# Dev Bootcamp Day One - Intro



### **MATERIAL**

https://github.com/bootcamp-joinville/javascript-syllabus

### **SYLLABUS**

### Dia 1 - 16 de Julho

- Introdução ao curso
- Git/GitHub/GitHub Education
- Introdução ao HTML

Instrutor: Luan Milnitz

### Dia 3 - 18 de Julho

- JavaScript no browser
- DOM
- Eventos
- HTTP e Ajax

Instrutor: Diogo Trentini

#### Dia 2 - 17 de Julho

- Introdução ao JavaScript
- Valores, tipos, operadores
- Funções
- Arrays e Objetos

Instrutor: Lucas Tomkiel

### Dia 4 - 19 de Julho

- Introdução ao
  - Terminal
- Introdução ao Node.js

Instrutor: Alan R. Fachini

### **SYLLABUS**

### Dia 5 - 23 de Julho

- Introdução ao Projeto
- Hands on

Instrutor: Luan Milnitz

### Dia 7 - 25 de Julho

Hands on

Instrutor: Lucas Tomkiel

### Dia 6 - 24 de Julho

Hands on

Instrutor: Diogo Trentini

### Dia 8 - 26 de Julho

Hands on

Instrutor: Alan R. Fachini

### **INSTRUTORES**



Alan Fachini alfakini



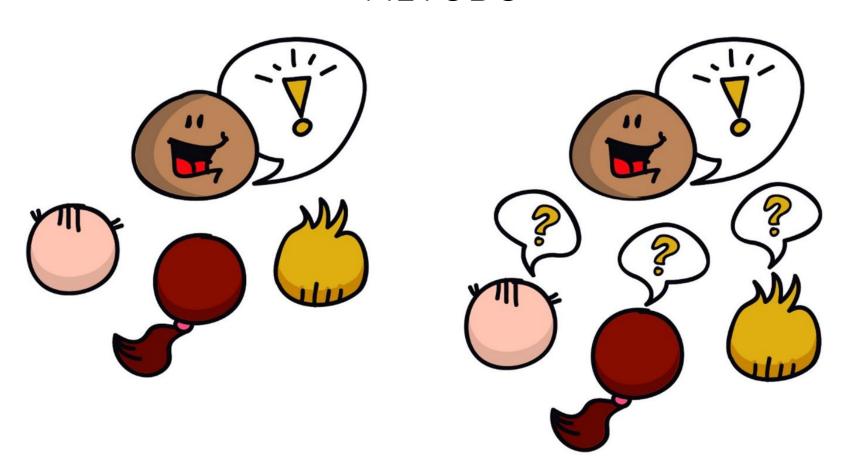
Diogo Trentini diogotrentini

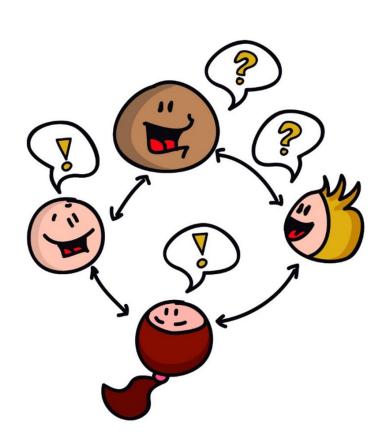


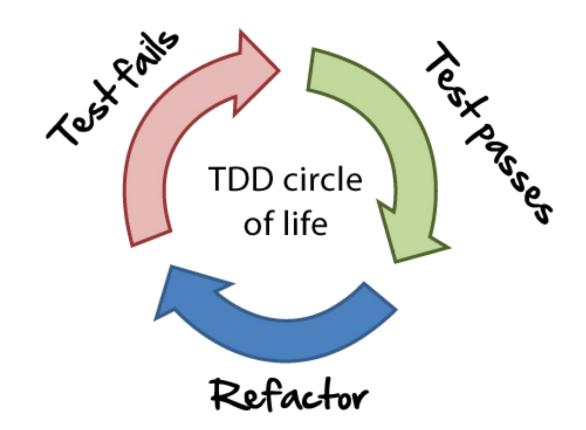
Lucas Tomkiel



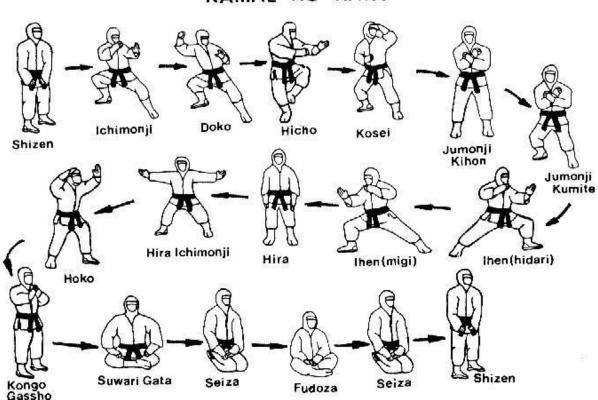
**Luan Millnitz** millnitzluan





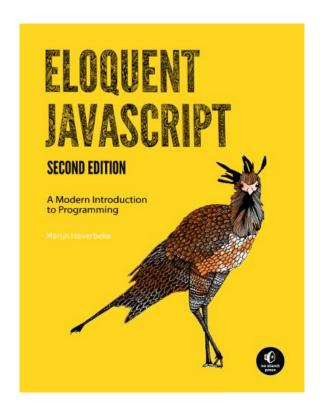


