

**Java technology zone technical podcast series: Season 2**

**Scott Davis, perhaps the Grooviest guy on Earth, shifts his spotlight and discusses HTML5**

**Episode date: 01-25-2011**

developerWorks: This is a special series of the developerWorks podcast with Andy Glover, a developer, author, speaker and entrepreneur. Andy is managing partner at Beacon50, a regular contributor to several technical publications, including developerWorks. He writes the Java Development 2.0 column.

He's also creator of the award-winning easyb, a behavior-driven development network. And here he talks to a variety of experts in the Java and related technologies community on topics ranging from closure to open source business intelligence and even .NET. Here's Andy.

GLOVER: So Scott Davis is the founder of thirstyhead.com. He's the author of multiple books including a recent one on Grails. What was the name of that book?

DAVIS: Getting Started with Grails. It's a free download available from infoq.com.

GLOVER: And then Groovy recipes. He also wrote a GIS book.

DAVIS: GIS for Web Developers.

GLOVER: You've written tons of articles for IBM developerWorks and spoken at conferences around the world, including the No Fluff series where you and I met, I believe, years ago.

DAVIS: Absolutely, yes. Years ago.

GLOVER: Excellent. And so what you've been talking about lately is HTML5. And I don't know much about HTML5, but the little I've heard and the little kind of chat we had, HTML5 seems really interesting.

And so I thought it would be interesting as well for you to kind of expound upon what it is and help us, those of us listening to that that have little to zero clue about what HTML5 means for us. So where do we start?

DAVIS: Where we start is I'm a Java guy from way back. You're a Java guy from way back. And you know when we talk about Java we talk about write once, run anywhere.

GLOVER: Yes.

DAVIS: And HTML is that same technology. I mean, it really is write once, run anywhere. Now, we scoff a little bit about the write once. Definitely there are browser differences and JavaScript differences, but those differences are really going away.

I mean, with HTML5 we finally have a spec that all the major browser vendors are competing on how well they implement it and how fully they implement it and how quickly they've implemented it.

GLOVER: So you mean even...

DAVIS: Even Microsoft. Even IE, even IE has announced with IE9 they're trying to be as standard compliant as the web browsers, Chrome and Safari, and of course the gecko browser with Firefox as well.

It's an exciting time that HTML is becoming that kind of write once, run anywhere, write once run anywhere platform that we've enjoyed for years as Java developers.

GLOVER: Let's take a step back then as Java developers and having learned HTML back when like the Internet started, right? And everyone made the website with the blink, you know? And also had the little, the construction, image of the guy like constantly pounding on the rock.

DAVIS: Little guy digging. Absolutely. I want to mine those and put them on my websites now. Super hips, cool retro.

GLOVER: Exactly. That was cool back then. So HTML as far as I'm concerned is just, is a bunch of tags. Fairly boring back in those days. Why is HTML5 so hip?

DAVIS: Well, so, first of all, the first thing about HTML5 that's really cool it's the third major era of HTML. Don't worry about the numbers. But we're used to that as Java developers as well. What is it, Java 2 or Java 1.5. It's just numbers.

But HTML got really popular in the mid-90s, '95, '96, when it got really exciting. But Tim Berners-Lee came out with the original specs a little bit early 90s, '91, '92, right around there.

So for about the first 10 years or so of HTML and the first three series of HTML, they were saying HTML is kind of like SGML. It's SGML but it's less and we've got different tags and things like that. But it's kind of like SGML.

Then the next 10 years, starting in about 2000 or so, was, of course, the era of XHTML, saying no, HTML is really like XML. We were doing these kinds of things. For the first 10 years we were saying it's kind of like SGML. For the second 10 years we were saying it's kind of like XGML.

Hard to believe it's been around for 20 years. But now I mean now here we are now in HTML5 land. They're saying is HTML is not XML. HTML is not SGML. HTML is HTML. It's finally come into its own. If you look at the doc type, the doc type at the top of that...I'm giving you error...error tags right now.

GLOVER: Yes, the greater and less than.

DAVIS: But if you go look at the doc type right now, the doc type of HTML5 is just HTML. Not a version number. Not transitional. Not strict. Not anything. It's just HTML. And what's amazing is how pervasive HTML5 is right now. If you go do a view source on Google.com right now, they're serving up HTML5.

GLOVER: Really? Okay.

DAVIS: If you go to view source on Apple.com, they're doing HTML5. If you do it on Mozilla.org, they're doing HTML5.

