

Basics of JavaScript

JS

Introduction

The World's Most Misunderstood Programming Language

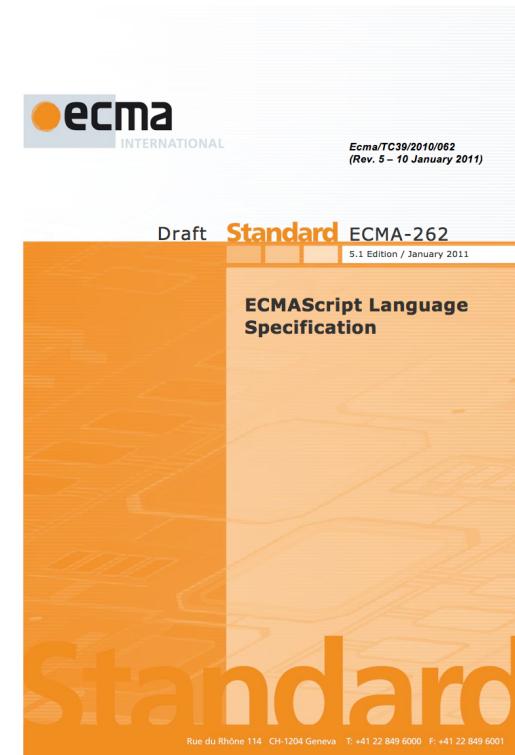
- Has nothing to do with JAVA
- Object Oriented and Functional [Lisp in C's clothing]
- Designed as a scripting language for NetScape Navigator
- Traditionally abused for Form validation
- Now runs giants like walmart / paypal / linkedin

Introduction

JavaScript is a **prototype-based scripting language** with dynamic typing and **first-class functions**.

This makes it a mix of features makes it a **multi-paradigm** language, supporting **object-oriented**, **imperative**, and **functional** programming styles.

Originally designed for the browser . But now used literally everywhere.



History

- Created in 1995 by Brenden Eich
- First version was created in 10 days !
- Microsoft releases JScript for IE after few months
- Netscape goes to ECMA for getting standards
- First formal standards released in 1999 as ECMAScript 3- Has been stable ever since
- Second coming happened after Google popularised concept of AJAX for their web apps.
- Latest version (ES6) released last month with a lot of new features - Classes , generators etc
- Google/Mozilla working on a project to make assembly in web possible. (ASM.js / WebAssembly)
- Today its the most popular programming language on Github



Who uses JS?

In the Browser

Everyone who has a modern webpage/web app.

Best examples can be Google products.

And literally every website that you can think of.

Desktop

Windows 8 metro UI was built using it .

iOS uses a webkit engine for the great UI . Same thing that is used by chrome for rendering.



Who uses JS?

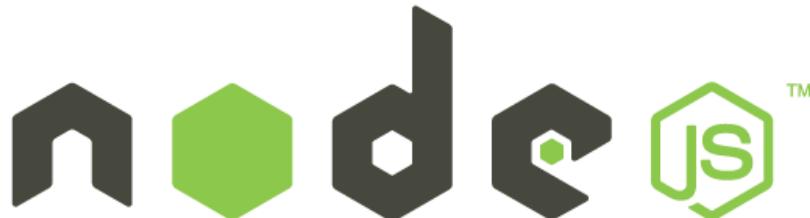
Server Side

Walmart.com,Paypal,Linked in, Apple were early adopter

Now most companies are moving to node / something similar for their content serving

