A large yellow square containing the letters 'JS' in a bold, dark grey sans-serif font. The square is positioned in the center of the slide, overlapping a white background on the left and a dark grey background on the right.

JS

@vtaquette
@rzuquim





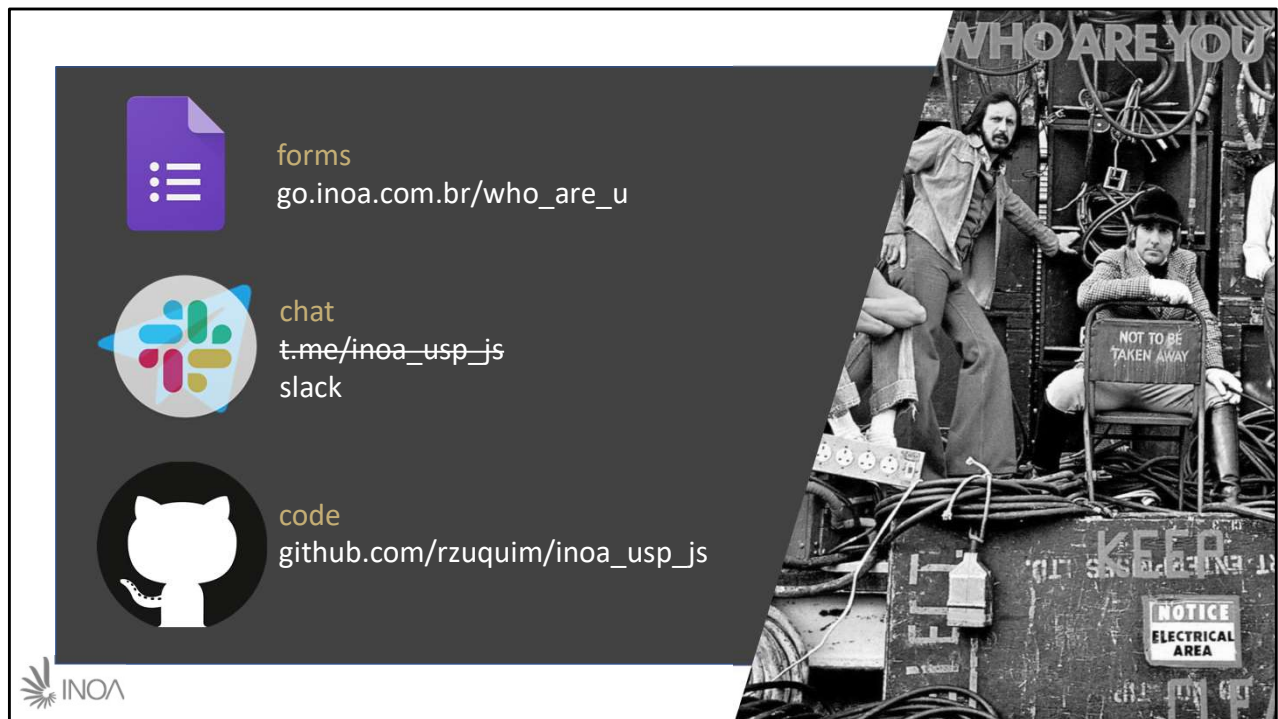
@vtaquette

eng. UFRJ
mestre UFRJ
sócia/analista INOA





sócio/dev INOA
eng. UFRJ


@rzuquim




WHO ARE YOU

 **forms**
go.inoa.com.br/who_are_u

 **chat**
t.me/inoa_esp_js
slack

 **code**
github.com/rzuquim/inoa_esp_js

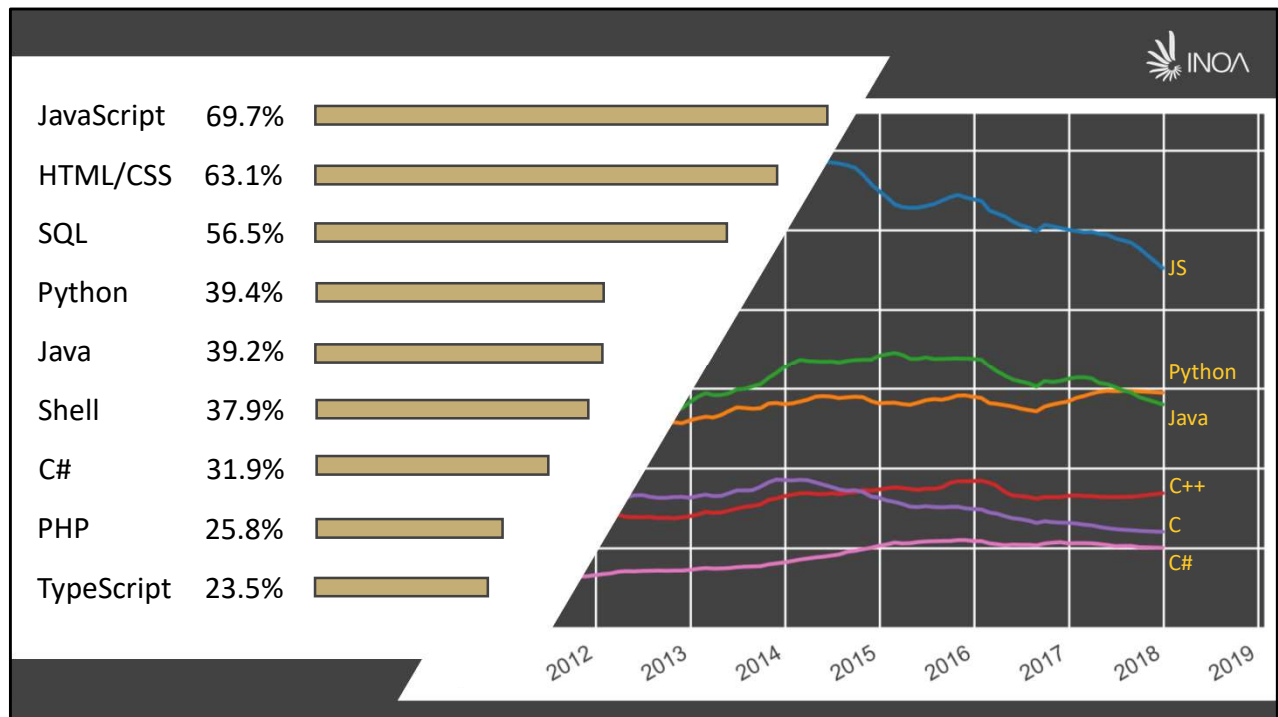
 INOVA

NOT TO BE TAKEN AWAY

KEEP

NOTICE ELECTRICAL AREA

Canais de comunicação pelos quais a gente vai trocar ideias, o material das aulas, etc.



Porque aprender js?

Porque você não tem opção...

- js é a linguagem mais popular e pervasiva do mundo
- a popularidade é um ciclo virtuoso
- o ferramental passa a ser muito bom
- o custo de contratação / projeto diminui
- a produtividade aumenta
- js é uma linguagem interessante, poderosa e constantemente em evolução

Referências:

https://insights.stackoverflow.com/survey/2019#technology_-_programming-scripting-and-markup-languages

<https://www.benfrederickson.com/ranking-programming-languages-by-github-users/>

<https://insights.stackoverflow.com/survey/2020>



js everywhere

Nosso curso é sobre js pra web, então a gente vai focar em desenvolvimento web. Na web foi onde tudo começou e onde a linguagem é mais usada, mas por causa da popularidade js pode ser usado em servidores, desktops, apps de celulares e jogos. **Já existe uma lib pra qualquer coisa que você quiser**

Server side: Node / Deno

Games: Unity

Desktop: Electron

Mobile: Cordova / React Native

Referências:

<https://www.amazon.com.br/dp/B084KMH9JP/>

<https://nodejs.org/>

<https://unity.com/>

<https://deno.land/>

<https://cordova.apache.org/>

<https://reactnative.dev/>

<https://www.electronjs.org/>

hello_world.js

```
1 <script type="text/javascript">
2     alert( 'Hello world!' );
3 </script>
```



Mas o curso é de javascript para WEB

Então o navegador é o nosso principal alvo na hora de escrever javascript. E todo o navegador tem um interpretador js em que a gente pode brincar. No chrome você tem acesso a esse interpretador clicando em F12.

