

# **AULA 04 - DANDO VIDA AOS SISTEMAS COM JAVASCRIPT**

Prof. Fabricio Freire, MSc in Cybersecurity

# METAS DE APRENDIZAGEM

- 🔗 Criar scripts integrados ao HTML
- 🔍 Inspecionar páginas no navegador
- 📦 Conhecer as diferenças entre variáveis
- ♣️ Entender o conceito de DOM
- ⚡ Manipular elementos do DOM
- 📊 Estruturar dados em arrays e objetos
- 🔧 Criar funções em JavaScript

# POR QUE JAVASCRIPT?

A linguagem que dá vida à web

Páginas estáticas versus dinâmicas

- HTML + CSS = Estrutura + Visual
- JavaScript = Interatividade + Dinamismo

Criada em 1995

- Originalmente LiveScript
- Parceria Netscape + Sun Microsystems
- Padrão ECMAScript (ECMA-262)

# ONDE O JAVASCRIPT ATUA

Transformando experiências web

Exemplos práticos:

- ✓ Validação de formulários em tempo real
- ✓ Animações e efeitos visuais
- ✓ Carregamento dinâmico de conteúdo
- ✓ Jogos e aplicações interativas
- ✓ Respostas imediatas a ações do usuário

# ADICIONANDO JAVASCRIPT AO HTML

## 1. Inline (não recomendado)

```
<button onclick="alert('Olá!')">Clique</button>
```

## 2. No cabeçalho com <script>

```
<head>  
  <script>  
    // Código JavaScript aqui  
  </script>  
</head>
```

## 3. Arquivo externo (recomendado)

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

# PROPRIEDADE DEFER

## Controlando o carregamento

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

## O que faz o defer?

- ⏸ Adia a execução do script
- ✓ Espera o HTML carregar completamente
- 🔄 Evita erros de elementos não encontrados
- 📍 Ideal para scripts no <head>

# FERRAMENTAS DO DESENVOLVEDOR

DevTools - Seu melhor amigo!





Como acessar:

- Clique direito → "Inspecionar"
- Ou pressione F12

Console JavaScript:

```
console.log("Bem-vindos à programação frontend!");
```

Para que serve:

-  Depurar código
-  Testar comandos
-  Ver mensagens de erro
-  Monitorar variáveis

# VARIÁVEIS EM JAVASCRIPT

Guardando dados temporariamente

Três formas de declarar:

✗ Não use mais  
`var rodada = 0;`

☑ Para valores que mudam  
`let rodada = 0;`

☑ Para valores constantes  
`const totalPerguntas = 17;`

**Dica: Use sempre const por padrão, mude para let quando necessário!**



# TIPOS DE DADOS

JavaScript é dinamicamente tipado

Tipos Primitivos:

- **null** - valor nulo
- **undefined** - indefinido
- **string** - texto
- **number** - número
- **boolean** - verdadeiro/falso
- **symbol** - símbolo único

Tipos Derivados:

- Objetos
- Arrays
- Funções

# COMPARAÇÕES EM JAVASCRIPT

Cuidado com os operadores!

Comparação fraca (ignora tipos)

`1 == "1" // true` ☒

Comparação estrita (compara valor E tipo)

`1 === "1" // false` ✗

**Recomendado: sempre use `===` e `!==`**

# DOCUMENT OBJECT MODEL - DOM

DOM é:

- 🌳 Uma árvore hierárquica
- 📄 Representação do HTML
- 🔗 Interface para JavaScript

## Document

```
└─ html
   └─ head
      └─ title
   └─ body
      └─ h1
         └─ p
```

# SELECIONANDO ELEMENTOS

## Principais métodos do DOM

*// Por ID*

```
const elemento = document.getElementById("meuId");
```

*// Por classe*

```
const elementos = document.getElementsByClassName("minhaClasse");
```

*// Por tag*

```
const paragrafos = document.getElementsByTagName("p");
```

*// Com seletores CSS*

```
const elemento = document.querySelector(".classe #id");
```

```
const todos = document.querySelectorAll("div p");
```

