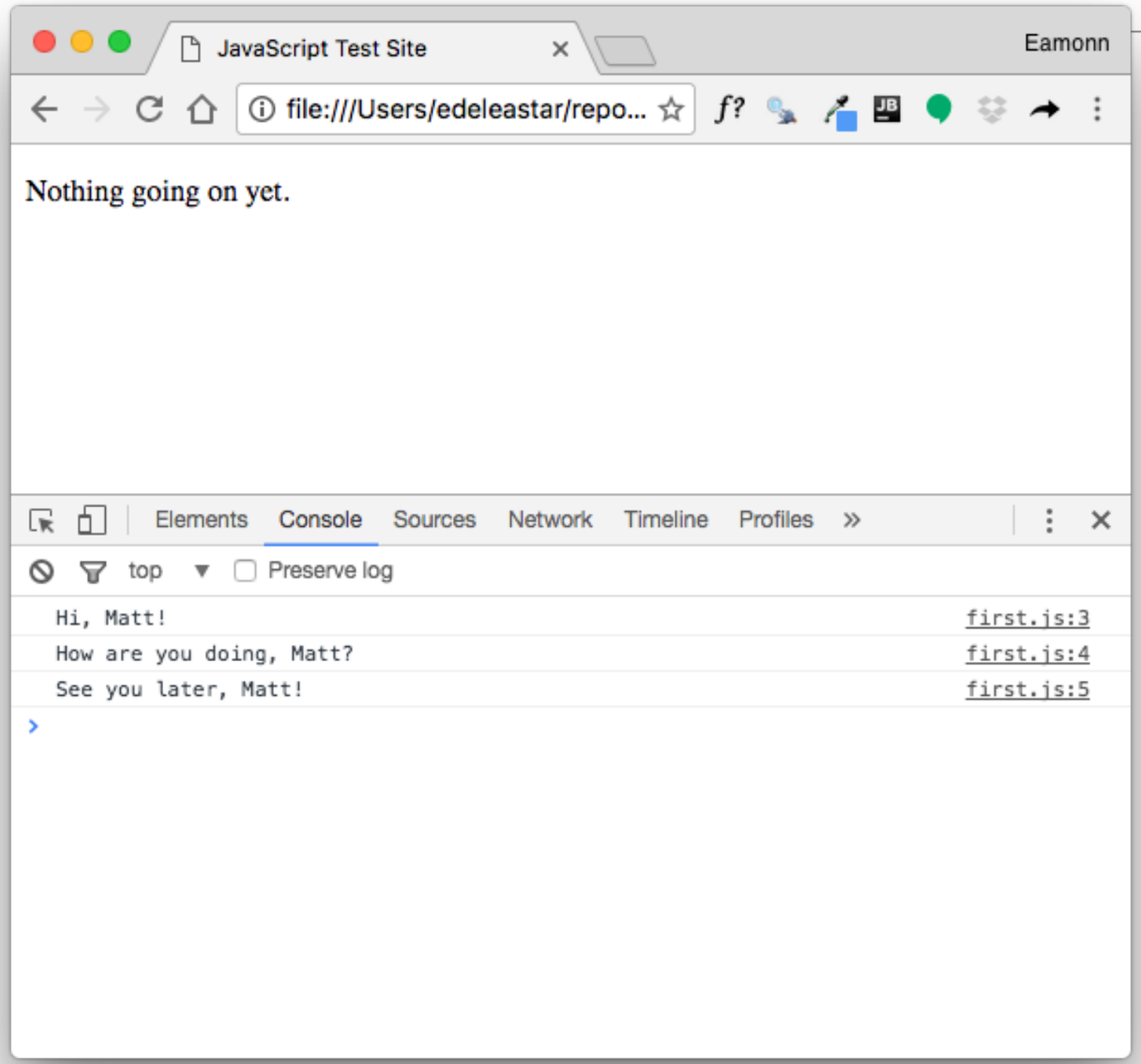
Javascript: Variables

Variables

- Initialize and assign variables in JavaScript
- Store variables using the prompt function
- Write comments in your JavaScript code
- List all of the data types in JavaScript
- Compare and contrast primitive data types with objects

