Announcements (0:00 – 7:00)

More Ajax: XML, Closures (7:00 – 38:30)

- See ajax4.html
- Now the user may input a comma separate list of symbols and get quotes for all of them
- Using live HTTP headers we see that there was a request to quote3.php?symbols=yhoo%20msft%20goog
- If we actually go to the webpage the showed up in HTTP headers, we get a page full of XML
- XML is a markup language in which you tag data in a way that is convenient to you so that another program knows how to understand it
- Here we see tags for price, high, low, etc.
- See quote3.php.
- The first thing to notice is that there is now code to "massage" the input into CSV form in case different users input data differently
- In particular, we would like to tolerate symbols separated by commas or spaces, and any casing
- This is done by replacing all commas with single spaces (str_replace), moving all characters to upper case, splitting on whitespace (preg_split), and finally putting commas in all the places where whitespace previously was (implode)
- Next is a header indicating the content-type, or preparing the browser for what is to come next
- Then we get the quote from Yahoo! Finance using fopen
- This gives us back a list of rows containing comma-separated values
- In fact, we can use Firefox to go to the URL that we are fetching the quote from and we see that it gives us back a CSV file that we can look at in Excel
- Now we take that CSV file and generate the XML using a loop:
    - Get each row using fgetcs
    - Index into the row using array notation
    - The function htmlentities to rewrite something with symbols like AT&T as AT&T
- The XML is then processed by ajax4.html
- We see that the function quote() is just like the one we used last week to fetch the quote up until the quotes come back, but the handler is different
- In this handler, we generate some XHTML using the XML
- We can think of the XML as a tree, and we can think of reponseXML as a pointer to this tree
- This tree may have many different children marked by many different tags
- When we get the elements with tag "quote" this gives us back an array of nodes marked by the tag "quote"
- Then we can print out the contents of each quote with appropriate formatting and spacing

- We build up the XHTML that we would like to print using a string xhtml
- Notice that if we do view source we only see the javascript, not things like <bold>Yahoo!</bold>. This is because the XHTML is being written client side.
- Now see ajax5.html
- In this we make a text area that will serve as a debugging window and give it the id "debugging"
- Then, when the XML comes back, before printing out the quotes, we dump the contents of the response text into document.getElementById("debugging")
- This is a useful debugging feature; now we can see what the XML looks like when it comes back
- Next look at ajax6.html
- Recall that in ajax4 we set the event handler by saying request.onreadystatechange = handler and then defining handler below
- In ajax6, we can simply set request.onreadystatechange equal to the function definition
- This is called a closure, because one function is enclosed in another
- We might do this because we don't intend to reuse the function, or because we want the inner function to be able to access the local variables of the outer function

Intro to Google Maps (38:30 – 57:30)

- Lucky for us, Google Maps API is in javascript
- See map1.html, copied and pasted from Google Maps tutorial. This is all that is required to embed a map in a webpage
- At top we include a piece of source, sort of like #include in C
- In the load function, we check for comptaibilty, and then plant the map wherever there's an element with id "map"
- In map3.html, we add a control called typeControl and add it to the map
- Notice that map is an object, or a chunk of memory with both data and methods associated with it. addControl() is a method that takes a pointer to the control.
- We could also just stick in the call to new and the constructor in the call to addControl, which we do with the second control that we add
- Displaying map3.html, we see that we now have some useful controls
- In map4.html, we plant a marker by making a point, centering on that point, and then creating a marker and overlaying it on the map
- Recall in Scratch that we could broadcast and listen for events in order to make certain functionality be executed when some particular event occurred
- In javascript we add an event listener for the click event using the method addListener
- Now, when the marker is clicked on, an info bubble pops up with a Wikipedia link
- In map5.html, instead of hard coding something to appear in the info window, we put dynamic text: we grab the current price of the Google stock and display that

- We do this by grabbing the same XML we grabbed in our ajax stuff on the click event, and then getting the price out of it
- We can make an ajax call using the GXmlHttp library provided by Google

Compilation and Assembly (57:30 – 70:00)

- Compiling means taking source code and outputting assembly language
- Assembly language is not zeroes and ones; it's still human readable
- Assembly language is then translated to zeroes and ones ("assembled")
- Finally, this output is linked to other compiled files (libraries that you're including) and merged into a single binary
- For instance, we can output the assembly corresponding to hello.c using the –s flag on gcc
- Notice that the hello.c assembly makes a call to printf
- This means that the object code needs to be linked up with the object code containing printf's definition
- We indicate to gcc that this needs to be done when we say #include <stdio.h>
- This tells gcc to go and also compile and assemble stdio.c
- Errors like "undefined reference to foo" mean that you have failed to tell gcc where a function is defined
- (Actually standard libraries are precompiled into shared libraries so that they don't need to be compiled again and again.)
- Registers are temporary memory on a CPU. There are usually on 16 or 32 on the CPU.
- Program counter keeps track of where you are in the program