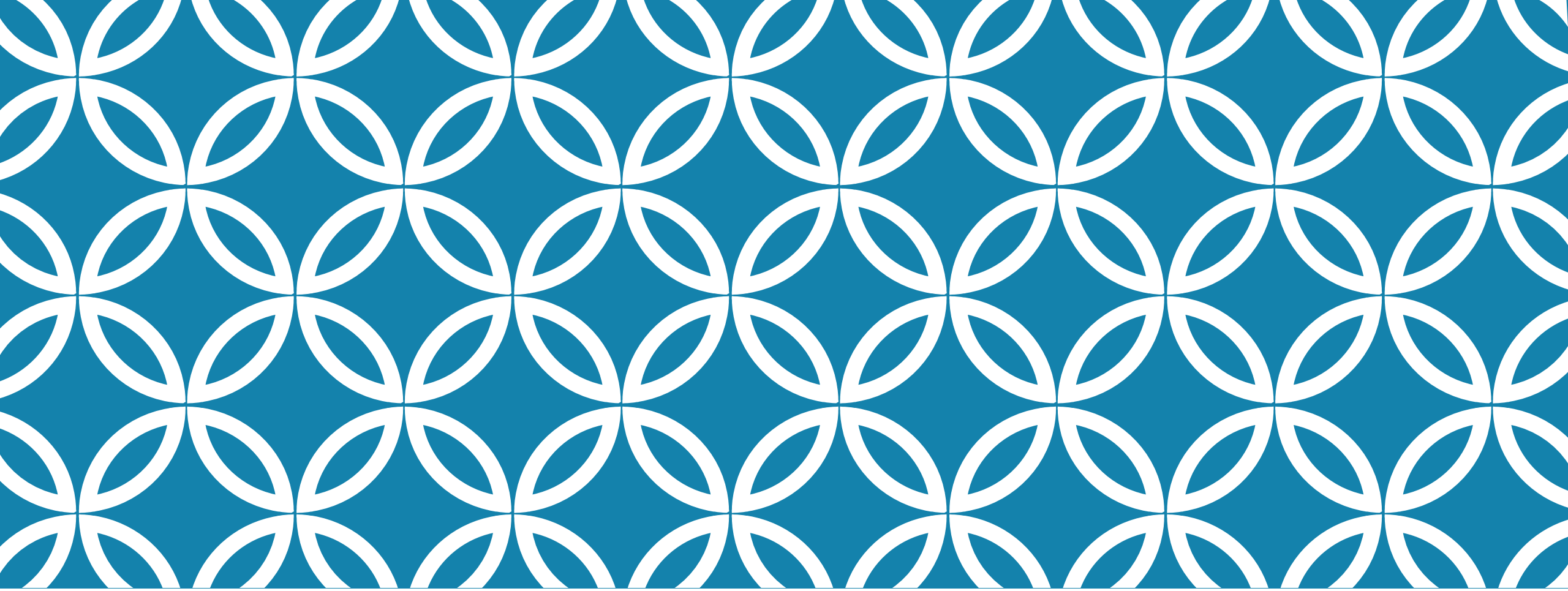
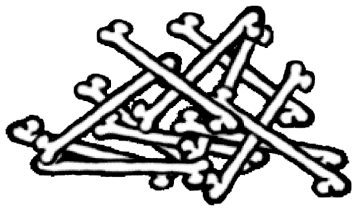


