

OPERADORES LÓGICOS E ESTRUTURAS CONDICIONAIS



DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

AGENDA

- Operadores Lógicos
- Estruturas condicionais
- if e else
- if ternário
- Switch

OPERADORES JAVASCRIPT

- **Operadores matemáticos:** são os operadores utilizados para realizar operações matemáticas com dados do tipo number

Operador	Operação	Exemplo
+	Soma	<code>1 + 1 // 2</code>
-	Subtração	<code>10 - 4 // 6</code>
*	Multiplicação	<code>5 * 1000 // 5000</code>
**	Exponenciação	<code>2 ** 8 // 256</code>
/	Divisão	<code>1024 / 8 // 128</code>
%	Resto	<code>18 % 2 // 0 ... 9 % 4 // 1</code>

OPERADORES JAVASCRIPT

- **Operador de atribuição:** o sinal de atribuição pode ser utilizado em conjunto com os operadores matemáticos. Ex: `var qtdItens = 120`

Operador	Operação	Exemplo
<code>+=</code>	Soma e reatribuição	<code>qtdItens += 25 // 145</code>
<code>-=</code>	Subtração e reatribuição	<code>qtdItens -= 20 // 100</code>
<code>*=</code>	Multiplicação e reatribuição	<code>qtdItens *= 1000 // 120000</code>
<code>**=</code>	Exponenciação e reatribuição	<code>qtdItens **= 2 // 14400</code>
<code>/=</code>	Divisão e reatribuição	<code>qtdItens /= 60 // 2</code>
<code>%=</code>	Resto e reatribuição	<code>qtdItens %= 40 // 0</code>

OPERADORES JAVASCRIPT

- **Operadores de comparação:** são operadores que comparam os valores e **sempre retornam um boolean**

Operador	Operação	Exemplo
<code>==</code>	Igual a	<code>"1" == 1 // true</code>
<code>===</code>	Estritamente igual a	<code>"1" === 1 // false</code>
<code>>, <</code>	Maior que, menor que	<code>20 > 20 // false</code>
<code>>=, <=</code>	Maior ou igual, menor ou igual	<code>20 <= 20 // true</code>
<code>!=, !==</code>	Diferente de	<code>1 != 2 // true</code>

OPERADORES JAVASCRIPT

- **Operadores lógicos:** são operadores que comparam **booleans** e retornam um valor de acordo com sua regra de validação:

Operador	Operação	Exemplo
&&	AND (e)	<code>false && true // false</code>
	OR (ou)	<code>false true // true</code>
!	NOT (não)	<code>!true // false</code>

Os operadores lógicos do JavaScript são um pouco mais poderosos (e complexos) do que a clássica *"tabela verdade"* da lógica matemática.

O operador **OR**, por exemplo, avalia cada um de seus operandos convertendo para **boolean**, e retorna o valor daquele operando que retornar **true** primeiro.

OPERADORES JAVASCRIPT (EXERCÍCIO)

Teste as seguintes afirmações e verifica no `console.log()` qual será a sua saída.

