Aula 6

Introdução a JavaScript

Prof. Sandino Jardim

CC-UFMT-CUA

O que é JavaScript?

O que já vimos:

- Linguagem de programação que adiciona interatividade e comportamento às páginas
- Terceiro pilar do desenvolvimento front-end

Adicionalmente:

- *Client-side scripting language*
  - Portanto, dependente das capacidades e configurações do navegador
- Fracamente tipada

O que **não é** JavaScript

Nada tem a ver com Java

- criado para o Netscape em 1995, chamado originalmente de *LiveScript*

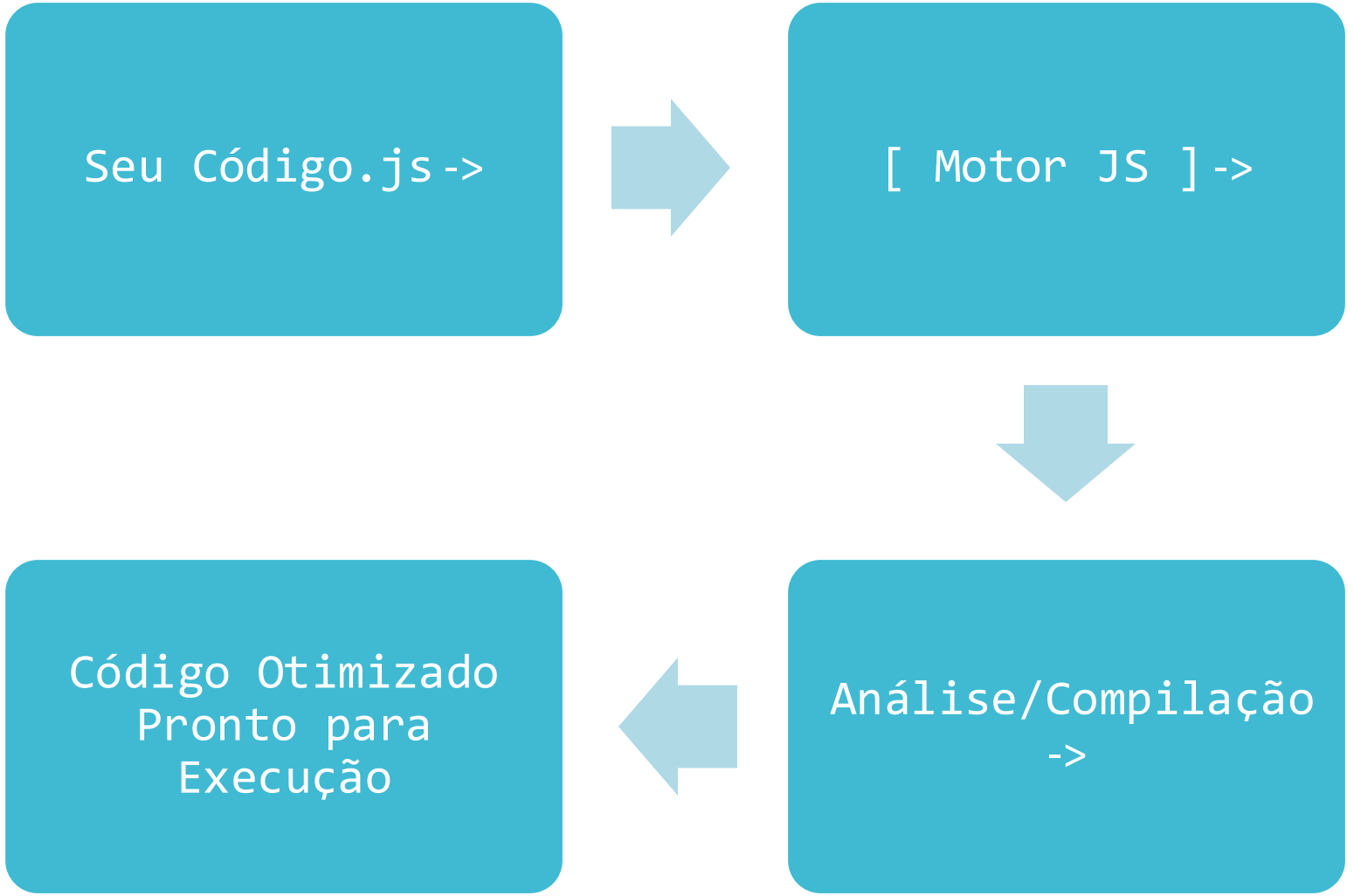
Não tem culpa pela sua má reputação

- Por um tempo foi sinônimo de vulnerabilidade
- Browsers permitem frear as capacidades do código

