

# Curso Desenvolvimento Web



# Desenvolvimento WEB

*Introdução ao desenvolvimento de  
aplicações WEB com NodeJS*

1.Introdução

2.JavaScript

3.NodeJS





UFMG STUDENT CHAPTER

# 1.

## Introdução

*Por que NodeJS?*



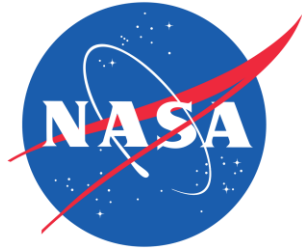
# Por que NodeJS?

*NodeJS é um interpretador de código JavaScript que roda da perspectiva do servidor.*

Criado em 2009, o NodeJS trouxe o JavaScript tradicional implementado no lado do cliente para os servidores, de forma que essa linguagem pôde ser utilizada para arquitetar o *backend* de aplicações de alta escalabilidade.



Quem utiliza



UBER

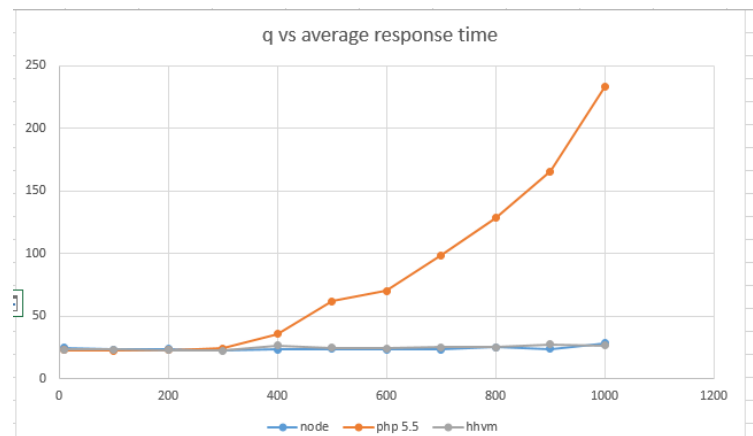
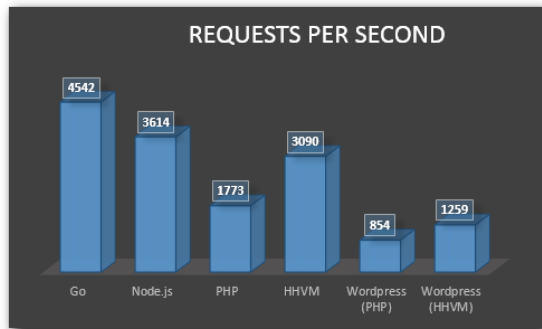
NETFLIX

GROUPON<sup>®</sup>

The  
New York  
Times



# PHP vs NodeJS



650k +  
packages



183k  
packages



“

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

Jeff Atwood, 2007

Co-founder of StackOverflow



UFMG STUDENT CHAPTER

2.

JavaScript

*Write Everywhere*

JS



# Declaração de Variáveis

O JavaScript permite dois tipos de comando para criação de variáveis: `const` e `let`.

Como a abreviação sugere, o comando `const` é utilizado para criação de constantes cujo valor não pode ser alterado após sua instanciação.

Já o comando `let` é utilizado para criação de variáveis restritas ao escopo de onde são declaradas.

```
const x = 6; // Constant
let y = 8; // Outer Variable

x = 8; // Not Allowed!
y = 1; // Allowed!

function sum(a, b) {
    let y = a + b; // Local Variable
    console.log(y); // Prints a+b
}

sum(3,5); // Prints 8
console.log(y); // Prints 1
```

# Tipos de Dados

O JavaScript é uma linguagem implicitamente tipada, ou seja, todas as variáveis são declaradas utilizando o mesmo comando de linguagem, porém, o seu tipo é definido internamente a partir do valor que é associado a essa.

```
let evening = true; // Boolean
let x = 6; // Integer
let name = "Tulio"; // String
let dogs = ["Pitbull", "Golden Retriever"]; // Array

x = "New Type"; // Automatic coercion to another type

console.log("Elementary, " + "my dear Watson!"); // String concatenation
console.log("Number" + 16); // Prints Number16
console.log(dogs.join(",")); // Prints Pitbull,Golden Retriever
```



# Objetos

Objetos em javascript são declarados seguindo o **JavaScript Object Notation (JSON)**, comumente utilizado por **Application Programming Interfaces (API's)** para retornar os dados, uma vez que pode ser interpretado por praticamente todas as linguagens de programação.

```
let person = { // Object
  name: "Henrique Milagres",
  age: 31,
  sex: "Male",
  nicknames: ["Milagrinho", "Milagroso", "Miracle"],
  hasDriverLicense: true
}

console.log(person.name); // Prints Henrique Milagres
console.log(person.height); // Prints Undefined
person.height = "1.80m";
console.log(person.height); // Prints 1.80m
```