Exemplos trabalhados:

- alert
- confirm
- aritimética
- aritimética de ponto flutuante

Referência:

https://wiki.theory.org/YourLanguageSucks#JavaScript_sucks_because



Conforme a gente for aprendendo JS vamos ter várias perguntas a respeito da linguagem. E sempre é interessante catalogar nossas dúvidas.

Por isso, ao longo desse curso, vamos desenvolver uma aplicação web chamada **faq.js**.

Onde a gente vai catalogar o próprio curso.

No final, o objetivo é ter uma aplicação que armazene essas dúvidas e fique disponível para consulta quando quiserem.

<code />

```
1 import { git } from './trilha_devs/bloco_1.js';  
2 import { html } from './trilha_devs/bloco_1.js';  
3 import { css } from './trilha_devs/bloco_2.js';
```



Vamos criar a base do nosso projeto

- projeto no github
- html
- css

Ferramentas

- git / github - <https://github.com/>
- gitignore - <https://github.com/github/gitignore>
- <https://tldrlegal.com/>
- vscode - <https://code.visualstudio.com/>
- bulma.css - <https://bulma.io/>
- fontawesome 4 - <https://fontawesome.com/v4.7.0/icons/>

Primeira Guerra dos Browser (1995 – 2001)

Tim Berners-Lee's - HTML / HTTP / WWW
Marc Andreessen - Mosaic / Netscape
Microsoft - Sistema de TV Interativa



Mosaic primeiro navegador com interface gráfica pra Web.

Mosaic vira Netscape por questões legais.

Em 2 anos do lançamento, o Netscape 1.0 dominava o mercado.

Microsoft pensava em TV interativa (via cabo) quando foi surpreendida pelo sucesso do Netscape.

Vendo o sucesso da Web, a MS tenta comprar o Netscape sem sucesso, e decide partir para o ataque.

Começa guerra pra saber quem vai ser a porta de entrada da internet.

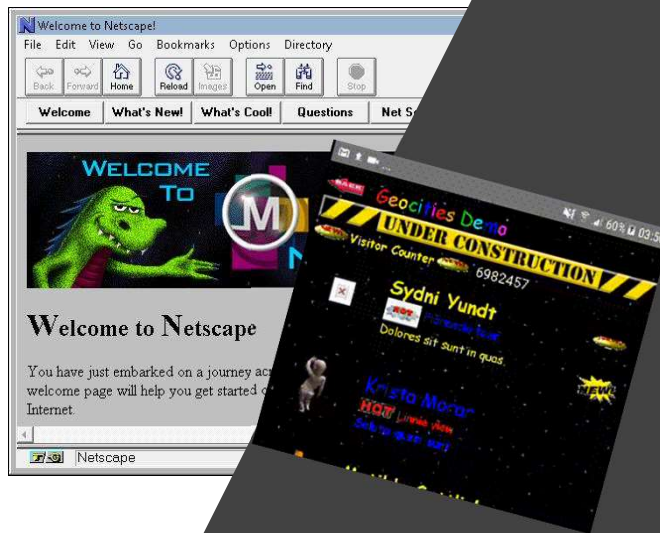
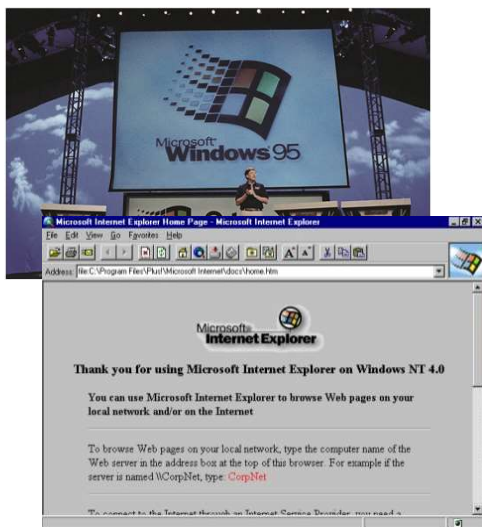
Referências:

https://en.wikipedia.org/wiki/Browser_wars

<https://medium.com/@traciemasek/a-brief-history-of-the-original-browsers-and-the-first-browser-war-7823fdf756fe>