### KAMAE NO KATA





The first 20 hours - how to learn anything: Josh Kaufman



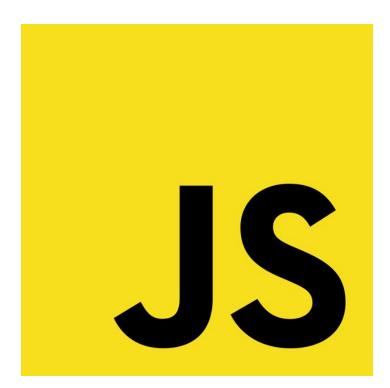
http://www.magrathealabs.com/eloquente-javascript

### **MATERIAL**

https://classroom.github.com/a/QYmlNgh0

### **DICAS**

function



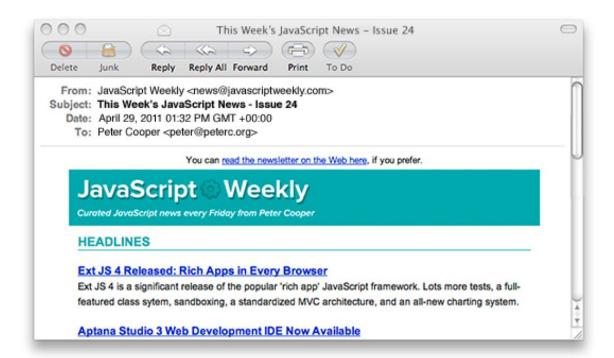
https://www.meetup.com/pt-BR/Javascript-Joinville

### **DICAS**



https://www.meetup.com/pt-BR/React-Joinville

### **DICAS**



http://javascriptweekly.com

# Dev Bootcamp Day One - HTML/CSS

## **HTML**

- HTML
  - Conteúdo
  - Dados e estrutura
- CSS
  - Apresentação
  - Formatação, layout, cores, fontes, posicionamento
- JavaScript
  - Comportamentos
  - Programação

# Hyper Text Markup Language

- Inventada por Tim Berners-Lee no fim da década de 80
- Nasceu junto com a Web
- Linguagem de Marcação Baseada em Hipertexto
- Versão: 5 (atual)



# Quem define

- <u>W3C</u>: World Wide Web Consortium
- WHATWG: Web Hypertext Application Technology Working Group



# Linguagem de Marcação

- Utiliza tags ou etiquetas
- Tag é um elemento que indica estrutura em um documento HTML
- Usam os sinais < e >
- Cada tag tem um significado diferente



