intro to JavaScript

class one

course adapted from:
Teaching Materials: http://www.teaching-materials.org/javascript/
Oz Girl Develop It: http://cathylill.net/courses/js101/
Girl Develop It: http://www.girldevelopit.com/materials

what is JavaScript?

- programming language used to add interactivity to web sites
- also used for games, animation, apps
- it's becoming a "full stack" development language
- along with HTML and CSS, a foundational part of modern web
- ***not*** Java

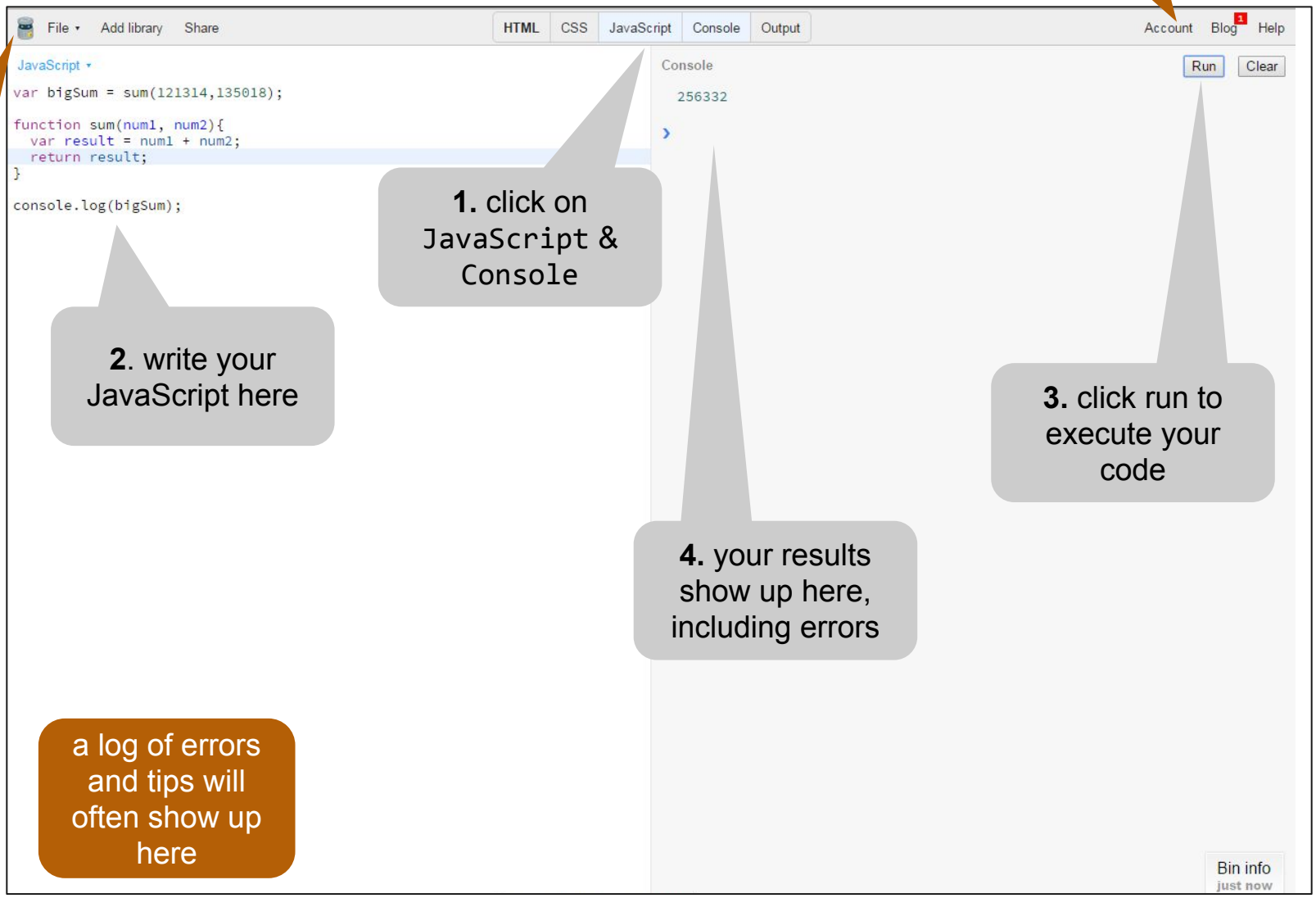
statements

1. a program is a list of instructions.
2. each instruction is a statement.
3. let's write our first one!

