

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARAÍBA

JavaScript
Operadores e Estrutura de controle de fluxo

JavaScript

Operadores e Estrutura de controle de fluxo

Prof. Mr. Fabio Abrantes Diniz

Operadores

Operadores

- Possui tanto operadores **binários** quanto **unários** e um operador **ternário**

- **Sintaxe Unário**

- ✓ operador operando: `x++`
- ✓ operando operador: `++x`

- **Sintaxe Binário**

- ✓ operando1 operador operando2
- Ex: `x*y`

- **Sintaxe ternário**

✓ condicao ? valor1 : valor2

– Ex: let status = idade >= 18 ? "adulto" : "menor de idade"



Operadores

Operadores

- Principais grupos de operadores
 - Atribuição
 - Aritmético
 - Comparação
 - Lógico

- Operadores bit a bit
- Operadores relacionais
- Operador ternário
- Operadores Unários



Operadores de atribuição •

Atribui um valor ao operando à sua esquerda •

Usado para atribuir valores a variáveis.

✓ O operador de atribuição básico é o igual (=)



Operadores Aritméticos • Os **operandos** tem que ser **numéricos** (sejam literais ou variáveis)

- Retornam um **único** valor **númeroico**.



Operadores de comparações

- Compara seus operandos e retorna um valor lógico ▪ Que pode ser **true** ou **false**
- Os operandos podem ser **numéricos, strings, lógicos** ou **objetos**.





Operadores bit a bit lógicos

- Realizam operações **bit a bit**,
- Comparam os bits dos operandos um a um.
- ✓ Antes da operação, os números são convertidos para **inteiros de 32 bits**.





Operadores

Lógicos

- Utilizados **tipicamente** com valores **booleanos** que retornam um valor booleano
 - **true** ou **false**





