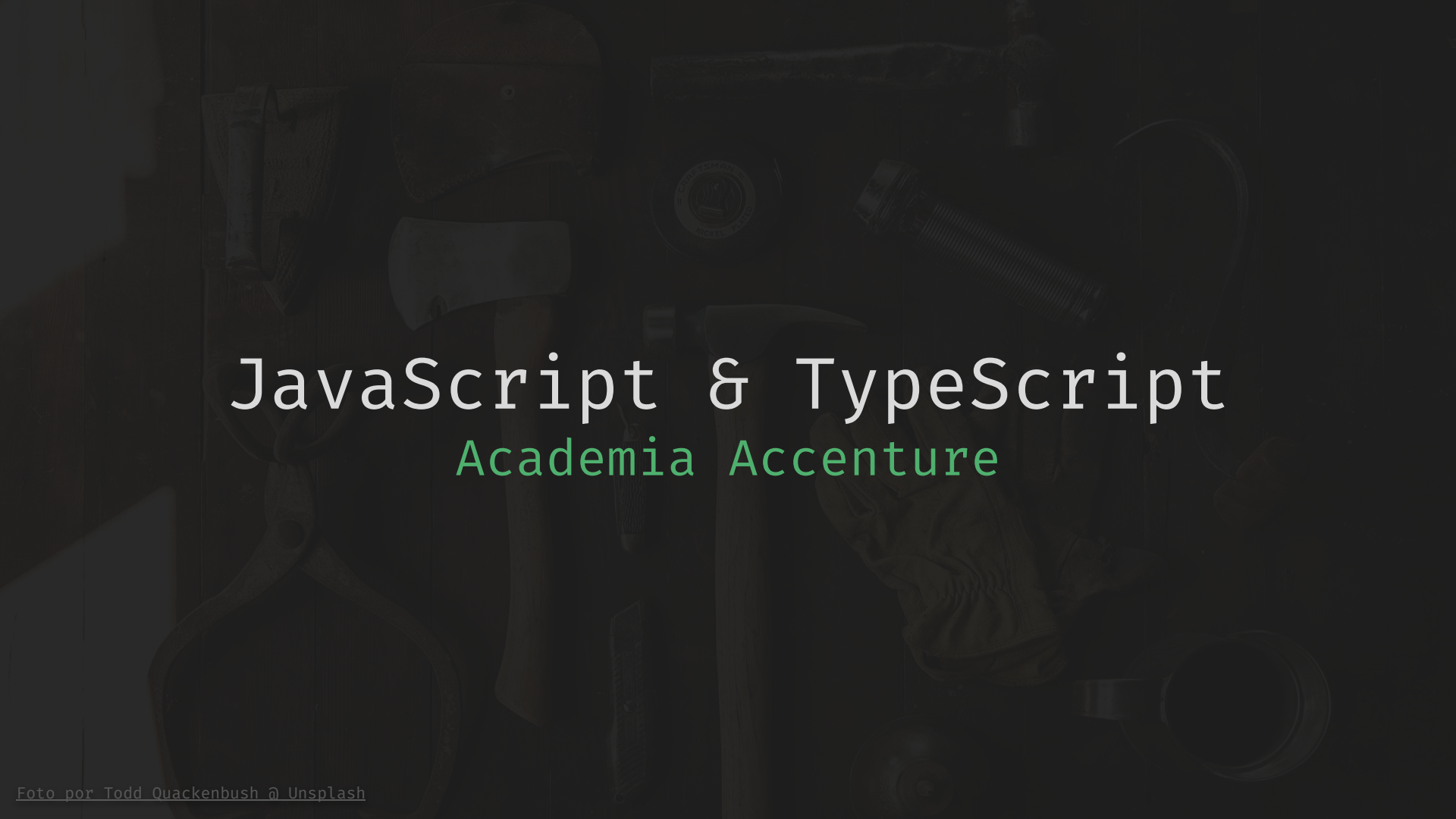


JavaScript & TypeScript

Academia Accenture

Foto por Todd Quackenbush @ Unsplash

Oi :)

@gabrieluizramos

gabrieluizramos.com.br

Gabriel Ramos

Desenvolvedor Front-end @ Loft | Mentor @ Laboratória | Instrutor @ Caelum

Dedicando os últimos anos a trabalhar com React e Node, recentemente estudando mais afundo sobre testes em JavaScript e outros assuntos que envolvem a linguagem.
Fotógrafo por hobby e "o doido dos gatos" (só 3).