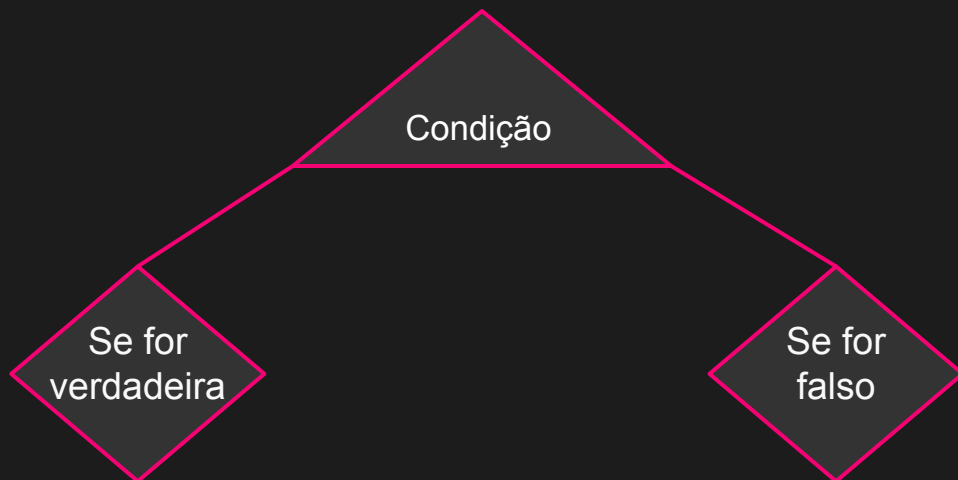
```
"alimentação" == "alimentação";  
"alimentação" == "refeição";  
"0" == "0";  
"0" == 0;  
"0" === 0;
```

```
0 > -2 || 10 < 9  
0 > -2 && 10 < 9  
!true == false  
true == false
```



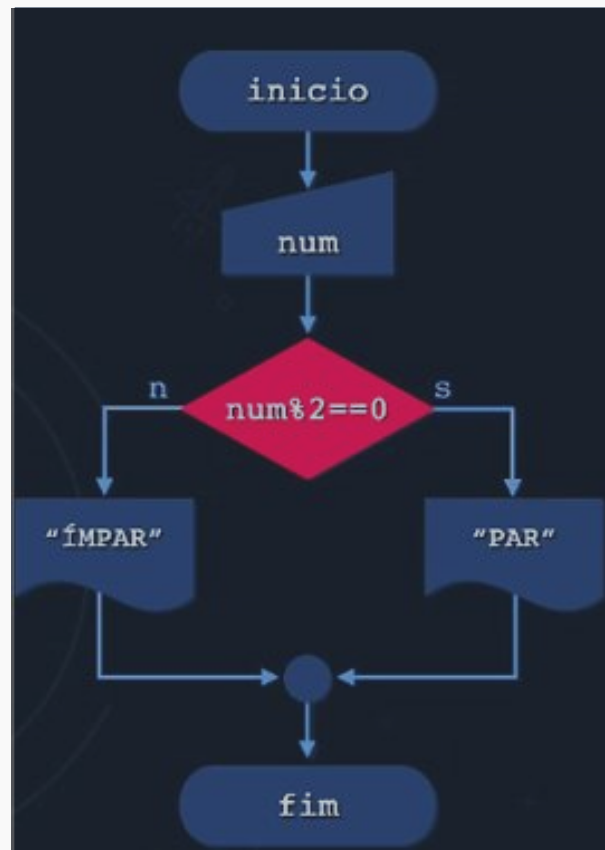
ESTRUTURAS CONDICIONAIS

- Estruturas condicionais são estruturas da programação que dependendo de uma condição, pode ou não executar um bloco de código.
- É possível definir uma ação específica para diferentes cenários e obter exatamente o resultado esperado para cada cenário.



ESTRUTURAS CONDICIONAIS

Vamos praticar a leitura do código !!



- A condicional **if** (se) é uma estrutura condicional que executa a afirmação, dentro do bloco, se determinada condição for verdadeira.
- Observe o exemplo abaixo:

```
var idade = 20;  
if (idade >= 18) {  
    console.log('É maior de idade');  
}
```

ESTRUTURA CONDICIONAL - EXERCÍCIO

- Construa um código com a estrutura condicional simples “if” que irá imprimir em um `console.log()` a mensagem “Parabéns você tirou a média mínima, não esqueça de continuar estudando :)” se determinada pessoa possui a média maior ou igual a 7.
- A média pode ser passada através de um prompt ou inicializada em uma variável.



INTERVALO DE AULA

DEV!

Finalizamos o nosso primeiro período de hoje. Que tal descansar um pouco?!

Nos vemos em 20 minutos.