It's amazing how much HTML5 is floating around out there. It's absolutely ready for prime time. That's what's so exciting about it.

GLOVER: So you mentioned kind of the evolution of HTML and somewhere I got really confused. What was DHTML?

DAVIS: So, you know, that's a phrase that kind of Microsoft popularized a little bit. Dynamic HTML. Of course we call it AJAX now.

But what they were trying to capitalize on was, hey, this is a unique way, new way of combining JavaScript and CSS and HTML to give us a more kind of dynamic experience, a more desktop experience.

GLOVER: Okay. But...keep going.

DAVIS: I was going to say we literally have Microsoft to thank for the AJAX revolution. They were the first one that put that XHR object in their browser to support their online version of Exchange.

They said, hey, we want to have an inbox floating around up there and have that unread message count to be able to flicker independently of the page refresh.

And so they were the first one that baked it into their browser. The Mozilla folks said, hey, that's kind of cool; we're going to bake that into our browser as well.

And that's kind of how those standards evolve with HTML is that it was a de facto standard. By the time everyone had implemented it the WWWC, the World Wide Web Consortium, came back and said this is a good thing that should be standard across all browsers.

GLOVER: You kind of mentioned AJAX and CSS and certainly kind of the Web 2.0 buzzword you can attach a lot of things to it but one of them certainly is kind of the AJAX revolution and the browser is almost...or a web page being more like a desktop.

How is all that related to HTML5? Are they different things? Is AJAX now somehow enhanced in HTML5?

DAVIS: Well, what's nice is that AJAX is still AJAX.

GLOVER: Okay.

DAVIS: Except for the X part, right? I mean, we're doing less and less throwing around XML and more and more JSON. So it is asynchronous. It is JavaScript. But asynchronous and JavaScript and JavaScript. AJAX, that doesn't roll off [the tongue] nearly as nicely.

But these web technologies, whether you're talking WebOS, the Palm phones, those web technologies are all the way down to the bare metal. That's what's driving that phone.

I'm working on a gig right now where I'm working for a major cable company. And the cable set-top boxes are sending JSON out. The listing of all the DVR content, the recorded content. And JSON of all the channel lineups and JSON of all the schedules for two weeks and things like that.

There are actually browsers baked into the TV right now that are kind of catching that JSON and displaying it. That's the gig I'm working on right now.

GLOVER: Tell me more about that. From a TV standpoint these are major TVs from like whatever Sony or something like that?

DAVIS: Yeah, exactly that kind of thing. I can't discuss the brand name specifically. But, yes. So that's what I'm talking about. This write once, run anywhere. If you've got JSON coming off of a set-top box and we have a web browser in a TV that can display all the stuff that you've recorded, all the channels and things like that.

I'm also working on Android Galaxy tablet that is taking that same information off of the set top box and displaying it on your tablet.

The same thing on iPads as well. So that's the write once run anywhere right there. Is because it's HTML, because it's JavaScript because it's those kinds of things I can be slinging that data around on all these different devices and grabbing it and displaying it.

GLOVER: From what little I know about HTML5, there are some new aspects to it. You mentioned it before we started recording this, this ability to store. There's a local data store. Let's talk about it. What does HTML5 introduce that's so cool?

DAVIS: So HTML5, on one hand HTML5 is a bunch of new elements.

GLOVER: Okay.

DAVIS: And that's cool in of itself. On a bunch of new form elements. So I can say input type equals URL, or input type equals email or input type equals date.

And I can start do those sorts of things. And the real cool thing about that, I don't know if you ever notice, I know you're an iPhone guy. If you're on a website and they prompt you for a URL and your keyboard changes so you get the big dotcom button.

If you are plugging in an email address and keyboard changes and you get the @ symbol and all that it's all based on HTML5. It's all based on input type equals URL and the device is absolutely able to display the keyboard that's appropriate for that kind of data entry.

GLOVER: Just curious, so you said all the major browsers are supporting HTML5. That would be, for example, IE9. So how is the backward compatibility working with that?

I guess in the case of like input type equals URL it becomes a text box or whatever.

DAVIS: Yes. So in those particular cases input type equals, the beautiful thing is every browser, no matter what, even IE6, if you give it an input type equals FU and it doesn't recognize it, it will roll back to input type equals text. So there's zero danger in using these new HTML5 form elements because they all degrade nicely back to what you're used to dealing with.

