RQ: shellshock

On the radio, the announcer talked about how there was a new virus that attacked programs with Unix used in it. (called Shell Shock?) Considering how many programs use Unix, what can one do to prevent getting the virus? How do anti-virus systems work?

“In short, the advice to consumers is this: watch for security updates, particularly on OS X… **Do be cautious of emails requesting information or instructing you to run software** – events like this are often followed by phishing attacks that capitalise on consumers’ fears…”

http://www.troyhunt.com/2014/09/everything-you-need-to-know-about.html
Speaking of Apple devices…

Why won't apple allow their users to use JavaScript/flash on their devices? Is it a programming problem or simply just for security reasons?
RQ: scratch

Is the Scratch program in Lab 2 a perfect parallel of JavaScript, or are there some differences in how they function? Also, how would HTML be integrated with the Scratch program the same way it is integrated with JavaScript?

http://wiki.scratch.mit.edu/wiki/Programming_Language
Speaking of the Scratch lab...

Let’s take a look at something that’s a teeny part of what you did, but makes a big change.
Variables in Action

A **variable** is a named quantity.
Playing with variables

- The following examples work pretty much the same no matter which programming language you use.
- That being said, it’s helpful to have something to test it out for you.
- So we’re going to be using JavaScript.
  - Side note: despite the name, this is a completely different programming language from Java.
The Assignment Statement

<variable> <assignment symbol> <expression>;

- **<variable>** is any declared variable in the program
- **<assignment symbol>** is the language’s notation for the assignment operation
- JavaScript’s <assignment symbol> is the equal sign (=)
- **<expression>** is a kind of formula telling the computer how to compute the new value
- Like any other statement, an assignment statement is terminated by a semi colon.
RQ

When is "=" used and when is "==" used when coding with JavaScript (For example, in a compound statement)?
Tips and Tricks

In computer science, the symbol “=“ can be used in two ways:
1) to express equality (as in math),
   e.g., 3=5 (false) or 3=5-2 (true)
   (I read this as “equals” – as usual)
2) to assign values to variables,
   e.g., X=5 (after doing this, variable X
   has value 5, and the equality X=5 is true)
   (I like to read this as “gets”)

age = 38;
days_in_year = 365.25;
age_in_days = age * days_in_year;
All of you who have a computer with you have a JavaScript interpreter with you

- Look at the schedule page for specifics for other browsers
- I’ll walk you through Chrome
  - Tools
  - JavaScript Console
- Now enter the previous commands:
  
  ```javascript
  age = 38;
days_in_year = 365.25;
age_in_days = age * days_in_year;
  ```
Tips and Tricks

• In HTML a statement like X=5 always refers to a variable (actually, attribute) assignment.

• Likewise, in JavaScript, a statement like X=5 always denotes variable assignment, while equality is written differently (X==5).

• In other languages, ‘=‘ is used to denote equality, and ‘:=‘ or ‘<-’ to denote variable assignment.
A few words on vocabulary

- ()s = parentheses
- []s = square brackets
- {} = curly braces
Exercise: balance

What is the value of the variable “balance_in_CAD”, after the following sequence of variable assignments?

```c
CAD_per_EUR = 1.32;
balance_in_EUR = 100;
balance_in_CAD = balance_in_EUR * CAD_per_EUR;
```
But why have variables in the first place?

• Sure, you could do these examples without any variables
• But in actual code we may use these values over and over and over and over again
• Plus we need to have some way to handle how these values change
Exercise: conversion

Modify the following to convert 100 CAD (Canadian dollars) into EUR (Euros)?

```plaintext
CAD_per_EUR = 1.32;
balance_in_EUR = 100;
balance_in_CAD = balance_in_EUR * CAD_per_EUR;
```

Connecting with Computer Science
www.ugrad.cs.ubc.ca/~cs101
Exercise: conversion

Modify the following to convert 100 CAD (Canadian dollars) into EUR (Euros)?

\[
\text{CAD\_per\_EUR} = 1.32; \\
\text{balance\_in\_EUR} = 100; \\
\text{balance\_in\_CAD} = \text{balance\_in\_EUR} \times \text{CAD\_per\_EUR};
\]