IOT - JS is becoming the go to language

RealTime - We are launching chat soon built fully in node

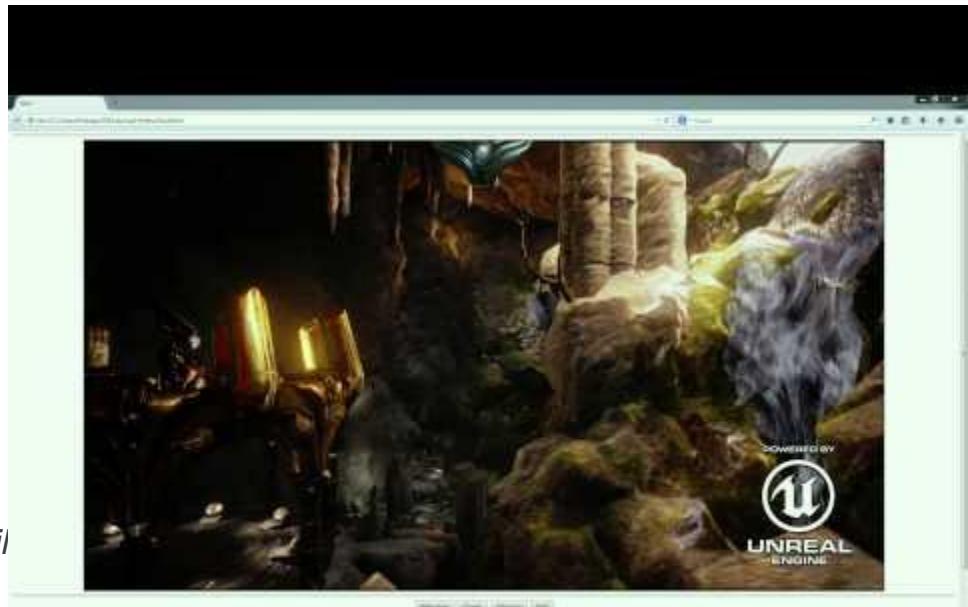


What can JS do today ?

- Run a VM inside a browser
- Run a game inside the browser
- Serve 300 million users without shooting up the CPU
- Help in making PPTs online
- Make real time chat possible

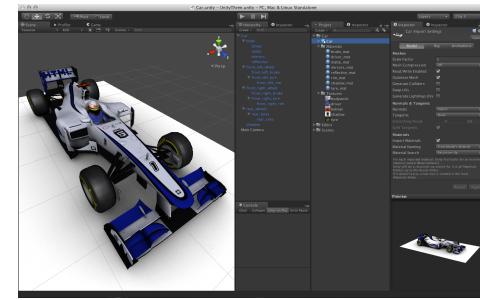
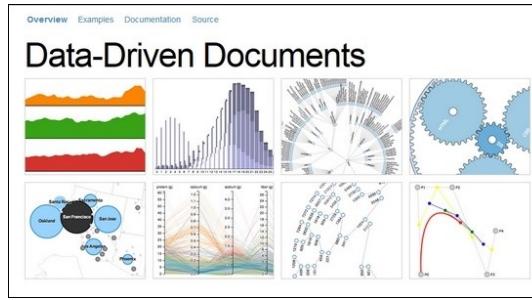
Atwood's Law:

“any application that *can* be written in JavaScript, *will* eventually be written in JavaScript.”



What can JS do ?

- Make cool graphics
- Make sophisticated dashboards
- Car dashboard panels
- and of course validation

