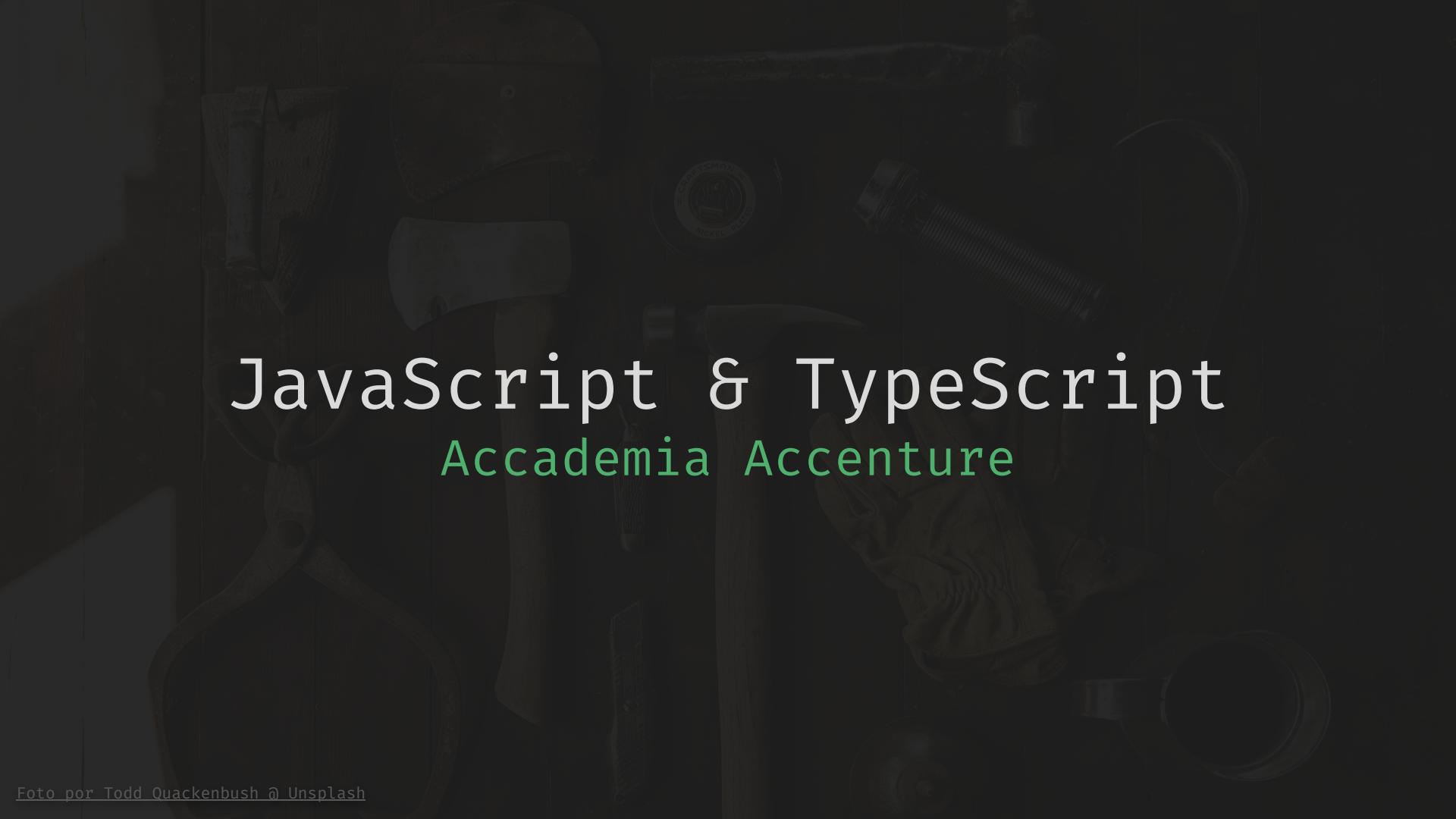


JavaScript & TypeScript

Accademia 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 (4h)

- 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

Dia 2 (8h)

- Condicionais e Operadores
- Laços de repetição
- Funções
  - Declaração
  - Expressão (ou Literal)
  - Arrow
- Escopo
- Funções construtoras e classes
- Template string
- Boas práticas (styleguide/DRY)
- Operador ternário e &&, ||

Dia 3 (4h)

- Formulários
- Eventos
- Desestruturação
- Spread/rest
- Callbacks

Agenda

Dia 4 (4h)

