JavaScript Problem Solving

Why learn more about solving programming problems?

- Most beginners can program a basic web page, but troubleshooting interactions and algorithmic thinking are more rare
- These kinds of problems are much harder to Google for and teach you to think on your feet
- Your code and thought process will be better organized
- Many coding interview questions ask you to solve problems

What is an algorithm?

- A set of simple instructions to complete a task
- Computers aren't smart, they have no intuition: an algorithm tells them everything step by step in a language they can understand

Peanut Butter + Jelly Sandwich algorithm

```
// Get the jelly

// Get the peanut butter

// Get the bread

// Spread peanut butter on one side and jelly on the other

// Combine the two sides
```

Here's an example of an algorithm. What steps could we add before, after, or in between steps?

As applied to a web development problem

How do I make an image slide off the screen after 3 seconds?

```
// Delay the script by 3 seconds

// Select the image with a jQuery selector,

// Move the image across the screen somehow (positioning? margins?),
// by adding a certain amount of space every x milliseconds
```

How can we break up the above algorithm even further?

What is some actual code that can help us solve each step?

Another problem

```
// Problem: Display an array's contents along
// with the index number of each array item.

// First, we'll need an array with some items in it

// Next, we'll need to iterate (loop) over the array

// Within our loop, we'll log each array item to the console
// within this log statement, we'll concatenate (add) together
// the array item along with the index number we're currently on
```

Adding code to our algorithm

```
// First, we'll need an array with some items in it
var myArr = ["giraffe", "zebra", "antelope"];
// Next, we'll need to iterate (loop) over the array
for(i = 0; i < myArr.length; i++){


// Within our loop, we'll log each array item to the console
// within this log statement, we'll concatenate (add) together
// the array item with the index number we're currently on
  console.log(i + " " + myArr[i]);
}
```

Exercise

- Write a pseudo-code algorithm (a list of steps in comments) that checks a form's fields for validity
- Break the problem down into as many steps as possible
- Begin researching the code behind each of the steps

Debugging Tips

Stack Trace

- Many times, you'll find the answer to the bug in the error message you get
- Occasionally it will even indicate the exact file and line number the problem was caused by:

```
console.lg("hi")
Uncaught TypeError: undefined is not a function script.js:10
// You tried to call something that wasn't a function
// In this case, "lg" is not a function of the console, "log" is
```

- Read error messages closely to see if any of this identifying information, known as a "stack trace" is available, so you can reference your script directly for the error

Debugger

- The keyword, 'debugger' allows for breaks in your scrips.
- The function will stop your code, with the option of continuing on in the developer tools.

```
console.log("hello world")
debugger;
alert('I think I found the bug!')
// the alert will not fire while debugger is on (developer tools)
```

- With the debugger turned on, the third line will not be invoked immediately.

Debugger Exercise

- Try adding a debugger statement into a script then opening the web page.
- By default the debugger functionality is turned off.
- To turn it on, open the developer tools, and refresh the page!

Subtraction

- Sometimes, it can be a bit harder to find a piece of troublesome code
- Try removing pieces of the offending code line by line to more easily isolate exactly what is causing the problem

```
for(var x = 0; x < 10; x++) {
  console.log(x)
  // Comment out the below line
  // to see what effect is has on output
  // x = z + 10;
  var y = x + 2;
  // Comment out this line to only see
  // the first log statement
  // console.log(y)
  return y
}
```

A final note

- Remember that nobody has all the answers or knows literally everything that there is to know about programming
- Don't be ashamed to use Google (and Stack Overflow), ask a fellow student, or ask your instructor if you have a question. The only way to get better is to learn from those around you!