

# Transitioning Computer Courseware to Mobile Web Apps

Thomas F. Fletcher, DVM, PhD

RE: <http://vanat.cvm.umn.edu/mNeuroConcepts/>

## *Synopsis:*

To exploit the learning potential of mobile devices, we are converting existing web based computer courseware into mobile web apps. The conversion necessitates screen redesign and the selection of a new development platform. Screen redesign is discussed in the context of particular courseware: Neurobiology Concepts Checker. An overview of the jQuery Mobile development platform is presented.

## *Note:*

This PDF document is a chapter in the eBook edited by A.H. Duin, E. Nater, & F. Anklesaria. (2012). *Cultivating Change in the Academy: 50+ Stories from the Digital Frontlines* at the University of Minnesota in 2012. The eBook is free: <http://purl.umn.edu/125273>

## **Minnesota Veterinary Anatomy Web Site**

Our Minnesota Veterinary Anatomy web site (<http://vanat.cvm.umn.edu/>), which received 97,842 visits from 163 countries/territories in 2011, serves as an umbrella site for accessing over two dozen individual courseware web sites dealing with carnivore gross anatomy, embryology, and neurobiology. We recently added a mobile device courseware section to the web site to accommodate our web app development.

To exploit the learning potential of smart phone and tablet mobile devices, we have begun transitioning our existing computer-based courseware web sites to mobile web apps. This Chapter explains our approach and recommendations regarding web app development.

## **Mobile Device Web Apps**

Mobile devices are becoming a dominant technology with significant potential as learning tools. Because smart phone and tablets are so convenient, ubiquitous, and advantageous for web technology, their utilization by students is rapidly expanding.

To exploit the Veterinary Anatomy learning potential of tablets and smartphones, it makes efficient sense for us to transition our existing computer-based courseware assets to mobile web apps.

In contrast to native apps, **web apps** are web sites designed for mobile browsers. Compared to the effort and skills required to develop and market native apps for various device platforms, web apps can be produced, tested and delivered with relative ease.

## Screen Design and Development Considerations

In preparing to convert existing web courseware designed for computer screens to mobile device web apps, primary considerations are re-design for small screen size and the selection of the development environment.

Large screen layouts must be redesigned in a meaningful way for the small screen mobile domain. Also, mobile devices use touch screen interfaces that have different requirements than the traditional mouse-driven interface.

Our most interactive existing courseware depends on browsers using the Flash Player plug-in (.swf files) for interface animation and XML processing. Because Adobe has announced that they will no longer support the Flash Player on mobile devices, the development environment that we used for interface animation and XML processing had to be replaced. We chose jQuery Mobile and Dreamweaver as the development environment and we are quite pleased with the results.

The following discussion focuses on the transition of one neurobiology courseware product from a web site designed for a computer screen to a mobile device web app. After which, an overview of jQuery Mobile development environment is presented.

## Neurobiology Courseware Description

(<http://vanat.cvm.umn.edu/neuroConcepts/>)