Variable Fundamentals



index.html

```
<!DOCTYPE html>
<html>
  <head>
  <title>JavaScript Test Site</title>
    <script src="first.js"></script>
  </head>
  <body>

    <p>Nothing going on yet.</p>

  </body>
</html>
```

first.js

```
console.log("Hi, Matt!");
console.log("How are you doing, Matt?");
console.log("See you later, Matt!");
```

- what if we want to change the person's name from "Matt" to something else?

var keyword

declare a variable

```
var firstName = "Matt";
console.log("Hi, " + firstName + "!");
console.log("How are you doing, " + firstName + "?");
console.log("See you later, " + firstName + "!");
```

use a variable

Primitive Data Types

- 6 Primitive Data Types
- JavaScript is known as a "weakly" typed language.
- This means is that when you create variables and assign them to values, you do not have to specify the type of data you are working with.

```
// String
var greeting = “hello";

// Number
var favoriteNum = 33;

// Boolean
var isAwesome = true;

// undefined
var foo;
var setToUndefined = undefined;

// null
var empty = null;
```

Strings

```
// a string is a set of characters enclosed in quotes.
//A string can be defined using double quotes:
var greeting = "Hello Whiskey”;

// or using single quotes:
var greeeting = 'Hello World';

// if We want quotes in a string, we can mix them, keeping them balanced:
var phrase = 'Matt said, "I have not been to Chile", the other day.';
```

Numbers

```
//JavaScript numbers can be positive:
var num = 5;

// or negative:
var num = -25;
```

Decimal Numbers

```
var piApproximation = 3.14159265;

var x = 1 + 3;
var a = 4.5;
var b = 5.9;
var c = Math.sqrt(a * a + b * b);
var studentTeacherRatio = 4 / 1;
```

Boolean

```
// A boolean type can only be in one of two states,
// true or false.

var pizzaIsGood = true;
var pizzaIsBad = false;
```

Undefined

```
// Any variable that is created in JavaScript
// that is not assigned a value is undefined:
var noValue;  // The value here will be undefined

//You can also explicitly set a variable to undefined:
var favoriteFood = "Candy";

// Changed your mind
var favoriteFood = undefined;
```

Null

```
// Null is not the same as undefined.
It signifies an intentional absense of data.
var secondEmailAddress = null;
```

- It is important to remember that null and undefined are different types in JavaScript

- This can be a confusing feature of JavaScript, even for people who know other programming languages.

- The distinction can seem somewhat arbitrary when you're first learning the language, but as you get more comfortable the distinction will become clearer.

Figuring out a variable's type

- In JavaScript, we have a keyword called typeof that returns the type of the variable.

```
typeof "";              // - "string"
typeof 5;               // - "number"
typeof false;           // - "boolean"
typeof undefined;       // - "undefined"
typeof null;          // this is not what we expect,
                      // it returns "object"!
```

Converting to a string: toString

- The toString method will convert any value which is not undefined or null into a string

```
var num = 5;
var bool = true;

num.toString(); // "5";
bool.toString(); // "true";
```

Converting to a number using *parse*

- There are several ways you can convert a value to a number.
- One way is to parse the number, using parseInt or parseFloat:
- Each function will look at a string from left to write and try to make sense of the characters it sees as numbers.

```
parseInt("2"); // 2
parseFloat("2"); // 2
parseInt("3.14"); // 3
parseFloat("3.14"); // 3.14
parseInt("2.3alkweflakwe"); // 2
parseFloat("2.3alkweflakwe"); // 2.3
parseInt("w2.3alkweflakwe"); // NaN (not a number)
parseFloat("w2.3alkweflakwe"); // NaN (not a number)
```

Converting to a number using *Number*

- This doesn't parse, it simply tries to convert the entire string directly to a number

```
Number("2"); // 2
Number("3.14"); // 3.14
Number("2.3alkweflakwe"); // NaN
Number("w2.3alkweflakwe"); // NaN
```

Converting to a number using +

- This doesn't parse, it simply tries to convert the entire string directly to a number.

```
+"2"; // 2
+"3.14"; // 3.14
+"2.3alkweflakwe"; // NaN
+"w2.3alkweflakwe"; // NaN
```