Lógicos

Operadores



Operadores

Lógicos

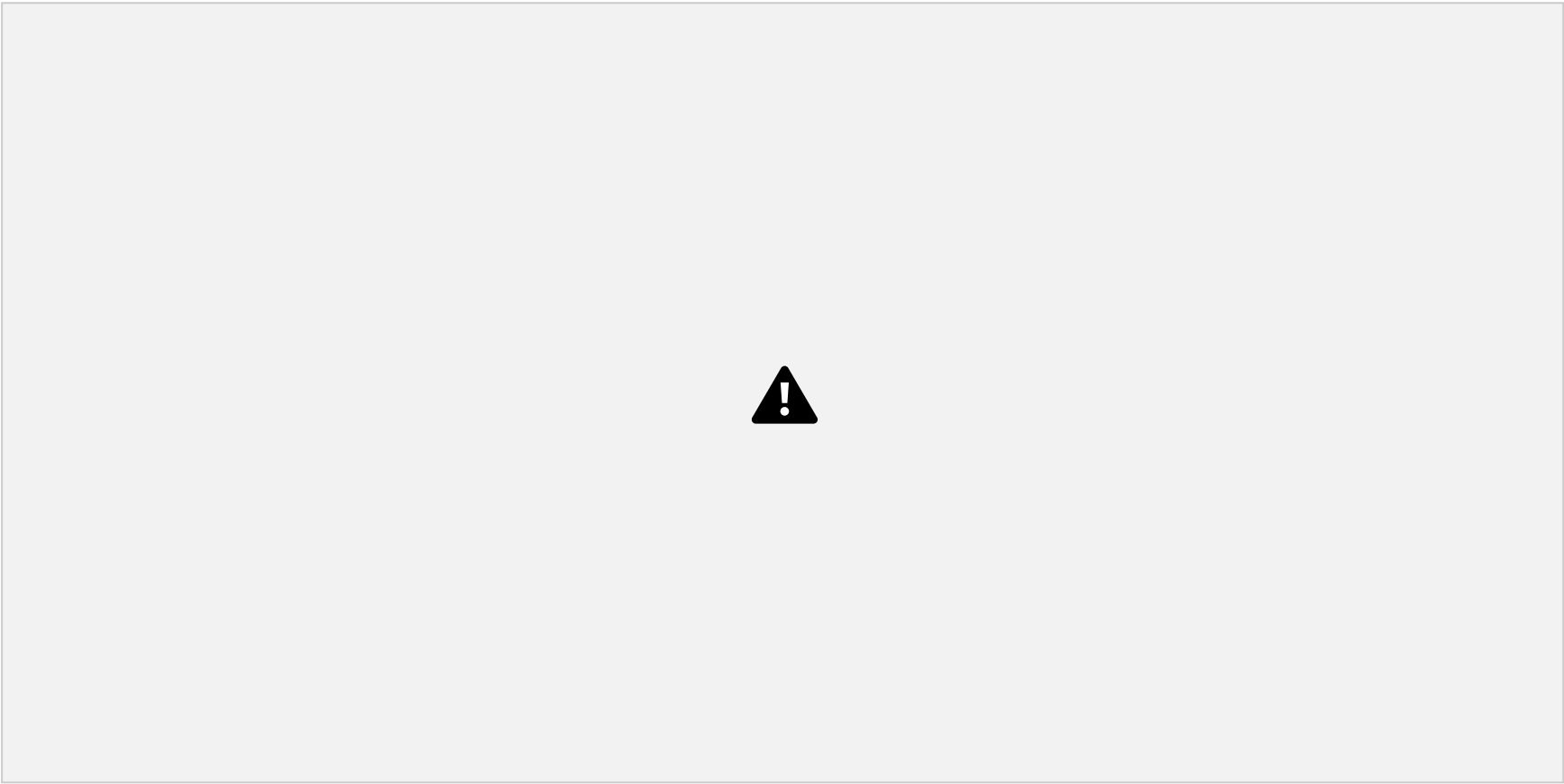




Operadores

Lógicos

- Caso os operandos **não** sejam **booleanos** ■
Possam **retornar** um valor **não-booleano**



Operadores Lógicos

- **Se os valores dos operandos não forem booleanos**
 - Retornam o valor de um dos operandos especificados
 - ✓ Se for false entre eles. Retorna o operando da **esquerda** ✓ Se for verdadeiro, retorna o operando da **direita**





Operadores

Lógicos

- **Avaliação de curto-circuito**
 - As expressões lógicas são avaliadas da esquerda para a direita



- Exemplos



Operador Ternário

- Usa **três operandos** – é o único operador ternário em JavaScript.
 - condição ? valor_se_verdadeiro : valor_se_falso



Operadores Relacionais especiais

- Compara seus operandos e retorna um valor **booleano**
 - São o **in** e **instanceof**





Operadores

unários

- Uma operação com

apenas um
operando.

- `typeof`

- ✓ Retorna o tipo do
operando





Operadores unários

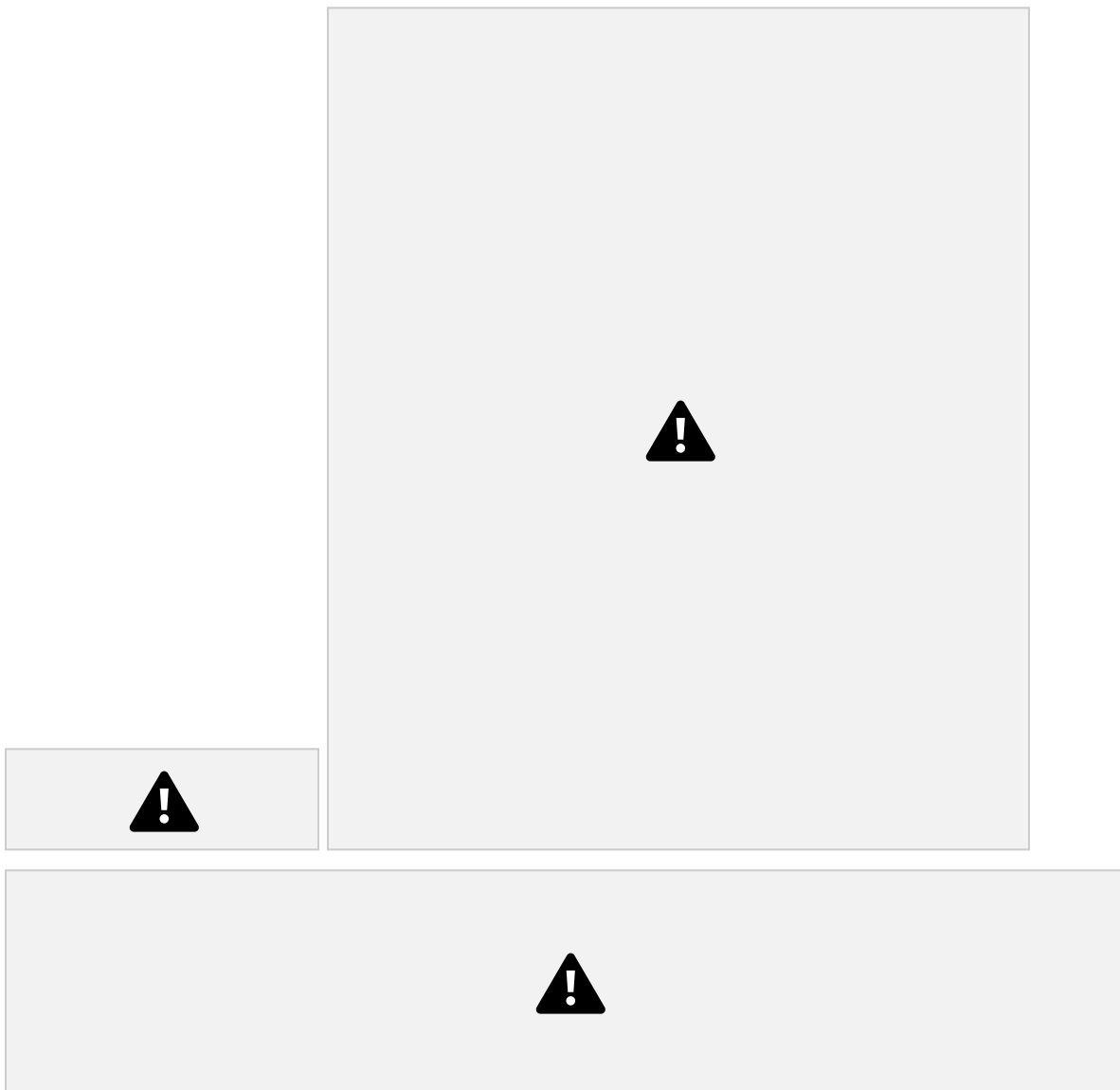
- **delete**:
 - Apaga um **objeto**, uma **propriedade** de um objeto
 - ✓ Se for bem-sucedido,
 - define a propriedade ou elemento para **undefined**.
 - ✓ Retorna **verdadeiro** se a operação for possível;





