LISTEN UP's

<u>VERY VERY VERY</u>

QUICK GUIDE TO JAVASCRIPT

BY: KIRSTEN SZETO AND LILLIAN TRAN

OVERVIEW

JavaScript is one of the most (if not *the* most) popular scripting languages. Supported by all web browsers, it's the standard for client-side scripting. The main purpose is to make websites more interactive, or more hands-on for things like animated graphics, etc. Other uses include creating pop-ups to display warning messages (which is something we'll dive into in a little bit), or manipulating user input and displaying the results.

Keep in mind: JavaScript can also be used server-side. While client-side typically responds to user events because it is run on the client machine, server-side allows an application to communicate with a larger database because the code is run on the server (as the name implies). It's typically used to create more dynamic pages and it's more secure for data because the source code is not visible to the user.

JAVASCRIPT AND JAVA

The good news is that JavaScript and Java are somewhat similar. JavaScript mostly follows Java syntax (which is why it was renamed from LiveScript), but there are a few key differences. JavaScript doesn't have as many requirements. This chart from Mozilla[1] touches on some of the most important ones:

JavaScript	**Java**
Object-oriented. No distinction between types of objects. Inheritance is through the prototype mechanism, and properties and methods can be added to any object dynamically.	Class-based. Objects are divided into classes and instances with all inheritance through the class hierarchy. Classes and instances cannot have properties or methods added dynamically.
Variable data types are not declared (dynamic typing, loosely typed).	Variable data types must be declared (static typing, strongly typed).
Cannot automatically write to hard disk.	Can automatically write to hard disk.

[1] https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction#what_is_javascript

CODE

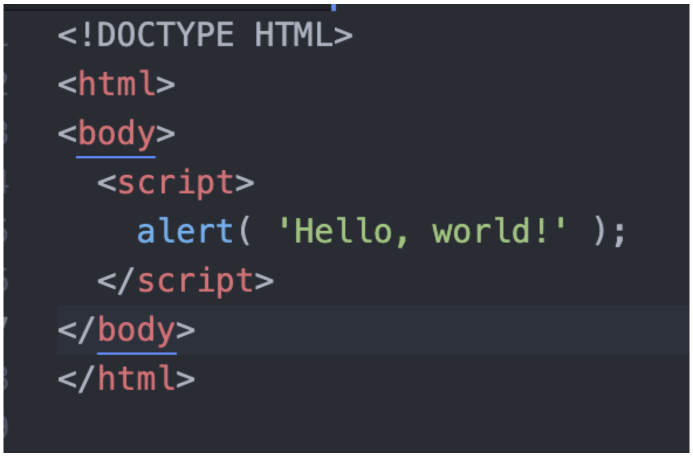
We'll be going over some of the *very basics* for JavaScript, mainly covering the methods that we discuss in our presentation. Javascript has a lot of different functions which are differentiated by different tags. All tags must have a closing tag (Ex: <script> command </script>). Let's go over them now!

TAG 1: <script> </script>- PACKAGES EXECUTABLE CODE

What's helpful about JavaScript is that you can insert an executable set of commands virtually anywhere in an HTML file using the <script> tags. This is internal JS.

You can also insert external JS, including your code in a .js file and then linking to it on the HTML file you want to add to your code. We'll be going over internal JS for the sake of our presentation.

It's as simple as this:



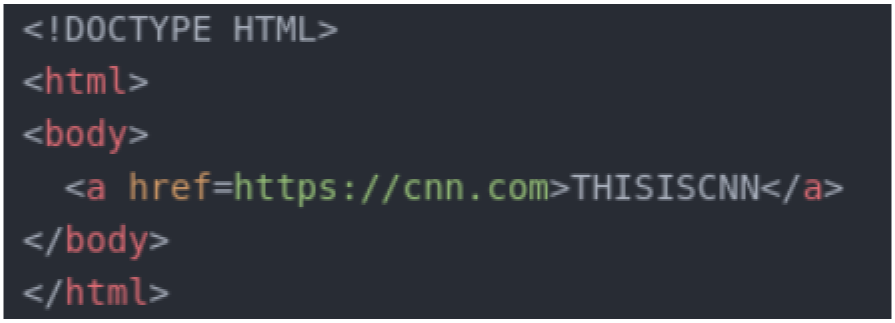
```
<!DOCTYPE HTML>
<html>
<body>
  <script>
    alert( 'Hello, world!' );
  </script>
</body>
</html>
```

In this example we see the "alert()" function surrounded by opening and closing script tags with a string parameter of "Hello, world!" The syntax is identical to what you would expect from a print statement in Java, the function is called and is completed with a semicolon at the end of the statement.

