

```
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');
```

COMMAND LINE INTERFACE: A UTILITY TO TYPE COMMANDS TO YOUR COMPUTER RATHER THAN CLICKING

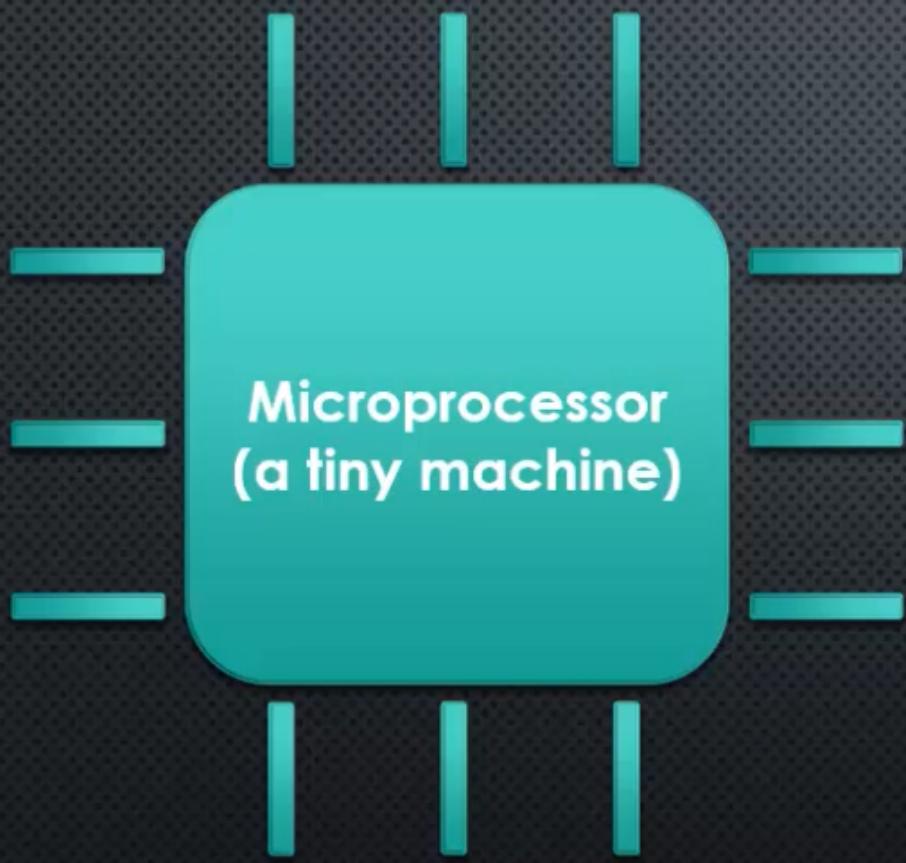
Bash on Linux,
Terminal on Mac,
Command Prompt on Windows,
and other replacements...

ARGUMENTS: VALUES YOU GIVE YOUR PROGRAM THAT AFFECT HOW IT RUNS

Essentially passing parameters to a function

V8

THE JAVASCRIPT ENGINE



Instructions

IA-32

x86-64

ARM

MIPS

MACHINE CODE (LANGUAGE): PROGRAMMING LANGUAGES SPOKEN BY COMPUTER PROCESSORS

Every program you run on your computer has been converted (compiled) into machine code.

000018A45438100	0	55	push rbp
0000018A45438101	1	4889e5	REX.W movq rbp,rsp
0000018A45438104	4	56	push rsi
0000018A45438105	5	57	push rdi
0000018A45438106	6	41ff75a8	push [r13-0x58]
0000018A4543810A	10	56	push rsi
0000018A4543810B	11	49baf9552c7e8f010000	REX.W movq r10, object: 0000018F7E2C55F9
0000018A45438115	21	4152	push r10
0000018A45438117	23	6a00	push 0x0
0000018A45438119	25	b803000000	movl rax,000000000000

Level of Abstraction

Javascript

C/C++

Assembly Language

Machine Language



Node is written in C++

V8 is written in C++

ECMASCRIPT: THE STANDARD JAVASCRIPT IS BASED ON

Needed a standard since there are many engines.

**A JAVASCRIPT ENGINE:
A PROGRAM THAT CONVERTS
JAVASCRIPT CODE INTO
SOMETHING THE COMPUTER
PROCESSOR CAN UNDERSTAND**

And it should follow the ECMAScript standard on how the language should work and what features it should have.