**Answer:**

\[
\text{CAD\_per\_EUR} = 1.32; \\
\text{balance\_in\_CAD} = 100; \\
\text{balance\_in\_EUR} = \text{balance\_in\_CAD} / \text{CAD\_per\_EUR};
\]
On readability of code…

**Note:** The following sequence of assignments has exactly the same function as the one shown previously, but is much less human-readable, and therefore inferior:

\[
x = 1.32; \\
y = 100; \\
z = y * x;
\]

(In fact, you could use ‘alice’ for ‘x’, ‘bob’ for ‘y’, etc., and still get the same function.)
Exercise: balance some more

What is the value of \texttt{balance} at the end of the following sequence of assignments?

\begin{verbatim}
balance = 123;
balance = balance + 100;
balance = balance - balance;
\end{verbatim}
RQ: learning JavaScript

Judging by the midterm, it seems at least some memorization of html is necessary. Will we be required to do the same for JavaScript? If so, do you have any tips for making the process easier (that is, in addition to practice)?

There is very precise and proper syntax in JavaScript and it seems to take a very long time to learn. How long/how much practice does it usually take the average person to learn JavaScript? What is the fastest way to learn and perfect JavaScript?
And now programming with JavaScript

- At this point, we’ve created webpages using HTML
- We’ve learned how to make them look better and have better coordination using CSS
- But we may want to do more…
Consider the class webpage

- The HTML for the top menu is in a separate file that we include in each page.
- But what if we wanted to have the top menu tell you which page we’re on?

Now:

- schedule
- syllabus
- project
- piazza
- exercises
- contacts

Desired:

- schedule
- syllabus
- project
- piazza
- exercises
- contacts

In CSS I can make an “active” class that can change the background to be purple and the foreground to be white.

But how can I know which webpage I’m on?
At this point, I feel like we have the HTML skills to design a solid 90's-era website. So does that mean that all the things that make modern websites look modern are due to adding other languages (like JavaScript) or is there some kind of advanced HTML that we haven't learned about?
Let’s look at the menu now:

<!--the menu for each page, here in one spot for easy changes-->

<div id="header">
    <div id="header_inner" class="fixed">
        <div id="logo">
            <h1>CPSC101</h1>
            <h3>Connecting with Computer Science…</h3>
        </div>
    </div>
</div>

<div id="menu">
    <ul>
        <li><a href="schedule.shtml">Schedule</a></li>
        <li><a href="syllabus.shtml">Syllabus</a></li>
        …
    </ul>
</div>
Let’s look at the menu now:

<!--the menu for each page, here in one spot for easy changes-->
<div id="header">
  <div id="header_inner" class="fixed">
    <div id="logo">
      <h1> CPSC101 </h1>
      <h3>Connecting with Computer Science…</h3>
    </div>
  </div>
</div>

<div id="menu">
  <ul>
    <li><a href="schedule.s.shtml">Schedule </a></li>
    <li><a href="syllabus.s.shtml">Syllabus</a></li>
    …
Let’s start by changing the menu list to JavaScript

```html
<ul> <!-- HTML so far-->
  <script> // Start script – JavaScript by default
    // comments in JavaScript
    ...
  </script>
</ul>
```
Let’s start by changing the menu list to JavaScript

```html
<ul> <!-- HTML before now-->
<script> // Start script – JavaScript by default
   // <-- comments in JavaScript
   document.write(
     //Document is a built in variable meaning the
     //webpage being worked on.
     //Anything after a . following a variable means
     //it’s either a sub variable, or (if it’s followed
     //by parentheses) something we can do to the
     //variable – this is called a function; we’ll look
     //at it this more later. What’s in ()s is the value
     //that the function will do something with. In this
     //case, it’s what will be written to the screen.
```

Connecting with Computer Science

www.ugrad.cs.ubc.ca/~cs101
Let's start by changing the menu list to JavaScript

```html
<ul> <!-- HTML before now-->
  <script> // Start script - JavaScript by default
    // <-- comments in JavaScript
    document.write("<li><a href="schedule.shtml"\">\n  //JavaScript uses "s for its syntax, much like html
  //where html would use &quot; to show the "s,
  //JavaScript uses the \ as an escape character
  //so document.write("\""), prints a " to the
  //current page.
  //So far this writes "<li><a href="schedule.shtml"\n  // to the current page

...```
Let’s start by changing the menu list to JavaScript

```html
<ul>
<!-- HTML before now-->

<script>
    // Start script – JavaScript by default
    // comments in JavaScript
    document.write("<li><a href="schedule.shtml"]")
    // comments deleted for extra room
    document.write(">Schedule </a></li>");

    document.write("<li><a href="syllabus.shtml"]")
    document.write(">Syllabus</a></li>");

    ...