Início: 20:20

Retorno: 20:35



ESTRUTURA CONDICIONAL COMPOSTA IF...ELSE

- A condicional **if** (se) é uma estrutura condicional que executa a afirmação, dentro do bloco, se determinada condição for verdadeira. Se for falsa, executa as afirmações dentro de **else** (senão).

```
var idade = 20;  
if (idade >= 18) {  
    console.log('É maior de idade');  
} else {  
    console.log('É menor de idade');  
}
```

ESTRUTURA CONDICIONAL ANINHADA IF...ELSE IF... ELSE

- Múltiplas condicionais if ... else podem ser aninhadas quando necessário. Observe que não existe elseif (em uma palavra). O correto é a instrução com espaços (**else if**), conforme abaixo:

```
var idade = 20;
if (idade === 18) {
    console.log('Acabou de ser maior idade');
} else if (idade > 18) {
    console.log('É maior de idade');
} else {
    console.log('É menor de idade');
}
```

ESTRUTURA CONDICIONAL ANINHADO - EXERCÍCIO

- Construa um código que pede duas notas de um aluno. Em seguida ele deve calcular a soma e imprimir em um `document.write` observando as seguintes condições:
 - A mensagem "Aprovado", se a soma alcançada for maior ou igual a sete;
 - A mensagem "Reprovado", se a soma for menor do que sete;
 - A mensagem "Você arrasou, aprovado com excelência", se a soma for igual a dez.



OPERADOR TERNÁRIO

- É um pequeno bit de sintaxe que testa uma condição e retorna um valor / expressão se for `true`, e outro caso seja `false`;
- É útil quando temos apenas duas opções escolhidas entre uma condição `true/false`;
- Ocupa muito menos código que um bloco `if...else`.

(condicao) ? 'executa esse código se for verdadeiro' **:** 'executa este código se for false'

OPERADOR TERNÁRIO

- Exemplo de operador ternário

```
var chuva = true;  
chuva ? console.log('Leve guarda-chuva') : console.log('Não leve guarda-chuva');
```

```
[Running] node "c:\U  
Leve guarda-chuva
```

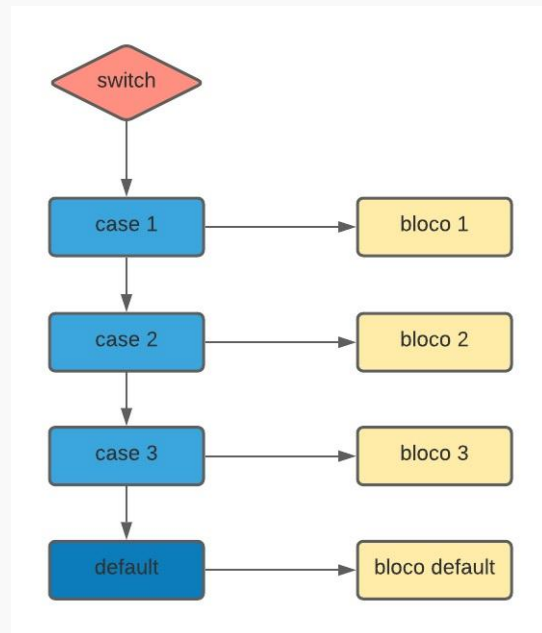
```
var chuva = false;  
chuva ? console.log('Leve guarda-chuva') : console.log('Não leve guarda-chuva');
```

```
[Running] node "c:\Use  
Não leve guarda-chuva
```

SWITCH

- A condicional **switch** avalia uma **expressão**, observando o **valor de entrada da expressão** para um cláusula **case**, e executa as instruções associadas a ela.

```
var fruta = 'laranja';  
switch (fruta) {  
  case 'uva':  
    console.log('Uva: R$10,00');  
    break;  
  case 'laranja':  
    console.log('Laranja: R$5,00');  
    break;  
}
```



SWITCH

- O bloco switch é mais recomendado para estruturas condicionais onde possua a entrada da condição única, também não é permitida múltiplas condições em um único case.
- Observe o caso abaixo:

```
var fruta = 'laranja';  
switch (fruta) {  
  case 'uva':  
    console.log('Uva: R$10,00');  
    break;  
  case 'laranja':  
    console.log('Laranja: R$5,00');  
    break;  
}
```

Valor a ser analisado

Bloco a ser executado quando o valor de entrada for igual ao caso.