**V8
UNDER THE HOOD**

ADDING FEATURES TO JAVASCRIPT

Javascript Code

V8 (C++)

Machine Code

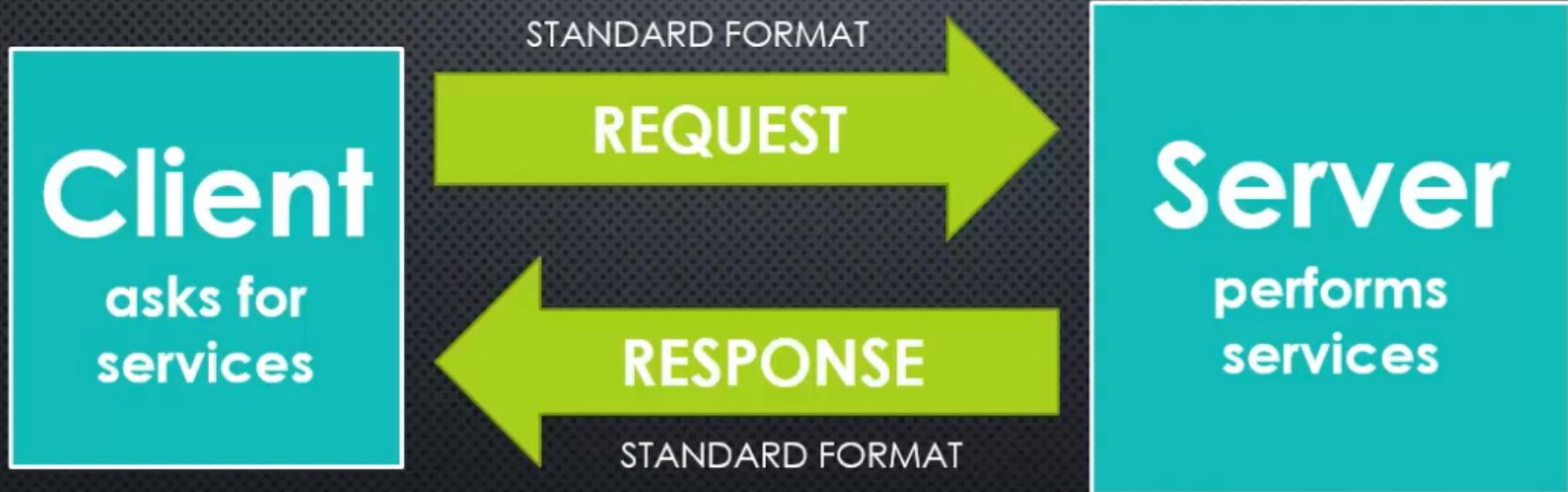
Javascript Code

My C++ Program

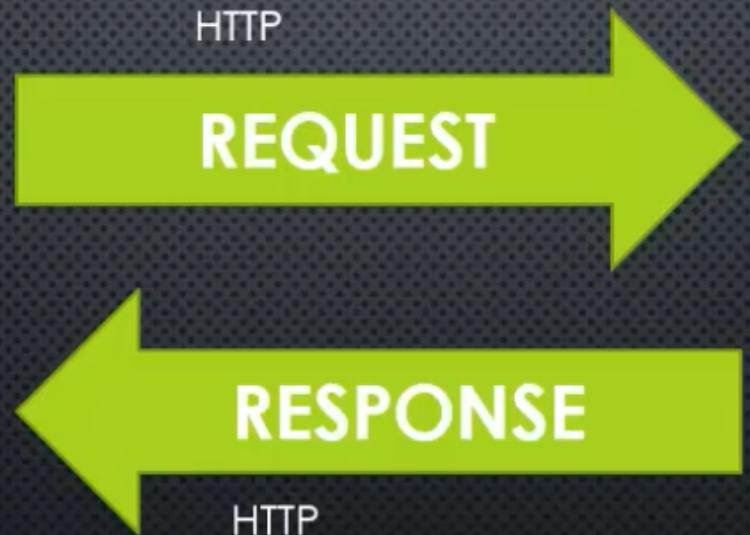
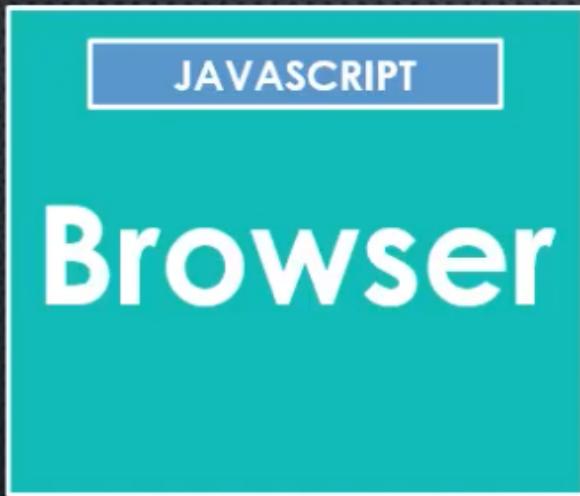
V8 (C++)