# Tags

Exemplo:

```
 Moro em:
     <address>Rua <abbr>USA</abbr>, 23 </address>
```

 Toda a abertura de tag deve estar associada a um único fechamento

## Tags

• Certo:

```
<cite>...</cite><button></button>
```

• Errado:

```
<cite>...</cite><button>
```

## Estrutura básica

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
</head>
<body>
</body>
</html>
```

## Estrutura básica

- <html> e </html>, indicam que se trata de um documento em HTML;
- <head> e </head>, delimitam o cabeçalho da página;
- <title> e </title>, definem o título da página; e
- <br/> **body>** e **body>**, delimitam o conteúdo a mostrar ao utilizador;
- e , delimitam um parágrafo.

# Tipos de tags

- Estrutura: <html>, <head>, <body>...
- Metainformações: <meta>, <title>...
- **Texto**: , <code>, <br>...
- **Links**: <a>, <base>...
- Imagens: <img>, <area>....
- Objetos: <object>, <param>...

# Tipos de tags

- Listas: , , <dl>...
- **Tabelas**: , , ...
- Formulários: <form>, <select>, <input>...
- Scripting / Programação: <script>, <noscript>
- Apresentação / Formatação: <em>, <strong>...
- Desenho: <canvas>

## Atributos de tags

<a href="http://bootcamp.joinville.br" title="Site bootcamp">Site do Bootcamp</a>;

- O atributo href indica qual será a página ou local dentro da própria página que o link levará;
- O valor dentro da abertura e fechamento, indica qual é o texto do link que será exibido;

## Exemplos de atributos

- <a href="mailto:contato@magrathealabs.com">Ema il da MLabs</a>
- Este texto resume tudo
- <input type="text" value="Caixa de Texto"/>

## Atributo ID

- Atributo id: identifica um elemento;
- Deve ter um valor único para toda a página;
- Exemplo:

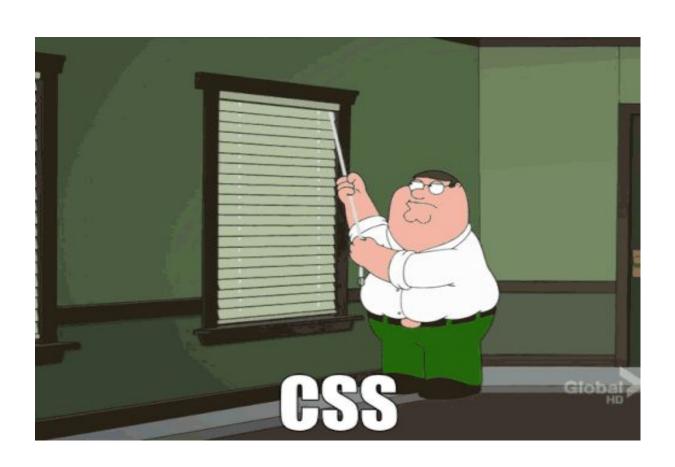
```
<span id="destaque">
</span>
<div id="cabecalho">
</div>
```

# Se aprofundando



https://developer.mozilla.org/pt-BR/docs/Aprender/HTML/Introducao ao HTML

# CSS



# **Cascading Style Sheets**

- Inventada por **Hakon Wium Lie** em 1994
- Nasceu com a **necessidade** de estilizar paginas html.
- Versão: 3 (atual)

## **CSS**

- Funciona diferentemente em cada browser.
- Deixa o código HTML mais limpo.
- Quando bem estruturado é fácil de dar manutenção.
- Ajuda na escala/reuso dos estilos.

# Tipos de seletores

- Tags: <body>, <div>, , <footer>...
- Id: id="meu-id" adiciona um estilo por id da tag.
- Classes: class="minha-class"É possível adicionar mais de uma classe por tag html

## Como usar

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" type="text/css"</pre>
  href="style.css"/>
</head>
</html>
```

# Estrutura arquivo .css

```
body {
  background-color: #ffe4c6;
  font-family: Arial, Helvetica,
sans-serif;
.classe1 { ... }
.classe1:hover > button{ ... }
ul li:nth-child(n+1) { ... }
#id { ... }
```