<http://www.nethistory.info/History%20of%20the%20Internet/browserwars.html>

Nascimento do Javascript



O business plan da Netscape focava em fornecer um navegador gratuito para uso pessoal e acadêmico e serviços pagos para empresa.

A estratégia da MS foi usar seu poder de mercado para asfixiar o concorrente. Junto com o Win95 ela liberou o Internet Explorer “gratuitamente”. Junto com o Windows Server entregou o IIS.

A contra-estratégia de Andreessen é entregar mais que um browser. A cada 3 semanas uma versão nova.

Barra de busca, transferência de arquivos, alertas de e-mail... e bugs, muitos bugs (que eram corrigidos rapidamente).

Netscape é um dos primeiros softwares com modelo de releases ágeis (a cada 3 meses - comparado com janelas de 3 anos da MS, por exemplo)

Mas contra o IE eles precisariam de algo revolucionário.

O Netscape 2.0 teria interatividade através de uma linguagem de programação.

O time to market seria essencial, o Netscape 2.0 tinha que sair antes do Windows 95.

Java era o candidato natural. Em 1995 Java era o futuro: OOP, multiplataforma, com rios de dinheiro que a Sun jogou em marketing.

Mas o Netscape sacou que Java era complicado demais pra Internet.

Nascimento do Javascript

Dialeto de Scheme / LISP
Inspirado em C, Java e Modula 2
Herança por prototipagem
HyperCard -> DOM

ESCRITO EM 10 DIAS



A missão era escrever uma linguagem:

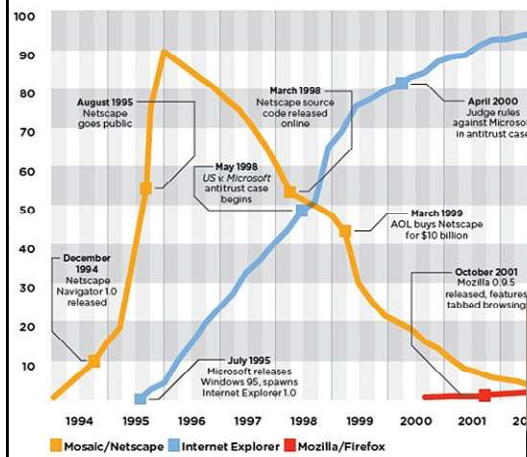
- Simples (curva de aprendizado simples) – Script / Interpretada
- Foco em hobbistas, não em profissionais
- Familiaridade (sintaxe próxima de C)
- Mas que preservasse alguns conceitos de OO para trazer o apelo do Java

A linguagem foi entregue por Brendan Eich e adotada por um consórcio de 28 empresas entre elas a Netscape, AOL, AT&T e Sun.

Referência

- <https://www.redhat.com/en/command-line-heroes/season-3/creating-javascript>

Primeira Guerra dos Browser (1995 – 2001)



Falta de padrões abertos
Batalhas judiciais

1997 – ECMAScript (ES)

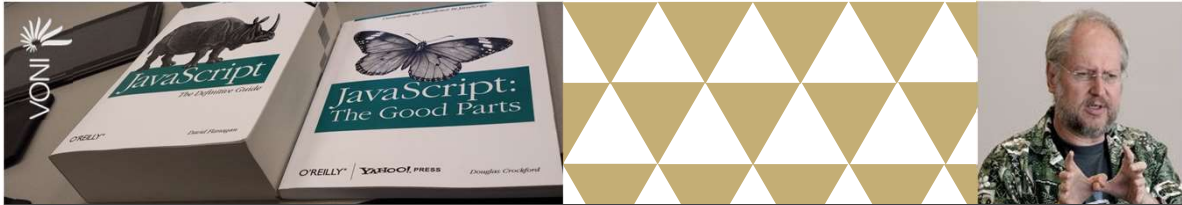


Época de falta de padrões

- Applets
- JavaScript
- JScript / JCSS / ActiveX
- ActionScript / Flash

Referências:

- <https://sudonull.com/post/80379-Never-give-up-how-Netscape-waged-an-unequal-battle-with-Internet-Explorer>
- <https://www.wired.com/1995/12/andreessen/>



- Importando js em HTML
- Aritimética
- Manipulação de strings
- Variáveis
- Tipagem
- Comentários

Agenda

Javascript tem tipagem fraca, conversões entre tipos podem gerar comportamentos inesperados.

Referências:

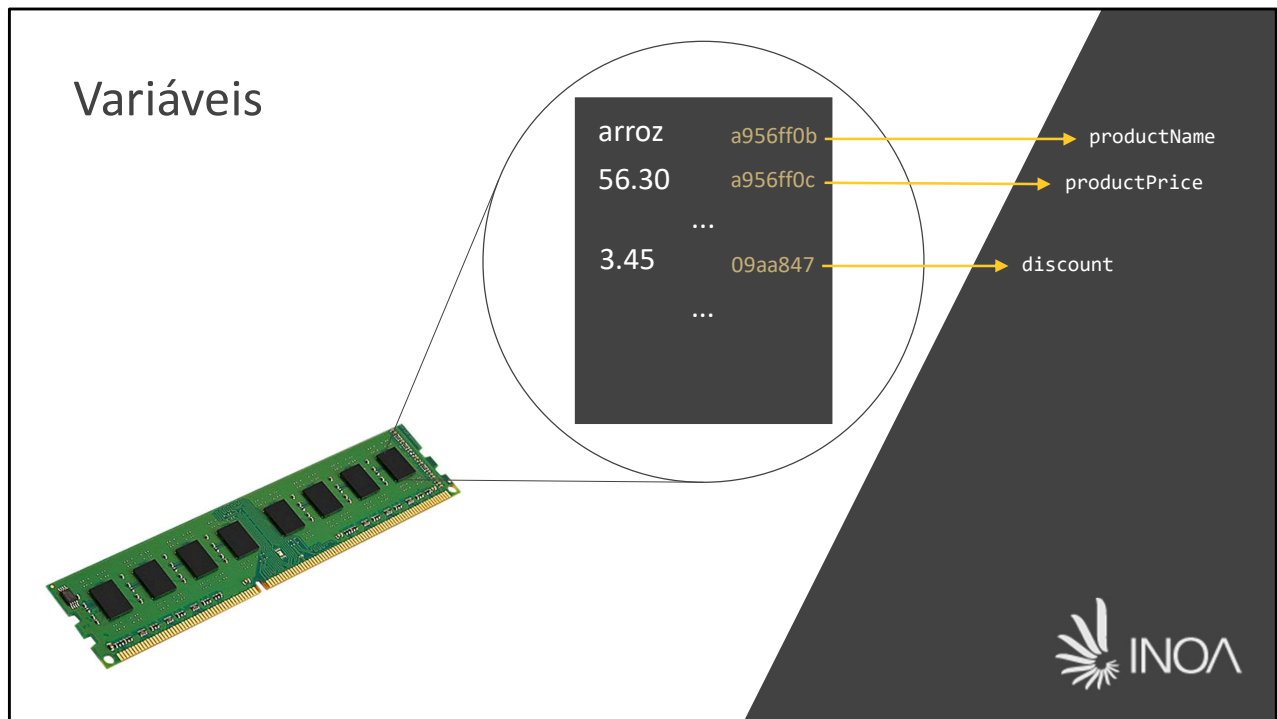
https://www.amazon.com.br/dp/B0026OR2ZY/ref=dp-kindle-redirect?_encoding=UTF8&btkr=1

https://wiki.theory.org/YourLanguageSucks#JavaScript_sucks_because

- Importando js em HTML
- Aritimética
- Manipulação de strings
- Variáveis
- Tipagem
- Comentários