Operadores unários

- **delete:**
 - Apaga um elemento no índice especificado de um array;



Operadores unários

- Spread

- é usado para **espalhar elementos** de arrays, objetos ou até strings em **outra estrutura**





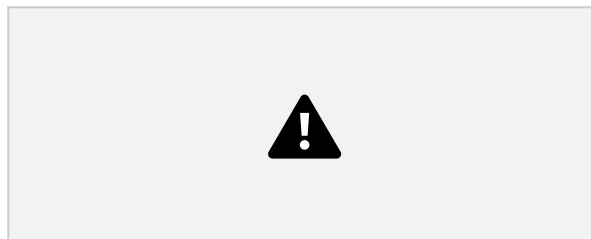
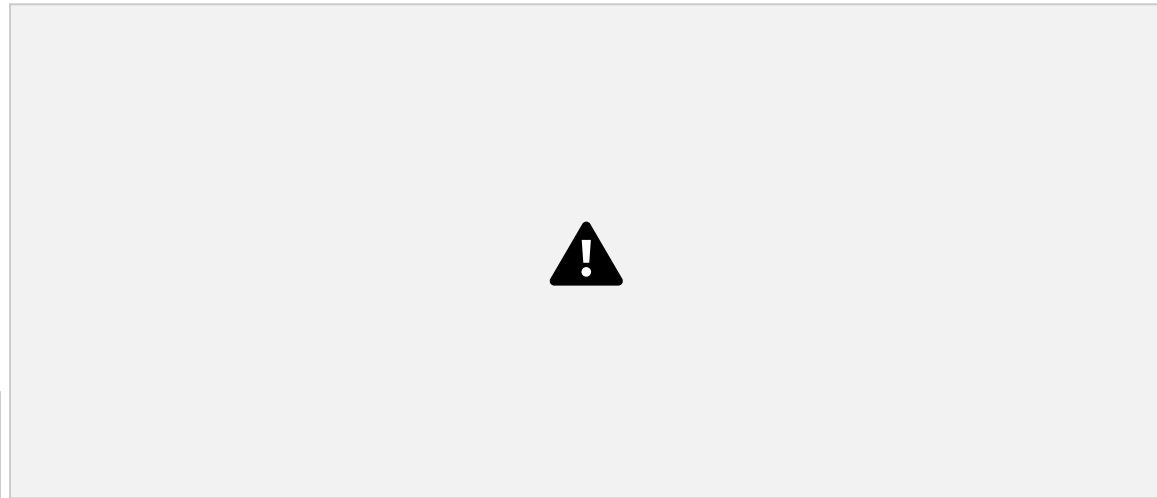
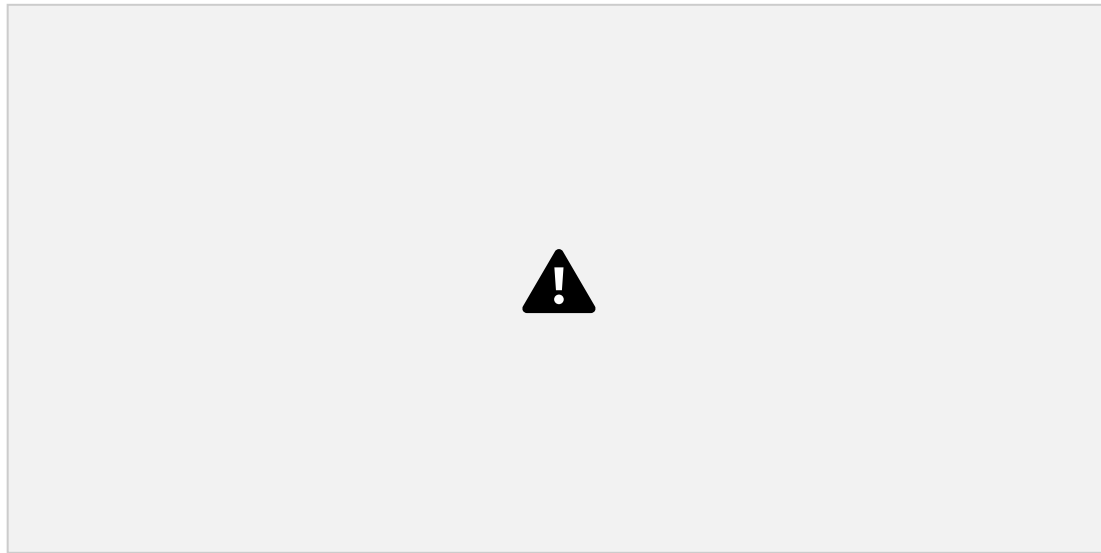
Operadores unários