A "Lei" do JavaScript: ECMAScript

- ECMAScript é a **especificação oficial** da linguagem. JavaScript é a **implementação** mais famosa dessa especificação.
- Versões importantes que você ouvirá falar:
  - **ES5 (2009):** A base do JavaScript moderno.
  - **ES6 / ES2015:** A maior atualização da história, introduziu `let`, `const`, *arrow functions*, etc.

O Mito da "Linguagem Interpretada"



Por Que a Compilação Importa?

- **Conceito:** "O motor do JS 'lê' seu código pelo menos duas vezes."
  - **1ª Passada (Compilação):** Registra todas as declarações de variáveis (var) e funções.
  - **2ª Passada (Execução):** Executa o código de fato.
- **Spoiler:** É por isso que você pode chamar uma função antes de ela ser declarada no código. Veremos isso em detalhes na próxima aula!

Os Pilares do Código: Tipos de Dados

Tipos primitivos	Tipo Objeto
String	Object ([ ] e { })
Number	
Boolean	
Null	
Undefined	
symbol	

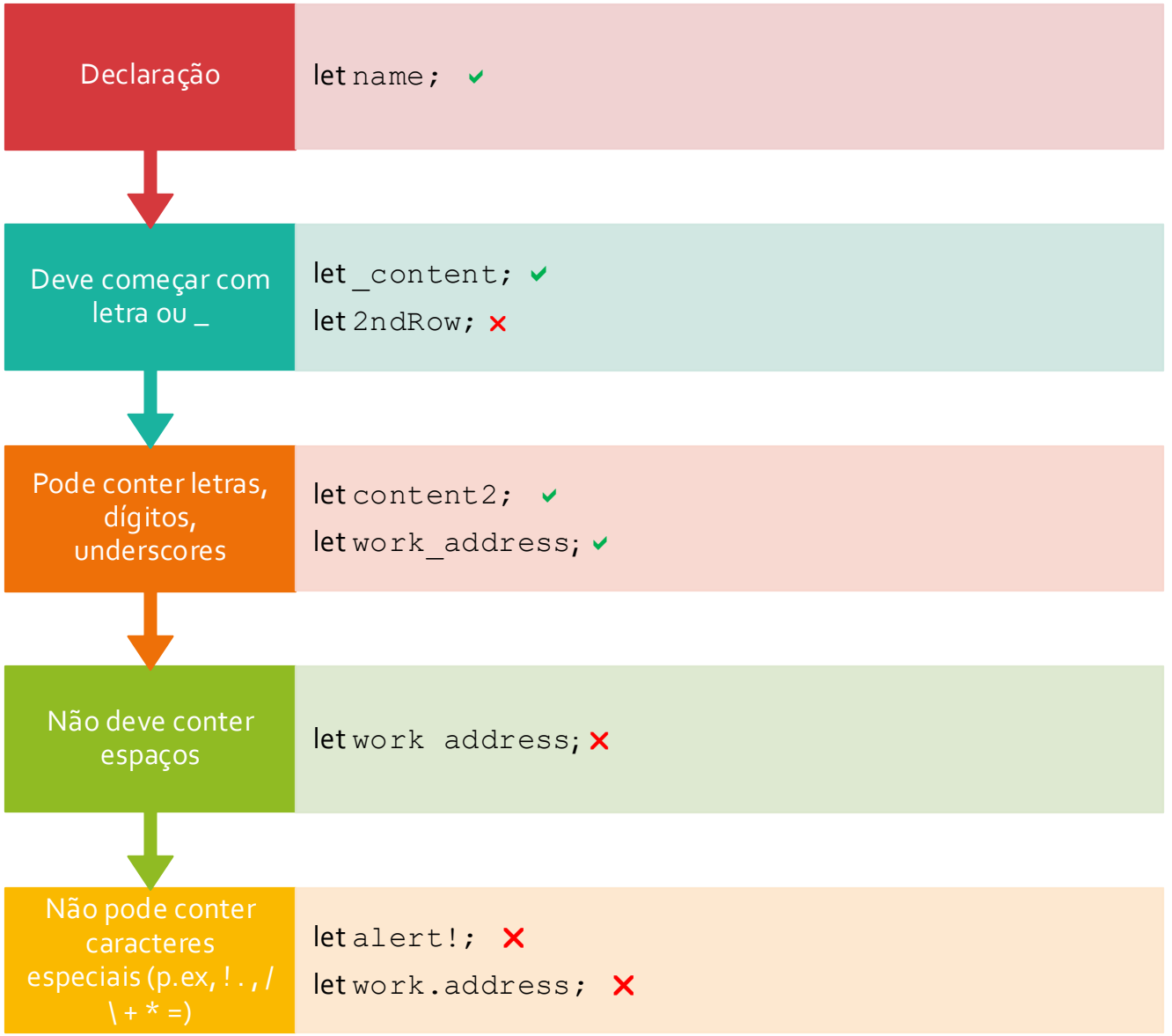
Guardando valores

- `var`: **Evite!** Tem escopo de função, pode causar bugs. (O jeito antigo)
- `let`: Use para variáveis que **precisam mudar** de valor. Tem escopo de bloco.
- `const`: **Prefira sempre!** Para valores que **não serão reatribuídos**. Também tem escopo de bloco.
- **Regra de Ouro:** "Comece com `const`. Se precisar reatribuir, mude para `let`."

```
for (var i = 0; i < 3; i++) {
  setTimeout(function() {
    console.log(i); // O que acha que isto vai imprimir?
  }, 1000);
}
```

Variáveis

- Entrada:
  - prompt() - entrada no browser
- Saída
  - console.log(value) - saída no console
  - alert(value) - saída no browser