Agenda

Dia 1 (8h)

- Características da Linguagem
- Como programar em JavaScript?
- Variáveis
- Strings e seus métodos
- Números e operadores
- Booleans e seus métodos
- Arrays e seus métodos
- Objetos e seus métodos
- Condicionais e Operadores
- Laços de repetição
- Funções
  - Declaração
  - Expressão (ou Literal)
  - Arrow
- Escopo
- Funções construtoras e classes
- Console
- Boas práticas (styleguide/DRY)

Dia 2 (4h)

- Formulários
- Eventos
- Template string
- Operador ternário e &&, ||
- Spread/rest
- Desestruturação
- Callbacks
- Métodos HTTP (GET/POST/PUT/DELETE/PATCH)
- APIs
  - XHR com Open/Send
  - XHR com Fetch

Dia 3 (4h)

- Promises
  - Thenable
  - Async/Await
- Debugging
  - Cliente
  - Servidor
- Progressive Web-Apps (PWA)
- Single Page Applications (SPA)

Agenda

Dia 4 (4h)

- NPM
- Yarn
- Babel
- Webpack
- Web Components
- Funções como Componentes

Dia 5 (4h)

- Introdução ao TypeScript
- Instalação e utilização
- Tipos básicos
  - String
  - Number
  - Boolean
  - Array
  - Tuple
  - Enum
  - Unknown
  - Any
  - Void
  - Null e Undefined
  - Never
  - Object

Dia 6 (4h)

- Union Type
- Intersection Type
- Type Alias
- Interfaces
- Valores opcionais

Agenda

Dia 1 (8h)

- Vantagens e características
- Variáveis
- Strings e seus métodos
- Números e operadores
- Booleans e seus métodos
- Arrays e seus métodos
- Objetos e seus métodos
- Condicionais
- Console
- Laços de repetição
- Funções
- Escopo
- Funções construtoras e classes

Dia 2 (4h)

- Boas práticas (styleguide/DRY)
- Formulários
- Eventos
- Template string
- Operador ternário e &&, ||
- Spread/rest
- Desestruturação
- Callbacks
- Métodos HTTP (GET/POST/PUT/DELETE/PATCH)
- APIs
  - XHR com Open/Send
  - XHR com Fetch

Dia 3 (4h)

- Promises
  - Thenable
  - Async/Await
- Debugging
  - Cliente
  - Servidor
- Progressive Web-Apps (PWA)
- Single Page Applications (SPA)

Teremos vários encontros e conversaremos sobre um **monte** de coisas.
Então é **muito** importante que ninguém saia com dúvidas.
Interrompam a qualquer momento e vamos construindo nosso **bate-papo juntos**, combinado? :)

Bora lá!

<dia 1>

Características da Linguagem

- JavaScript? O que é um Script?
- Diferente de linguagens compiladas, é muito fácil de começar a trabalhar com **JS**
- Você não vai precisar de uma nave alienígena para desenvolver e vai ver que é tudo muito prático

Como programar em JavaScript?

Para essa receita você vai precisar de:

- 1 editor de texto qualquer (**Word não vale, hein!**)
- 1 navegador qualquer

Eu, particularmente, uso VSCode

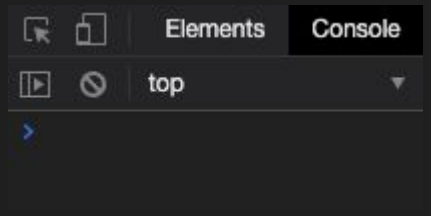
Como programar em JavaScript?

Em uma página, no navegador

```
<script>
    // seu código vai aqui
</script>
<!-- ou -->
<script src="caminho/do/arquivo.js"></script>
```

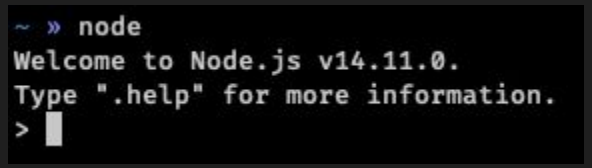
Como programar em JavaScript?