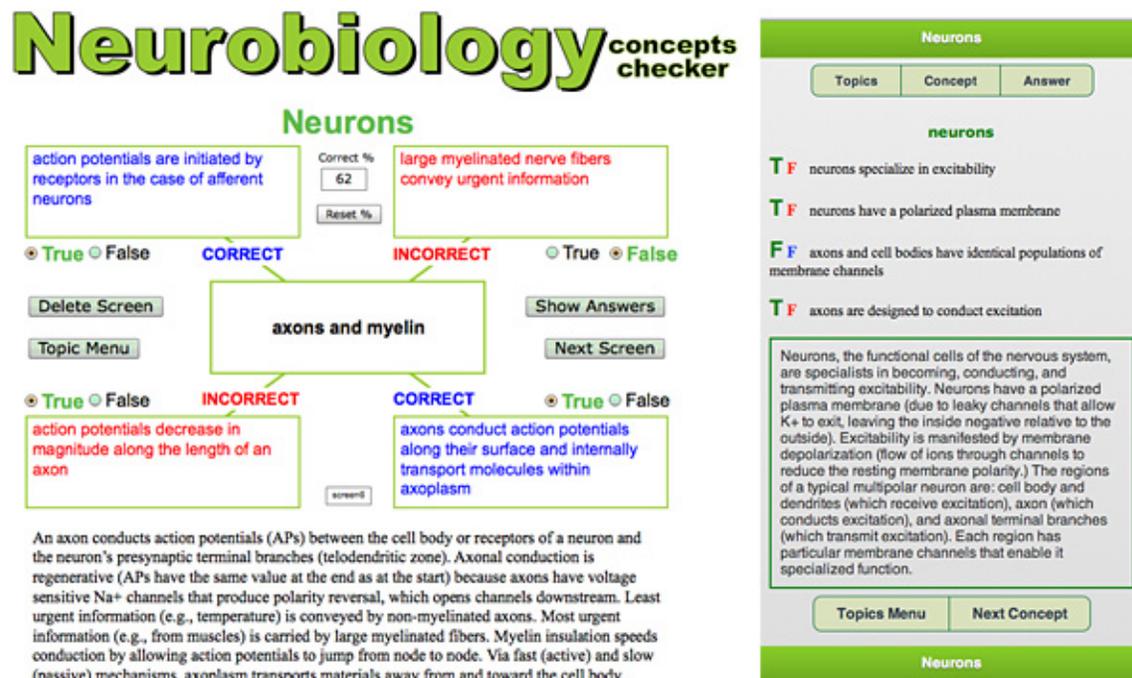
Our existing computer-based courseware, *Neurobiology Concepts Checker*, is a web site that gives students an opportunity to clarify their conceptual knowledge of veterinary neurobiology. As students confirm their conceptual understanding, via self-assessment, they also develop personal reassurance that they know the required subject matter. Overall the Concepts Checker presents 150 concepts within 14 topical categories. Each concept is associated with four True/False statements and a Concept Explanation paragraph. Concepts are stored as individual XML nodes per XML topic file.

For the *Neurobiology Concepts Checker*, students select from a menu of 14 topics (e.g., Neurons, Cerebellum, Hypothalamus, etc.). Per topic, a series of concept screens is presented randomly. Each concept screen displays a centrally placed concept statement surrounded by four randomly positioned phrases that are either true or false. Random display was used to make repeated questions (XML nodes) appear novel.

Typically, a student would read the concept statement and answers the four questions by clicking **True/False** buttons. When the student clicks a **Show Answers** button, True/False answers are graded and a paragraph explaining the concept is displayed. A **Delete Screen** button (to be used when a concept is sufficiently known) eliminates the concept from the pool of randomly displayed concept screens. A **Next Screen** button randomly presents the next concept and a **Topic Menu** button presents the menu of 14 topics.

Thus, for this computer-based courseware, the screen is designed to simultaneously present all concept-related information in one view. Students are able to see all four

True/False questions at the same time and their answers can be changed based on reconsideration of total screen information. For the mobile web app version of this courseware, re-design to a small screen was necessary (see Fig. 1).



**Fig. 1.** Comparable concept screens from the computer screen web site (left) and the mobile web app (right). Each screen shows what is displayed after the **Show Answers** or **Answer** button is clicked/tapped.

### Mobile Web App Description

(<http://vanat.cvm.umn.edu/mNeuroConcepts/>)

In re-designing the *Neurobiology Concepts Checker* for small screens, we opted for a chronological approach to conserve screen area. A student sees only one True/False question at a time. The question must be answered immediately and it is graded before the next question is revealed. At any time the **Answer** button may be tapped to show the questions and their correct answers along with the explanation paragraph.

**Concept** or **Topics** buttons can be tapped at any time to view the next concept or revert to the Topics Menu. The Delete Screen button provided for the computer web site was eliminated in the web app to conserve space. For each topic selected in the web app, the concept screen that appears initially is randomly chosen. But, unlike the computer web site, the display order of following concepts and order of True/False questions per concept is not random. Randomness could be added in the future, but the chronological approach makes it easier to have repeated questions appear novel.

