

```
40
41
42 $(function){cards();});
43 $(window).on('resize', function(){cards();});
44 function cards(){
45     var width = $(window).width();
46     if(width < 750){
47         cardssmallscreen();
48     }else{
49         cardsbigscreen();
50     }
51 }
52 function cardssmallscreen(){
    var cards = $('#card').length;
    var height = 0;
    var card2 = 1;
    while(card2 <= cards){
        $('#card'+card2).height(height);
        $('#card'+card2).width(750);
        card2++;
    }
}
```

# Aprendendo Javascript

Aprenda a programar usando a linguagem de programação mais popular do mundo

Iniciar

# Visão geral

Neste curso, você aprenderá os fundamentos da programação usando a linguagem Javascript. Desde os conceitos básicos até os tópicos avançados, você desenvolverá habilidades de programação essenciais para criar aplicações web interativas e dinâmicas.

01 Introdução



## Introdução ao Javascript

01 | Introdução ao Javascript

## O que é JavaScript?

JavaScript é uma linguagem de programação de alto nível, interpretada e multiparadigma (POO, funcional, imperativa etc..). Foi desenvolvida originalmente para ser executada nos navegadores web, permitindo a criação de sites dinâmicos e interativos. Com o passar do tempo, o JavaScript também passou a

ser utilizado em outras áreas, como o desenvolvimento de aplicações móveis e desktop.

## Por que aprender JavaScript?

Aprender JavaScript é essencial para quem deseja se tornar um "software developer". Com ele, você poderá criar sites interativos, adicionar efeitos visuais, validar formulários, criar aplicativos para celulares, computadores etc. Além disso, é uma linguagem de fácil aprendizado, pois possui uma sintaxe simples e intuitiva. Empresas como Google, Facebook, Amazon, Tesla possuem aplicações criadas com o uso de Javascript e possibilitam a evolução e disseminação da linguagem através de uma vasta comunidade.

## Sintaxe básica

Antes de começar a escrever código em JavaScript, é importante entender a sintaxe básica da linguagem. O JavaScript é case-sensitive, ou seja, diferencia letras maiúsculas de minúsculas. Além disso, cada comando ou instrução deve ser encerrado com ponto e vírgula (;).

## Variáveis e tipos de dados

Em JavaScript, podemos armazenar valores em variáveis. Uma variável é um espaço de memória reservado para guardar um valor. Existem vários tipos de dados em JavaScript, incluindo números, strings, booleanos, arrays e objetos. Vamos aprender como declarar e utilizar variáveis e explorar esses diferentes tipos de dados.

# Operadores

Assim como em outras linguagens de programação, o JavaScript possui operadores que nos permitem realizar operações aritméticas, comparar valores e atribuir valores a variáveis. Neste tópico, vamos explorar os operadores básicos do JavaScript e como utilizá-los corretamente.

## Estruturas de controle

As estruturas de controle nos permitem controlar o fluxo de execução do programa. No JavaScript, temos as estruturas condicionais (como if-else e switch) e as estruturas de repetição (como for e while). Vamos entender como utilizar essas estruturas para tomar decisões e repetir blocos de código.

## Funções

As funções são blocos de código que podem ser reutilizados em diferentes partes do programa. No JavaScript, podemos criar nossas próprias funções e utilizar as funções pré-definidas da linguagem. Vamos aprender como declarar e utilizar funções em JavaScript, passando parâmetros e retornando valores.

## Manipulação do DOM

O Document Object Model (DOM) é uma representação da estrutura em árvore dos elementos HTML de uma página web. No JavaScript, podemos utilizar o DOM para manipular os elementos HTML, alterando seu conteúdo, estilo e

comportamento. Vamos explorar como acessar e manipular os elementos HTML utilizando JavaScript.

## Considerações finais

Neste tópico, introduzimos os conceitos básicos do JavaScript, desde sua sintaxe até a manipulação do DOM. Ao dominar esses princípios fundamentais, você estará pronto para desenvolver aplicações web mais complexas e explorar todo o potencial do JavaScript.

Continue estudando e praticando para aprimorar suas habilidades em JavaScript e aprofundar seu conhecimento na linguagem.

[Conclusão](#) [Introdução ao Javascript](#)

O curso de programação usando a linguagem Javascript é uma introdução abrangente a essa linguagem de programação popular. Ao longo do curso, os alunos aprenderão os conceitos básicos do Javascript, incluindo a sintaxe, variáveis e tipos de dados. Além disso, os alunos serão introduzidos às estruturas de controle e repetição do Javascript, permitindo-lhes criar programas complexos e funcionais.