# Funções

É possível armazenar em uma variável um trecho de código de forma a permitir a sua execução em diversas seções do seu algoritmo por meio da invocação de seu rótulo, facilitando a reutilização e melhorando a legibilidade do código. Essa variável é chamada de função.

```
let sum = (a, b) => { // Named function sum
  return a+b;
}

let alarmFunction = (func) => { // Functions passed as parameters
  setTimeout(func, 1000); // Execute the function after 1000ms
}

console.log(sum(1,3)); // Prints 4
z = alarmFunction(() => { // Unnamed function
  return sum(1,3);
});
console.log(z); // Prints 4 after 1000ms
```



# Controle de Fluxo

```
const x = Math.random(); // Random number between 0 and 1
if (x >= 0.7) {
  console.log("Top");
}
else if(x >= 0.3) {
  console.log("Mid");
}
else {
  console.log("Bottom");
}

const names = ["Milagres", "Tulio", "Breno", "João", "Thaís"];

names.forEach((name) => { // Iterate over names
  console.log(name);
});

// OR

for (let i = 0; i < names.length; i++) { // Iterate over names using the index increment method
  console.log(names[i]);
}
```

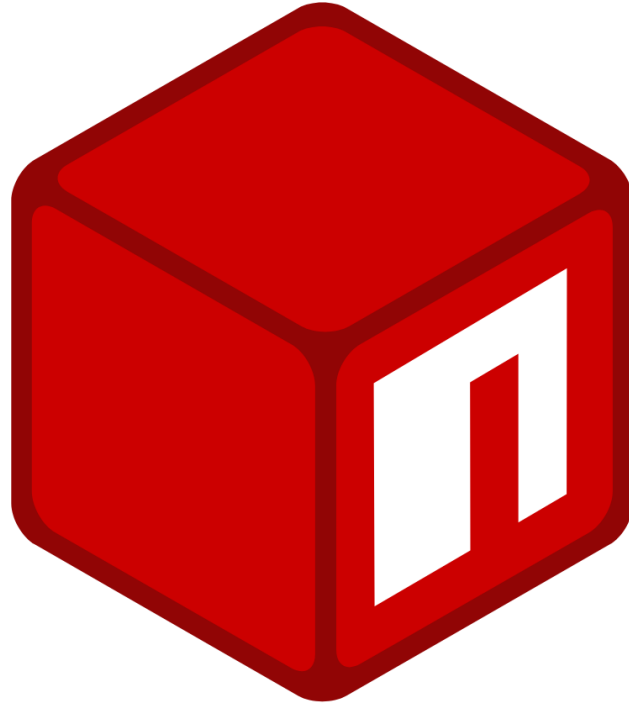




A Oitava Maravilha do Mundo.

# NPM

O **Node Package Manager (NPM)** é o gerenciador de pacotes para o JavaScript e o maior registro de software do mundo, contendo mais de 650 mil repositórios públicos e possibilitando a modularização de código para facilitar a construção e integração de componentes de uma aplicação JavaScript.



# Principais Comandos

Os principais comandos do NPM estão listados abaixo. Os comandos de inicialização e instalação de novos módulos serão amplamente utilizados durante o desenvolvimento de nossa aplicação:

```
npm init

npm install package-name

npm install package2 --save

npm install global-package -g

npm uninstall thrash-package --save

npm start
```



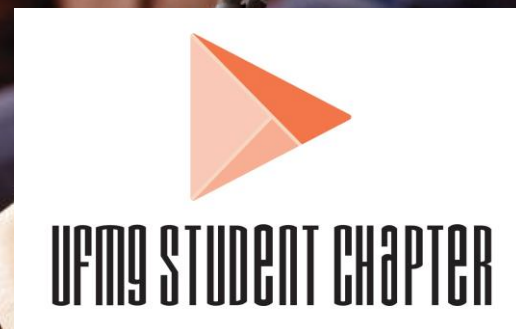




Obrigado a todos por  
apoiar a nossa causa!

Para maiores informações:

- [www.ufmgstudentchapter.org](http://www.ufmgstudentchapter.org)
- [ufmgstudentchapter@gmail.com](mailto:ufmgstudentchapter@gmail.com)
- [tulio.ao@gmail.com](mailto:tulio.ao@gmail.com)
- [milagres.henrique@gmail.com](mailto:milagres.henrique@gmail.com)
- [brenosilva1@gmail.com](mailto:brenosilva1@gmail.com)



Obrigado!

*Um agradecimento especial de todo o grupo UFMG Student Chapter por acreditar e apoiar a nossa causa!*

- Para outros cursos visite: [www.ufmgstudentchapter.org](http://www.ufmgstudentchapter.org)
- Em caso de dúvidas: [ufmgstudentchapter@gmail.com](mailto:ufmgstudentchapter@gmail.com)



UFMG STUDENT CHAPTER