Why is it So Popular?

By: Eric Benedetto

A Little About Node.js

- Written in 2009
- Open Source
- Cross-Platform
- Server-Side JavaScript Runtime Environment
- Asynchronous Driven I/O
- Functional and OOP Conventions

Strongly Typed Languages

- Conventional OOP Languages use Strong Typing
  - Variable declarations contain the data structure type
  - Methods/Functions contain a type signature
  - Complex data structures require templates
- JavaScript is a Weak Typed Language
  - Variable declarations are not limited to one type
  - Methods/Functions do not have a type signature
  - Complex data structures do not require templates
  - Type Checking in Logical Operations is possible
- TypeScript is a Strong Typed JavaScript
  - Transpiled to Vanilla JavaScript
  - All JavaScript is valid TypeScript, but not all TypeScript is valid JavaScript

Examples of Strong vs. Weak Typing

Java (This Code Won't Run due to Strong Types)

```
char cur = 'a';

private boolean compareChar(char cur, char
cur_two) {

        return cur == cur_two;

}

cur = new ArrayList<Integer>(Arrays.asList(1,2,3));

System.out.println(cur);
```

JavaScript (Perfectly Valid JavaScript)

```
var cur = 'a';

function compareChar(cur, cur_two) {

        return cur == cur_two;

}

cur = [1,2,3];

console.log(cur);
```

*semicolons are optional in JavaScript

ES6

- Using ES6 Syntax allow the simplifying of functions with cleaner/more concise code similar to Python and other Languages' lambdas.

Java

```
char cur = ‘a’;

private boolean compareChar(char cur, char cur_two) {

        return cur == cur_two;

}
```

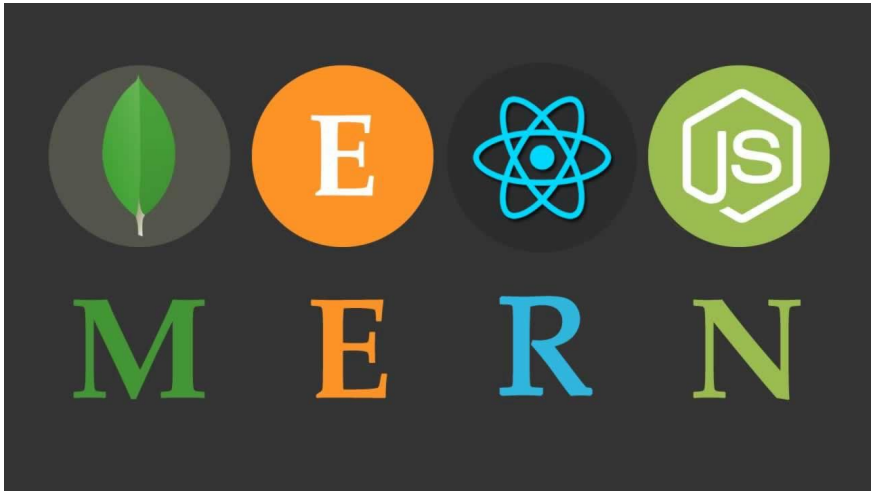
JavaScript

```
let cur = ‘a’

const compareCur = (cur, cur_two) => cur == cur_two
```

Completing the Stack with Node.js

- Popular Front-End Frameworks are built on JavaScript
- NoSQL Databases like MongoDB use JSON
- Template Engines allow Server-Side Rendering in Node.js
- Frameworks allow Desktop Application Development in Node.js



Scalable PWAs

- PHP is a main competitor for Node.js
- Both can create robust platforms for many purposes.
- Scalable Progressive Web Applications need to be dynamic, and conform to many devices and standards with flexibility to read and render data to clients quickly.
- Many older languages are built to fit specific niches and target direct needs and platforms--making them hard to scale and progress.
- Node.js is dynamic, lightweight, and **very** fast by leveraging the use of modules/packages.