#### **DICAS**

Developer Mozilla: <a href="https://developer.mozilla.org/">https://developer.mozilla.org/</a>

W3schools: https://www.w3schools.com

BEM: <a href="http://getbem.com/">http://getbem.com/</a>

CSS-Tricks: <a href="https://css-tricks.com/">https://css-tricks.com/</a>

Bootstrap: <a href="https://getbootstrap.com/">https://getbootstrap.com/</a>

Bulma: https://bulma.io/

# Dev Bootcamp Day One - Exercícios

#### **MATERIAL**

https://github.com/bootcamp-joinville



https://facebook.github.io/jest

#### **JEST**

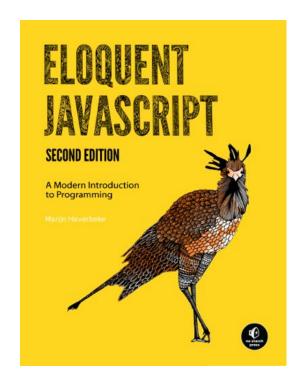
- Usado para testar aplicações JavaScript
- Paraleliza os testes por padrão
- Code coverage com --coverage

https://repl.it/@amasad/try-jest

# Dev Bootcamp Day Two

#### DAY 2

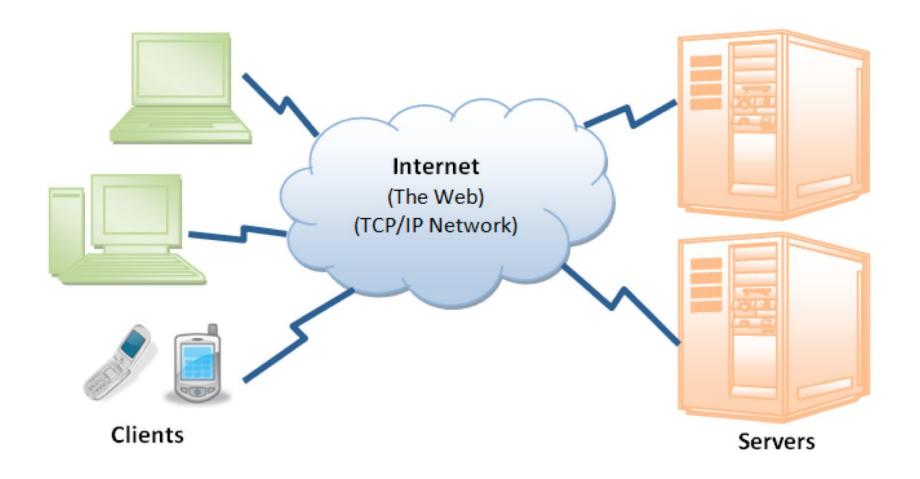
- Introdução ao JavaScript
- Valores, tipos, operadores
- Funções
- Arrays e Objetos



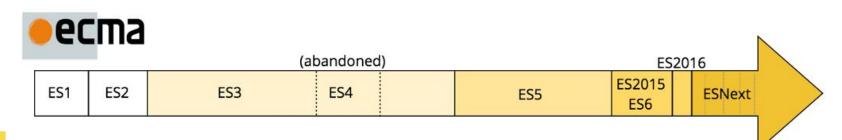
http://www.magrathealabs.com/eloquente-javascript

