

Aula 2 – HTML e JavaScript

Programação Mobile
Professor Bruno Maion

Introdução

Uma página Web é composta basicamente três componentes

- Conteúdo (informação)
- Apresentação (visual, formatação e design – apresentação)
- Comportamento (mudanças dinâmicas)

Podemos dividir esses componentes em três tecnologias básicas:

- HTML (conteúdo)
- CSS (apresentação)
- JavaScript (comportamento)

HTML

O que é ?

- **HTML** (*HyperText Markup Language*) é a linguagem de **marcação** usada para **estruturar** o conteúdo de páginas web.
- **Define elementos** como títulos, parágrafos, imagens, tabelas e links.
- **Não** é uma linguagem de programação → É uma linguagem de estruturação de dados.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Minha Página</title>
  </head>
  <body>
    <h1>Olá, mundo!</h1>
    <p>Bem-vindo ao site.</p>
  </body>
</html>
```

JavaScript

- **JavaScript** é uma linguagem **interpretada**
 - Não exige **compilação** para executar programas
 - O navegador precisa apenas de um **interpretador** de JavaScript **embutido**
 - Foi desenvolvida por Brendan Eich, em 1995, para o **Netscape Navigator** (versão 2.0)
 - Posteriormente, foi incorporada ao Internet Explorer sob o nome de JScript
- **JavaScript não é Java!**
 - Embora tenham sintaxe semelhante (inspirada em C/C++), são linguagens com origens diferentes
 - **Java** é uma linguagem mais **robusta** e complexa, com **propósitos distintos**

O que pode ser feito com JavaScript?

- **JavaScript fornece uma linguagem de programação para interagir com HTML.**
 - É uma linguagem simples e de fácil aprendizado.
- **Permite inserir texto dinâmico em uma página.**
 - Exemplo: `document.write("<h1>" + name + "</h1>")` → escreve o valor de uma variável diretamente em uma página HTML.
- **Executa ações quando algo está prestes a acontecer, como:**
 - O carregamento completo da página;
 - O clique do usuário em algum elemento da página; entre outros.
- **Possibilita ler e modificar elementos HTML.**
 - Por exemplo, para alterar o conteúdo de um elemento na página.

O que pode ser feito com JavaScript?

- **Validar dados de formulário**
 - Antes de enviar os dados do formulário para o servidor, o JavaScript pode ser utilizado para validar os campos, evitando processamento extra no servidor.
- **Detectar o navegador do usuário**
 - JavaScript permite ajustar a apresentação da página para o navegador específico do usuário ou carregar uma versão da página que funcione melhor nele.
- **Criar *cookies***
 - A linguagem possibilita armazenar e recuperar informações salvas no computador do usuário, ajudando a personalizar a experiência.

O que é um algoritmo?

- Uma sequência fininita e ordenada de passos para a resolução de um problema específico.

Algoritmo para fritar ovos

1. pegar frigideira, ovo, óleo e sal
2. colocar óleo na frigideira
3. acender o fogo
4. levar a frigideira ao fogo
5. esperar o óleo esquentar
6. quebrar o ovo
7. colocar o ovo
8. colocar sal no ovo
9. retirar quando pronto
10. apagar o fogo

O que é uma variável?

- Na Matemática, variável é o nome que se dá ao termo que substitui um outro desconhecido, como numa equação: $y = 2x + 1$.
- O valor de uma variável pode ser conhecido ou não (ponteiro).
- Uma variável é uma referência a um local na memória do computador, onde um valor pode estar armazenado.
- Exemplo:
 - Idade, que armazena o valor **inteiro** 18;
 - Nome, que armazena a **string** João;

Variáveis

- O nome de uma variável
