

NodeJS

Under the hood!



Don't just code, learn



Akash Saxena

CTO & Co-Founder, GetSetGo Fitness

I believe that code should not be imitated. It should be understood!

TO START
WITH ...

LET'S LOOK AT WHAT
I MEAN

**LAST STEP BEFORE
WE START**

LET'S DO A SMALL EVALUATION ...

THE COMMAND LINE INTERFACE

LET'S LOOK AT WHAT WE WILL BE
DEALING WITH A LOT WHILE CODING
WITH NODEJS

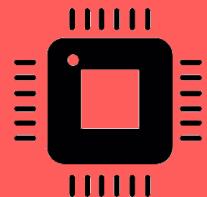
PROCESSORS, MACHINE CODE AND C++

Javascript

C++

Assembly Language

Machine Language



IA-32
X86-64
MIPS
ARM

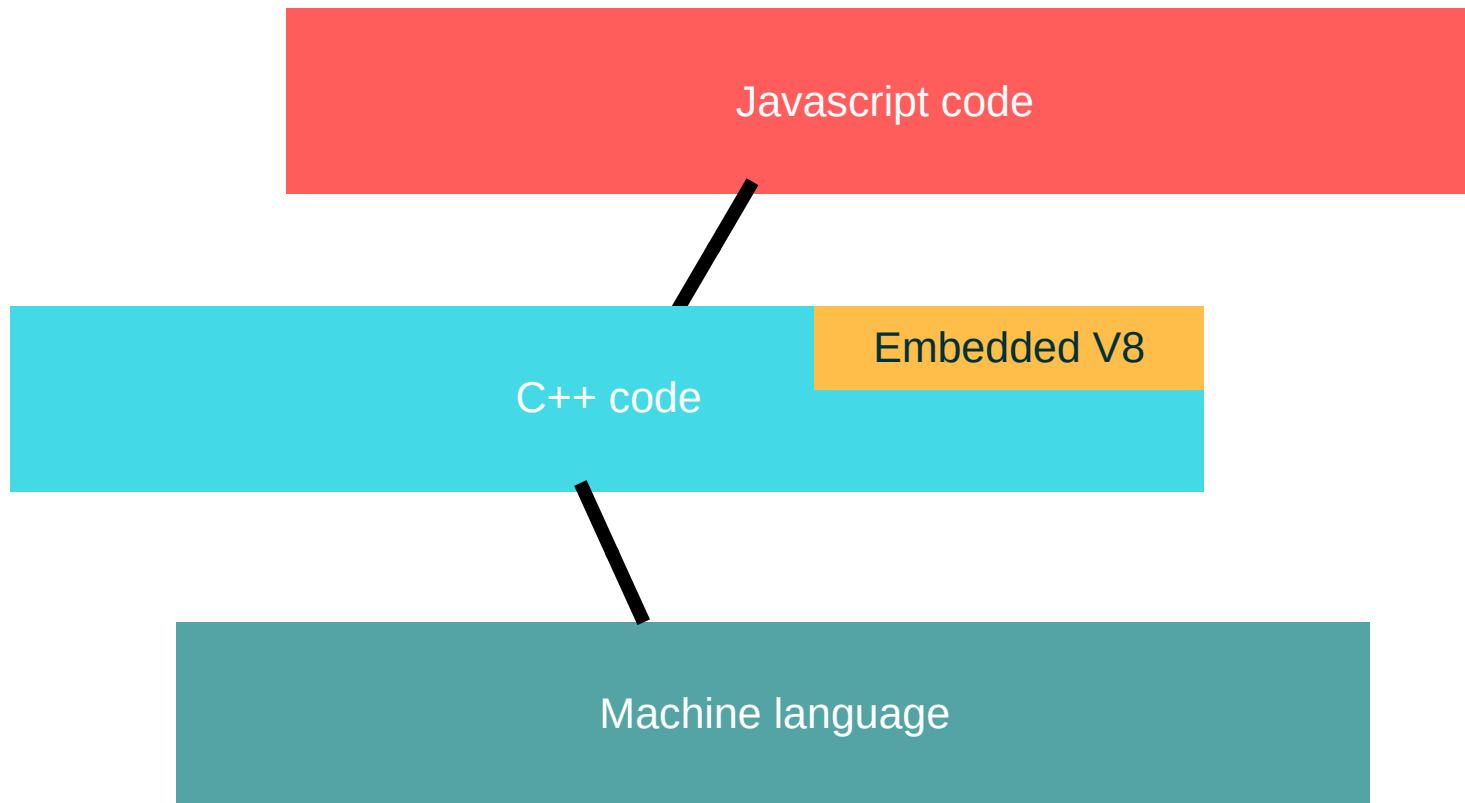
ECMA 6

SPECIFICATION

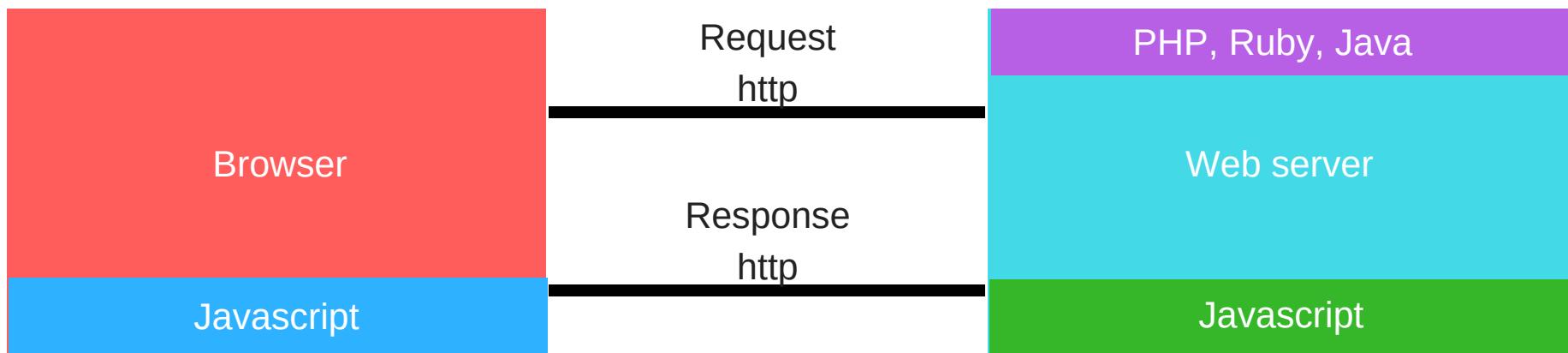
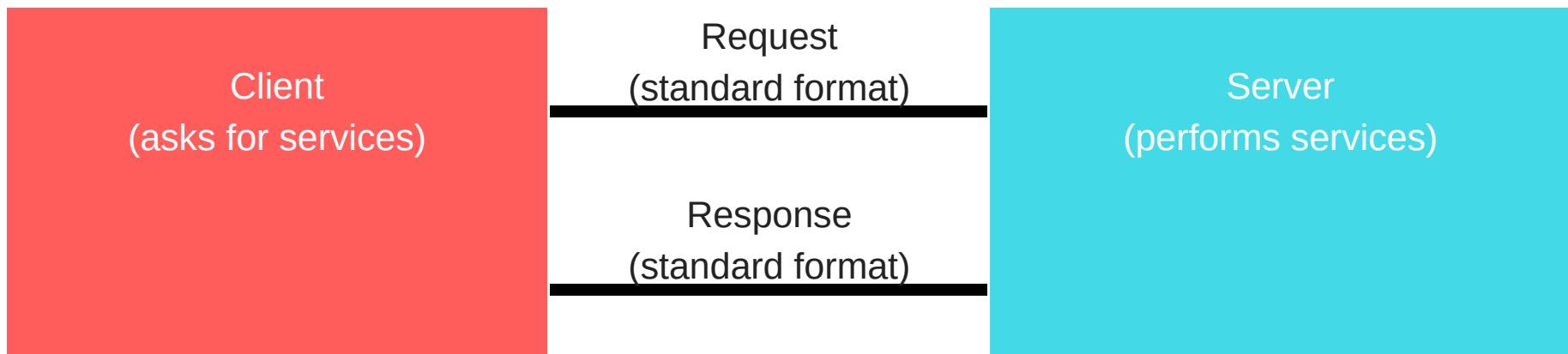
JAVASCRIPT SPECIFICATION

A specification to standardize the javascript engines over the world

ADDING FEATURES TO JAVASCRIPT



CLIENT-SERVER MODEL



WHAT DOES JAVASCRIPT NEED TO BE A WEB- SERVER

- Better ways to organize your code into reusable pieces
- Ways to deal with files
- Ways to deal with databases
- Ability to communicate over the internet
- Ability to accept requests and send responses in standard http format
- A way to deal with work that take a long time

THE C++ CORE

Let's look at the nodeJS repository

THE JAVASCRIPT CORE

Let's look at the nodeJS repository and check the 'lib' folder

LET'S RUN OUR
FIRST JS CODE IN
NODE

MODULE

A REUSABLE CODE THAT DOES NOT ACCIDENTALLY AFFECT OTHER CODE

It's not only important to create a profitable business, it's also important that you're happy managing and growing it day in and day out.

