



Operadores

Nesta aula, exploraremos os operadores em JavaScript, que são fundamentais para a manipulação e controle de dados no desenvolvimento de software. Iniciaremos com os operadores aritméticos, que permitem realizar cálculos matemáticos básicos e avançados diretamente no código. Em seguida, abordaremos os operadores de string, essenciais para a manipulação de dados textuais, e os operadores lógicos, que possibilitam a criação de condições e fluxos de controle robustos. Por fim, veremos os operadores de atribuição, que combinam operações matemáticas com a atualização de variáveis, tornando o código mais conciso e eficiente.

Operadores Aritméticos

Nesta primeira parte da nossa aula, vamos explorar o tema “Operadores Aritméticos” no contexto do JavaScript. Este é um tópico fundamental para entender como realizar operações matemáticas dentro de um código. Os operadores aritméticos são essenciais, pois permitem a realização de cálculos diretamente no código, o que é uma habilidade crucial para qualquer desenvolvedor.

Os operadores aritméticos incluem as operações básicas da matemática: adição, subtração, multiplicação e divisão. Esses operadores são amplamente utilizados e fáceis de aplicar em variáveis numéricas. Por exemplo, se temos duas variáveis, A e B, com valores 10 e 5 respectivamente, podemos somá-las utilizando o operador de adição (+), subtraí-las com o operador de subtração (-), multiplicá-las com o operador de multiplicação (*) e dividi-las com o operador de divisão (/). A simplicidade desses operadores é que, ao inseri-los no código, o JavaScript automaticamente executa a operação desejada.

Além desses operadores básicos, o JavaScript também oferece operadores aritméticos mais avançados. O operador exponencial (**`^`**), **por exemplo, eleva um número a uma determinada potência, como `10^5`**, que resulta em 100000. Outro operador útil é o de módulo (**`%`**), que calcula o resto de uma divisão entre dois números. Por exemplo, `10 % 5` retorna 0, já que 10 é divisível por 5 sem deixar resto.


Dois outros operadores que vale mencionar são o de incremento (**`++`**) e o de decremento (**`--`**). Eles ajustam o valor de uma variável somando ou subtraindo 1, respectivamente. Por exemplo, se A inicialmente vale 10, após aplicar o operador de incremento, seu valor será 11. De forma semelhante, se B vale 5, após aplicar o operador de decremento, seu valor será 4.

Compreender e saber aplicar esses operadores aritméticos é essencial para o desenvolvimento em JavaScript, pois eles são ferramentas básicas e poderosas para manipulação de dados e controle de fluxo dentro do código. A prática com esses operadores facilitará muito o entendimento de operações matemáticas dentro de seus projetos de programação.

Operadores de String

Na continuação da nossa aula sobre operadores em JavaScript, vamos explorar o uso dos operadores de string. Esses operadores são essenciais quando trabalhamos com dados textuais, permitindo concatenar, manipular e comparar strings de maneira eficiente no desenvolvimento de software.

Um dos operadores mais comuns é o operador de adição (**`+`**), que em strings é utilizado para concatenar, ou seja, unir duas ou mais strings em uma única. Por exemplo, se temos duas variáveis, `primeiroNome = "João"` e `sobrenome = "Silva"`, podemos utilizar o operador `+` para criar uma nova variável `nomeCompleto = primeiroNome + " " + sobrenome`, resultando em `"João Silva"`. Esse operador é bastante intuitivo e fácil de aplicar em diversas situações no desenvolvimento de aplicações web.

Além do operador de adição, temos o operador `+=`, que também pode ser utilizado para concatenar strings. Por exemplo, se inicialmente `'saudacao = "Olá"'`, podemos  usar `'saudacao += " Mundo!"'` para alterar o valor de `'saudacao'` para `"Olá Mundo!"`. Esse operador é útil quando precisamos acrescentar texto a uma string existente de forma prática e eficiente.

Outro operador relevante é o operador de comparação (`==`), que verifica se duas strings possuem o mesmo conteúdo. Por exemplo, `'if (nome1 == nome2)'` verifica se `'nome1'` e `'nome2'` são iguais. Esse operador é amplamente utilizado em condições e estruturas de controle, onde a lógica do código depende da comparação entre strings.


Para ilustrar a aplicação desses operadores, considere o seguinte exemplo prático: ao criar uma aplicação que solicite o nome do usuário e verifique se ele já está cadastrado, podemos usar o operador de adição para construir uma mensagem de boas-vindas personalizada, e o operador de comparação para verificar se o nome inserido corresponde a algum nome previamente armazenado no sistema.