- **Spread**



Operadores unários

- **Spread**



palavras-chaves

especiais



- **new**

- criar uma instância de um tipo de objeto definido pelo usuário
 - ✓ ou de um dos tipos de objeto predefinidos: Array, Date, Object, .

- **super**

- Utilizada para chamar a função pai de um objeto

- **this**

- se referir ao objeto atual. ou seja, quem está chamando a função ou método.



Estrutura de controle de fluxo

- If e else





<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/if...else>



Estrutura de controle de fluxo • If e else

- Observação importantíssimo:
 - ✓ Se o valor da condição do **if for**:
 - **undefined, null, 0, NaN**, ou uma string **vazia** ("") o resultado é **falsy**



<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/if...else>



Estrutura de controle de fluxo

- switch
 - compara valor e tipo (===)



<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/switch>

Estrutura de controle de repetição

- for

- Cria um loop que consiste em três expressões opcionais



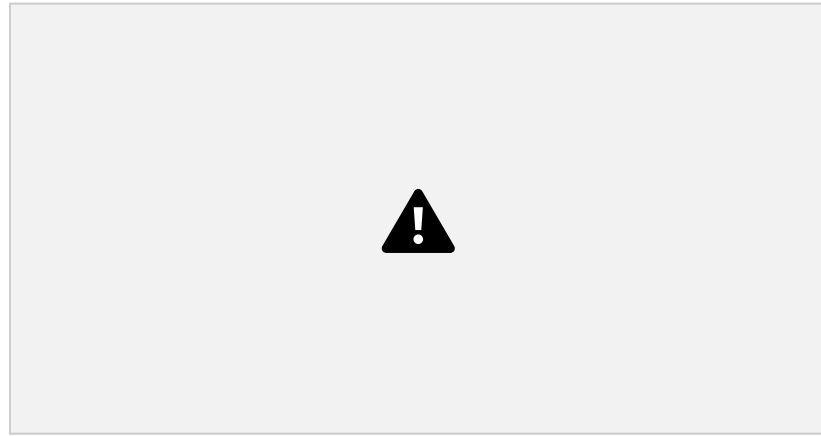


<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/for>

Estrutura de controle de repetição

- `for..of`
 - Para array e strings (objetos iteráveis)





<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/for..of>

Estrutura de controle de repetição

- `for..in`
 - Interage sobre **propriedades** de um objeto



<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/for...in>



Instruções

(while)



- Repete
 - Enquanto a condição de teste for avaliada como verdadeira.



