ARIA Week 5 Tutorial:

Creating an offline capable application

Offline capable applications allow us to access important information on our mobile devices or desktop.

PART I

In Part I, we're going to write a file to detect if we're online or offline.

1) Using your choice of editor, create a new blank HTML page called testOffline.html

2) Give the page a title and add a heading of whatever you'd like

3) Add a <p> element on the page which says "Current network status:". Inside the <p> element, add a <span> element with the id of "status".

4) Inside the span give some default text. Something like "checking status". This text will show up while our function is running.

5) Create a <script> block after your <p> elements to hold some JavaScript.

6) Create global variable inside your script tags. Set the variable to be equal to the span element you created.

If you don't know how to do this, look up "getElementById" on w3schools.com for the syntax.

7) Now that we have the element, we need to create a function to check the network status.

Create a function that takes no arguments.

8) The method "navigator.onLine" will return true if your browser has a network connection and false if it does not. Call that method from inside your function and set the answer (true or false) to a new variable.

9) Lastly, we can change the value of an element like a <p> element or a <div> element or a <span> element by setting its innerHTML property. Set your global variable's innerHTML to the value returned by navigator.onLine. This should update the page.

10) Now run the example in the browser. Make sure it is telling the truth! Disconnect your network and reload the page. Is it working correctly?

PART II

You're going to create an app manifest file to ensure your application can work offline.

1) Download starter files **offlineExample.html** & **clock.js** to your desktop.

2) Double-click the file **offlineExample.html** to make sure you are able to see the current time in your browser.
3) Now you need to create an app manifest file. Create a new blank document on the desktop and save it as **“clock.appcache”**

WHAT IS A MANIFEST FILE?

The first line of every cache manifest file is this:

```
CACHE MANIFEST
```

After that, all manifest files are divided into three parts: the “explicit” section, the “fallback” section, and the “online whitelist” section. Each section has a header, on its own line.

If the manifest file doesn’t have any section headers, all the listed resources are implicitly in the “explicit” section. Try not to dwell on the terminology, lest your head explode.

Here is a valid manifest file. It lists three resources: a CSS file, a JavaScript file, and a JPEG image.

```
CACHE MANIFEST
/clock.css
/clock.js
/clock-face.jpg
```

Now, knowing what you know about basic cache manifest files, add the right information to your cache.manifest file so that the browser can use your clock application offline.

Now with that file saved, try reloading your clock to make sure it still works.

If we put this on a server and tried to access it now, what would happen?

If you lost connectivity to the server, what would happen?

The answer is not a lot at this point. We haven't linked the manifest file and our clock page yet!

Edit the offlineExample.html page so that it includes the cache manifest file. If you don't remember how to do this, take a look at http://www.w3schools.com/html5/att_html_manifest.asp

TEST IT

If you have a server you can upload this to, go ahead and do that. Upload all of the files and then try and access offlineExample.html.

If that works, turn off your network access and try to access the file again. It should show you the exact same results as the files have now been cached locally.