how to use jsbin.com

create an account to save your bins

click the bin for more options



1. click on JavaScript & Console

2. write your JavaScript here

3. click run to execute your code

4. your results show up here, including errors

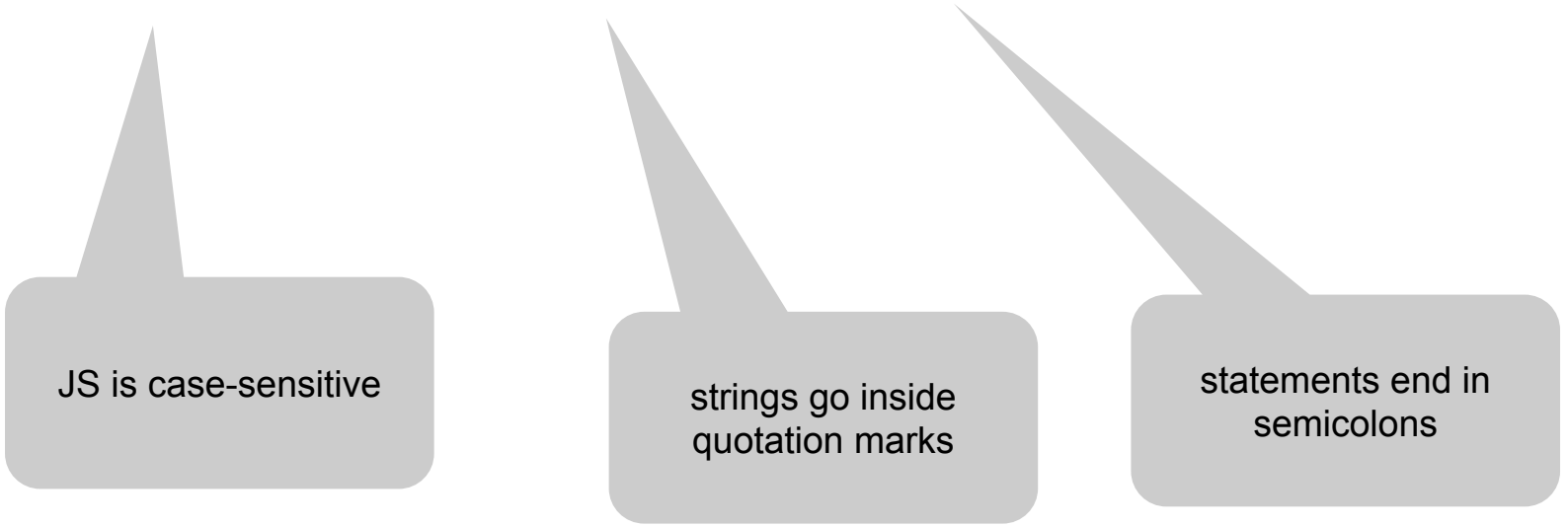
a log of errors and tips will often show up here

say hello with `alert`

```
alert('Hello world!');
```

syntax

```
alert(‘Hello world!’);
```



leaving //comments

```
//to write something to the console
console.log('Hello world!');

/*this is really far too long a
comment. I should learn to be more
concise*/
console.log('Hello world!');
```

variables

use them to store values.

declare a variable

```
var firstOne;
```

initialize a variable

```
firstOne = “apples”;
```

naming variables

- begin with letters, _, or $
- names are case-sensitive
- standard to use camelCase (`aLongName` vs. `a_long_name`)
- can't use reserved words
- choose for clarity and meaning
- pick convention and stick with it

try it

```
var name;
name = 'Cody';
console.log(name);
```

declare a variable

initialize a variable

see the value in the console

an easier way

declare and initialize all in one step:

```
var secondOne = 5678;
```

variables change value

Called “reassigning the value”:

```
var bucket = 'oranges';
bucket = 'apples';
bucket = 'weasels';
bucket = 'the cold hand of vengeance';
```

create the bucket

but we can change what is in the bucket at any time

data types

- string:
  - `var eternalTruth = 'I love coffee';`
  - `var iNeedToUseApostrophe = "Cody's things";`
- number:
  - whole: `var numOfHighFives = 48621;`
  - floating point: `var pi = 3.14;`