Resumo e um Chá





Variáveis são formas de armazenamento de dados dentro de algum context ou escopo. Se você está fazendo um FAQ, uma pergunta é uma variável e uma resposta também. Se você está vendendo um produto, o preço é uma variável. Se sua aplicação é um jogo, pode ser a velocidade atual do personagem ou a pontuação. Fisicamente variáveis são armazenadas na memória do computador em algum endereço de memória. Com variáveis conseguimos dar nomes inteligíveis a esses endereços de memória. Assim comunicamos nossas regras de negócios não só para o computador mas também para outros programadores.

Em javascript ao declarar uma variável não precisamos alocar espaço explicitamente e podemos declarar uma variável sem atribuir uma valor a ela. E como já vimos é uma linguagem tipagem dinâmica.

Um ponto importante é que essa memória é finita, e portanto as variáveis tem **escopos**. Um endereço de memória deve ser liberado o mais rápido possível quando você não precisa mais daquela informação, dessa forma não esgotamos. Em js não controlamos explicitamente essa limpeza e liberação de espaço, mas indicamos ao runtime quando queremos que as variáveis sejam limpas de acordo com **como** declaramos uma variável.



Variáveis são chamadas assim porque elas podem mudar.

O valor de uma variável declarada com **let** e **var** pode ser trocado com o operador de atribuição (assign) '='.

No entanto quando elas são declaradas com **const**, você declara não uma variável, mas uma constante que não vai aceitar novas atribuições de valor e precisa ter o valor atribuído na declaração.

Um prática comum é sempre declarar constantes e só alterar pra **let** o que sabidamente vai mudar.

let e **const** são formas de atribuição relativamente novas (2016), e são os indicados para uso por terem um comportamento previsível.

Mas muito código legado em js vai usar **var** pra declaração de variáveis.

Como js precisa ser compatível com códigos legados, foi decidido usar novas formas de declaração pra desambiguar comportamentos.

Um ferramenta que nos ajuda a escrever código moderno, mas gerar código compatível com legado é o Babel.

Referências:

<https://babeljs.io/repl>



Referências:

<https://www.npmtrends.com/flow-bin-vs-typescript-vs-elm-vs-coffee-script>



JS FATIGUE

- jQuery
- Cleave.js
- D3.js
- Glimmer
- moment
- Polymer
- Multiple.js
- Omniscient
- Parsley
- Yeoman
- Premonish
- ReactJS
- Slick
- TaffyDB
- TweenJS
- Three.js
- Voca
- Aurelia
- Angular
- Ember
- Ionic
- Mocha
- Next.js
- Vue.js
- Webix
- Babel
- ESLint
- Gatsby
- iziModal
- Meteor
- Shave
- StencilJS
- TypeScript
- Webpack
- Chart.js
- Granim.js

Não uma... Várias...

O sucesso tem seu preço, com o crescimento da linguagem, num ambiente open source MUITAS biblioteca.

Paralisia de escolha é um problema real e a constante sensação que você aprende e domina um framework pra semana que vem um novo e mais colorido tomar seu lugar



JS FATIGUE

- ESTUDE OS FUNDAMENTOS
- FERRAMENTAS SÃO SÓ FERRAMENTAS
- JS É SÓ UMA LINGUAGEM DE PROGRAMAÇÃO

• jQuery • Multiple.js • Slick • Angular • Webix • Shave
• D3.js • D3.js • Tween.js • Babel • StencilJS
• Glimmer • Teoman • Three.js • Mocha • TypeScript
• Polymer • ReactJS • Aurelia • Vue.js • Meteor • Granim.js

- Importando js em HTML
- Aritimética
- Manipulação de strings
- Variáveis
- Tipagem
- Comentários

Resumo e cabô

É ISSO AÍ

CABÔ O PROGRAMA