JAVASCRIPT

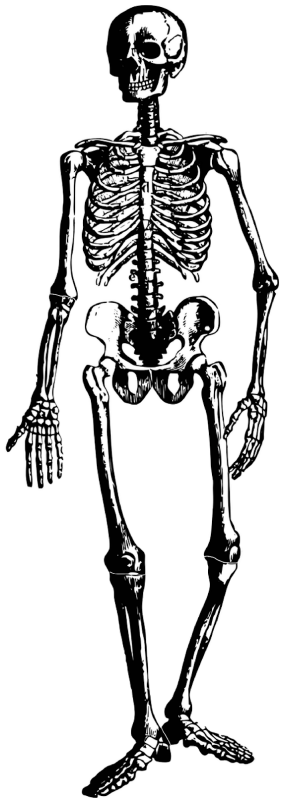
Jamal Nasir

Learn IT Skills

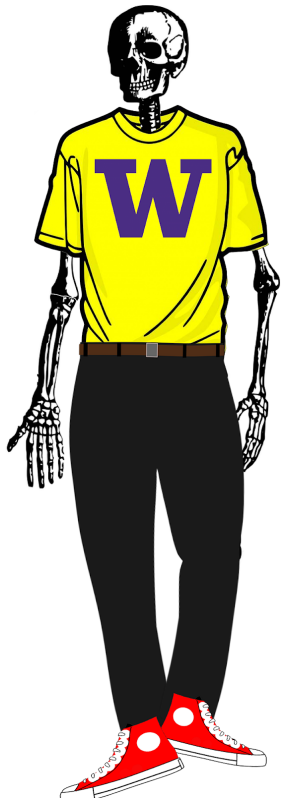
WHAT IS A WEB PAGE REALLY?



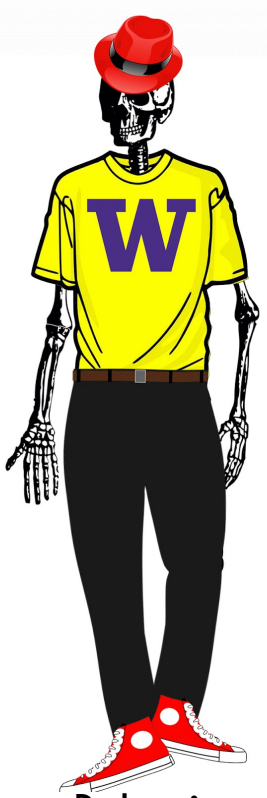
Content



Structure



Style

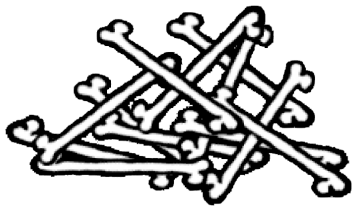


Behaviour

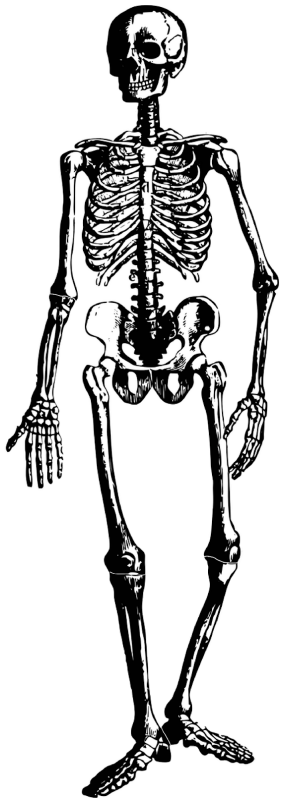
Credits: Andrew Fitz at University of Washington

Learn IT Skills

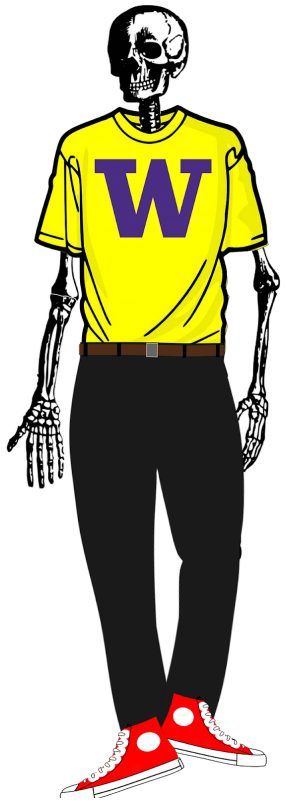
WHAT IS A WEB PAGE REALLY?



