

IW-II INTERFACES WEB II

Prof. Márcio

Competências

1. Desenvolver sistemas para Internet com foco na interface com o usuário e programação lado cliente.

Habilidades

1. Codificar software em linguagem para a Web.
2. Utilizar linguagem de script para Web.
3. Utilizar interface baseada em navegador para interação com usuário.

Bases Tecnológicas

Processamento script lado cliente (Javascript)

- Sintaxe básica, variáveis, tipos e escopo;
- Controle de fluxo e manipulação de erro;
- Laços e interação;
- Funções e manipulação Document Object Model (DOM);
- XML e JSON;
- Solicitações assíncronas (AJAX);
- Cookies.

Bases Tecnológicas

Biblioteca Javascript cross-browser (JQuery)

- Instalação, função \$() e seletores;
- Eventos;
- Manipulação.

Critérios de Avaliação

- Exercícios práticos
- Pesquisa Técnica
- Avaliação Prática
- Projeto Prático
- Assiduidade e Participação em sala de aula

Critérios de Avaliação

I : menos de 50%

R : de 50% até 70%

B : de 70% até 90%

MB : de 90% a 100%

Material das aulas

- Github:

<https://github.com/Marciobarrosetec/2026/tree/main/IW>

Divisão das Bases

1º Bimestre - Fundamentos do JavaScript

- Introdução ao JavaScript
- Variáveis e Tipos de Dados
- Operadores e Expressões
- Controle de Fluxo
- Laços de Repetição
- Funções em JavaScript
- Escopo e Hoisting
- Manipulação de Arrays e Objetos
- Tratamento de Erros
- Mini-projeto

Divisão das Bases

2º Bimestre - Manipulação do DOM e Eventos Avançados e Delegação

Eventos

- Introdução ao DOM
 - Eventos no DOM
 - Manipulação de CSS com JavaScript
 - Criando e Removendo Elementos
 - Formulários e Validação
- Introdução ao LocalStorage
 - Manipulação Avançada de Arrays e DOM
 - Mini-projeto 2

Divisão das Bases

3º Bimestre - Comunicação Assíncrona e Efeitos e Animações

jQuery

- Introdução ao JSON
- Requisições AJAX com Fetch API
- Consumindo APIs Públcas
- Introdução ao jQuery
- Eventos com jQuery
- Manipulação do DOM com jQuery
- Validação de Formulários com jQuery
- Mini-projeto 3

Divisão das Bases

4º Bimestre - Aplicação Prática e Projeto • Consumo de APIs no projeto

Final

- Introdução ao Projeto Final
- Estruturando o HTML e CSS do projeto
- Implementando a lógica com JavaScript (parte 1)
- Implementando a lógica com JavaScript (parte 2)
- Implementação de jQuery para interatividade
- Testes e Debugging do código
- Refinamento e otimização do projeto
- Apresentação dos projetos pelos alunos
- Entrega final e feedback do professor

Aula 1: Introdução ao JavaScript

O que é JavaScript?

JavaScript é uma linguagem de programação de **alto nível, dinâmica e interpretada**, amplamente utilizada para adicionar interatividade a páginas da web.

Juntamente com HTML e CSS, é **uma das três principais tecnologias da Web**. Originalmente desenvolvida por Brendan Eich em 1995, no contexto do navegador Netscape Navigator, a linguagem evoluiu significativamente desde então.

O que é JavaScript?

No ambiente do navegador, o JavaScript permite manipular o **Document Object Model (DOM)**, possibilitando a atualização e modificação do conteúdo exibido sem a necessidade de recarregar a página. Além disso, a linguagem é utilizada em outros contextos, como no desenvolvimento de aplicativos de servidor com o **Node.js**, permitindo a criação de aplicações completas utilizando apenas JavaScript.

O que é JavaScript?

Uma característica notável do JavaScript é seu suporte a múltiplos paradigmas de programação, incluindo programação **orientada a objetos, funcional e imperativa**. Isso proporciona flexibilidade aos desenvolvedores na abordagem de diferentes problemas e na implementação de soluções diversas.

Com o tempo, a linguagem expandiu seu uso para além dos navegadores, sendo empregada em ambientes como **servidores, aplicativos móveis e até mesmo em sistemas embarcados**. Essa versatilidade consolidou o JavaScript como uma ferramenta essencial no desenvolvimento moderno de software.

Diferenças entre JS e outras linguagens

JavaScript é uma linguagem de programação amplamente utilizada, especialmente no desenvolvimento web. Embora compartilhe algumas semelhanças com outras linguagens, como Java e Python, possui características distintas que influenciam sua aplicação e comportamento.

