request/response nature of HTTP is also important. Many subtle bugs can result if you ignore the differences between the GET and OIst methods when configuring an XMLHttpRequest and HTTP response codes when processing callbacks.
JavaScript is the client-side glue, in a sense. JavaScript is used to create the XMLHttpRequest Object and trigger the asynchronous call. JavaScript is used to parse the returned content. JavaScript is used to analyze the returned data and process returned messages. JavaScript is used to inject the new content into the HTML using the DOM API and to modify the CSS.

Question 10 :

Do I really need to learn JavaScript?

Answer :
Basically yes if you plan to develop new AJAX functionality for your web application. On the other hand, JSF components and component libraries can abstract the details of JavaScript, DOM and CSS. These components can generate the necessary artifacts to make AJAX interactions possible. Visual tools such as Java Studio Creator may also use AJAX enabled JSF components to create applications, shielding the tool developer from many of the details of

AJAX. If you plan to develop your own JSF components or wire the events of components together in a tool it is important that you have a basic understanding of JavaScript. There are client-side JavaScript libraries (discussed below) that you can call from your in page JavaScript that abstract browser differences. Object Hierarchy and Inheritance in JavaScript is a great resource for a Java developer to learn about JavaScript objects.

Quick Links

Contact Us (http://www.indiaparinam.com/contact-us)
Ask Question (http://www.indiaparinam.com/contact-us)
Send Query (http://www.indiaparinam.com/contact-us)
Disclaimer (http://www.indiaparinam.com/Disclaimer)
SiteMap

Newsletter

Email Address...

Subscribe

Follow us :

Facebook (https://www.facebook.com/infoindiaparinam)
Google PLus (https://plus.google.com/+Indiaparinam)
Twitter (https://twitter.com/indiaparinam)