# Dev Bootcamp Day Three

#### A Internet e a Web



# JavaScript e o navegador











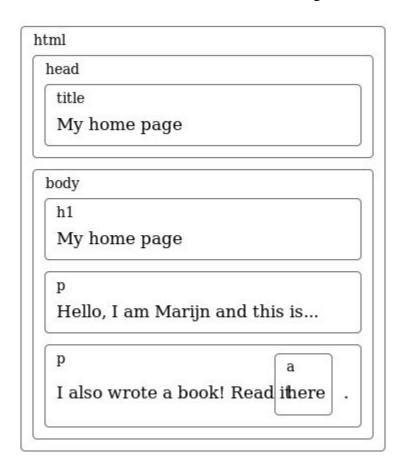




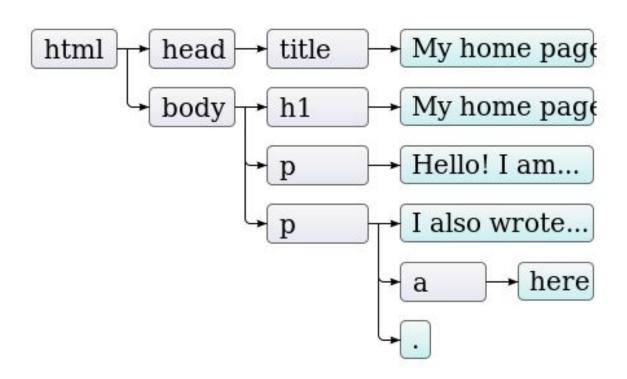
# DOM (Document Object Model)

```
<html>
  <head>
    <title>Minha home page</title>
  </head>
  <body>
    <h1>Minha home page</h1>
    Olá, eu sou Marijn e essa é minha home
page.
    Eu também escrevi um livro! leia-o
      <a
href="http://eloquentjavascript.net">aqui</a>.
  </body>
</html>
```

## DOM (Document Object Model)



### Árvore

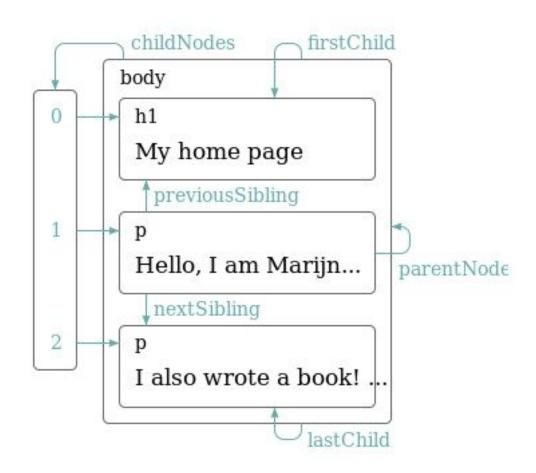


#### document

- document ▶ #document document.documentElement <-<html data-cast-api-enabled="true" lang="en" class="gr</pre> ▶ <head>...</head> ▶ <body dir="ltr" role="application" class itemscope i ▶ <div>...</div> </html>

document.body

# Movimentando-se pelo DOM



#### **Encontrando elementos**

- getElementsByTagName
- getElementById
- getElementsByClassName
- querySelectorAll
- querySelector