JS FUNCTIONS

**FUNCTIONS IN
JAVASCRIPT ARE
FIRST-CLASS**

MODULE

LET'S BUILD A MODULE

IMPORTANT JS CONCEPTS

- OBJECT LITERALS
- PROTOTYPICAL INHERITANCE
- PASS VALUE BY REFERENCE
- IIFE (POTENTIAL ISSUE WITH HOISTING)

MODULES

HOW DO THEY REALLY WORK?

Let's step into the JavaScript require function and see how it works

JSON

JAVASCRIPT OBJECT NOTATION

a standard for structuring data that is inspired by JS object Literals

REQUIRE FROM MULTIPLE FILES

Require from a single folder. Include json configuration. JSON file is read and its contents are converted to a JS object

MODULE PATTERNS

Let's look at different patterns that modules can be declared with
Remember to look at the cached module and the revealing pattern

MODULE

EXPORT VS MODULE.EXPORT

As soon as an object is assigned a new value, the reference (pointing to the same location) is broken. Also, let's look at how it can be resolved with mutation

REQUIRING NATIVE/CORE MODULES

Check module.js for native module
Check util.js

REQUIRING AND ES6

Let's check the code

NODEJS

EVENTS

**SYSTEM
EVENTS**

C++ core
libuv

**CUSTOM
EVENTS**

Javascript core
Event emitter

First-class function array

- Let's declare an array of functions, parse them one-by-one and invoke them

EVENT Emitter

LET US WRITE A SIMPLE EVENT Emitter

Remember that this is nothing but a simple array of functions that are executed in sequence

EVENT Emitter

LET US USE THE NODEJS EVENT Emitter

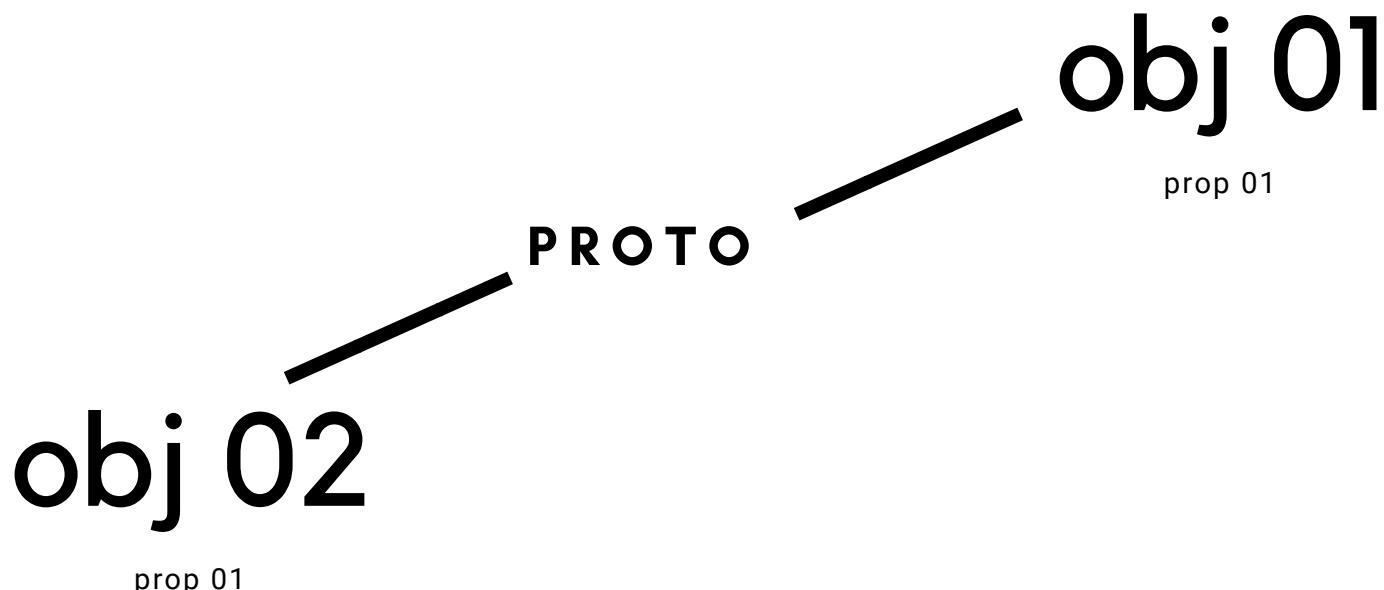
Remember to look at events.js before you use it
Also, define the events in a central config and load from there

OBJECT.CREATE ... AND PROTOTYPE

Let us use `Object.create` with an object literal and see how the prototypical inheritance functions

EVENT Emitter

INHERITING FROM EVENT Emitter



Let's look at util.js for inherits function
Let us also look at passing data to the listener while emitting