Usando o Console **DevTools**



Como programar em JavaScript?

Usando **NodeJS** (vamos deixar opção para depois)



Variáveis

Variáveis são estruturas capazes de armazenar algum valor.

É comum pensarmos em variáveis como se fossem pequenas **caixinhas**, onde cada uma pode armazenar algum dado.

Existem basicamente 3 formas de se criar uma variável na linguagem, e elas são através de:
var, **let** e **const**

Variáveis (var)

Variáveis declaradas com **var** são do tipo mais primitivo da linguagem. Podem ser reatribuídas e também afetam o **escopo global** (vamos falar sobre isso já já!) de sua aplicação.

Variáveis (let)

Algumas pessoas pensam como se fosse "a novar **var**". Foi introduzida mais recentemente na linguagem e, embora possa ser reatribuída, possui um escopo de bloco.

O que podemos armazenar em variáveis?

- String (textos);
- Number (números)
- Boolean (verdadeiro/falso);
- Array (sequência de valores);
- Objeto (um "conjunto" de várias variáveis).

Strings 'hello'

Estrutura utilizada para representar caracteres de texto. Alguns métodos interessantes.

- Criação e utilização
- Concatenar strings com +
- Métodos
  - String.replace e String.replaceAll
  - String.split
  - String.trim
  - String.startsWith e Strings.endsWith

Numbers 1234

Estrutura utilizada para representar caracteres numéricos. Alguns métodos interessantes.

- Criação e utilização
- Number.toFixed
- Conversão de String para Number com:
  - parseFloat
  - parseInt
  - Number
- Ponto e vírgula nos números

Booleans true false

Estrutura utilizada para representar valores de verdadeiro/falso (true/false). Muito utilizado no contexto de operações e condicionais (que veremos logo).

Arrays []

Estrutura sequencial, atuando como uma lista de vários valores.

- Criação e utilização
- Acessando valores
- Quantidade de itens
- Métodos:
  - Array.push e Array.pop
  - Array.slice e Array.splice
  - Array.join
  - Array.forEach
  - Array.reverse
  - Array.map
  - Array.filter
  - Array.reduce
  - Array.find
  - Array.concat

Objetos {}

Conjunto que pode conter as diversas estruturas que já vimos.

- Criação e utilização
- Dot notation e bracket notation
- Métodos:
  - Object.keys
  - Object.entries

Condicionais e Operadores

Condicionais são estruturas utilizadas para dividir a execução de nosso código em várias partes. Podem ser estruturadas utilizando valores **booleanos**.

- if/else
- switch/case

Condicionais e Operadores

Operadores são caracteres utilizados para realizar diversas operações com variáveis.

- Comparação
  - Igualdade (== e ===)
  - Diferente (!= e !==)
  - Maior e Maior igual (> e >=)
  - Menor e Menor igual (< e <=)
- Lógicos
  - Ou (||)
  - E (&&)
  - Negação (!)

Para ler mais

Laços de Repetição (ou loops)

Estruturas que permitem que façamos repetições no nosso código.

- For
- While
- Do/While

Funções

São pequenos trechos de código que podemos criar e reutilizar da forma como melhor entendermos.

- Declaração
- Expressão (ou literal)
- Arrow
- Parâmetros

Escopo

É um contexto onde determinado trecho de código é executado.

- Global
- Local e blocos
- Funções (e como variáveis com **var/let/const** se comportam)
- Funções dentro de funções

Funções Construtoras e Classes

Formas diferentes de criar objetos em JavaScript.

- Utilização da palavra **new**
- Classes são funções, mas com uma notação diferente
- Métodos (funções) em objetos

Console

Métodos interessantes do console para nossa utilização

- log
- error
- table
- time
- timeLog
- timeEnd

Boas Práticas (styleguide/DRY)

Existem vários padrões e guias de estilo para código no mercado (Google). O importante é manter a consistência dentro de um projeto.

DRY significa *Don't Repeat Yourself* e é o esforço de tentar organizar abstrações coerentes, evitando repetição de código.

</dia 1>

<dia 2>

</dia 2>

<dia 3>

</dia 3>

<dia 4>

</dia 4>

<dia 5>

</dia 5>

<dia 6>

</dia 6>