data types, continued

- boolean:
  - `var iRule = true;`
  - `var youDrool = false;`
- undefined
  - `var notDefinedYet;`
- null
  - `var daysToNotDrinkCoffee = null;`

loose typing

JS bases type on value, and type can change.

```
secondOne = "Now I’m some text!";
console.log(typeof secondOne);
```

find the data type with `typeof`

```
console.log(typeof secondOne);
```

math-y expressions

variables can also contain the result of any piece of code that evaluates to a value. these are called expressions.

```
var x = 42 + 37;
var y = 356 - 204;
var total = x + y;
```

operators

+	Addition	`2 + 2 = 4`
-	Subtraction	`2 - 2 = 4`
*	Multiplication	`2 * 2 = 4`
/	Division	`2 / 2 = 1`
%	Modulus (the remainder)	`5 % 2 = 1`
++	Increment (+1)	`++2 = 3`
--	Decrement (-1)	`--2 = 1`

text-y expressions

We can also use expressions to combine strings.

```
var x = 'coffee';
var y = 'beer';
var z = 'I love ' + x + ' and ' + y;
```

loose typing in expressions

```
var x = '8';
var y = 3;
```

What is the value and data type of the expressions `x+y`, `x-y`, `x*y`, and `x/y`?

stuck?

you could try making a new variable to hold each expression, like:

```
var addition = x + y;
```

or do it all at once

```
console.log(x + y + " is a " + typeof
(x+y));
// and so on...
```

exercise time!

exercise 1: age calculator

- Store the current year in a variable.
- Store someone’s birth year in a variable.
- Calculate their 2 possible ages based on the stored values.
- Output them to the console like so: "They are either NN or NN"

exercise 2: geometry

find the circumference and area of a circle

- Store a radius into a variable.
- Calculate the circumference based on the radius (pi*diameter), and output "The circumference is NN".
- Calculate the area based on the radius (pi*r^2), and output "The area is NN".

You can have JS remember pi for you by using the `Math.PI` function.

functions

are reusable collections of statements.

```
function myFunc() {
   //a bunch of statements
}
```

after it's declared, call a function like this:

```
myFunc();
```

arguments

are the values the function needs to do what you need it to.

here's your argument, like a variable just for this function

```
function sayMyName(name) {
  console.log('Hi, ' + name);
}
sayMyName('Cody');
```

then you call the function, providing a value for the argument.

multiple arguments

```
function sum(num1, num2) {
   var result = num1 + num2;
   console.log(result);
}

sum(123,456);
```

variables as arguments

```
var aBigNum = 12893241;

sum(aBigNum, 42);
```

return

the ‘return’ keyword sends the result of the function back to whoever called it.

```
var bigSum = sum(121314,135018);
console.log(bigSum);
```

what happens?

return

add a return to the sum function.

```
return result;
```

return exits the function - anything added after a return isn’t run.

scope

- variables in JS are only visible inside the function they're declared in - they have "local scope"
- variables declared outside any function have "global scope"

local scope

```
function sum(num1, num2){
    var result = num1 + num2;
    console.log("The local result is "                    );
}

sum(4,2);
console.log(“The global result is “ + result);
```

global scope

```
var result;

function sum(num1, num2){
    result = num1 + num2;
    console.log("The local result is " + result);
}

sum(4,2);
console.log(“The global result is “ + result);
```

outside the function

homework!

lifetime supply: html

In JS Bin, close the Console and click on HTML & Output. In the HTML window, paste this markup:

```
<!DOCTYPE html>
    <html>
        <head>
            <title>JAVASCRIPT 4EVA</title>
        </head>
        <body>
            <p>Check out my sweet webpage. Here's some
            stuff I learned how to do with JavaScript.
            <p>
        </body>
    </html>
```

lifetime supply: js

Write a function named `calculate` that figures out how much of your favorite food or drink you will consume in your lifetime. Some of the necessary steps include:

- Store your age in a variable
- Store your maximum age in a variable
- Store an estimated amount of snacks per day in a variable
- Calculate how many you would eat/drink for the rest of your life.
- Output the result in an alert (`alert(“text”);`) like so: "`You will need NN to last you until the ripe old age of NN`"

Note: Click on **Run with JS** in the **Output** window to initiate your JS.

run the function

To your HTML, add the following:

```
<button onclick="calculate()">Figure out your
lifetime supply!</button>
```

the `onclick` event lets us call a function when the user clicks on this link