# Trabalhando com variáveis e tipos de dados

02 | Trabalhando com variáveis e tipos de dados

Nesta seção do curso, vamos explorar o conceito de variáveis e tipos de dados em JavaScript. As variáveis são como "containers" que armazenam valores, permitindo que possamos manipulá-los e utilizá-los em nossos programas.

## Variáveis em JavaScript

Uma variável em JavaScript é declarada utilizando a palavra-chave `var`, seguida pelo nome da variável desejada. Por exemplo:

```
var nome;
```

Aqui, `nome` é o nome da variável que declaramos. No entanto, uma variável não terá um valor até que atribuamos um a ela. Podemos atribuir um valor usando o operador de atribuição `=`. Por exemplo:

```
nome = "João";
```

Agora, a variável `nome` contém o valor "João". Podemos também declarar e atribuir um valor à variável ao mesmo tempo:

```
var idade = 25;
```

Aqui, a variável `idade` é declarada e inicializada com o valor 25.

## Tipos de dados em JavaScript

JavaScript suporta diversos tipos de dados, incluindo:

- **Números:** utilizados para representar valores numéricos. Podem ser inteiros ou decimais.
- **Strings:** utilizadas para representar texto. Podem ser declaradas usando aspas simples (") ou aspas duplas (").
- **Booleanos:** representam valores lógicos, verdadeiro (true) ou falso (false).
- **Arrays:** utilizados para armazenar múltiplos valores em uma única variável.
- **Objetos:** representam um conjunto de propriedades e métodos relacionados.

- Null: representa a ausência intencional de qualquer objeto ou valor.
- Undefined: indica que uma variável foi declarada, mas não tem um valor atribuído a ela.

Para determinar o tipo de dados armazenado em uma variável, podemos utilizar o operador `typeof`. Por exemplo:

```
var numero = 10;  
console.log(typeof numero); // irá imprimir "number"  
  
var texto = "Olá";  
console.log(typeof texto); // irá imprimir "string"  
  
var verdadeiro = true;  
console.log(typeof verdadeiro); // irá imprimir "boolean"
```

É importante entender e utilizar os tipos de dados corretos em cada situação, pois isso nos ajudará a escrever programas mais eficientes.

## Conversão de tipos de dados

Em JavaScript, também é possível converter um tipo de dado para outro. Podemos converter um número para uma string, uma string para um número e assim por diante.

Para converter um número em uma string, podemos utilizar o método `toString()`, por exemplo:

```
var numero = 10;  
var numeroString = numero.toString();  
console.log(typeof numeroString); // irá imprimir "string"
```



Da mesma forma, podemos converter uma string em um número utilizando a função `parseInt()` ou `parseFloat()`, como no exemplo a seguir:

```
var numeroString = "20";  
var numero = parseInt(numeroString);  
console.log(typeof numero); // irá imprimir "number"
```

Ao trabalhar com variáveis e tipos de dados em JavaScript, é importante entender seus conceitos e como utilizá-los corretamente para alcançar os resultados desejados em nossos programas.

#### Conclusão Trabalhando com variáveis e tipos de dados

Trabalhar com variáveis e tipos de dados é uma parte essencial da programação em Javascript. Neste tópico, os alunos aprendem como declarar variáveis, atribuir valores e realizar operações em diferentes tipos de dados, como números, strings e booleanos. Eles também são introduzidos aos conceitos de conversão de tipos e escopo de variáveis. Compreender esses conceitos é crucial para a construção de programas funcionais e eficientes em Javascript.



# Estruturas de controle e repetição

## 03 | Estruturas de controle e repetição

As estruturas de controle e repetição são fundamentais para o desenvolvimento de programas em linguagens de programação como o Javascript. Essas estruturas permitem que um conjunto de instruções seja executado de forma condicional ou repetitiva, possibilitando o controle do fluxo de execução do programa.