 - **não** deve conter espaços ou acentos.
 - **não** deve conter operadores (<, >, == ...);
 - **não** deve ser só um número
- Algumas linguagens de programação diferenciam letras maiúsculas de letras minúsculas, logo: **Nome** pode ser diferente de **nome**
- Variáveis cujo nome seja um substantivo composto, como “data de nascimento”, podem ser nomeadas com o padrão **camelCase**:
 - **DataDeNascimento, salaTurma;**
- Ou underline (_), entre cada palavra, **snake_case**:
 - **data_de_nascimento, sala_turma**

Variáveis

- O nome de uma variável
 - **não** deve conter espaços ou acentos.
 - **não** deve conter operadores (<, >, == ...);
 - **não** deve ser só um número
- Algumas linguagens de programação diferenciam letras maiúsculas de letras minúsculas, logo: **Nome** pode ser diferente de **nome**
- Variáveis cujo nome seja um substantivo composto, como “data de nascimento”, podem ser nomeadas com o padrão **camelCase**:
 - **DataDeNascimento, salaTurma;**
- Ou underline (_), entre cada palavra, **snake_case**:
 - **data_de_nascimento, sala_turma**

Variáveis

Atribuir um valor a uma variável é dar um valor a uma variável. É o mesmo que guardar um valor em um espaço da memória.

- **a = 4;** pode ser lido como: variável a recebe 4 .
- **nome = "José";** pode ser lido como: variável nome recebe José

Este é o padrão, mas pode variar!!

Tipo de Variáveis

- **Inteiro** (integer): número inteiro. Ex.: 10 ou 50 .
- **Real** (float): número real. Ex.: 10.5 ou 0.8 .
- **Caractere** (character): uma letra do alfabeto, algarismo, sinal de pontuação ou símbolo. Ex.: 'A' ou '2' ou '\$'.
- **Cadeia (string)**: números, letras, palavras, frases e/ou outros símbolos. Ex.: "A" ou "abc123" ou "* é um asterisco."
 - Valores do tipo cadeia geralmente são representados entre aspas duplas ("texto"); embora em JavaScript se possa utilizar também aspas simples ('texto').
- **Lógico** (boolean): verdadeiro ou falso (true e false em JavaScript)

Tipo de Variáveis

Embora não se possa fixar o tipo de uma variável em JavaScript, os tipos de valores existentes nesta linguagem são:

- números (number)
- cadeias de caracteres (string)
- lógicos (boolean)
- objetos (object)
- nulo (null)
- indenido (undefined)

Operadores

Aritméticos

- Soma (+):
 $1 + 3$
- Subtração (-):
 $5 - 4$
- Multiplicação (*):
 $8 * 8$
- Divisão (/):
 - $40 / 5$
- Resto/Módulo (%):
 $20 \% 3$
- Potenciação (**): $2 ** 2$

Relacionais

- Igual (==):
 $5 == 5$
- Diferente (!=):
 $8 != 80$
- Maior que (>):
 $12 > 11$
- Menor que (<):
 $11 < 12$
- Maior ou igual (>=):
 $18 >= 17$
- Menor ou igual (<=):
 $16 <= 16$

Lógicos

- E/Conjunção (&&):
 $5 == 5 \&\& 9 > 7$
- Ou/Disjunção (||):
 $5 > 3 || 7 >= 1$
- Não/Negação (!):
 $!(3 > 2) == \text{false}$

Operadores

- E e Ou (em JS, **&&** e **||**)

X	Y	X e Y	X ou Y
F	F	F	F
F	V	F	V
V	F	F	V
V	V	V	V

Precedência de operadores

- 1. Parênteses e funções
- 2. Potência e resto
- 3. Multiplicação e divisão
- 4. Adição e subtração
- 5. Operadores relacionais
- 6. Operadores lógicos

```
3 + 4 * 9;  
3 + 36;  
39;
```

```
10 < 9 e 6 + 3 > 10  
10 < 9 e 9 > 10  
F e F  
F
```


Declaração de variáveis

- **let** é utilizado para declarar uma variável com escopo de **bloco**, ou seja, uma variável que só é acessível dentro do bloco em que foi declarada;

```
let nome = João;
```
- O termo **var** para declarar uma variável com escopo **global**. Variáveis assim, mesmo se declaradas dentro de um bloco, serão acessíveis do lado de fora dele.