Diferenças entre JavaScript e Java:

Tipagem: Java é uma linguagem de tipagem estática e fortemente tipada, o que significa que as variáveis devem ser declaradas com seus tipos específicos e não podem mudar de tipo. Em contraste, JavaScript é de tipagem dinâmica e fracamente tipada, permitindo que variáveis mudem de tipo durante a execução.

Diferenças entre JavaScript e Java:

Compilação vs. Interpretação: Java é uma linguagem compilada; o código fonte é compilado em bytecode, que é então executado pela Máquina Virtual Java (JVM). JavaScript, por outro lado, é interpretado diretamente pelos navegadores da web, sem a necessidade de uma etapa de compilação.

Diferenças entre JavaScript e Java:

Aplicações: Java é amplamente utilizado no desenvolvimento de aplicativos de grande escala, como sistemas corporativos, aplicativos móveis (especialmente para Android) e aplicações de desktop. JavaScript foi inicialmente projetado para adicionar interatividade às páginas web, mas com o advento de tecnologias como Node.js, seu uso se expandiu para o desenvolvimento de aplicativos no lado do servidor.

Diferenças entre JavaScript e Python:

Sintaxe: Python é conhecido por sua sintaxe clara e legível, que enfatiza a legibilidade do código. JavaScript possui uma sintaxe mais complexa, que pode ser menos intuitiva para iniciantes.

Paradigmas de Programação: Ambas as linguagens suportam múltiplos paradigmas de programação, incluindo programação orientada a objetos, funcional e imperativa. No entanto, Python é frequentemente preferido para aplicações científicas, análise de dados e aprendizado de máquina, enquanto JavaScript é predominante no desenvolvimento web.

Diferenças entre JavaScript e Python:

Ambiente de Execução: Python é uma linguagem de propósito geral que pode ser executada em diversos ambientes, desde scripts simples até aplicações complexas de inteligência artificial. JavaScript foi originalmente projetado para ser executado em navegadores web, mas com o Node.js, também pode ser utilizado no lado do servidor.

Estrutura Básica

Criando os arquivos básicos

Agora que temos uma noção básica sobre o que é a linguagem JavaScript e como se constitui uma página web, vamos partir para um pouco mais de prática e criar a nossa primeira estrutura web.

Criando os arquivos básicos

Agora que temos uma noção básica sobre o que é a linguagem JavaScript e como se constitui uma página web, vamos partir para um pouco mais de prática e criar a nossa primeira estrutura web.

Criando os arquivos básicos

```
aula01 > <> index.html > ...
 1  <!DOCTYPE html>
 2  <html lang="pt-br">
 3  <head>
 4  |  <meta charset="UTF-8">
 5  |  <meta name="viewport" content="width=device-width, initial-scale=1.0">
 6  |  <title>JavaScript Básico: Estrutura Básica</title>
 7  </head>
 8  <body>
 9  |  <h1>Primeira estrutura básica</h1>
10  </body>
11  </html>
```

Criando os arquivos básicos

Agora que temos a estrutura básica do nosso HTML, vamos aprender como colocar um JavaScript em nossa página. Há duas formas de fazermos isto. A primeira forma é colocarmos uma tag script e dentro desta escrevermos nosso código JavaScript:

Criando os arquivos básicos

```
aula01 > ◁ index.html > ...
1   <!DOCTYPE html>
2   <html lang="pt-br">
3   <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>JavaScript Básico: Estrutura Básica</title>
7       <script>
8           |   console.log("Olá Mundooooo");
9       </script>
10      </head>
11      <body>
12          <h1>Primeira estrutura básica</h1>
13      </body>
14      </html>
```

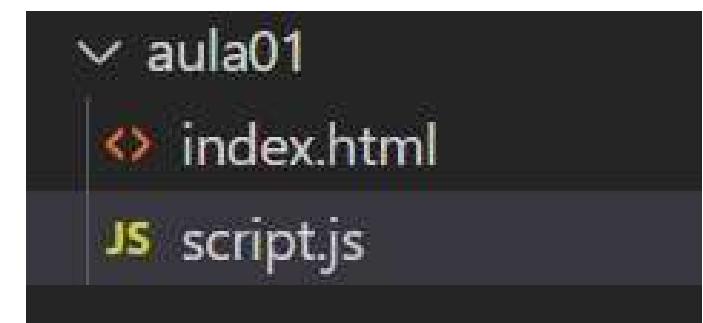
Criando os arquivos básicos

A segunda forma é criarmos um arquivo do tipo JavaScript chamado **script.js** (ou um outro nome que preferir!). Após criar o arquivo você irá referenciar o mesmo em seu HTML utilizando uma **tag script** e utilizar o atributo **src** desta:

Criando os arquivos básicos

```
aula01 > index.html > ...
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>JavaScript Básico: Estrutura Básica</title>
7      <script src="script.js"></script>
8
9  </head>
10 <body>
11     <h1>Primeira estrutura básica</h1>
12 </body>
13 </html>
```

```
aula01 > script.js
1  console.log("Olá Mundooooo");
2
```



Console

O Console é um dos componentes que constituem as ferramentas de desenvolvimento do Google Chrome, assim como todo o navegador tem um Console e suas próprias ferramentas de desenvolvimento.