#### Alterando o documento

- appendChild
- removeChild
- insertBefore
- replaceChild

#### Criando novos elementos

- createTextNode
- createElement

#### Alterando atributos

- element.attributeelement.style
- getAttribute
- setAttribute

### **Eventos**

# Manipulando eventos

# Propagação

# Ações padrão

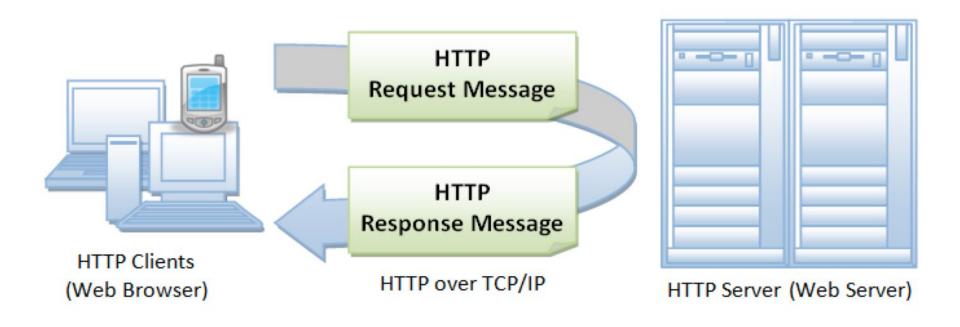
# Tipos de eventos

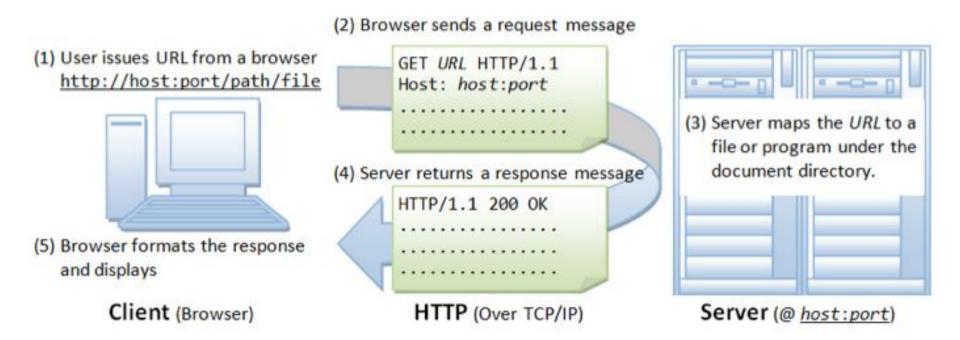
### Cronograma de execução

•

# **Temporizadores**

## **HyperText Transfer Protocol**





### **Uniform Resource Locator (URL)**

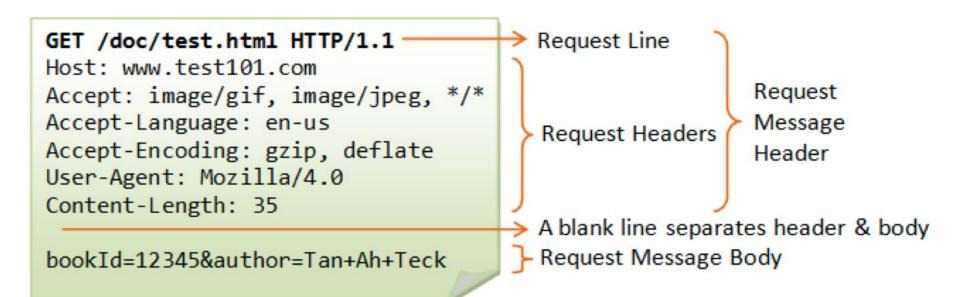
protocol://hostname:port/path-and-file-name

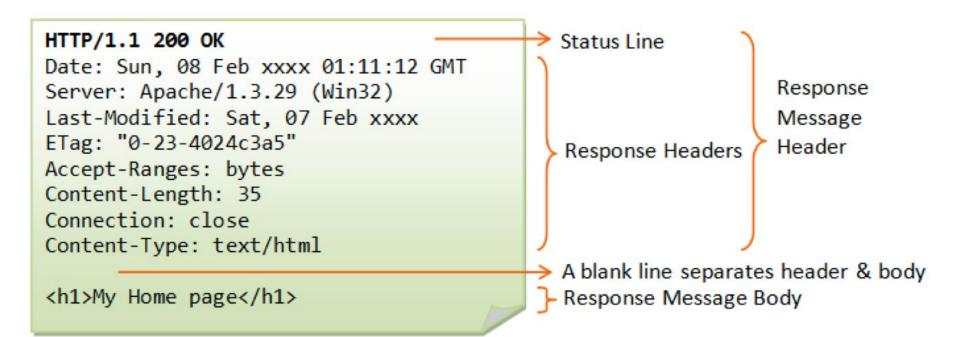
- Protocol: protocolo no nível de aplicação, e.g., HTTP, FTP, and telnet.
- Hostname Domínio DNS (e.g., www.nowhere123.com) ou endereço IP (e.g., 192.128.1.2) do servidor
- Port: a porta TCP
- Path-and-file-name: a localização

#### **HTTP Protocol**

Ao requisitar http://www.nowhere123.com/doc/index.html, o browser vai enviar a mensagem:

```
GET /docs/index.html HTTP/1.1
   Host: www.nowhere123.com
   Accept: image/gif, image/jpeg, */*
   Accept-Language: en-us
   Accept-Encoding: gzip, deflate
   User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT
5.1)
   (blank line)
```