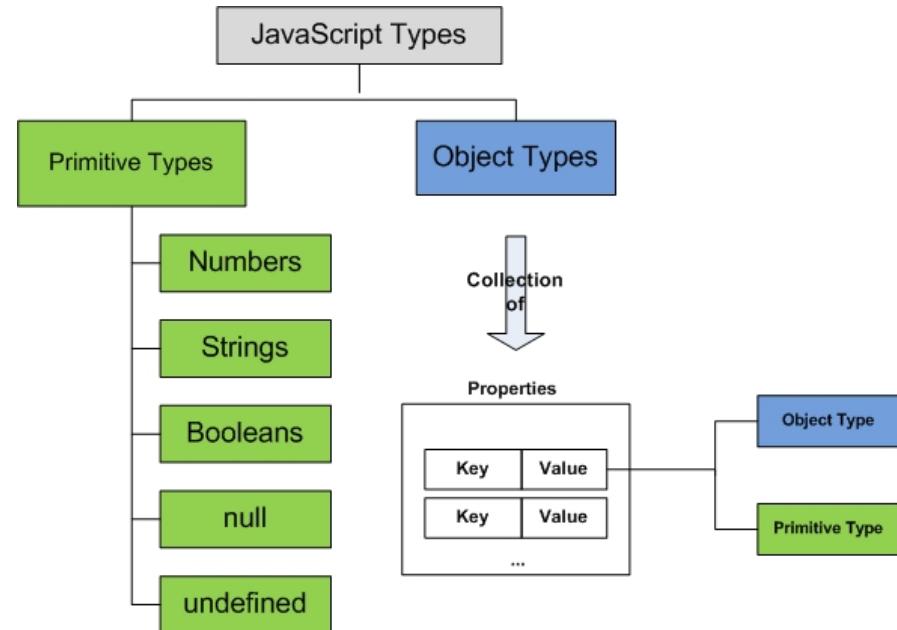


A screenshot of a web form with several validation errors. The fields are outlined in red, and error messages are displayed next to them. The fields include Name, Email, Phone, Fax, Address, Postcode, Country, D.O.B., Sex, and Disclaimer. The 'Email' field has the message 'Please enter your email'. The 'Phone' field has the message 'Phone Number'. The 'Address' field has the message 'Address'. The 'Postcode' field has the message 'Postcode'. The 'Country' dropdown has the message 'please select'. The 'D.O.B.' field has the message 'Date of Birth'. The 'Sex' radio buttons have the message 'Male Female Please select an option'. The 'Disclaimer' checkbox has the message 'Disclaimer'.

Language fundamentals

Data Types

- **Number**
- **String**
- **Boolean**
- **null**
- **undefined**
- **Object**
 - **Function**
 - **Array**
 - **Date**
 - **RegExp**



Language fundamentals

Data Types

- **Number**
- **String**
- **Boolean**
- **null**
- **undefined**
- **Object**
 - **Function**
 - **Array**
 - **Date**
 - **RegExp**

Numbers

- Double-precision 64-bit format [Int and floats]
- Leads to problems like
$$0.1 + 0.2 == 0.30000000000000004$$
- All standard arithmetic operators (+ , - , * , / , %)
- **Math Object** , Math.sin() , Math.round() , Math.floor() etc
- parseInt() , parseFloat() for parsing string to numbers
- Special Numbers - **NaN** , **Infinity**

String

- Sequences of 16 bit Unicode chars . It will support any language . हिन्दी , తమిల్
- String has a lot of built in functions, properties . DEMO

Boolean

- Coerce any thing into Boolean using Boolean()
- **Falsy** Values :false, 0, "",NaN, null, and undefined
- **Truthy** Values: Everything else

Language fundamentals

Data Types

- Number
- String
- Boolean
- null
- undefined
- Object
 - Function
 - Array
 - Date
 - RegExp

Declaring a variable

```
var a;  
var b = 10;  
b = "Hello World"
```

`console.log(a)` // Would show **undefined** - Means its declared but not defined

null

Is an assignment value.

Can be used for explicitly saying at a point of execution that variable is not available or doesn't have an actual value.

null absence of value for a variable;

undefined absence of variable itself;

Language fundamentals

Data Types

- Number
- String
- Boolean
- null
- undefined
- Object
 - Function
 - Array
 - Date
 - RegExp

Object

- Most important part of JS .Everything is an object in JS . Even Functions
- Simple collections of name-value pairs
- Primitives are immutable

```
var obj = new Object();
var obj = {};
```



```
function Person(name, age) {
    this.name = name;
    this.age = age;
}
var p1 = new Person("John Doe", 24);
```

TIP: All object assignments are References . i.e when you do

```
var p2 = p1 // This will point to same place in memory
```

Language fundamentals

Data Types

- Number
- String
- Boolean
- null
- undefined
- Object
 - **Function**
 - **Array**
 - **Date**
 - **RegExp**

Functions

- First class citizens

```
function add(x, y) {  
    var total = x + y;  
    return total;  
}
```

```
var add = function(x, y) {  
    var total = x + y;  
    return total;  
}
```

This that you can do :

add(3,4)	// will return 7
add("hello","world")	// will return "helloworld"
add(3,4,5)	// ??
add(3)	// ??
add()	// ??