Variáveis e Tipos de Dados

Variáveis e Tipos de Dados

Na linguagem JavaScript, toda a vez que desejarmos declarar novas variáveis devemos utilizar a seguinte sintaxe:

var variavel = valor;

Variáveis e Tipos de Dados

Sabendo agora como declarar uma variável vamos declarar algumas variáveis em nosso arquivo JavaScript e apresentá-las no Console do navegador.

A linguagem JavaScript faz diferenciação entre caracteres maiúsculos e minúsculos, este conceito de análise tipográfica é chamado em inglês de **Case Sensitive**.

Variáveis e Tipos de Dados

aula01 > `exemplo_variavel.html` > html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Exemplo Variaveis</title>
7      <script src="variaveis.js"></script>
8  </head>
9  <body>
10     <h1>Exemplo Variáveis</h1>
11  </body>
12  </html>
```

aula01 > `JS variaveisjs` > ...

```
1  var nomeDaVariavel = "valor da variável";
2  var outraVariavel = 'valor da outra variável';
3  var variavelIdade = 22;
4  var variavelTenhoCabeloBranco = false;
5  var variavelIndefinida ;
6
7  console.log(nomeDaVariavel);
8  console.log(outraVariavel);
9  console.log(variavelIdade);
10 console.log(variavelTenhoCabeloBranco);
11 console.log(variavelIndefinida);
12 console.log(variavelNula);
```

▼ aula01

`exemplo_variavel.html`
`index.html`
`JS script.js`
`JS variaveisjs`

valor da variável	variaveis.js:7
valor da outra variável	variaveis.js:8
22	variaveis.js:9
false	variaveis.js:10
undefined	variaveis.js:11
✖ Uncaught ReferenceError: variavelNula is not defined at variaveis.js:12:13	variaveis.js:12

>

Tipos de dados

Ao declararmos as variáveis em nosso exemplo é possível notar que os valores atribuídos as variáveis são diferentes, alguns com aspas, alguns sem aspas, números, etc. Isto ocorre, pois os diferentes tipos de dados necessitam que a sintaxe de atribuição seja diferente. Para melhor entender o assunto vamos dar uma olhada nos tipos de dados existentes na linguagem JavaScript.

Tipos de dados - Primitivos

Numérico (Number)

O tipo de dado numérico na linguagem de programação JavaScript será sempre decimal, mesmo que você o declare como 2 a linguagem de programação sempre o entenderá como 2.0, mesmo que você não veja isto.

Em algumas linguagens de programação, denominadas linguagens tipadas, há como definir se uma variável do tipo numérico será inteira ou decimal, mas isto não é necessário no JavaScript.

Para atribuir este tipo de dado a uma variável não é necessário utilizar aspas.

Tipos de dados - Primitivos

Texto (String)

Tipo de dado texto são basicamente sequências de caracteres como por exemplo o nome de alguém ou uma mensagem de bom dia.

Para atribuir este tipo de dado a uma variável é necessário colocar o texto entre aspas ou aspas duplas, conforme o último exemplo.

Tipos de dados - Primitivos

Indefinido (Undefined)

Tipo de dado indefinido é o valor que uma variável recebe automaticamente, pela linguagem JavaScript, ao ser declarada sem que seja definido um valor para a mesma, ou seja uma variável não populada é preenchida com o tipo de dado indefinido, entenderemos melhor isto no decorrer do curso.

Em nosso exemplo foi declarado uma variável chamada *variavelIndefinida* que ao apresentarmos ela no console o resultado será ***undefined***.

Tipos de dados - Primitivos

Nulo (Null)

O tipo de dado nulo é similar ao indefinido, porém este realmente não existe.

Em nosso exemplo estamos tentando apresentar no console uma variável chamada *variavelNula* , porém esta variável não foi declarada previamente, então ela não existe e consequentemente é preenchida com o valor *null* .

Tipagem Dinâmica

Diferente de algumas linguagens de programação, onde ao definir uma variável é necessário especificar o tipo de dado desta, a linguagem JavaScript possui algo chamado tipagem dinâmica, significa que ao declarar uma variável não é necessário definir ou restringir o tipo de dado desta, pois ela irá se adaptar a cada novo tipo de dado atribuído a ela.