# Form

# Fetch

# Dev Bootcamp Day Four



# **DevTools**

Ctrl + Shift + I

https://developer.chrome.com/devtools

# **Elements**

https://developers.google.com/web/tools/chrome-devtools/inspect-styles

# Console

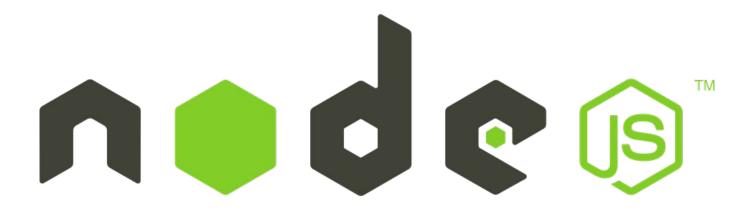
https://developers.google.com/web/tools/chrome-devtools/console

## CONSOLE

```
$("selector") = document.querySelector("selector")
$$("selector") = document.querySelectoryAll("selector")
$_ = última expressão
$0 = elemento selecionado
copy(variável) = copia para o clipboard
inspect(elemento) = abre o painel Elements
console.table(object, ["list", "of", "attributes"])
```

# Programando em JS no Browser

- 1. Abra o DevTools
- 2. Vá ao painel Sources, na barra a esquerda, mude para Snippets
- 3. Crie um novo snippet clicando na área abaixo e de um nome
- 4. Escreva seu código e clique em Run (>) ou Ctrl+ Enter



https://nodejs.org
http://bit.ly/guia-nodejs







**DOW JONES** 















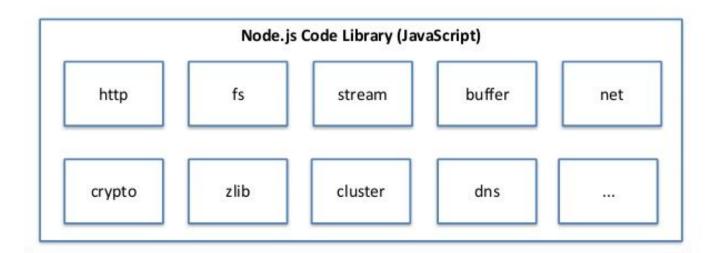




# Node.js



- Criado em 2009
- Roda em cima da V8
- Usa a biblioteca libuv para processamento assíncrono
- V8 é a VM JavaScript do Google Chrome
- Designed to build scalable network applications
- Asynchronous event driven JavaScript runtime
- Não existe DOM