Machine Code

**THE NODE
CORE**



Client-Server Model of Computing





**WHAT DOES
JAVASCRIPT NEED TO
MANAGE A SERVER?**

Better Ways to Organize Our Code Into Reusable Pieces

Ways to Deal with Files

Ways to Deal with Databases

The Ability To Communicate Over the Internet

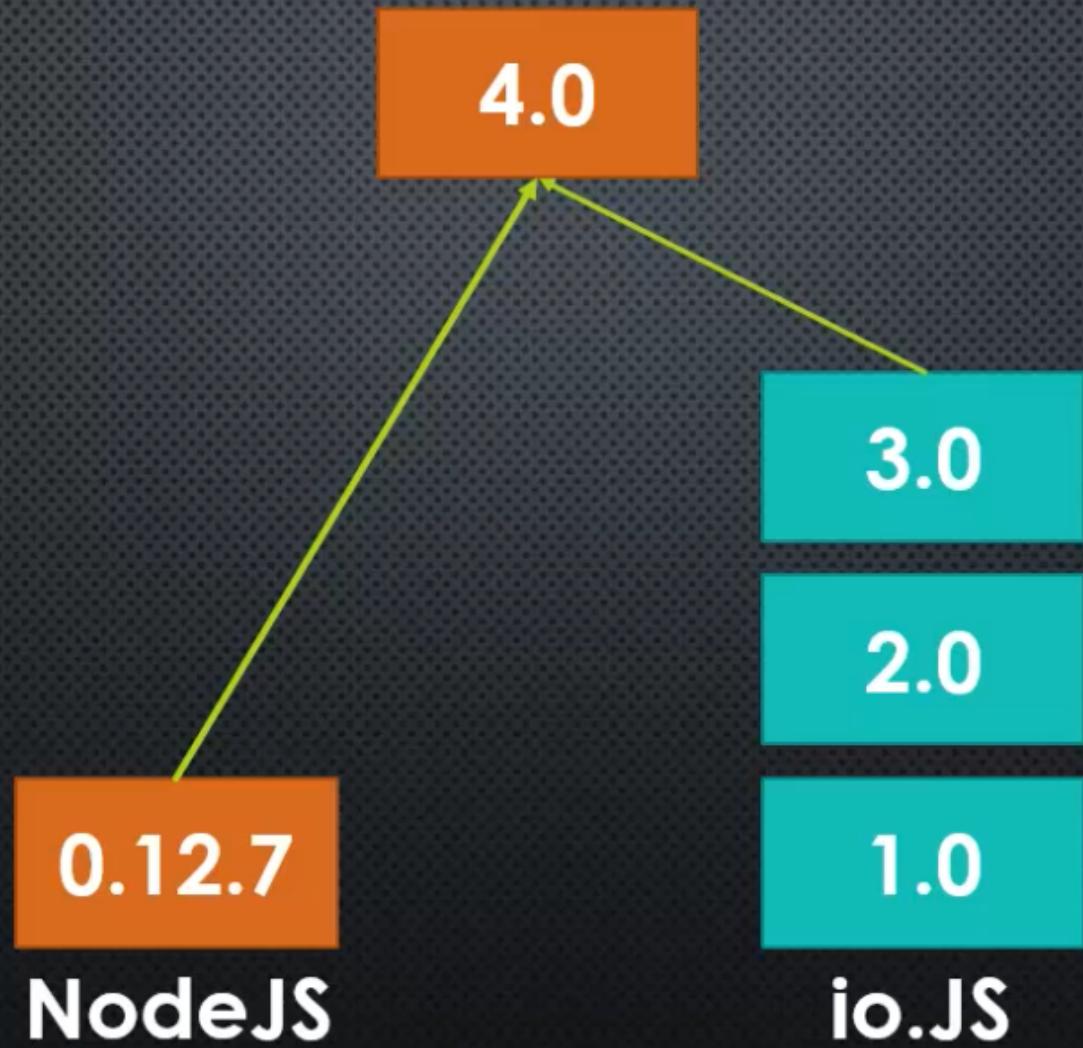
**The Ability to Accept Requests and Send Responses
(in the standard format)**