```
var nome = João;
```

Declaração de variáveis

- Em certos casos precisamos declarar variáveis que não terão seu valor ou referência modificados. Este tipo de variável é chamado de **constante**.
- Para declarar uma constante em JavaScript, utilize o termo **const**;

```
const nome = João;
```

Estruturas de Decisão

- Na programação, há momentos em que precisamos fazer decisões, definir **caminhos** diferentes para situações diferentes, e é por isso que existem as estruturas de decisão.
- As duas estruturas de decisão presentes na maioria das linguagens de programação são o se/senão (**if / else**) e o escolha/caso (**switch / case**).

“Se o aluno tirou nota igual ou maior que 70, está aprovado”

If / Else

- A estrutura **if** (se) é utilizada na maioria dos casos em que é necessário decidir que ação será tomada; Já o **else** (se não) adota um outro caminho caso o anterior seja falso.

```
let numero;  
numero = parseInt(prompt("Entre com um número inteiro: "));  
if (numero < 0) {  
    console.log("O número lido é negativo.");  
} else if (numero > 0) {  
    console.log("O número lido é positivo.");  
} else {  
    console.log("O número lido é nulo.");  
}
```

Switch / Case

Estrutura de controle que compara um valor com vários casos (case).
Útil para evitar muitos if...else.

- **switch** → Escolha múltipla.
- **case** → Opções possíveis.
- **break** → Impede que outros casos sejam executados.
- **default** → Executa quando nenhum caso corresponde.

```
switch (variavel) {  
    case valor1:  
        // código  
        break;  
    case valor2:  
        // código  
        break;  
    default:  
        // código caso nenhum valor corresponda  
}
```

Estruturas de repetição

- As estruturas de repetição permitem executar um bloco de código várias vezes, economizando trabalho e tornando o programa mais eficiente.
- São úteis quando precisamos processar listas, contar números, repetir ações até que uma condição seja atendida, entre outros cenários.
- No JavaScript, as principais são:
 - **for** – repetição com número fixo de vezes.
 - **while** – repetição enquanto a condição for verdadeira.
 - **do...while** – semelhante ao while, mas executa pelo menos uma vez.

Estrutura FOR

- O for é ideal quando sabemos exatamente quantas vezes queremos repetir uma ação.
- Ele possui três partes:
 - Inicialização – onde definimos a variável de controle.
 - Condição – define até quando o loop continuará.
 - Incremento – altera a variável de controle a cada repetição.

```
for (let i = 1; i <= 5; i++) {  
  console.log(i);  
}
```

Estrutura While

- O **while** executa um bloco de código enquanto a condição for **verdadeira**.
- Útil quando não sabemos o número **exato** de repetições.

```
let contador = 1;
while (contador <= 5) {
  console.log(contador);
  contador++;
}
```


Estrutura Do While

- O **do...while** é parecido com o while, mas ele executa o código pelo menos uma vez, mesmo que a condição seja falsa no início.

```
let numero = 1;  
do {  
    console.log(numero);  
    numero++;  
} while (numero <= 5);
```

Em JavaScript!

- Como JavaScript é uma linguagem interpretada pelo navegador, utilizaremos o console, ou a caixa de alerta;
 - Entrada:
 - Utilizar o **prompt("Insira aqui")**
 - Saída:
 - Para o **console.log(texto)** deve apertar F12, ou utilizar
 - **alert(texto)**

Exercício

- **Criar dois algoritmos com entrada e saída**
 - Um com **if, else if, else**
 - Outro com **switch case**
- **Criar dois algoritmos com entrada apenas**
 - Um com um **for**
 - Outro com **while** ou **do while**



Obrigado!