All functions will have access to special parameters inside its body like **arguments** , **this** etc.

Language fundamentals

Data Types

- Number
 - String
 - Boolean
 - null
 - undefined
 - Object
 - Function
 - **Array**
 - Date
 - RegExp

Arrays []

- Special type of Objects.
 - Has a special property called **length**
 - **length** is **not** the number of elements of array
 - It is one more than **highest index** in the array

Language fundamentals

Data Types

- **Number**
- **String**
- **Boolean**
- **null**
- **undefined**
- **Object**
 - **Function**
 - **Array**
 - **Date**
 - **RegExp**

Date

- Exact replica of Java date class.
- **new Date()** gives you the timestamp accurate to milliseconds from 1/1/1970
- **new Date(10-1-2015)** gives you a date object with that days timestamp
- Lots of date manipulation functions inbuilt. Also lots of good i18n functions

Regular Expressions

- One of the least exploited parts of JS.
- Good for form validation .
- Can be used along with String.replace method

Flow Control

Support for almost every flow control structure.

Including :

if then else

while

for

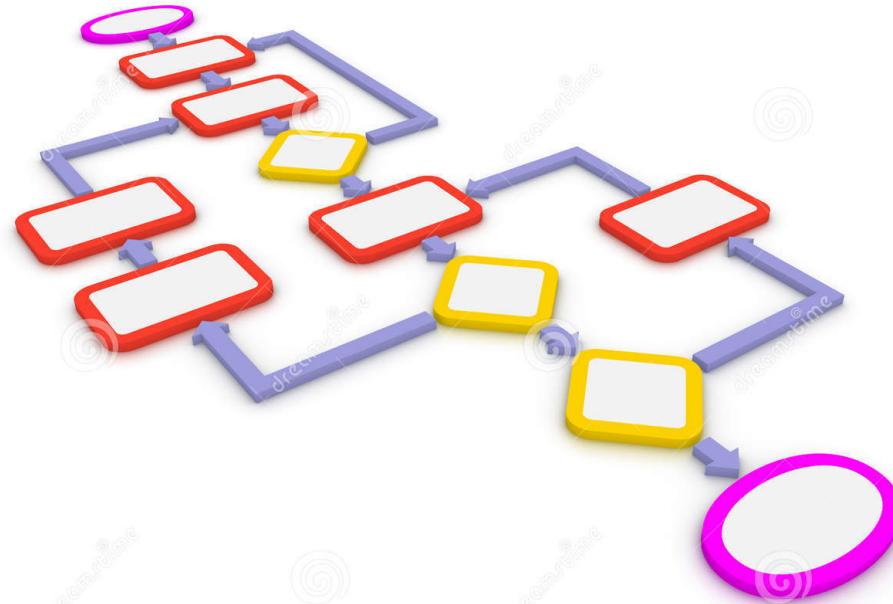
ternary operator

switch case

for in

break

continue



Error Handling

Best way to handle errors is to write good code.

Next best way is to use try catches

```
try {  
    Block of code to try  
}  
  
catch(err) {  
    Block of code to handle errors  
}
```

This ensures that rest of the code continues to execute . Otherwise your code will stop executing at the line where the error occurred . Leading to total disaster.