## jQuery Mobile Overview:

(<http://jquerymobile.com/demos/1.1.0/>)

The jQuery mobile framework, along with Adobe Dreamweaver, was chosen as our development platform for web app production, and we are quite pleased with the productivity they offer us. Also, all mobile browsers support HTML5, which offers a consistent browser interface along with additional features such as video and audio tags that we did not use. (HTML5 discussion is beyond the scope of this Chapter).

The jQuery Mobile framework targets a variety of mobile platforms and offers an assortment of enriched interface elements including navigation bars, sliders, accordions, layout grids, buttons, etc. The framework is freely available and well documented. The documentation includes basic page templates that are useful for getting started.

The JavaScript and CSS framework files that comprise jQuery Mobile are automatically downloaded by the browser, when `<script>` tags linking to a public Content Delivery Site are included in the HTML page. The script tags can be copied from template files.

The jQuery Mobile approach is particularly clever. The mobile browser downloads one physical HTML page into memory but it converts that single download into multiple mobile device pages based attributes that the developer can assign to `<div>` tags (e.g., *data-role = page*, *data-role = header*, *data-role = content*, *data-role = footer*, etc.). Also, the developer can assign *data-role = button* attributes to paragraph (`<p>`), list (`<li>`), or anchor tags (`<a>`) and the framework automatically converts them to elements that look and behave like buttons.

In other words, the developer only needs to provide a skeleton structure and the framework automatically provides the detailed appearance and behavior of a full featured web app. The framework achieves this transformation by augmenting the developer generated *data-* attributes with a variety of additional class attributes. The additional classes are inserted into the HTML code by framework JavaScript during the loading process before the code is interpreted and displayed by the browser (see Fig 2.) This automatic code generation greatly simplifies interface development requirements for the web app designer.

Finally, the jQuery Mobile web site offers access to a graphic tool (ThemeRoller) which generates a CSS file that customizes your web app appearance. Web app themes involve font-family, drop shadows, corner radii, colors for headers/footers and backgrounds, color gradients for button states, etc.

If you want to develop web apps, jQuery Mobile will simplify the process!

```

<!-- Start of first page: #one -->
<div data-role="page" id="one">

  <div data-role="header">
    <h1>Multi-page</h1>
  </div><!-- /header -->

  <div data-role="content" >
    <h2>One</h2>

```

```

<!-- Start of first page: #one -->
<div id="one" data-role="page" data-url="one" tabindex="0" class="ui-page ui-body-c
ui-page-active" style="min-height: 898px;">

  <div data-role="header" class="ui-header ui-bar-a" role="banner">
    <h1 class="ui-title" tabindex="0" role="heading" aria-level="1">Multi-page</h1>
  </div><!-- /header -->

  <div data-role="content" class="ui-content" role="main">
    <h2>One</h2>

```

Fig. 2. Excerpts from an HTML page as submitted by a developer (top) and as loaded by the browser following automatic attribute edits inserted by the jQuery Mobile JavaScript framework (bottom). The top image shows three attributes inserted by the developer to define a mobile page (underlined in green). The lower image shows the extra content that was added automatically by the jQuery Mobile framework to generate the appearance and behavior of a mobile web app page.

### Addendum:

*Neurobiology Concepts Checker* is one of a number of optional courseware web sites that we offer first-year veterinary students taking CVM 6120 Veterinary Neurobiology, which is taught during the first half of Spring semester. We have not tracked usage of optional courseware in CVM 6120, relying instead on anecdotal feedback regarding courseware value to individual students. Because the mobile version of *Neurobiology Concepts Checker* was launched after CVM 6120 ended this Spring (2012), student feedback is not available. (Via Google Analytics we know that the computer screen version of *Neurobiology Concepts Checker* received 544 visits in 2011.)

Tom Fletcher <[fletc003@umn.edu](mailto:fletc003@umn.edu)>



Professor of Veterinary Anatomy, College of Veterinary Medicine.  
Develops anatomy, embryology and neurobiology web-based courseware.