Esses operadores de string são fundamentais para o desenvolvimento em JavaScript, oferecendo flexibilidade na manipulação de dados textuais. Com o tempo e prática, o uso desses operadores se tornará natural, permitindo que você manipule strings de forma eficaz e clara em seus projetos.

Operadores Lógicos

Dando continuidade à nossa aula sobre operadores em JavaScript, vamos explorar o uso dos operadores lógicos. Esses operadores são essenciais para a construção de condições e fluxos de controle em um programa, permitindo tomar decisões com base em várias condições.

O primeiro operador lógico que vamos discutir é o operador AND, também conhecido como operador E, representado em JavaScript pelos símbolos `'&&'`. Ele retorna verdadeiro somente se ambas as condições comparadas forem verdadeiras. Por exemplo, considere a função `'selecionarFrutas'`, que recebe duas frutas como

parâmetros e verifica se ambas são as frutas desejadas, como “maçã” e “banana”. Se as duas frutas corresponderem, o operador AND retornará verdadeiro; c.  contrário, retornará falso. Este comportamento é fundamental para criar validações robustas em seu código.

O segundo operador que vamos abordar é o operador OR, representado pelos símbolos ||. Diferente do AND, o operador OR retorna verdadeiro se pelo menos uma das condições for verdadeira. Por exemplo, se desejamos saber se uma das frutas recebidas é “morango” ou “mamão”, utilizamos o operador OR para validar essa condição. Mesmo que apenas uma das frutas seja desejada, o operador OR retornará verdadeiro, mostrando sua flexibilidade na criação de condições menos restritivas.

Para ilustrar, vamos implementar uma função que utiliza ambos os operadores. Primeiramente, utilizaremos o operador AND em uma função que verifica se as frutas “maçã” e “banana” são as desejadas. Depois, aplicamos o operador OR em outra função para verificar se pelo menos uma das frutas, como “morango” ou “mamão”, corresponde às frutas desejadas. Ao testar essas funções com diferentes entradas, como “uva” e “pera” ou “maçã” e “uva”, podemos observar como cada operador lógico se comporta e como eles podem ser usados de maneira eficiente para controlar o fluxo do código.

Compreender e aplicar os operadores lógicos é essencial para qualquer desenvolvedor, pois eles permitem criar código mais inteligente e responsivo a diferentes situações. Sempre que trabalhar com condições em JavaScript, lembre-se de como os operadores AND e OR operam e aplique-os conforme necessário para garantir que seu código funcione conforme esperado.

Operadores de Atribuição

Na parte final da nossa aula sobre operadores em JavaScript, vamos explorar os operadores de atribuição. Eles são amplamente utilizados para atualizar o valor de uma variável, aplicando operações matemáticas e lógicas de forma eficiente. O

operador de atribuição mais comum é o sinal de igual (=), que atribui diretamente um valor a uma variável. Por exemplo, `x = 5` atribui o valor 5 na variável `x`.




Além do operador básico de atribuição, existem outros operadores que combinam atribuição com operações matemáticas. Por exemplo, `+=` soma um valor à variável e, em seguida, armazena o resultado na mesma variável. Se `x = 5`, então `x += 3` resultará em `x = 8`. Da mesma forma, temos operadores como `-=`, `*=`, `/=`, e `%=` que subtraem, multiplicam, dividem e calculam o módulo do valor da variável, respectivamente, e atribuem o resultado à própria variável. Esses operadores tornam o código mais conciso e fácil de ler, especialmente quando precisamos realizar operações repetitivas.

Para ilustrar o uso desses operadores, vamos criar uma variável `x` com valor 5 e aplicar uma série de operações utilizando os operadores de atribuição. Por exemplo, `x += 3` somará 3 ao valor de `x`, resultando em 8. Em seguida, podemos multiplicar `x` por uma outra variável `y`, e assim sucessivamente, aplicando operações de subtração, divisão e módulo. Cada operação modificará o valor de `x`, que será atualizado automaticamente.

Esses operadores de atribuição são essenciais para simplificar operações que seriam mais longas se feitas manualmente. Ao utilizar esses operadores, você pode tornar seu código mais eficiente e legível, facilitando a manutenção e o entendimento do fluxo lógico das operações realizadas.

Nesta aula, abordamos de forma abrangente os diferentes tipos de operadores em JavaScript, desde os aritméticos, passando pelos operadores de string e lógicos, até os operadores de atribuição. Compreender e aplicar esses operadores é essencial para o desenvolvimento de software eficiente e robusto.