A Way to Deal with Work that Takes a Long Time

THE C++ CORE

THE JAVASCRIPT CORE

**LET'S RUN SOME
JAVASCRIPT IN NODE**



**BREAKPOINT:
A SPOT IN OUR CODE WHERE
WE TELL A DEBUGGING TOOL TO
PAUSE THE EXECUTION OF OUR
CODE.**

So we can figure out what's going on....

**MODULES,
EXPORTS, AND
REQUIRE**

**MODULE:
A REUSABLE BLOCK OF CODE
WHOSE EXISTENCE DOES NOT
ACCIDENTALLY IMPACT
OTHER CODE**

Javascript didn't have this before.

**COMMONJS MODULES:
AN AGREED UPON STANDARD
FOR HOW CODE MODULES
SHOULD BE STRUCTURED**

FIRST-CLASS FUNCTIONS: EVERYTHING YOU CAN DO WITH OTHER TYPES YOU CAN DO WITH FUNCTIONS.

You can use functions like strings, numbers, etc. (i.e. pass them around, set variables equal to them, put them in arrays, and more)

AN EXPRESSION: A BLOCK OF CODE THAT RESULTS IN A VALUE

**Function expressions are possible in Javascript because
functions are first-class.**

INVOKE: RUN THE FUNCTION

We can also say ‘call’ the function.

PRIMITIVE: A TYPE OF DATA THAT REPRESENTS A SINGLE VALUE

Like a number or a string. In other words, **not an object**.

```
graph LR; a[a] -- "0x001" --> val1[Primitive Value]; b = a["b = a<br>(or pass to a function)"] -- "0x002" --> val2[Copy of Primitive Value]; b[b] -- "0x002" --> val2;
```

a

b = a

(or pass to a function)

b

0x001

0x002

Primitive
Value

Copy of
Primitive
Value

a

b = a

(or pass to a function)

b

by reference

0x001

0x001

Object

SCOPE:
**WHERE IN CODE YOU HAVE
ACCESS TO PARTICULAR
VARIABLE OR FUNCTION.**

HOW DO NODE MODULES REALLY WORK?: MODULE.EXPORTS AND REQUIRE

```
(function (exports, require, module, __filename, __dirname) {  
  
  var greet = function() {  
    console.log('Hello!');  
  };  
  
  module.exports = greet;  
  
});
```

```
(function (exports, require, module, __filename, __dirname) {  
  
    var greet = function() {  
        console.log('Hello!');  
    };  
  
    module.exports = greet;  
  
});  
  
fn(module.exports, require, module, filename, dirname);  
  
return module.exports;
```

require is a function, that you pass a ‘path’ too

module.exports is what the require function *returns*

this works because **your code is actually wrapped in a function** that is given these things as function parameters

HTTP AND BEING A WEB SERVER

**PROTOCOL:
A SET OF RULES TWO SIDES
AGREE ON TO USE WHEN
COMMUNICATING.**

Both the client and server are programmed to understand and use that particular set of rules. It's similar to two people from different countries agreeing on a language to speak in.