#### Node.js Bindings (C++)



```
const http = require('http');
const hostname = '127.0.0.1';
const port = 3000;
const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World\n');
});
server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

### **GETTING STARTED**

- Instalar o Node.js
- Abra o editor de texto
- Rode seu código com

node hello-world.js

## **GETTING STARTED - REPL**

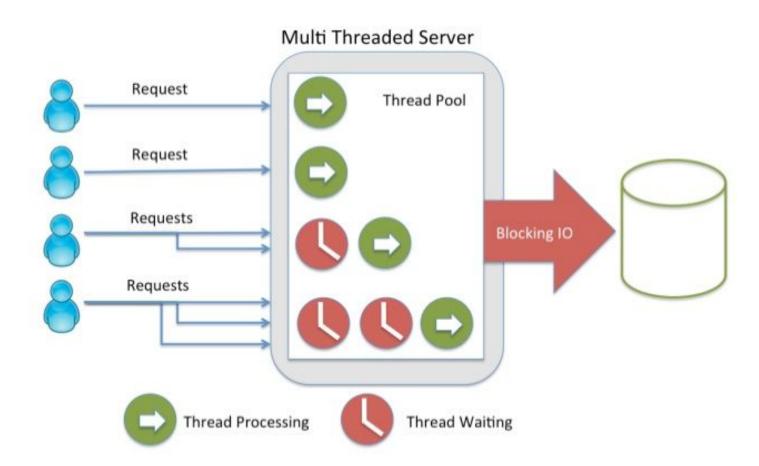
```
$ node
> console.log('Hello, World!');
Hello, World!
undefined
>
```

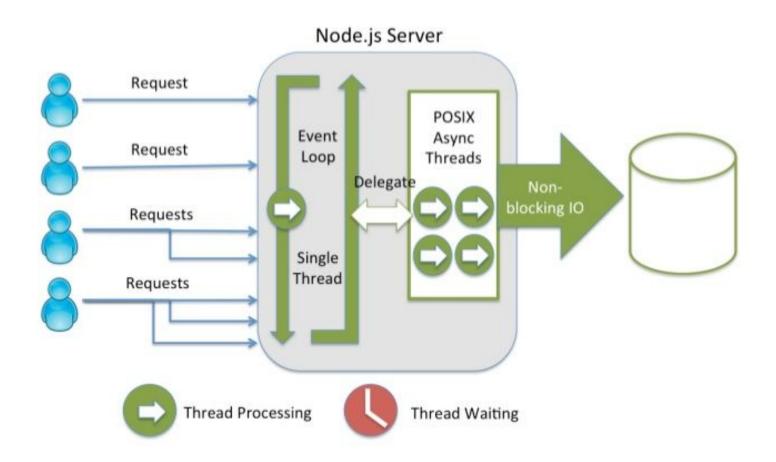
# **GETTING STARTED**

```
~/Code/Catolica/nodejs/hello-world.js
   hello-world.is
 1 setTimeout(function() {
 2 console.log('World!');
 3 }, 3000);
    console.log('Hello');
COMMAND MODE, W: 0 E: 0, Line 1, Column 1
```

# NON-BLOCKING I/O

```
~/Code/Catolica/nodejs/block-non-block.js
                  block-non-block.js
   // I/O TRADICIONAL
   function doSomething(result) {
   // NON-BLOCKING I/O
   var result = db.query('SELECT x FROM table');
   doSomething(result);
 9
   var result = db.query('SELECT x FROM table', function(result) {
   doSomething(result);
12 });
  , COMMAND MODE, W: 0 E: 0, Line 4, Column 1
```





#### Requisição Síncrona

- Requisição I
- Requisição II
- Requisição III

### Requisição Assíncrona



### **EXEMPLO**

```
. .
                                  ~/Code/Catolica/nodejs/block-non-block-example.js
    block-non-block-example.js
  1 var frase;
 3 v carregaFrase = function (callback) {
 4    setTimeout(function() {
  5    frase = "Minha frase obstrutiva";
 6 callback();
     }, 3000)
     imprimeFrase = function () {
     console.log(frase);
12
13
     carregaFrase(imprimeFrase);
     console.log("01á");
O , COMMAND MODE, W: 0 E: 0, Line 18, Column 1
                                                                                       Spaces: 2
                                                                                              JavaScript (Babel)
```



npm Enterprise

features

pricing do

cumentatio

support



find packages

Q

sign up or log in

# Build amazing things

npm is the package manager for JavaScript and the world's largest software registry. Discover packages of reusable code — and assemble them in powerful new ways.

Sign up for npm









npm Orgs is powerful collaboration — for free

## **ECOSSISTEMA**

- Maior repositório de pacotes do mundo
- Npm (Node Package Manager)

npm install-g nyancat

# DOCUMENTAÇÃO

#### Node.js

About these Docs

Usage & Example

**Assertion Testing** 

Async Hooks

Buffer

C++ Addons

C/C++ Addons - N-API

**Child Processes** 

Cluster

**Command Line Options** 

Console

Crypto

Debugger

#### Node.js v9.8.0 Documentation

#### **Table of Contents**

- About these Docs
- Usage & Example
- Assertion Testing
- Async Hooks
- Buffer
- C++ Addons
- C/C++ Addons N-API
- Child Processes
- Cluster
- · Command Line Options
- Console
- Crvpto

https://nodejs.org/api