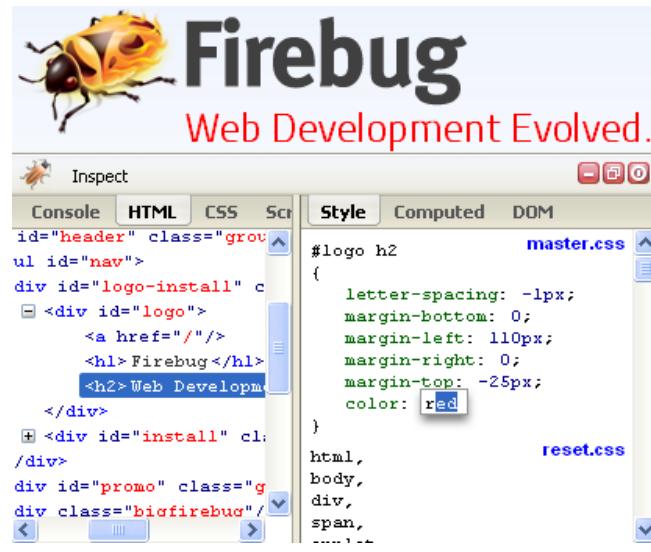
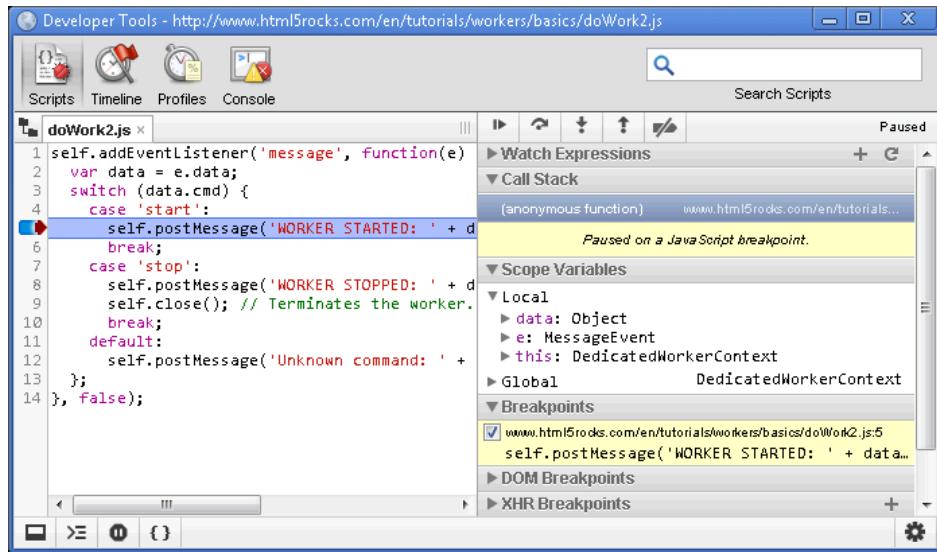
Error handling

CATCH ALL THE ERRORS!



Debugging

F12



Lets talk about the DOM

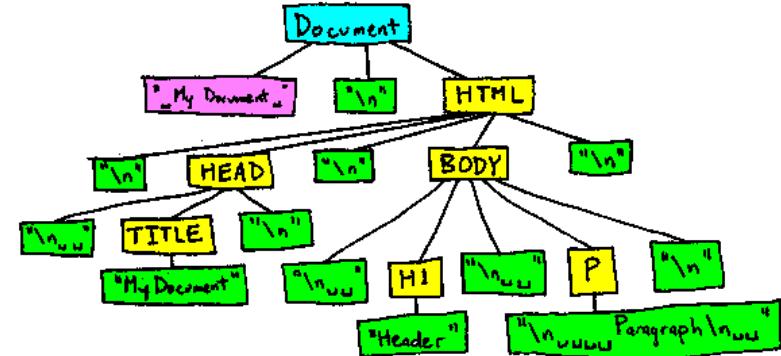
Document Object Model exposes many APIs to mess with its Objects.

eg : getElementByID("id") , getElementsByClassName().

