Session 108

What You Must Know to Select the Best Mobile Learning Development Tools

By Joseph Ganci

www.elearningjoe.com

joe@elearningjoe.com
What is HTML5?

Isn’t HTML5 just the latest version of the HyperText Markup Language, which has been around since the early 90s, the brainchild of Tim Berners-Lee, a Swiss contractor at CERN (Conseil Européen pour la Recherche Nucléaire”, or European Council for Nuclear Research). The early HTML was much simpler, allowing for 18 HTML “Tags” that allowed in the initial version. The hyperlink tag, was new. The rest were borrowed and modified from the earlier SGML, an in-house Standard Generalized Markup Language (SGML)-based format used at CERN to format documents. SGML is still in use today in the technical documentation field.

HTML’s features and power has increased with each new version, but version 5 is a true leap ahead. For the first time, we have certain tags that allow for the sorts of features that until now required the use of a plug-in, such as Flash and QuickTime. Now we have all sorts of new strengths in HTML5 that help us avoid plug-ins.

HTML5 is really a combination of three technologies.
Why is HTML5 important for mobile?

Simply put, most mobile devices don’t support Flash and HTML5 is the only real alternative. Why don’t mobile devices support Flash? Apple CEO Steve Jobs announced that it would not be allowed on iOS devices: iPhones, iPods and iPads.

Steve Jobs said: “Flash was created during the PC era – for PCs and mice. Flash is a successful business for Adobe, and we can understand why they want to push it beyond PCs. But the mobile era is about low power devices, touch interfaces and open web standards – all areas where Flash falls short... New open standards created in the mobile era, such as HTML5, will win on mobile devices (and PCs too).”

This was the beginning of the end. Adobe also found it very difficult to keep up with the many different Android hardware manufacturers, each offering a somewhat different take on the Android experience, hence there was no way to make a Flash version that would work flawlessly on all devices. In November 2011, Adobe announced it would stop developing Flash for mobile devices.

As mentioned, the only real alternative is HTML5.
Important Lessons I’ve Learned about Responsive Design

Simply put, too many are assuming that they need to create a responsive project when working on a new project. They may have heard that while nobody is using their devices now, in a year or so they will be and will likely want to access your lesson then.

1. It’s rare for any eLearning project to be used for years to come. It will probably need to be updated before then. It will probably take longer than expected to decide to let learners use their mobile devices (that has been my experience). Most of my clients are still not considering responsive. Your experience may differ.

2. Creating a responsive project will always take longer than one that is not, even if you follow all the best practices illustrated in this document and elsewhere. It doesn’t have to be a lot more, but it will be more. If you don’t need to allow devices to access your learning now, you may want to reconsider.

3. If there is a chance that you will want to allow devices to access your learning later, but not for a while, understand that you can take the slides you’ve created and import them into a new responsive project, then adjust the objects on the stage as necessary in each view. It’s not ideal, but it works.

4. HTML5 does not necessarily mean responsive, though responsive always means HTML5. What I mean by that is that you can create a standard file, if you design it correctly, and publish it to HTML5 so that I will play on mobile devices as well. This is much simpler to create than a multi-view responsive project, but you’ll want to be careful to keep mobile in mind when you do this. For instance, make sure buttons and other interactive elements are large enough to touch with a finger.

You cannot publish a responsive project out as Flash, though you can copy the slides out of a responsive project and paste them into a non-responsive blank project and then publish to Flash.

5. Speaking of HTML5, if you intend on using any Flash elements in your file, forget it. They won’t play in HTML5 and so they won’t play on mobile devices. Especially if you’re trying to update legacy files, look out for any Flash animations you may be using.

6. You don’t have to use all five views. If you want to limit learner to just the desktop view and a mobile landscape view, you can delete the views you don’t want. You should know what your learners need and not spend time finagling with views that they’ll never see.

7. You can resize each view to whatever width and height you wish. Know what devices you need to support and give learners the maximum height and width that are safe to use without going too big and forcing them to scroll or going too small and robbing yourself of stage real estate you can use.
Understanding Responsive Design

When websites were first created, smart phone and tablets were nowhere to be found. All web sites were built for desktop and laptop computers.

Once the mobile revolution started and folks tried to view websites on their smart phones and tablets, two revelations occurred:

1. Flash was not going to work well on those devices. It had become large, too large to run efficiently in the limited memory of mobile devices. It was also prone to security flaws.
2. Smart companies had to start paying attention to delivering on mobile devices.