There are a bunch of new elements like a nav element to display your navigation stuff. A section tag. A header tag. A footer tag. These kinds of things.

What's really cool about most browsers...most browsers, not IE...but most browsers, if they come across an element that they don't recognize, they don't care. They'll just go ahead and display it. And you can style it up however you want.

GLOVER: Okay.

DAVIS: So there's zero danger in using these new HTML5 elements on most browsers because they're like I don't care, again you could have an element FU and I'll just display it and expect you to style it up in CSS as you normally would.

With IE there's a bit of a catch. IE, if it doesn't recognize an element, it hides it. It hides it. So they are guaranteed to not be forward compatible out of the box. But there's a workaround for that as well.

And it's really, really simple. In JavaScript, you just add those elements to the DOM. You say you've got a whole bunch of new elements, NAV and header element and footer things like that. It makes your browsers HTML5 forward compatible even in IE9.

There's a snippet of JavaScript you can include, called an HTML5shiv. If you Google it, it will come up. You include that one liner in your code and that's what makes even your IE forward compatible in HTML5.

GLOVER: All right.

DAVIS: So the one aspect, I didn't mean to interrupt you. But that's the one cool thing all these elements, the keyboard and all that kind of stuff.

But the really cool stuff is what you asked me about. There are two new things that are especially interesting for mobile device development but still are really cool for desktop development as well and that is application cache and local storage.

GLOVER: Start with the cache.

DAVIS: So you know your browsers are caching up your HTML, your CSS, your JavaScript anyway. That's just how the Web works. When you go into Google Maps, the more you use Google Maps, the faster it gets because all of those map tiles are stored up on your local cache.

That's what's just so cool about the way Google Maps works. It naturally gets faster just based on the way the browsers work.

The problem with HTML 4 and below is there's no way to actively control that. The only way you could really control it is give server-side hints.

When you're sending down data, you can give what they call far future expire headers, thing. You know what, this logo expires in the year 3000. Then your browser is going to hold onto it.

If you don't give it far future expires headers the browser gives it a 24-hour time span on it and then throws it away. Every time you visit Google you'd have to download that same Google logo over and over again if they didn't say no hold onto this, our logo isn't going to be changing.

So historically the way to get your browser to hold onto that material, hold onto that raw material and use it, was giving hints from the server.

But what application cache allows you to do is provide a manifest file and write in your HTML header, excuse me, you body tag you say body manifest equals and you point it to a plain text file. And this manifest file says, all right, these are the things I want you to hold onto now.

Now I'm being explicit. I want you to hold onto this HTML file. Hold onto this CSS file, these images, this JavaScript, these kinds of things. And the manifest file has three sections. One of them is these are the things I want you to cache. Another section is labeled network saying you have to be online in order to do these kinds of things.

So if you wanted to like place an order, place an order on your shopping cart, things like that, you can throw URLs under the network section saying no, no, you've got to be online in order to do this kind of thing.

The third section is a fall-back section. It says, all right, now what I can do is I can say, all right, if you're online I'm going to display these images. But if you're offline I want you to display these images and those offline images, of course, you've already cached up.

GLOVER: Interesting. Okay.

DAVIS: When you're using this manifest and you're doing these kinds of things, on an iPhone and on an Android, that icon will show up on your desktop. It will look like it was a native application written objective C and everything else. But in fact it's all based on the big three technologies: HTML, JavaScript and CSS.

GLOVER: Wow. That's pretty neat. What about the second one?

DAVIS: So the second one is local storage. Now, if application cache is for all of your artifacts...again, HTML, CSS, JavaScript, images, MP3 files, all of these kinds of things, local storage is for your data.

And so you might think of it like a database. And, in fact, there is kind of a web database. But the Microsoft folks are not going to implement it.

And the Firefox folks, the Mozilla folks aren't going to implement it. They said we're not doing to deal with this database mentality. But every major browser, including IE8, Safari, Firefox, all of these guys, support local cache, which is simply a key value store.

GLOVER: Okay.

DAVIS: So it's the no SQL revolution going on inside your browser right there. The fact that let's say you have a game that you've written: All of your logic, all the images, all the everything will be dealt with in your manifest file and app cache. But all of your data, your high score, your user name, your password, all of that kind of information will be stored as key value pairs in local storage.

GLOVER: And so that local storage is part of the browser itself?

DAVIS: Think of it as cookies on steroids.

GLOVER: Okay. So you can access it...