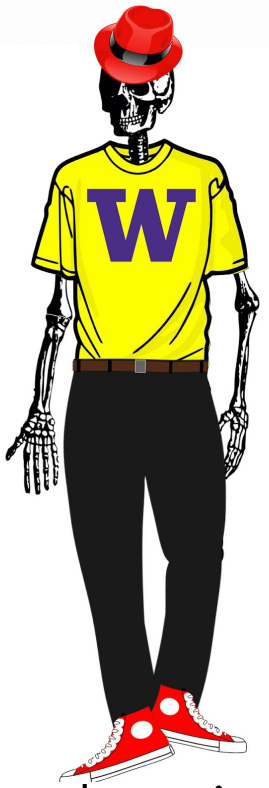
Words/Images



HTML



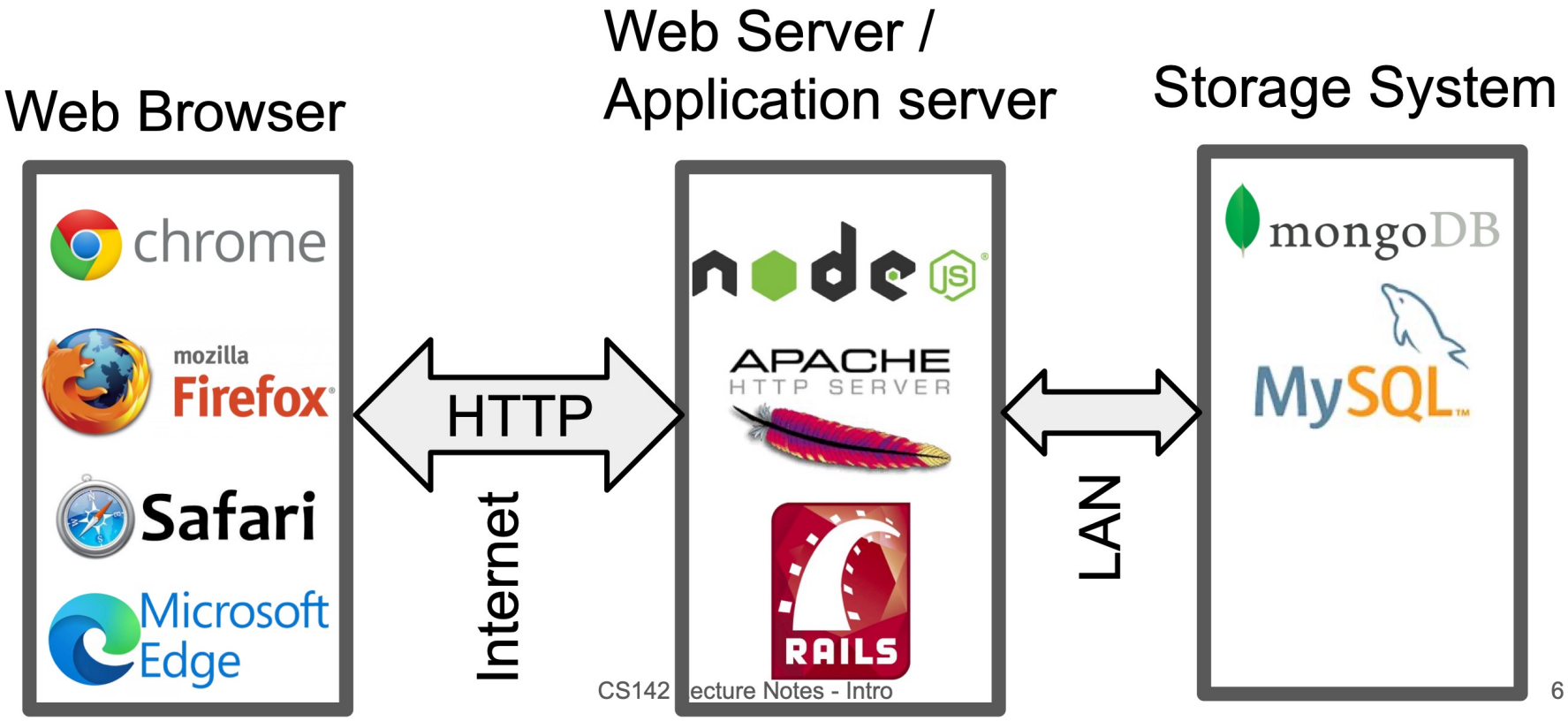
CSS



Javascript

Learn IT Skills

FULL STACK WEB APPLICATION ARCHITECTURE



Learn IT Skills

JAVASCRIPT IS TO JAVA AS ...

Grapefruit → Grape

Carpet → Car

Catfish → Cat

Learn IT Skills

WHAT IS JAVASCRIPT?