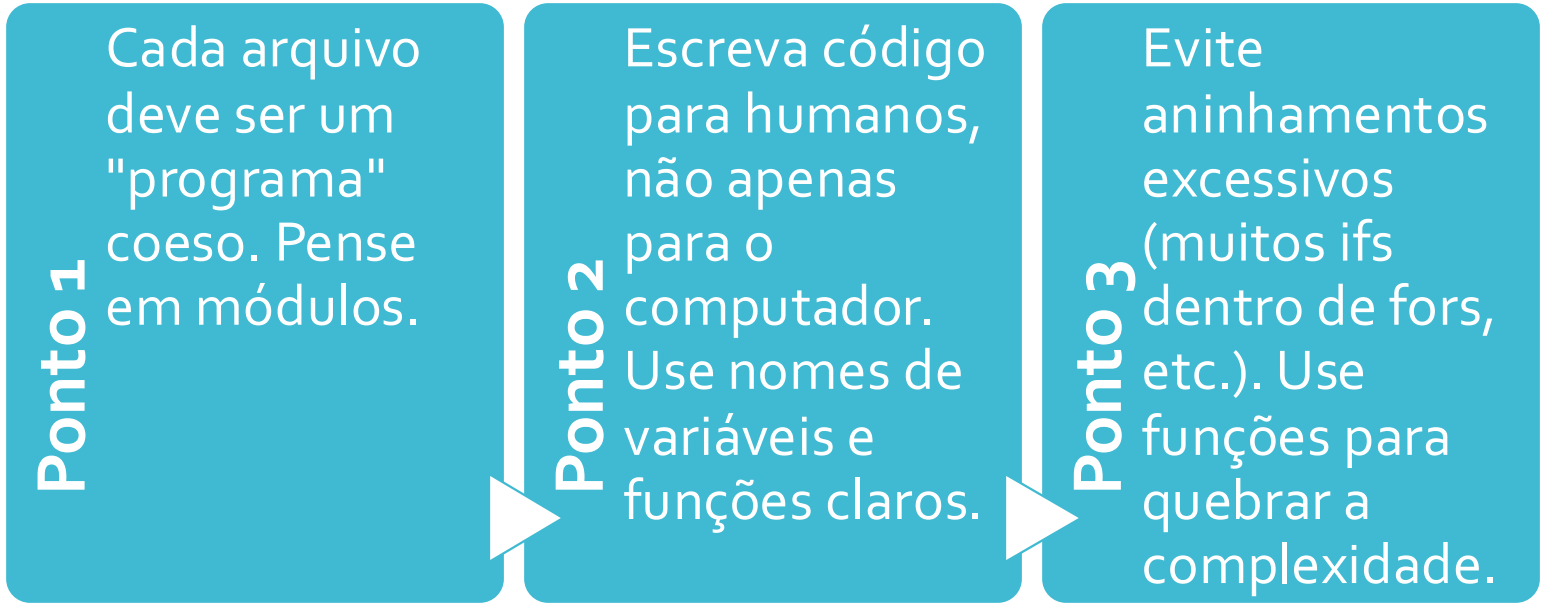
Funções

- **Definição:** Blocos de código nomeados e reutilizáveis.
- **Ponto Chave:** Em JS, funções são "cidadãos de primeira classe". Elas podem ser tratadas como qualquer outro valor (passadas como argumento, retornadas, etc.).
- **Exemplos de Sintaxe:**
- **Declaração:** `function soma(a, b) { ... }`
- **Arrow Function (ES6):** `const soma = (a, b) => { ... }`

Comparações

- **== (Igualdade Frouxa - Perigoso!)**
  - Tenta converter os tipos antes de comparar.
  - Exemplo: `5 == "5"` resulta em `true`. (Fonte de bugs!)
- **=== (Igualdade Estrita - Seguro!)**
  - Compara o valor **E** o tipo. Não faz conversão.
  - Exemplo: `5 === "5"` resulta em `false`.
- **Mantra do Desenvolvedor JS: SEMPRE USE ===!**

Organizando seu Código



"Qualquer um pode escrever um código que o computador entende. Bons programadores escrevem código que humanos entendem." – Martin Fowler

Adicionando JS a uma página

- Assim como em CSS, podemos incorporar um script de duas formas:
  - Script incorporado diretamente na página

```
<script>
  … JavaScript code goes here
</script>
```

  - Script externo

```
<script src="my_script.js"></script>
```

- O elemento <script> pode vir em qualquer lugar, sendo o mais comum dentro do elemento <head> e ao fim do elemento <body>

if/else

```
let theNumber = Number(prompt("Pick a number"));
if (!Number.isNaN(theNumber)) {