DAVIS: Every URL has its own local storage that you can turn around and start storing key values paired. They don't get passed back and forth from client to server like cookies do. But they are tied to that URL. So every URL has this unique local store that they can start taking advantage of.

GLOVER: That's pretty interesting. It's the URL is almost like the name space or the schema or something like that. Okay. And you can access this local store via JavaScript or whatever it is, it doesn't matter.

DAVIS: Exactly. Think about the ubiquitous example that everyone reaches for, the Twitter client. Now what you could do is you could have all the look and feel all the HTML artifacts stored locally on your cache. So when I'm sitting on an airplane, I'm in airplane mode, I can start entering in new tweets.

And as I click on save and save and save, they're all being stored in local storage. Your browser is now network aware. It now knows when it comes back online again. It would be up to you to write the code to

actually synchronize the local storage with the server. But it gives you the full kind of Twitter experience in pure HTML.

GLOVER: Wow. Interesting. So where do you get started with HTML5? Obviously listen to this podcast. Where does one start?

Obviously this is no longer just kind of, oh, I know HTML, whatever, I learned HTML back in '98. It sounds to me that...and that's probably when I learned it. Probably before that, actually. Gosh.

But I haven't necessarily looked in detail at HTML since then. But it seems to me there's new elements. This cache stuff, the data store, there's substantial new things to learn about HTML. Where do you start?

DAVIS: Well, absolutely. And this is the real irony of the technology. You wouldn't sit down and say, wow, I'm interested in Grails, this web framework written in Groovy. I'm just going to sit down and start writing it.

You're going to read articles. You're going to read books. You're going to do that kind of stuff. As a Java developer. No one would give you your first job and say just learn it as you go going. You'll pick it up. You're a smart guy. But that's exactly what we've been forced to do with HTML all along. They say how hard can it be, no formal training, don't read anything, just pick it up as you go.

GLOVER: Yes.

DAVIS: That's why some of the HTML you look at is so frightening. The JavaScript that you find is copy and pasted. And it's a cancer that spreads across the Internet.

Someone has written crappy JavaScript code and then copy and pasted it in theirs and that gets copied and pasted around all that kind of stuff. With HTML5, this is the first time you might want to sit down and start reading and doing some actual studying on it.

It's not hard stuff. But it's really cool. And it is very different than what you're looking for. The best resource I've found is O'Reilly's HTML5 Book by Mark Pilgrim. It's available for free on the Web as well. If you go to [diveintohtml5.org](http://diveintohtml5.org). [Diveintohtml5.org](http://diveintohtml5.org). It's the entire content of the book free online. But I'm just trying to look up the official title of the book, the O'Reilly book.

GLOVER: Well, I'm actually on [diveintohtml5.org](http://diveintohtml5.org). And this is wonderful. This is wonderful. Excellent. Excellent. [ED NOTE: Site no longer available.]

DAVIS: HTML5 Up and Running. That is the official O'Reilly title. HTML5 Up and Running. But [diveintohtml5.org](http://diveintohtml5.org) is the online free version of that stuff.

GLOVER: Tell us about [thirstyhead.com](http://thirstyhead.com). You have some HTML5 training.

DAVIS: Absolutely. And I've...kind of practicing what you preach, eating my own dog food and everything. If you go to thirstyhead.com that's an HTML5 site. And you start watching the video interviews. The video interviews on the desktop, if you hit there and there, that's being served up by simple Flash player like YouTube and everything else will be served up. But if you go, hit it on a mobile device, an iPhone or an iPad or any of these kinds of things, it's actually using the new HTML5 video tag.

GLOVER: Okay. What's special about that?

DAVIS: The HTML5 video tag, what's nice is that's it's baked into the browser once again. We don't have to worry about plug-ins.

You think of it as the image tag. We just assumed that we've got an image tag in there and we can display pings or JPEGs or GIFs and we don't have to think twice about it. The video tag and audio tag both give you that same kind of capabilities. Allows you to kind of view video or audio natively in the browser.

Now, the browsers don't all support the same codecs right now, which is kind of a bummer. But the HTML5 that I'm serving up is using HTTP live streaming, which is really cool. And all of your Safari browsers are going to be able to view this. Not only desktop but also, as I said, iPhone and iPad and all the good stuff out of that.

But what's really cool about HTTP live streaming, is that rather than doing a progressive download, which is how YouTube works right now, you've got a big, let's say, 500-meg file up on the server.