Conteúdo Bônus

Para auxiliar nos estudos sobre operadores, recomendo assistir ao vídeo "Aprenda a Programar - #03 Operadores Aritméticos", que está disponível no canal  Andrade no YouTube.

Referência Bibliográfica

ASCENCIO, A. F. G.; CAMPOS, E. A. V. de. Fundamentos da programação de computadores. 3.ed. Pearson: 2012.

BRAGA, P. H. Teste de software. Pearson: 2016.

GALLOTTI, G. M. A. (Org.). Arquitetura de software. Pearson: 2017.

GALLOTTI, G. M. A. Qualidade de software. Pearson: 2015.

MEDEIROS, E. Desenvolvendo software com UML 2.0 definitivo. Pearson: 2004.

PFLEEGER, S. L. Engenharia de software: teoria e prática. 2.ed. Pearson: 2003.


SOMMERVILLE, I. Engenharia de software. 10.ed. Pearson: 2019.

Prática Integradora Desenvolvimento de Software

Atividade Prática 9 - Explorando Operadores em JavaScript

Objetivos:

- Compreender e aplicar operadores aritméticos, de string, lógicos e de atribuição em JavaScript.
- Desenvolver habilidades práticas na manipulação de variáveis e controle de fluxo utilizando operadores.

- Implementar e testar funções que utilizem diferentes tipos de operadores para realizar operações matemáticas, manipular strings e controlar o fluxo do programa. 

Materiais, Métodos e Ferramentas:

- Computador com um editor de código (Visual Studio Code, Sublime Text, etc.)
- Navegador web (para executar e testar o código JavaScript)
- Acesso à documentação do JavaScript (MDN Web Docs ou similar)

Atividade Prática

Primeiramente, leia atentamente o texto a seguir:

Nesta atividade, você aplicará e explorará diferentes tipos de operadores em JavaScript. A prática envolverá a criação de funções que utilizam operadores aritméticos, de string, lógicos e de atribuição. Você será desafiado a implementar e testar funções para manipulação de dados e controle de fluxo, aplicando os conceitos aprendidos na aula.

Agora, vamos praticar!

PASSO A PASSO DETALHADO DA ATIVIDADE:

- **Operadores Aritméticos:**
 - Crie um arquivo HTML e inclua um arquivo JavaScript.
 - No arquivo JavaScript, defina uma função chamada `operacoesAritmeticas` que recebe dois números e realiza as seguintes operações:

Soma



Subtração

Multiplicação

Divisão

Exponenciação

Módulo

Incremento e Decremento (de uma variável inicial)

- A função deve retornar um objeto contendo os resultados dessas operações e exibi-los no console.

- **Operadores de String:**


- No mesmo arquivo JavaScript, crie uma função chamada `manipularStrings` que recebe duas strings e realiza as seguintes operações:

Concatenação das duas strings

Comparação para verificar se as duas strings são iguais

Adição de um texto adicional a uma das strings

- A função deve retornar um objeto contendo os resultados dessas operações e exibi-los no console.
- **Operadores Lógicos:**

- Crie uma função chamada verificarFrutas que recebe duas frutas e utiliza operadores lógicos para verificar as seguintes condições: 

Se ambas as frutas são “maçã” e “banana” (utilizando o operador AND).

Se pelo menos uma das frutas é “morango” ou “mamão” (utilizando o operador OR).

- A função deve exibir no console se cada condição foi atendida ou não.

- **Operadores de Atribuição:**

- Defina uma variável valor com o valor inicial de 10.
- Aplique uma série de operações utilizando operadores de atribuição (+, -, *, /, %) e exiba o resultado da variável valor após cada operação no console.
- Crie uma função chamada operacoesAtribuicao que faz o seguinte:

Incrementa valor em 5.

Subtrai 2 de valor.

Multiplica valor por 3.

Divide valor por 4.

Calcula o módulo de valor com 7.

- A função deve retornar o valor final de valor e exibi-lo no console.



- **Operadores Aritméticos:** A função `operacoesAritmeticas` deve corretamente realizar e exibir os resultados de todas as operações aritméticas.
- **Operadores de String:** A função `manipularStrings` deve realizar a concatenação, comparação e adição de texto corretamente e exibir os resultados esperados.
- **Operadores Lógicos:** A função `verificarFrutas` deve corretamente avaliar e exibir os resultados das condições lógicas.
- **Operadores de Atribuição:** A função `operacoesAtribuicao` deve aplicar corretamente os operadores de atribuição e exibir o valor atualizado da variável valor após cada operação.

Certifique-se de que todas as funções sejam testadas e que os resultados estejam corretos de acordo com as especificações fornecidas.

Ir para exercício