## Estruturas de controle condicional

As estruturas de controle condicional permitem que um bloco de código seja executado apenas se uma condição for verdadeira. Em Javascript, as estruturas de controle condicional mais comuns são:

Estrutura `if`

A estrutura `if` permite a execução de um bloco de código caso uma condição seja verdadeira. Veja o exemplo abaixo:

```
if (condicao) {  
    // bloco de código a ser executado se a condição for verdadeira  
}
```

## Estrutura `if-else`

A estrutura `if-else` possibilita a execução de um bloco de código caso uma condição seja verdadeira, e outro bloco de código caso a condição seja falsa. Veja o exemplo abaixo:

```
if (condicao) {  
    // bloco de código a ser executado se a condição for verdadeira } else {  
    // bloco de código a ser executado se a condição for falsa  
}
```

## Estrutura `switch case`

A estrutura `switch case` permite a execução de diferentes blocos de código dependendo do valor de uma expressão. É uma alternativa ao uso excessivo de várias estruturas `if-else`. Veja o exemplo abaixo:

```
switch (expressao) {  
    case valor1:  
        // bloco de código a ser executado se expressao for igual a valor1 break;  
    case valor2:  
        // bloco de código a ser executado se expressao for igual a valor2 break;  
    default:  
        // bloco de código a ser executado se expressao não corresponder a nen break;  
}
```

# Estruturas de repetição

As estruturas de repetição permitem que um conjunto de instruções seja executado várias vezes enquanto uma condição for verdadeira. Em Javascript, as estruturas de repetição mais utilizadas são:

## Estrutura `while`

A estrutura `while` executa um bloco de código enquanto uma condição for verdadeira. Veja o exemplo abaixo:

```
while (condicao) {  
  // bloco de código a ser repetido enquanto a condição for verdadeira  
}
```

## Estrutura `do-while`

A estrutura `do-while` executa um bloco de código pelo menos uma vez, e em seguida verifica se uma condição é verdadeira para repetir o bloco. Veja o exemplo abaixo:

```
do {  
  // bloco de código a ser repetido  
} while (condicao);
```

## Estrutura `for`

A estrutura `for` é uma forma mais compacta de criar uma estrutura de repetição. Ela permite especificar a inicialização, a condição de continuação e o incremento/decremento em uma única linha. Veja o exemplo abaixo:

```
for (inicializacao; condicao; incremento) {  
    // bloco de código a ser repetido  
}
```

### Conclusão Estruturas de controle e repetição

Estruturas de controle e repetição são tópicos essenciais a qualquer linguagem de programação. Neste tópico, os alunos aprendem como utilizar as principais estruturas em Javascript.



## Exercícios Práticos

Vamos colocar seus conhecimentos em prática

## 04 | Exercícios Práticos

Nesta lição, colocaremos a teoria em prática por meio de atividades práticas. Clique nos itens abaixo para conferir cada exercício e desenvolver habilidades práticas que o ajudarão a ter sucesso na disciplina.

### 1- Primeiros passos em Javascript

Neste exercício, você vai começar a escrever código em Javascript. Crie um arquivo HTML com um script em Javascript que exiba a mensagem 'Olá Mundo!' no console do navegador.

### 2- Manipulando variáveis

Neste exercício, você vai praticar a manipulação de variáveis em Javascript. Crie um arquivo HTML com um script em Javascript que

declare duas variáveis: 'nome' e 'idade'. Atribua um valor para cada variável e exiba a mensagem 'Olá, {nome}! Você tem {idade} anos.' no console do navegador, substituindo {nome} pelo valor da variável 'nome' e {idade} pelo valor da variável 'idade'.

### 3 - Calculadora de média



Neste exercício, você vai criar uma calculadora de média em Javascript. Crie um arquivo HTML com um script em Javascript que peça ao usuário para digitar as notas de três provas e calcule a média. Em seguida, exiba a média no console do navegador. Utilize a estrutura de controle 'if' para verificar se o aluno foi aprovado ou reprovado, considerando a média mínima de 7. Se o aluno foi aprovado, exiba a mensagem 'Parabéns, você foi aprovado!' no console. Caso contrário, exiba a mensagem 'Infelizmente, você foi reprovado.'.

Correção exercícios



## 1- Resolução:



Basta salvar esse código em um arquivo com extensão ".html" (por exemplo, "index.html") e abrir o arquivo em um navegador da web. Ao abrir o console do navegador (geralmente pressionando F12 e indo para a guia "Console"), você verá a mensagem "Olá Mundo!" impressa no console.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exercício - Olá Mundo!</title>
</head>
<body>

  <!-- Seu script em JavaScript -->
  <script>
    // O código a seguir exibirá a mensagem 'Olá Mundo!' no console do navegador
    console.log('Olá Mundo!');
  </script>

</body>
</html>
```



## 2- Resolução



Basta salvar esse código em um arquivo com extensão ".html" (por exemplo, "index.html") e abrir o arquivo em um navegador da web. Ao abrir o console do navegador (geralmente pressionando F12 e indo para a guia "Console").

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Manipulação de Variáveis</title>
</head>
<body>
  <script>
    // Declaração das variáveis
    var nome = "SeuNome"; // Substitua "SeuNome" pelo valor desejado
    var idade = 25; // Substitua 25 pelo valor desejado

    // Exibição da mensagem no console do navegador
    console.log("Olá, " + nome + "! Você tem " + idade + " anos.");
  </script>
</body>
</html>
```

### 3- Resolução

Basta salvar esse código em um arquivo com extensão ".html" (por exemplo, "index.html") e abrir o arquivo em um navegador da web. Ao abrir o console do navegador (geralmente pressionando F12 e indo para a guia "Console").