And all the headers, all the metadata, everything that your player needs to know, the length of the video and the size and all that kind of stuff is in the very front of the browser, in the very front of the file.

So as the file starts downloading, your player has enough to begin playing and it's downloading in the background as it's playing. But that's why you get that buffering all the time. If the download isn't quite keeping...but it's one big 500 meg file.

With HTTP live streaming, what you do is you take that 500 meg file and you whack it up into 10 second segments. And so this is now streaming to your browser in 10-second segments.

GLOVER: Interesting.

DAVIS: So what's really cool about this is you don't need specialized software. You can serve this up off of a standard Apache installation, off of a standard Tomcat installation. I'm serving it up off an Amazon S3 data store.

GLOVER: How do you break it up into 10-second intervals?

DAVIS: There's actually...it's a media segmenter you download. If you're part of the Apple developer union, you get that kind of toolsets. But

they're open source toolkits as well that will allow you to kind of chop it up.

But what's especially cool about this is now we've got video coming off of plain old web server, nothing special about that. As I said, I'm doing it off of S3.

What's really cool is because it's streamed up like that, if I am on my iPhone and I'm in a Wi-Fi zone and I'm listening on a really nice high bandwidth connection and I walk outdoors and start walking down the street and flip over to 3G, the HTTP live streaming will flip to the bandwidth appropriate stream.

So I can have down simplified versions of this video and up simplified versions and versions based on the different things like that. There's no way you can do that with progressive download. It's one big 500-meg nut.

And so when you start playing it and you get out into the street and your fat pipe goes to a drinking straw, you know, that 500-meg file is still trying to squeeze through the drinking straw.

And HTTP live streaming solves that problem. Just the players seamlessly flips streams to the new streaming, to the new bandwidth appropriate stream and starts picking up.

GLOVER: You said this live streaming, though, is...it's supported by Safari or any of the Safari browsers.

DAVIS: So I'm conflating two things on you right now. So your confusion is well founded.

GLOVER: I'm always confused.

DAVIS: The video tag itself has nothing to do with HTTP live streaming. So let me start over. The video tag, what's cool about that is because the browsers don't all support the same codecs, you can start lining up your video in order and you can say here's the MP4 version of this video, here's the Aug version of this video, here's the Flash version of this video.

And the browsers are smart enough to start at the top of the list. If they know that codec, they'll start playing it. If they don't, they'll fail over to the next and the next down the line.

That's straight up HTML5 out of the box what you get. Now HTTP live streaming is an Apple-only technology right now but they've submitted it to EITF. So it's going through the standardization process right now.

If you've watched any of the Apple events, any of the Steve Jobs keynotes or any of that kind of stuff it's using HTTP live streaming. So anything that can play QuickTime, if you've got a QuickTime plug-in for IE or Firefox you can do HTML5 live streaming.

If you've got Safari browsers, or desktop or mobile, you can do that kind of thing. And HTTP live streaming is what whacks it up in those 10 second segments. Those combination of technologies are what makes things exciting.

Think about what we've talked about here, we've got new form elements. Local storage. Application cache. This video tag. I am super, super jazzed about what's going on in the HTML space right now. This is really, really exciting stuff.

GLOVER: Yes, you're exciting me. Going back to the live streaming and you mentioned Apple, and you specifically mentioned Steve Jobs. For those of us that were perhaps confused or wondering kind of about the whole Flash thing, you know, back when Apple said, hey, look we're not going to support Flash. It seems to me this is somehow related to this HTML live streaming.

DAVIS: It is kind of interesting. Flash served a great purpose. I'm not going to knock Flash at all. It's what brought video to the Web in that kind of XHTML era of things, since there wasn't a native video element, since there wasn't a lot of this native support. We had a plug-in that sprung up really nicely. And the plug-in is all but ubiquitous. It's all over the place.

I say all but, because it's not native on the iPhone. It's not native on the iPad, these kinds of things. So what makes me so excited about HTML5 is we have a real contender. We can get back to pure web development that truly is write once, run anywhere, that gets us out of depending on proprietary plug-ins and gets us back to native, open standard technology.

GLOVER: Okay. And I guess you had mentioned earlier CSS, JavaScript, HTML. Just to clarify for everyone listening, those are all three, quote/unquote, standards that are addressed by different, shall we say, organizations.

So there's no relationship between HTML5 and let's say CSS 2.0 or whatever the current version of CSS is, correct?