By using / misusing the APIs we can bring magic/disaster to web pages.

DOM Manipulation

DOM Manipulation is slow. Depends based on the browser /OS/ System resources and implementation of DOM. After each manipulation depending on what changes you made , browser has to do a rerender / repaint.



Over to JSBin

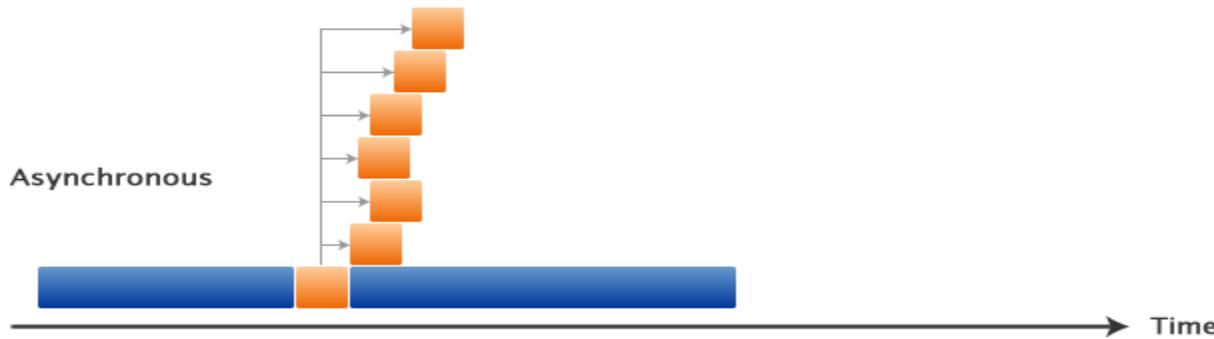
Sync Vs Async

- Synchronous code runs line by line
- Async code runs parallelly
- [DEMO](#)

Synchronous



Asynchronous



Hoisting

Variable Hoisting

```
var declaredLater;  
  
// Outputs: undefined  
console.log(declaredLater);  
  
declaredLater = "Now it's defined!";  
  
// Outputs: "Now it's defined!"  
console.log(declaredLater);
```

Function Hoisting

```
// Outputs: "Definition hoisted!"  
definitionHoisted();  
  
// TypeError: undefined is not a function  
definitionNotHoisted();  
  
function definitionHoisted() {  
    console.log("Definition hoisted!");  
}  
  
var definitionNotHoisted = function () {  
    console.log("Definition not hoisted!");  
};
```

JSON

- Universal Data exchange format of web
- Invented initially for representing JS objects
- Supports all the basic data types

```
{  
    "firstName": "John",  
    "lastName": "Smith",  
    "isAlive": true,  
    "age": 25,  
    "address": {  
        "streetAddress": "21 2nd Street",  
        "city": "New York",  
        "state": "NY",  
        "postalCode": "10021-3100"  
    },  
    "phoneNumbers": [  
        {  
            "type": "home",  
            "number": "212 555-1234"  
        },  
        {  
            "type": "office",  
            "number": "646 555-4567"  
        }  
    ],  
    "children": [],  
    "spouse": null  
}
```

A large, three-dimensional logo for "JSON". The letters are rendered in a dark gray/black color with blue outlines and shadows, giving them a metallic, rounded appearance. They are enclosed within blue curly braces {}, which are also rendered in a similar 3D style.

Scope

Scope is the set of variables you have access to.