  console.log("Your number is the square root of " +
              theNumber * theNumber);
} else {
  console.log("Hey. Why didn't you give me a number?");
}
```

```
let num = Number(prompt("Pick a number"));

if (num < 10) {
  console.log("Small");
} else if (num < 100) {
  console.log("Medium");
} else {
  console.log("Large");
}
```

Loops

```
for (let number = 0; number <= 12; number = number + 2)
  console.log(number);
}
// → 0
// → 2
```

```
let result = 1;
let counter = 0;
while (counter < 10) {
  result = result * 2;
  counter = counter + 1;
}
console.log(result);
// → 1024
```

```
let yourName;
do {
  yourName = prompt("Who are you?");
} while (!yourName);
console.log("Hello " + yourName);
```

Arrays

Grupo de múltiplos valores atribuídos a uma única variável, indexado a partir do número o

```
var foo = new Array();
foo[0] = 5;
foo[1] = "cinco";
foo[2] = "5";
```

Ou:

```
var foo = [5, “cinco", "5"]
```

Iterando sobre objetos

```
let pessoa = { nome: "Ana", idade: 25, cidade: "São Paulo" };

  for (let chave in pessoa) {
    console.log(chave + ": " + pessoa[chave]);
  }


  Object.keys(pessoa).forEach(chave => {
  console.log(chave + ": " + pessoa[chave]);
  });


  Object.values(pessoa).forEach(valor => {
  console.log(valor);
  });

  Object.entries(pessoa).forEach(([chave, valor]) =>
  {
  console.log(chave + ": " + valor);
  });
```