Palavra reservada para a quebra (stop) da blocagem.


Fluxo:

- Primeiro procura por um caso (**case**) que a expressão dentro dos () tenha o mesmo valor do input da expressão (usando a comparação de igualdade estrita, ===);
- Depois transfere o controle para a cláusula encontrada e em seguida executa as instruções associadas;
- Caso nenhum caso seja correspondido, então o programa procura pela cláusula opcional **default**.

SWITCH

- Caso nenhum caso (case) seja correspondido, é executado as instruções contidas na cláusula opcional **default**.
- Por convenção, a cláusula default é a última, mas não é algo obrigatório.

```
var fruta = 'laranja';  
switch (fruta) {  
  case 'uva':  
    console.log('Uva: R$10,00');  
    break;  
  default:  
    console.log('Fruta não encontrada');  
}
```




```
[Running] node "c:\Us  
Fruta não encontrada
```

SWITCH

- A instrução opcional **break** associada com cada case garante que o programa saia da condicional switch assim que a instrução correspondente for executada. Caso break seja omitido, o programa continua a execução para a próxima instrução dentro de switch, independentemente do critério ter sido correspondido ou não.

```
var fruta = 'uva';  
switch (fruta) {  
  case 'uva':  
    console.log('Uva: R$10,00');  
  case 'laranja':  
    console.log('Laranja: R$5,00');  
  default:  
    console.log('Fruta não encontrada');  
}
```



```
[Running] node "c:\Use  
Uva: R$10,00  
Laranja: R$5,00  
Fruta não encontrada
```

SWITCH - Exercício

Observando os três casos abaixo, qual dos códigos terá um peso de processamento menor na sua aplicação?

```
var categoria = "terror";
```

```
switch (categoria) {  
  case "drama":  
    console.log("Este é um filme de drama!");  
    break;  
  case "ação":  
    console.log("Este é um filme de ação!");  
    break;  
  case "comédia":  
    console.log("Este é um filme de comédia!");  
    break;  
  case "terror":  
    console.log("Este é um filme de drama!");  
    break;  
}
```

```
if(categoria == "drama"){  
  console.log("Este é um filme de drama!");  
}else if(categoria == "ação"){  
  console.log("Este é um filme de ação!");  
}else if(categoria == "comédia"){  
  console.log("Este é um filme de comédia!");  
}else if(categoria == "terror"){  
  console.log("Este é um filme de terror!");  
}
```

```
if(categoria == "drama"){  
  console.log("Este é um filme de drama!");  
}  
if(categoria == "ação"){  
  console.log("Este é um filme de ação!");  
}  
if(categoria == "comédia"){  
  console.log("Este é um filme de comédia!");  
}  
if(categoria == "terror"){  
  console.log("Este é um filme de terror!");  
}
```


- <https://www.linkedin.com/feed/update/urn:li:activity:6948606332258107392/>
- <https://ricardo-reis.medium.com/operadores-l%C3%B3gicos-logical-operators-b0687819d1a5#:~:text=Os%20operadores%20l%C3%B3gicos%20s%C3%A3o%20importantes,executa%20outro%20bloco%20de%20c%C3%B3digo.>

AVALIAÇÃO DOCENTE

O que você está achando das minhas aulas neste conteúdo?

[Clique aqui](#) ou escaneie o QRCode ao lado para avaliar minha aula.

Sinta-se à vontade para fornecer uma avaliação sempre que achar necessário.





DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!



<LAB365>