DAVIS: When we're talking about HTML5, a lot of times people will talk about CSS 3 in the same breath. They are two different standards. But most browsers are providing really decent support for both of those kinds of things.

So CSS 3 gives us things like rounded corners, finally that's baked into the browser right now.

And 3D transformation. So when you're looking at that really cool cover flow effect that you get in iTunes, things like that, you can do that native in the browser now with CSS 3D transforms and JavaScript to handle the mouse-over events and things like that, and HTML5 to handle the actual elements themselves.

GLOVER: Cool. And so you had mentioned that there's a lot of websites currently like Apple and Google and then obviously you mentioned thirstyhead.com.

Where else...are there some cool sites out there that people should start looking at so they can see some of this stuff in action?

DAVIS: There's some really great websites. If you begin Googling just HTML5 tutorials, one that jumps to mind is html5rocks.com. So there are a lot of neat tutorials there.

And on other technologies we haven't had a chance to talk about yet, the geolocation API. The fact that in your browser now on your mobile devices especially, you know, they can report where your latitude/longitude is.

So imagine going into Google and just searching on Caribou Coffee. Not Caribou Coffee near Denver, Colorado, just typing in, baking in these kinds of location aware kinds of things. Geolocation API is baked into that kind of HTML5 umbrella as well.

GLOVER: Really? That's pretty interesting.

DAVIS: Canvas is something that's baked into this. So Canvas is where you start looking at...people have implemented things like Doom in the browser in the Canvas.

GLOVER: Doom.

DAVIS: Doom or Quake, one of those old school...

GLOVER: Doom precedes my knowledge of HTML. That was like college.

DAVIS: Imagine having that implemented like in your browser. So the canvas are really cool. SVG, scalable vector graphics, is now baked into the spec as well. Even Internet Explorer 9 is going to have SVG support in it as well. So there's just a cluster of technologies you can find. So html5rocks. HTML5 tutorials. Diveintohtml5.org, a lot of resources.

GLOVER: Seems to me that clearly the mobile device revolution, I'm looking for my phone, never mind, people on the podcast can't see the video here, right?

But clearly the ubiquitousness of computing, you mentioned the tablet and the iPhone, the iPad. They seem to be driving a lot of this innovation. It appears, right?

DAVIS: It's so funny, because mobile browsers used to be a kind of watered-down version of what you got on the desktop.

GLOVER: Yes, WAP, remember that?

DAVIS: Absolutely, WML.

GLOVER: Underwhelming.

DAVIS: Yes. Turning tables. The really innovative stuff that we're beginning to see is coming to the iPhone, coming to the iPad, coming to the Galaxy tablet, coming to Android phones, coming to a lot of those kinds of things, and almost kind of lapping what the desktop browsers are able to provide.

GLOVER: And earlier today, earlier in this podcast you mentioned that you're working on set-top boxes and televisions that now embed browsers, right? That's pretty darned cool.

DAVIS: Oh, yes. I mean, so Netflix, it sends silver platters to you in the mail. You're like that's kind of cool.

GLOVER: That's old cool.

DAVIS: Yes, exactly. And Netflix goes to watch instantly, where you can begin watching on your laptop. Now, that's cool. That got me really interested.

But the fact now that I can watch Netflix on demand baked into my TV or baked into my Blu-Ray player, baked into my Apple TV, you know, or baked into my Wii or baked into my PS3 or baked into my Xbox or iPad, again write once run anywhere, being so pervasive.

What's so cool I can begin watching a movie on my TV, hit pause, put the kids to bed, read them a story, whatever I need to do, go up and sit on my bed on the iPad hit resume and it will pick up right where I was playing. Completely different devices, completely different rooms in the house.

There's just so much coolness available to us once we get beyond what you can do to a desktop computer that's literally tethered down. Once you start looking at these mobile things, it totally changes the way you consider the Web and consider what these devices can do.

GLOVER: So let me ask you this. So for our listeners, you and I are having this conversation shortly before the new year. Chances are this would be published sometime hopefully early January.

So it's perfect timing then to ask you to pontificate a little, given that we've talked about HTML5 which is kind of new, hip, cool. We've talked about all these mobile devices. What do you see as 2011 as we approach this and as we listen to this when we're in it, what should we be looking at? What's next? Where should smart developers be looking?

DAVIS: Thirstyhead for years and years has focused on Groovy and Grail's development. And Groovy is a dynamic language, targeted at Java developers, but meant to really loosen things up, listen, hey we can add dynamically add new methods on the fly. We can dynamically add new fields to your classes, doing these kinds of things.