In JS there are mainly two scopes

- 1) Local Scope
- 2) Global Scope
- 3) Automatic Global

Any variable declared inside a function using var has local scope

Any variable declared outside it has global scope

Special Case : Any variable declared inside a function without the "var" keyword is assumed global and is assigned to global scope.

```
function foo() {  
    var a = 10; // Local scope  
}  
  
var b = 100; // Global scope  
  
function bar() {  
    c = 10; // Automatic global  
}
```

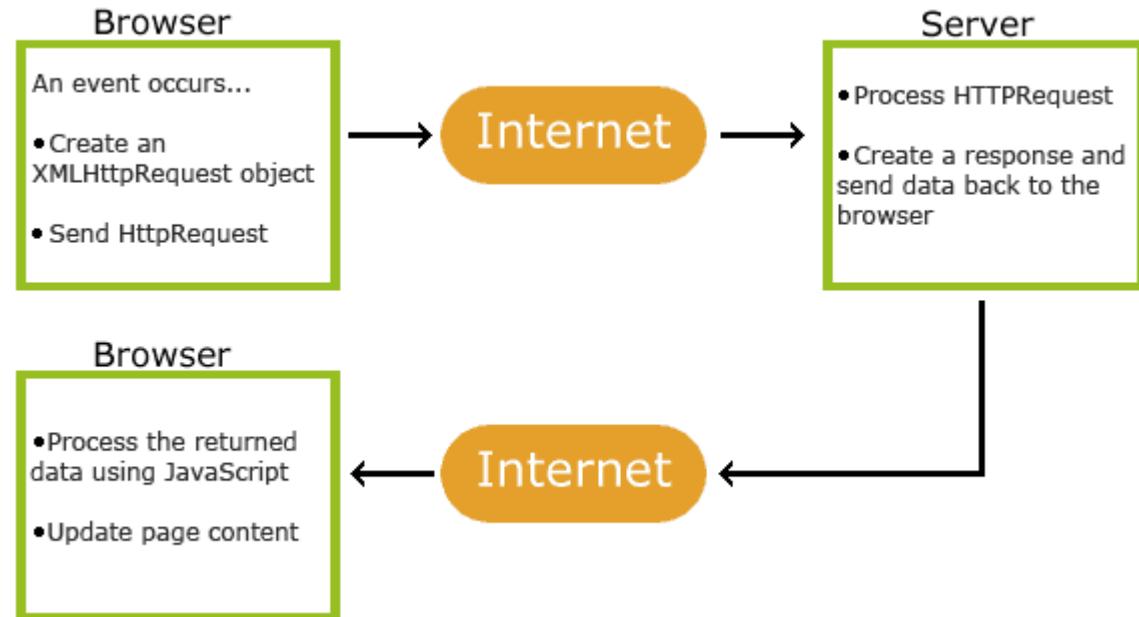
AJAX

AJAX - Asynchronous JavaScript and XML.

Something that [microsoft](#) did right :P

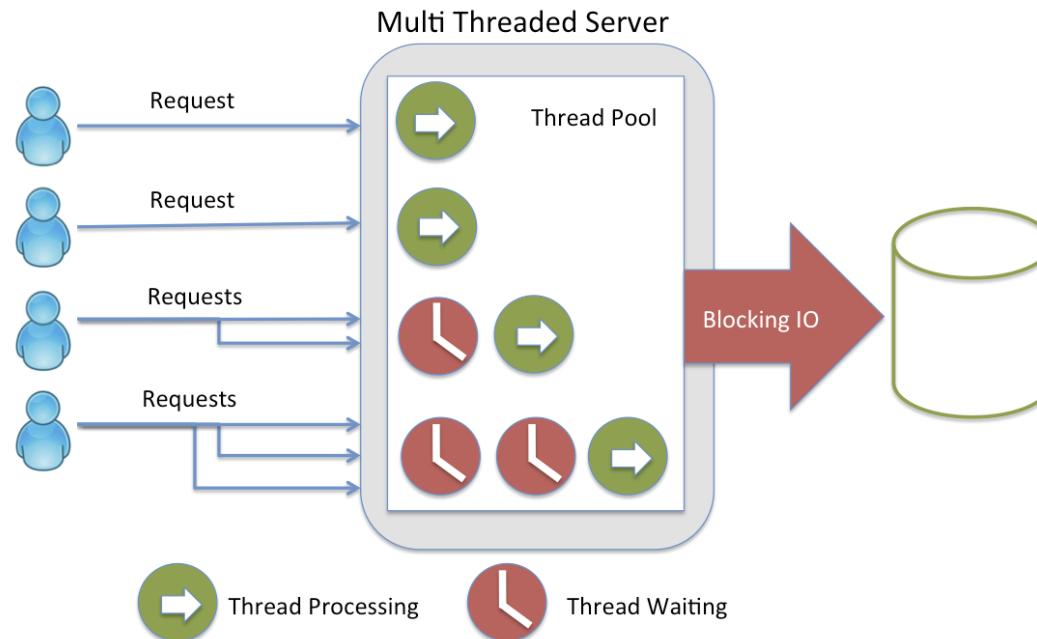
AJAX is the art of exchanging data with a server, and updating parts of a web page - without reloading the whole page.