- Métodos HTTP
- APIs
  - XHR com Open/Send
  - XHR com Fetch
- Promises
  - Thenable
  - Async/Await
- Progressive Web-Apps (PWA)
- Single Page Applications (SPA)
- Web Components
- Funções como Componentes
- Node e NPM
- Yarn
- Babel
- Webpack
- Console
- Debugging
  - Cliente
  - Servidor

Dia 5 (4h)

- Introdução ao TypeScript
- Instalação e utilização
- Motivos pelos quais utilizar ou não
- Tipos básicos
  - String
  - Number
  - Boolean
  - Array
  - Tuple
  - Null e Undefined
  - Any
  - Unknown
  - Void
  - Never
  - Object
  - Function

Dia 6 (4h)

- Enum
- Union Type
- Intersection Type
- Type Alias
- Interfaces e um pouco de OO
- Valores opcionais

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

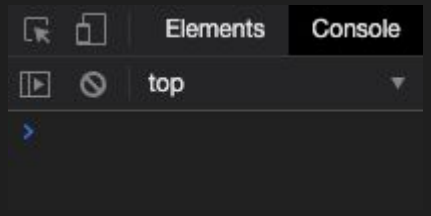
Como programar em JavaScript?

Em uma página, no navegador

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

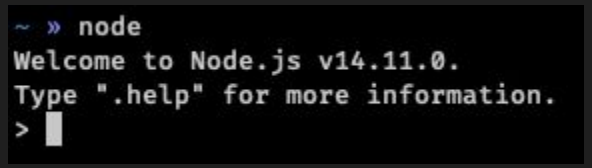
Como programar em JavaScript?

Usando o Console **DevTools**



Como programar em JavaScript?

Usando **NodeJS** e seu **REPL** (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.

Variáveis const

Funciona exatamente igual às variáveis declaradas com **let**, entretanto, não podem ser reatribuídas.
Isso não quer dizer que são, de fato, **constantes**, pois é possível mudar seu valor mesmo sem utilizar reatribuição.

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.

- 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.

- 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
- Estrutura chave: valor
- Dot notation e bracket notation
- Métodos:
  - Object.keys
  - Object.entries

</dia 1>

<dia 2>

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 e valores default

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

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.

Template strings `Olá ${nome}`

Chega de concatenar variáveis com strings.

Ternário, && e ||

Como escrever um "if reduzido" e como os operadores && e || retornam outros dados.

- Estrutura do ternário e seu retorno
- Retorno do operador &&
- Retorno do operador ||

Para ler mais

</dia 2>

<dia 3>

Formulários

Com certeza você já preencheu um! São as tags que nos permitem interagir com dados do usuário.

- Tag form
- Tag de inputs
- Tag select
- Tag textarea
- Placeholders

Eventos

São formas de reagir à algum acontecimento dentro de uma página.

- Inline
- Via atributo
- Via addEventListener
- Objeto event
- event.preventDefault

Desestruturação

Formas mais simples de criar variáveis direto ao acessar valores de objetos e arrays.

- Utilização
  - Array
  - Objeto

Spread/Rest ...

Uma mesma sequência de caracteres interpretando dois papéis parecidos (à primeira vista), mas diferentes na prática

- Rest utilizado para receber qualquer sequência parâmetros
- Spread utilizado para atribuir valores existentes à uma nova variável

</dia 3>

<dia 4>

Callbacks

São funções que possuem sua execução adiadas à um momento futuro.

- Por quê?
- Um pouco sobre código assíncrono

Métodos HTTP

Um resumo sobre o protocolo e alguns métodos que podemos utilizar

- GET
- POST
- PUT
- DELETE
- PATCH

Comunicações com AJAX (XHR)

Vamos entender mais à fundo como essas comunicações e requisições à APIS funcionam.

- O que é XHR?
- Como funciona o objeto XmlHttpRequest?
- Como realizar uma requisição com XmlHttpRequest?

Comunicações com AJAX (Fetch)

Vamos entender mais à fundo como essas comunicações e requisições à APIS funcionam.

- O que é o fetch?
- Introdução à Promises
- Como realizar uma requisição com fetch?

Promises

O que são, do que se alimentam, como são executadas?