- A lightweight "scripting" programming language
- Created in 1995 by Brendan Eich (original prototype created in 10 days and called LiveScript)
- Some ‘Good parts’, some not so good

Learn IT Skills

WHY JAVASCRIPT AND NOT ANOTHER LANGUAGE?

Popularity.

The early web browsers supported it as a lightweight and flexible way to add interactivity to pages.

Microsoft created their own version, called JScript, but the open source browsers (notably Firefox and Chrome) put all their effort into JavaScript.

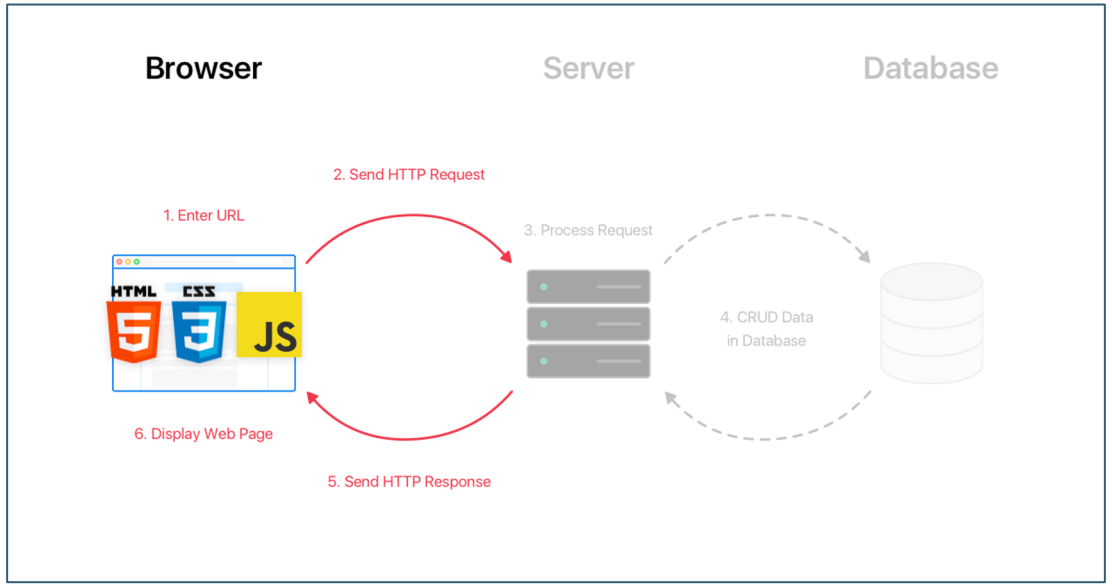
Now: If you want to run anything other than JavaScript in the browser… it's Very Hard™ (often impossible)

WEB PAGE BEHAVIOR WITH JAVASCRIPT

Now that we know how to add content and styles to a web page, let's explore how to add responsive behavior

We will use these building blocks to dynamically update what you see on a web page in response to clicks, text input, timers, etc.

TERMINOLOGY: CLIENT-SIDE SCRIPTING



Client-side script: Code that runs on the user's computer and does not need a server to run (just a browser!).
Client-side JavaScript is usually run after HTML and CSS have been loaded on the browser (e.g. from a server response).

Often, this JavaScript manipulates the page or responds to user actions through "event handlers".

Learn IT Skills

JS: ADDING BEHAVIOR TO HTML/CSS

We can use write JavaScript functions to...

- Insert dynamic text into HTML (ex: username)
- React to events (ex: page load, user's mouse click)
- Get information about a user's computer (ex: what browser they are using)

COMMENTS (SIMILAR TO JAVA)

```
// single-line comment

/**
* multi-line
* comment
*/
```

Identical to Java's comment syntax

Recall: 3 comment syntaxes

- `HTML: <!-- comment -->`
- `CSS/Java/JS: /* comment */`
- `Java/JS: // comment`

For functions and program files, we'll use JSDoc commenting with `@param` and `@returns`, which is covered in the Code Quality Guide here.