JavaScript has been, I mean, in this resurgence, you know, since 2005 and the AJAX revolution. But JavaScript is this same notion, this kind of dynamic, very flexible language that's going on.

I've been jazzed about HTML5, but I've also been jazzed about NoSQL. And I love the fact that even push back, people are jaded about NoSQL, they're beginning to get jaded about Web 2.0 and cyberspace and all these kinds of things.

But if we can get past the NoSQL moniker for a little bit, the fact is it's a flexible dynamic data store. And so whether you're looking at CouchDB. CouchDB has me really jazzed, because CouchDB is a schemeless key value store that uses RES to interact with it.

I do HTTP puts to get data into it. I do HTTP gets to get data out of it, and do HTTP deletes to delete stuff out of it. And I do HTTP restful interfaces and what's it returning? It's returning JSON.

So something like that a flexible data store that speaks HTML down to the metal, coupled with HTML running on phones, tablets, computers, TVs, you know, on and on like that.

And even that local data store that I was talking about, local data storage and HTML5, that's this flexible key value store, I think that what we're seeing is a lot of people really loosen up and we're really seeing this dynamic nature.

And my technology stack of choices is Groovy and Grails. But of course I could be talking about ruby and rails or I could be talking about Python and Django, all of these technologies, these dynamic technologies that have these flexible ways of not only expressing themselves in code, but we're getting to the place where we can have this flexible way of persisting data.

I see this trend really expanding in 2011.

GLOVER: Certainly, yes. And HTML5 itself is, I guess when you would say kind of like server side language agnostic, doesn't matter if you're a Rails guy, Grails guy, Microsoft guy, doesn't matter, right?

There's nothing specific to those languages in HTML itself. It's from that standpoint a ubiquitous, quote/unquote, technology that we'll all using and are probably are using without even knowing it.

DAVIS: Exactly. But, I mean, being able to move comfortably between these technologies, if I've got a Grails server that's written in Groovy that's serving up HTML5, that's taking advantage of app cache and local storage and these kinds of things, talking to a CouchDB server behind the scenes and everything, you've got a really rich, really flexible environment that you can kind of bend to your will and really make, do amazing things.

GLOVER: This is awesome. This is awesome. Scott, I know I speak for everyone listening when I say thank you very much for your time. And your

excitement around HTML5 has me very excited. I'm on [diveintohtml5.org](http://diveintohtml5.org) right now. I'm going to relisten to the podcast to get the other references you mentioned. Any parting words on HTML5, any last thoughts in terms of where we should go look for information?

DAVIS: The most important thing I want to get across is that HTML5 is right here and right now.

GLOVER: Clearly.

DAVIS: If you've got a smart phone in your pocket, it knows HTML5. If you've got a tablet, it knows HTML5. If you've got a modern browser, it knows HTML5.

But when you start looking at the Googles and the Apples and Mozillas of the world and folks like that are using it, if you're a developer you need to dive in now. It's not a wait-and-see technology. It's right here. That's what makes it so exciting.

GLOVER: I guess after this conversation I kind of see...I look at HTML in a different light now. Again, I tried to kind of enunciate or whatever earlier, articulate that to me HTML is just this kind of nasty markup language that you never really learn, you just kind of knew about it.

Like you said, you looked on the Internet. How do you make a form? Oh, yes, here's one, I'll just steal it. I'll view source, copy/paste. And now I have this excitement around like, well, I've got to learn this, because I'm working on web applications and I'm not using HTML5. But clearly I need to be.

DAVIS: Clearly. Clearly.

GLOVER: I'm old school. So this is awesome. So, again, I say thank you. I know our listeners will say thank you and thank you for all the references.

This is very exciting. And we'll look for more resources from you hopefully on IBM developerWorks. We'll continue to look at [thirstyhead.com](http://thirstyhead.com) for additional videos articles and obviously training on HTML5.

DAVIS: Outstanding. Thanks for having me, Andy. Always a joy talking to you.

GLOVER: Yes, sir. Thank you.

developerWorks: That was Andy Glover, and this is the developerWorks podcast. Follow us on iTunes and at [ibm.com/developerworks](http://ibm.com/developerworks). And specifically keep track of Andy's interview series at [ibm.com/developerworks/java](http://ibm.com/developerworks/java). Thanks for listening.