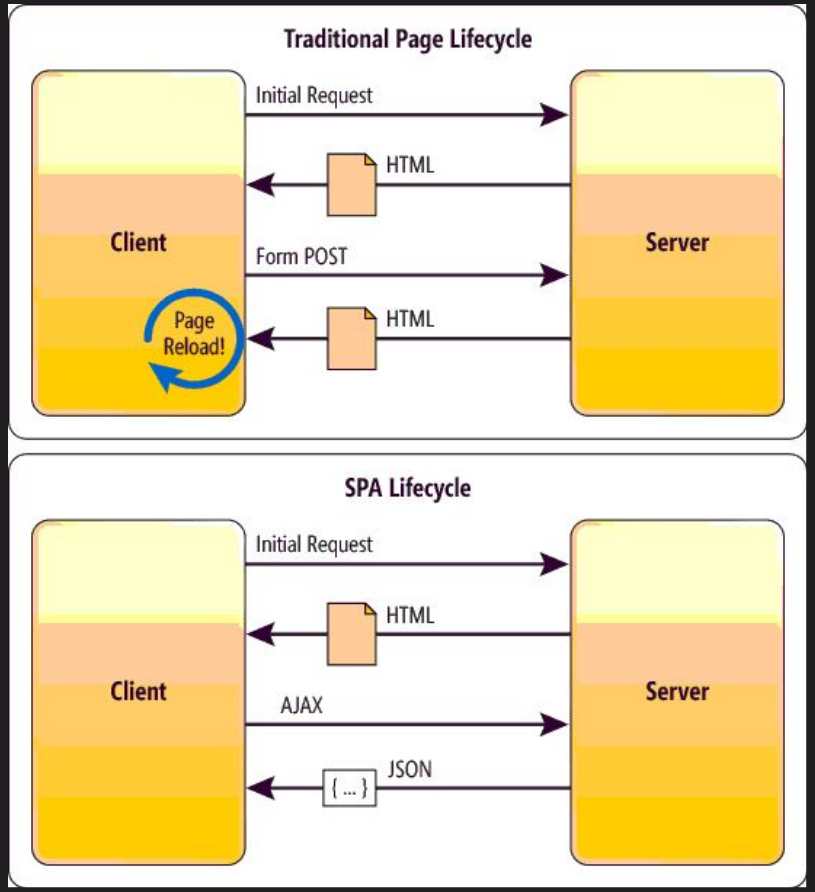
- Criação de uma Promise
- Estados de uma Promise
  - Pending
  - Fulfilled
  - Rejected
- Thenable (.then) ou Async/Await?
- Semelhança com Generators

Progressive Web-Apps (PWA)

Uma série de decisões e abordagens tecnológicas que podem adicionar alguns temperos à sua aplicação web.

- Adicionar à sua tela inicial
- Push notification
- Service Worker e trabalhar em modo offline
- Várias outras coisas, desde sua estrutura até sua experiência

Single Page Application (SPA)



Web Components

O que são e como funciona esse conjunto de tecnologias.

- Por quê desenvolver pensando em components?
- Como componentizar?
- Eu preciso de algum framework para aplicar boas práticas de componentização?

Para ler mais

Funções como Componentes

Vamos brincar um pouco com React?

- Entendendo por quê bibliotecas e frameworks foram adotadas
- Pontos positivos
- Pontos negativos
- Criando um componente de contador

Node e NPM

Entendendo um pouco do ecossistema JavaScript atualmente e como as coisas funcionam entre projetos

- O que são e como instalá-los
- Como criar um projeto
- O que é um registry
- Como funciona o package.json
- NPX
- Um passeio por módulos em JavaScript

Para ler mais
Para ler mais

Yarn

Outro gerenciador de pacotes? Pra quê?

- No que se assemelha e no que se difere do NPM
- Histórico e cache
- Workspaces, Monorepo e Multirepo
- Qual eu devo usar?

Babel

O que é um transpilador (ou compilador?) e como ele funciona

- Como ele funciona
- Por quê utilizar Babel?
- Experimentando seu playground
- Configurando em um projeto via CLI
- Configurando em um protótipo no navegador (**não** recomendado para projetos que irão para produção)

Webpack

O que é um empacotador de módulos e como ele funciona

- Como ele funciona
- Por quê utilizar Webpack?
- Configurando um projeto com Webpack
- Como funcionam os loaders

Console

Métodos interessantes do console para nossa utilização

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

Debugging

Algumas outras formas de depurar as aplicações cliente/servidor

- Um passeio pelo DevTools
- Utilizando debugger; e outras coisas mais
  - No navegador
  - No Node
- Source-maps

</dia 4>

<dia 5>

TypeScript

Uma visão geral da ferramenta e como utilizá-la

- Introdução
- Brincando em seu playground
- Instalação e utilização
- Tipos básicos
- Extra: tipagem usando JSDoc

</dia 5>

<dia 6>

TypeScript

Alguns tipos um pouco mais complexos e outras funcionalidades do superset.

- Trabalhando com Enum
- Union type
- Intersection type
- Type alias
- Interfaces, semelhanças com Type Alias e um pouco de OO
- Valores opcionais

</dia 6>

Obrigado!

@gabrieluizramos

gabrieluizramos.com.br