# MANIPULANDO ELEMENTOS

## Alterando conteúdo e propriedades

- **Selecionar elemento**  
`const enunciado = document.getElementById("enunciado");`
- **Alterar conteúdo HTML**  
`enunciado.innerHTML = "Quem descobriu o Brasil?";`
- **Alterar apenas texto**  
`enunciado.innerText = "Nova pergunta";`
- **Alterar estilos**  
`enunciado.style.color = "blue";`  
`enunciado.style.fontSize = "20px";`

# ARRAYS EM JAVASCRIPT

## Listas de valores

- **Criando arrays**  
`const frutas = ["banana", "limão", "pera", "jaca"];`
- **Acessando elementos (índice começa em 0)**  
`console.log(frutas[0]); // "banana"`  
`console.log(frutas[2]); // "pera"`
- **Propriedades úteis**  
`console.log(frutas.length); // 4`

# OBJETOS EM JAVASCRIPT

## Pares chave-valor

```
// Criando um objeto
const pergunta = {
  enunciado: "Qual o nome de batismo do Superman?",
  alternativas: ["Bruce Wayne", "Clark Kent", "Wally West"],
  resposta: "Clark Kent"
};

// Acessando propriedades
console.log(pergunta.enunciado);
console.log(pergunta["resposta"]);
```

# FUNÇÕES TRADICIONAIS

## Blocos de código reutilizáveis

*// Declaração de função*

```
function soma(a, b) {  
  return a + b;  
}
```

*// Chamando a função*

```
const resultado = soma(5, 3); // 8
```

*// Função com múltiplas linhas*

```
function exibePergunta(numero) {  
  const p = perguntas[numero];  
  enunciado.innerHTML = p.enunciado;  
}
```



# ARROW FUNCTIONS

## Sintaxe moderna e concisa

*// Função tradicional*

```
function soma(a, b) {  
  return a + b;  
}
```

*// Arrow function equivalente*

```
const soma = (a, b) => a + b;
```

*// Com múltiplas linhas*

```
const exibePergunta = (numero) => {  
  const p = perguntas[numero];  
  enunciado.innerHTML = p.enunciado;  
};
```

# PROJETO PRÁTICO

 Quem quer ser um milionário?!

Vamos construir:

- Um jogo de perguntas e respostas
- Interface interativa
- Sistema de pontuação
- Feedback visual

**Recursos que usaremos:**

- Manipulação do DOM
- Arrays e objetos, Funções e Eventos

# ESTRUTURA DO PROJETO

## Organizando nossos arquivos

projeto-trivia/

├ index.html      (estrutura)

├ styles.css      (visual)

├ script.js      (lógica)

└ extras.js      (funcionalidades extras)

# EXEMPLO: EXIBINDO PERGUNTAS

Aplicando o que aprendemos

```
const perguntas = [
  {
    enunciado: "Qual é o maior planeta?",
    alternativas: ["Júpiter", "Saturno", "Urano", "Netuno"],
    resposta: "Júpiter"
  }
];

function exibePergunta(numero) {
  const pergunta = perguntas[numero];

  document.getElementById("enunciado").innerText =
    pergunta.enunciado;

  // Exibir alternativas...
}
```

# EVENTOS E INTERATIVIDADE

Respondendo a ações do usuário







```
// Adicionar evento de clique
const botao = document.getElementById("btnPerguntar");

botao.addEventListener("click", () => {
  // Verificar resposta
  const selecionada =
document.querySelector(".selecionada");

  if (selecionada) {
    // Processar resposta
    verificarResposta(selecionada.innerText);
  }
});
```




# DICAS IMPORTANTES

## Boas práticas




-  Use `const` por padrão
-  Sempre verifique o console
-  Consulte a documentação MDN
-  Teste no console antes
-  Mantenha funções pequenas e focadas
-  Use nomes descritivos para variáveis

# RECURSOS PARA ESTUDO

## Documentação:

-  MDN Web Docs
-  W3Schools
-  JavaScript.info

## Prática:

-  CodePen
-  JSFiddle
-  GitHub

## Comunidade:

- Stack Overflow
- Dev.to
- Reddit r/javascript

# ATIVIDADE PRÁTICA

Mãos à obra! 🛠️

**Desafio:**

1. Baixe o projeto inicial
2. Adicione 5 novas perguntas
3. Implemente a verificação de respostas
4. Adicione feedback visual (certo/errado)
5. Bônus: Sistema de pontuação



**BONS ESTUDOS**