    //keep going until we’ve done all the links in the
    //menu
</script>
</ul>
```
Okay, so now it’s in JavaScript, but it hasn’t done anything!

Right. In order for that to happen, we need to know what page we’re accessing:

```javascript
var index = location.pathname.lastIndexOf("/") + 1;
//create a variable called “index”
//location is another built in variable. Which has
//a subvariable called pathname. Which has a
//function called "lastIndexOf" that looks for the
//last place that / appears.
//In other words, where the directories end
//adding one to it means the space after the /
//so this is looking for the start of the filename.
```
Okay, so now it’s in JavaScript, but it hasn’t done anything!

Right. In order for that to happen, we need to know what page we’re accessing:

```javascript
var index = location.pathname.lastIndexOf("/") + 1;

var filename = location.pathname.substr(index);
```

// create another variable called filename
// assign to that the substring of the location
// after the last /: the filename

document.write(filename);

// for now, write that filename to the current document so we can see its value
Great! Now we know what page we’re on!

But now we need to change what we’re doing based on which file is being loaded
We need a conditional statement
Conditional Statements

The conditional has the form:

\[
\text{if (}\langle\text{Boolean expression}\rangle) \\
\langle\text{then-statement}\rangle;
\]

The \langle\text{Boolean expression}\rangle is an expression evaluating to true or false

The \langle\text{then-statement}\rangle is any JavaScript statement

\[
\text{if (waterTemp < 0)} \\
\text{waterState = "Frozen";}
\]
I noticed that there are some similarities between JavaScript and Scratch in terms of the coding language. One example would be the (if...else) command... is Scratch an extension of JavaScript? Or are they completely different programming language?
if/else Statements

if ("<Boolean expression>")
   "<then-statement>";
else
   "<else-statement>";

• If the "<Boolean expression>’s outcome is true:
  • The "<then-statement>" is executed
  • The "<else-statement>" is skipped

• If the "<Boolean expression>’s outcome is false:
  • The "<then-statement>" is skipped
  • The "<else-statement>" is executed
An important detail

The “if statement” and “then statement” can only be ONE statement. To make them more than one, use {} Example

```java
if (date == October 31){
    Halloween = true;
    wearCostume = true;
}
else{
    Halloween = false;
    //note, you can still wear a costume!
}
```

Always use the {}s or you will shoot yourself in the foot
Clicker question: what is the value of x after executing the following?

```java
x = -10;
if (x < 10) {
    x = x * 195;
} else {
    x = x * 25;
}
```
What if we wanted to give an appropriate greeting based on time of day and there are more than two options? We can use `else if`:

```java
if (time < 12) {
    x = "Good morning!";
} else if (time < 17) {
    x = "Good afternoon!";
} else {
    x = "Good evening!";
}
```
More Conditional Statements

==  equal:
    if (serviceLevel == 10) {...}

!=  unequal:
    if (serviceLevel != 10) {...}

&&  and:
    if ((food > 7) && (service > 7)) {...}

||  or:
    if ((food < 3) || (service < 3)) {...}
But back to our webpage. Everyone remember the web page?

What we want is to say *if* the name of the page is “schedule.shtml” *then* we add in the information *class="active"*.

Do we need an else?
Putting it all together

... <script>
    var index = location.pathname.lastIndexOf("/") + 1;
    var filename = location.pathname.substr(index);
    document.write("<li><a href="schedule.shtml">

    if (filename=="schedule.shtml"){
        //Note uses ==, not =
        document.write(" class="active"");
    }

    document.write(">Schedule </a></li>");
</script>
A minor detail...

We’ve made it so that index.shtml and schedule.shtml are the same. So we need to use that logical or.

```javascript
... if (filename=="schedule.shtml" || filename=="index.shtml") {
    //Note uses ==, not =
    document.write(" class="active"\"\")
}

document.write(">Schedule </a></li>");
...
```