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Calculadora de Média</title>
</head>
<body>
  <script>
    // Solicitar ao usuário para digitar as notas das três provas
    var nota1 = parseFloat(prompt("Digite a nota da primeira prova:"));
    var nota2 = parseFloat(prompt("Digite a nota da segunda prova:"));
    var nota3 = parseFloat(prompt("Digite a nota da terceira prova:"));

    // Calcular a média
    var media = (nota1 + nota2 + nota3) / 3;

    // Exibir a média no console do navegador
    console.log("Média: " + media);

    // Verificar se o aluno foi aprovado ou reprovado
    if (media >= 7) {
      console.log("Parabéns, você foi aprovado!");
    } else {
      console.log("Infelizmente, você foi reprovado.");
    }
  </script>
</body>
</html>
```

# Resumo

Vamos revisar o que acabamos de ver até agora

05 | Resumo

- ✓ O curso de programação usando a linguagem Javascript é uma introdução abrangente a essa linguagem de programação popular. Ao longo do curso, os alunos aprenderão os conceitos básicos do Javascript, incluindo a sintaxe, variáveis e tipos de dados. Além disso, os alunos serão introduzidos às estruturas de controle e repetição do Javascript, permitindo-lhes criar programas complexos e funcionais. Com este curso, os alunos estarão preparados para iniciar sua jornada na programação web usando o Javascript.
- ✓ A introdução ao Javascript é o primeiro passo fundamental para aqueles que desejam se aventurar no mundo da programação. Neste tópico, os alunos aprendem sobre a história e os principais conceitos do Javascript. Eles também são apresentados à sintaxe básica do Javascript, incluindo declaração de variáveis, operadores e estruturas de controle. Ao final deste tópico, os alunos estarão familiarizados com os fundamentos do Javascript e prontos para mergulhar em tópicos mais avançados.

- ✓ Trabalhar com variáveis e tipos de dados é uma parte essencial da programação em Javascript. Neste tópico, os alunos aprendem como declarar variáveis, atribuir valores e realizar operações em diferentes tipos de dados, como números, strings e booleanos. Eles também são introduzidos aos conceitos de conversão de tipos e escopo de variáveis. Compreender esses conceitos é crucial para a construção de programas funcionais e eficientes em Javascript.
- ✓ As estruturas de controle e repetição são ferramentas poderosas para manipular o fluxo de um programa em Javascript. Neste tópico, os alunos aprendem como usar declarações condicionais, como if-else e switch, para criar lógica de ramificação em seus programas. Além disso, eles também são apresentados a estruturas de repetição, como while e for, que permitem executar um bloco de código várias vezes. Dominar essas estruturas é essencial para escrever programas eficientes e resolver problemas complexos em Javascript.



# Questionário

Verifique seu conhecimento respondendo a algumas perguntas

Pergunta 1/6

1- O que é Javascript?

- ☐ Um framework de desenvolvimento web
  - ☐ Uma linguagem de programação
  - ☐ Um sistema operacional
- 

Pergunta 2/6

2- Qual é o tipo de dado usado para representar números inteiros em Javascript?

- ☐ Boolean
  - ☐ String
  - ☐ Number
- 

Pergunta 3/6

3- Qual é a estrutura de controle que permite executar um bloco de código repetidamente até que uma condição seja satisfeita?

- ☐ If-else
- ☐ Switch
- ☐ While

Pergunta 4/6

4- O que é uma variável em Javascript?

- ☐ Um objeto complexo
  - ☐ Uma função
  - ☐ Um local de armazenamento para dados
- 

Pergunta 5/6

5- Qual é a função da palavra-chave 'let' em Javascript?

- ☐ Declarar uma variável
  - ☐ Definir uma função
  - ☐ Iniciar um loop
- 

Pergunta 6/6

6- Qual é o operador usado para concatenar strings em Javascript?

- ☐ +
  - ☐ -
  - ☐ \*
- 

Gabarito: 1- B 2- C 3-C 4-C 5-A 6-A