Instruções

(while e do while)



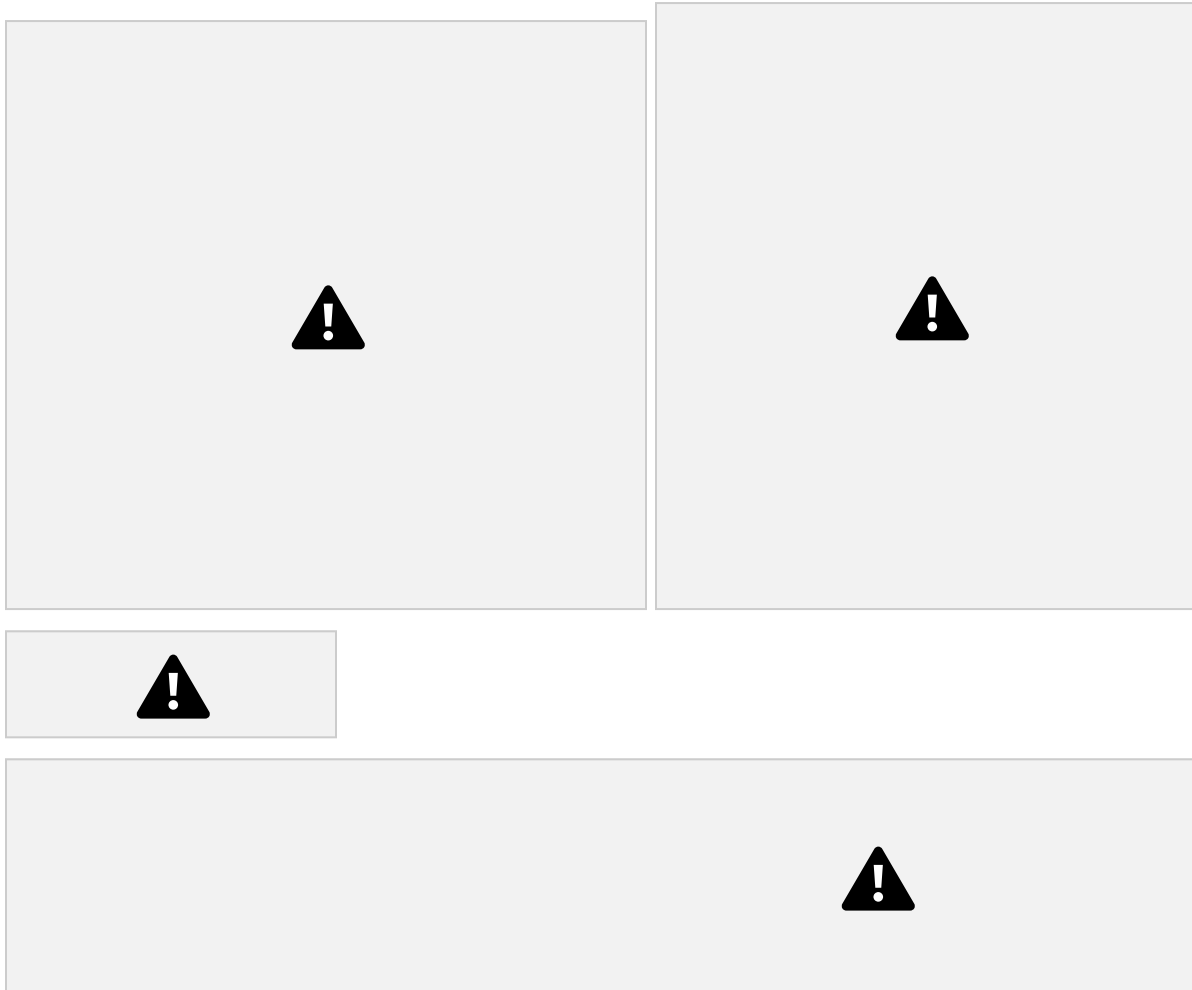
- Uma declaração seja executada pelo menos uma vez



Instruções (Break e continue)

- **break** encerra:
 - O loop atual, switch,

- E transfere o controle da execução do programa para o comando seguinte.v



Instruções (Break e

continue)

- **continue**

- Termina a atual iteração do laço
- ✓ E continua a execução deste laço com a próxima



iteração.



Instruções



(return)

- **return** finaliza a execução de uma **função** ■ E retorna o valor especificado
- ✓ Se omitido, **undefined** é retornado.





Instruções



(return)

- **return** finaliza a execução de uma **função** ■

Sem retornar valor (undefined)



Instruções (return)



- **return** finaliza a execução de uma **função**
 - Exemplo usando em loops





Tabela comparativa





Vamos

pensar?

- Quais as saídas desse código?