Implemented using the XMLHttpRequest
abstracted in jQuery using `$.ajax({})`



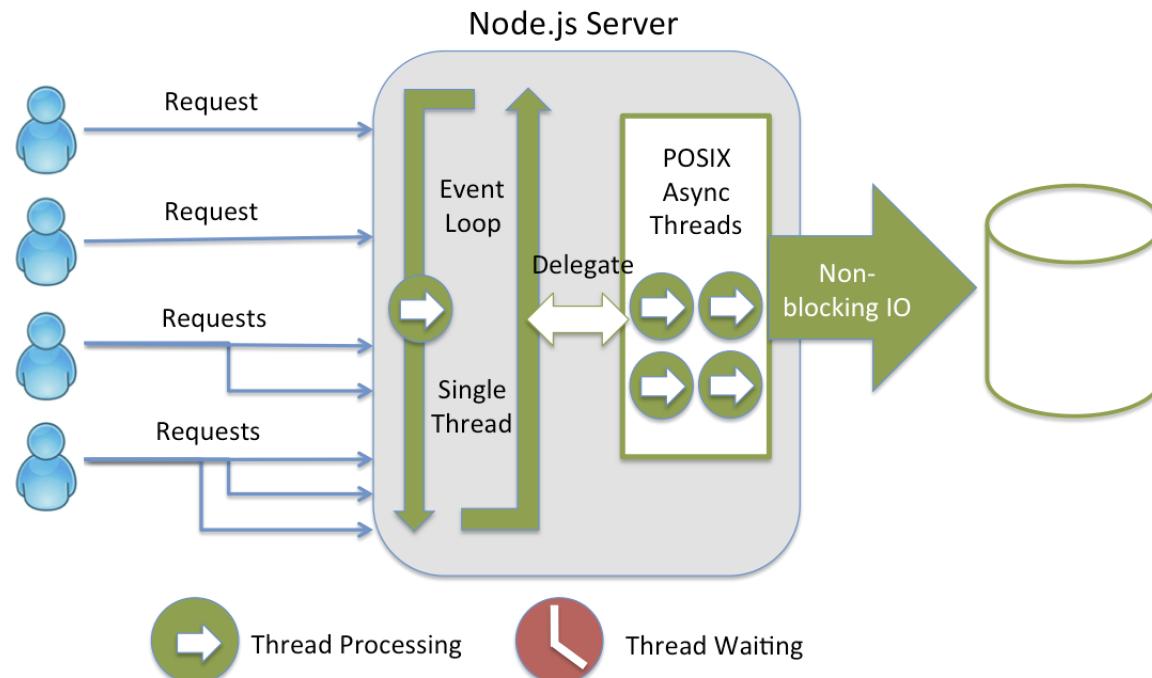
JS on the Server

Whats wrong with our servers ?



JS on the Server

How does node.js help ?



JS on the Server

Setting up a server using Node

```
var http = require('http');

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(1337, '127.0.0.1');

console.log('Server running at http://127.0.0.1:1337/');
```

JS on the Server

Setting up a server using Node



Thankyou !

Questions ?