TAG 2: <a></a> - DEFINES HYPERLINKS (HREF)

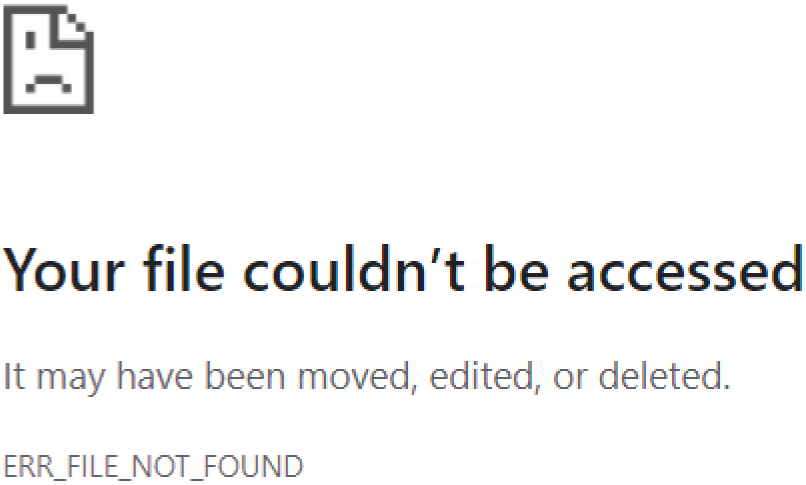
This should be familiar HTML: <a> tags define hyperlinks! You can use <a> tags to send the user ANYWHERE. Clicking on the link can bring you to another tab on the page or a new website entirely.

Here is an example of how to use the "HREF" attribute to display a link to bring the user to another website.



```
<!DOCTYPE HTML>
<html>
<body>
  <a href=https://cnn.com>THISISCNN</a>
</body>
</html>
```

One important note to remember is if the user is not being directed to a publicly accessible tab on the current website, the ENTIRE url must be included.



Your file couldn't be accessed

It may have been moved, edited, or deleted.

ERR_FILE_NOT_FOUND

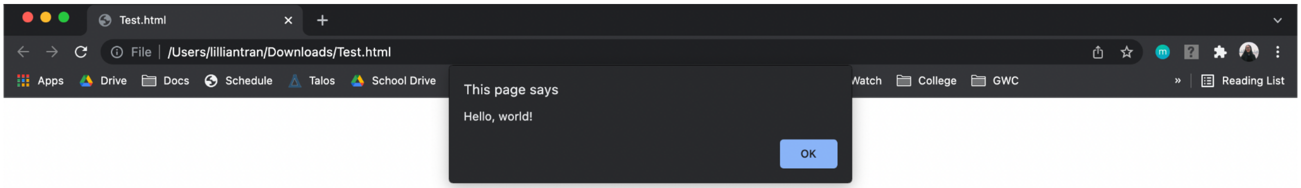
In this example, if the "https://" was not included in the attribute, then the computer would understand the code to be directing the user to a page on the website with the address cnn.com. (error message shown above).

There are a whole bunch of other tags that can be used to work with in JavaScript but you can learn that on your own… :))

Now let's review the major methods we will be using!

METHOD 1 : alert() - CREATING A POPUP

Like we mentioned above in the <script> section, one of the things JavaScript can do is create a popup. In this case, the alert() method will be what creates our popup.
If you try to open the super basic HTML file from above, the code will execute and you'll see a little something that looks like this:



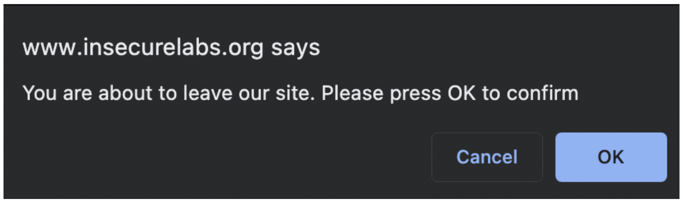
This page says

Hello, world!

METHOD 2: window.confirm() - POPUP +1

This is a method that displays a dialog with an optional message. It's a little different from a popup in that it will have to wait for the user to confirm or cancel the dialog. With this option, you can include other actions, like clicking a link. The syntax for this is pretty simple:

window.confirm(message);



www.insecurelabs.org says

You are about to leave our site. Please press OK to confirm

Here's what a window.confirm() page will look like. Notice how it's slightly different from the one of the alert() popup.

CONCLUSION

That's it! Hopefully, you have a bit of a better idea of how to use JavaScript. This guide only covers the absolute basics so you don't come into our presentation blind. There are countless online resources for further information. Linked are some of the more comprehensive ones, some of which we actually referenced!

https://www.tutorialrepublic.com/javascript-tutorial/
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction#what_is_javascript
https://javascript.info/intro
https://www.geeksforgeeks.org/introduction-to-javascript/
https://www.w3schools.com/js/js_intro.asp