Around 2006 or so, companies started to solve the problem by building two websites: one for desktop and one for mobile. You still see this approach used by many. Typically this is evident when your phone is automatically routed to a URL that starts with m. such as m.disney.com.

This has started to fall out of favor and instead most new websites are being built using the responsive design approach, and many older sites are rebuilding their sites using responsive design as well. (See this 2012 blog about why these sites had to die: http://www.mobify.com/insights/6-reasons-m-dot-websites-are-dead-ends/).

In short, responsive design sites adapt themselves to the width of your browser. If you’re running a browser on your mobile phone, it will be narrower than one you run on your tablet, which will be narrower than one you run on your desktop. In each case, the website automatically customizes itself to the browser. This is not magic: it takes some work to design a site this way.

eLearning typically runs in a browser. It too can be designed responsively or not. However, learners won’t have much patience with learning that simply shrinks down to the size of a cell phone screen. In most cases, it makes it difficult to read and difficult to click any included buttons because they will be too small and crowded together.

Creating a responsive design website requires an excellent understanding not just of HTML5, the standard for mobile and increasingly for desktop delivery of websites, but also of JavaScript and CSS, which stands for Cascading Style Sheets. They all fall under the HTML5 umbrella.

The Ins and Outs of Creating Responsive Design

Creating responsive design projects mainly means knowing how to the following:

1. Insert screen objects in ways that make them look correct across views. This may include text, images, video or any other object.
2. Decide when an object does not belong in one of the views and how to leave it out of that view.
3. Decide how to tie objects to other objects or to screen edges.

Let’s see how responsive design works:

1. You start with three breakpoints or viewpoints:
   a. Desktop, which can be up to 1,280 pixels wide.
   b. Tablet Portrait, which can be up to 1,175 pixels wide.
c. Mobile Portrait, which can be up to 1,075 pixels wide.

2. You can also enable two more:
   a. Tablet Landscape, which can be up to 1,220 pixels wide
   b. Mobile Landscape, which can be up to 1,125 pixels wide.

3. However, the above values are the maximum pixel widths. It’s always best to know what devices you need to support within your organization or within your client’s organization.

You will see this bar above the stage when you first create a responsive design lesson.

![Responsive Design Lesson Bar]

<table>
<thead>
<tr>
<th>Desktop</th>
<th>Tablet</th>
<th>Mobile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Portrait</td>
<td>Portrait</td>
<td></td>
</tr>
</tbody>
</table>

1. Clicking the first + sign will add a Tablet Landscape version.
2. Clicking the second + sign will add a Portrait Landscape version.

**Avoiding features that will cause problems in your mobile design**

There are several considerations that you must keep in mind when creating lessons for mobile delivery.

**When Delivering to both Desktop and Mobile Learning**

Certain interactions and features work on desktops but not on most mobile devices. What are they?

**Any kind of rollover state!**

Why? Well, how do you roll over an object with your finger? Simply put, you can’t. You can’t hover your finger above an object and make it interact. As soon as you touch the object, it registers as a click. On some devices, holding your finger for a few seconds on a mobile screen object will register as a rollover, but this is rare and not consistent. Therefore, rollovers should be avoided in mobile projects.

Use clicks or other interactions instead.
SWF (Flash) Animations

This is kind of a no-brainer, but it trips us up sometimes. When creating eLearning, we are used to the idea of inserting an animation. Whether it’s a cool example of a business process or showing how water flow through a pipe, we’ve become quite enamored of SWF animations.

The alternatives? Animated GIFs and HTML5 animations work on both desktop and mobile. This may mean that you can’t use all the nifty animations created in the past and there’s no easy way to convert a SWF without degradation, so for new projects, shift away from Flash animations and start building HTML5 animations instead.

So What About Mobile Features?
You know you can take advantage of mobile features in most apps that you run on your tablet or phone, including:

1. **Pinch and zoom**, where you use two fingers, bringing them together to zoom out and pulling them apart to zoom in.
2. **Geolocation**, so that apps know your whereabouts to recommend local restaurants and such.
3. **Swiping**, where you can navigate left and right (or up and down) by dragging your finger across the screen (as you often do with a library of photos)

All Right. What Questions Must You Ask Vendors of Mobile Learning Applications?
Use this handy guide to determine what questions you should ask.

---


Contents © 2016 by Joseph Ganci – All rights reserved
Note the bottom right question: “How much flexibility do I have in developing responsive?” Along with that question you should ask the usual ones:

1. How much does a license to your tool cost?
2. Are there any additional costs, such as cost per learner?
3. Where do I find your online community of users?
4. What version are you currently in?