Client
asks for
services

STANDARD FORMAT

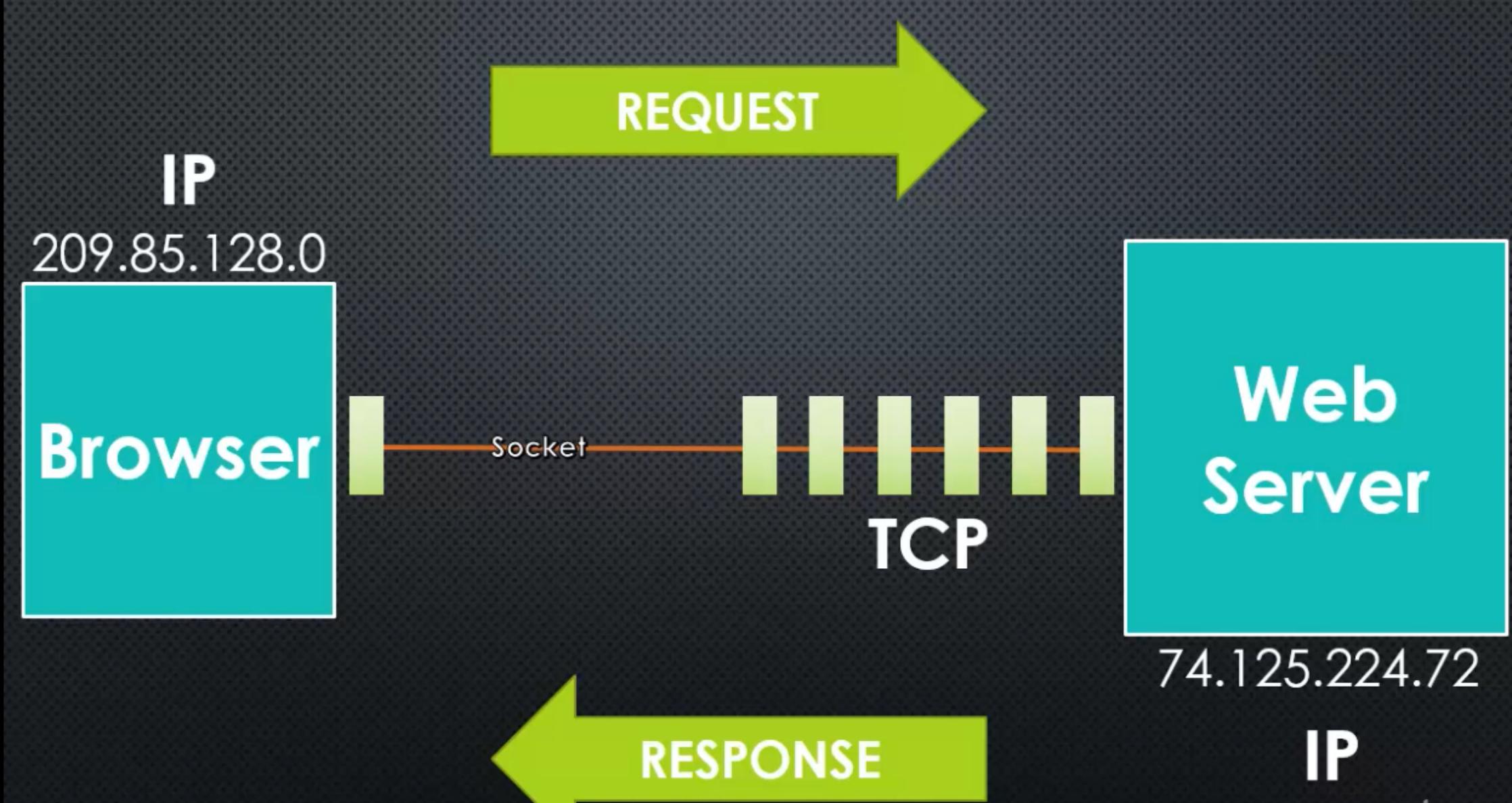
REQUEST

RESPONSE

STANDARD FORMAT

Server
performs
services

Client-Server Model of Computing



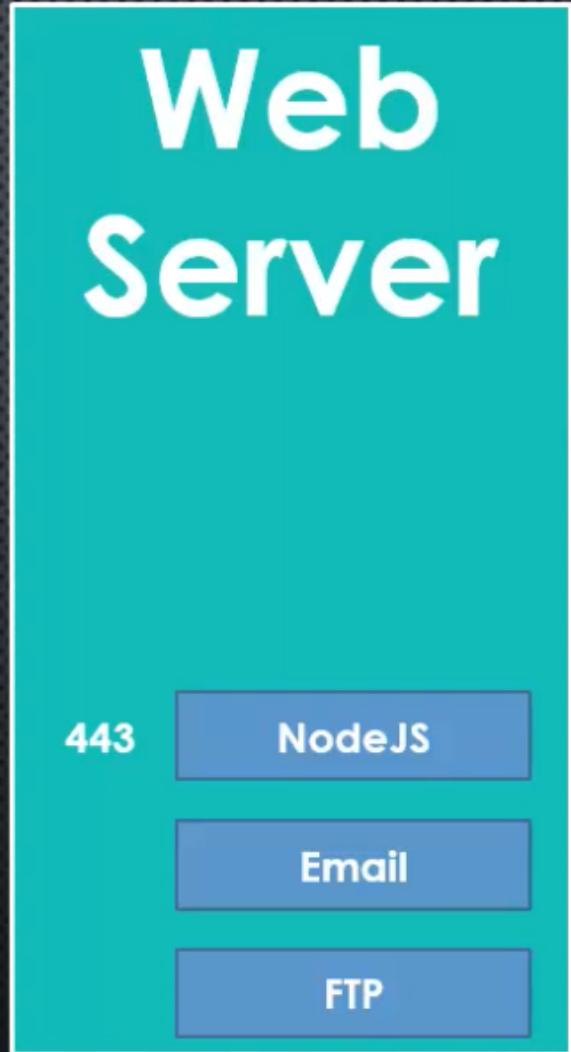
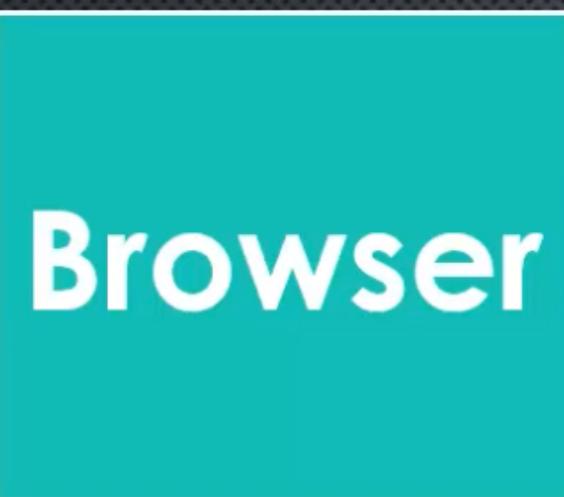
**PORT:
ONCE A COMPUTER RECEIVES
A PACKET, HOW IT KNOWS
WHAT PROGRAM TO SEND IT
TO.**

When a program is setup on the operating system to receive packets from a particular port, it is said that the program is 'listening' to that port.

78.132.160.4:443

Socket address

<https://www.google.com>



**HTTP:
A SET OF RULES (AND A
FORMAT) FOR DATA BEING
TRANSFERRED ON THE WEB.**

Stands for ‘HyperText Transfer Protocol’. It’s a format (of various) defining data being transferred via TCP/IP.

CONNECT www.google.com:443 HTTP/1.1
Host: www.google.com
Connection: keep-alive

HTTP/1.1 200 OK

Content-Length: 44

Content-Type: text/html

<html><head>...</head></html>

MIME type: A STANDARD FOR SPECIFYING THE TYPE OF DATA BEING SENT.

Stands for ‘Multipurpose Internet Mail Extensions’.

Examples: application/json, text/html, image/jpeg

HTTP_PARSER

**LET'S BUILD A
WEB SERVER IN NODE**

OUTPUTTING HTML AND TEMPLATES

**TEMPLATE:
TEXT DESIGNED TO BE THE
BASIS FOR FINAL TEXT OR
CONTENT AFTER BEING
PROCESSED.**

There's usually some specific template language, so the template system knows how to replace placeholders with real values.

API: A SET OF TOOLS FOR BUILDING A SOFTWARE APPLICATION.

Stands for ‘Application Programming Interface’. On the web the tools are usually made available via a set of URLs which accept and send only data via HTTP and TCP/IP.

ENDPOINT: ONE URL IN A WEB API.

Sometimes that endpoint (URL) does multiple thing by making choices based on the HTTP request headers.

OUTPUTTING JSON

**SERIALIZE:
TRANSLATING AN OBJECT
INTO A FORMAT THAT CAN BE
STORED OR TRANSFERRED.**

JSON, CSV, XML, and others are popular. ‘Deserialize’ is the opposite (converting the format back into an object).