VARIABLES

```
// template
let name = expression;

// examples
let level = 23;
let accuracyRate = 0.99;
let name = "Pikachu";
```

Variables are declared with the let keyword (case-sensitive). You may also see `var` used instead of `let` - this is an older convention- **DO NOT USE** `var` anywhere

Learn IT Skills

"TYPES" IN JAVASCRIPT

```
let level = 23; // Number
let accuracyRate = 0.99; // Number
let name = "Pikachu"; // String
let temps = [55, 60, 57.5]; // Array
```

Types are not specified, but JS does have types ("loosely-typed")

- `Number, Boolean, String, Array, Object, Function, Null, Undefined`
- Can find out a variable's type by calling typeof, but usually this is poor practice (why?)
- Note: Type conversion isn't always what you expect...

Learn IT Skills

NUMBER TYPE

```
let enrollment = 99;
let medianGrade = 2.8;
let credits = 5 + 4 + (2 * 3);
```

- Integers and real numbers are the same type (no `int` vs. `double`). All numbers in JS are floating point numbers.
- Same operators: `+ - * / % ++ -- = += -= *= /= %=` and similar precedence to Java.
- Many operators auto-convert types: `"2" * 3` is `6`
- `NaN` ("Not a Number") is a return value from operations that have an undefined numerical result (e.g. dividing a String by a Number).

Practice!

Learn IT Skills

STRING TYPE

```
let nickName = "Sparky O'Sparkz";                    // "Sparky O'Sparks"
let fName = nickName.substring(0, s.indexOf(" ")); // "Sparky"
let len = nickName.length;                           // 15
let name = 'Pikachu';                                // can use "" or ''
```

Methods: charAt, charCodeAt, fromCharCode, indexOf, lastIndexOf, replace, split, substring, toLowerCase, toUpperCase

MORE ABOUT STRINGS

Escape sequences behave as in Java: `\' \" \& \n \t \\`
To convert between Numbers and Strings:

```
let count = 10;                                 // 10
let stringedCount = "" + count;                 // "10"
let puppyCount = count + " puppies, yay!";      // "10 puppies, yay!"
let magicNum = parseInt("42 is the answer"); // 42
let mystery = parseFloat("Am I a number?");  // NaN
```

To access characters of a String s, use s[index] or s.charAt(index):

```
let firstLetter  = puppyCount[0];                                  // "1"
let fourthLetter = puppyCount.charAt(3);                           // "p"
let lastLetter   = puppyCount.charAt(puppyCount.length - 1); // "!"
```

SPECIAL VALUES: NULL AND UNDEFINED.

```
let foo = null;
let bar = 9;
let baz;

/* At this point in the code,
* foo is null
* bar is 9
* baz is undefined
*/
```

`undefined`: declared but has not yet been assigned a value
`null`: exists, but was specifically assigned an empty value or `null`. Expresses intentional a lack of identification.
A good motivating overview of `null` vs. `undefined`

Note: This takes some time to get used to, and remember this slide if you get confused later.

ARRAYS

```
let name = [];                              // empty array
let names = [value, value, ..., value]; // pre-filled
names[index] = value;                       // store element
```

```
let types = ["Electric", "Water", "Fire"];
let pokemon = [];          // []
pokemon[0] = "Pikachu";  // ["Pikachu"]
pokemon[1] = "Squirtle"; // ["Pikachu", "Sqiurtle"]
pokemon[3] = "Magikarp"; // ["Pikachu", "Sqiurtle", undefined, "Magikarp"]
pokemon[3] = "Gyarados"; // ["Pikachu", "Sqiurtle", undefined, "Gyarados"]
```

Two ways to initialize an array

`length` property (grows as needed when elements are added)

DEFINING FUNCTIONS

```
// template
function name(params) {
  statement;
  statement;
  ...
  statement;
}

// example
function myFunction() {
  console.log("Hello!");
  alert("Your browser says hi!");
}
```

The above could be the contents of basics.js linked to our HTML page
Statements placed into functions can be evaluated in response to user events

JS FUNCTION VS. JAVA METHOD

```
function repeat(str, n) {
  let result = str;
  for (let i = 1; i < n; i++) {
    result += str;
  }
  return result;
  }
  let repeatedStr = repeat("echo...", 3); // "echo...echo...echo..."
}
```

```
public static String repeat(String str, int n) {
  String result = str;
  for (int i = 1; i < n; i++) {
    result += str;
  }
  return result;
  }
  String repeatedStr = repeat("echo...", 3); // "echo...echo...echo..."
}
```