FMT - aula 03

Operadores lógicos e HTML





#### **AGENDA**

- Revisão operadores de comparação
- Operadores lógicos
- Listas HTML
- Div
- Formulários

# Revisando operadores de comparação





#### Operadores de comparação

 Operadores de comparação: são operadores que comparam os valores e retornam um tipo boolean

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

#### Operadores de comparação



o JavaScript trata os valores de **string** vazia, o **number** 0, **undefined**, **false** e **null** como **valores** falsos (*falsy*).

Todos os demais valores são tratados como valores verdadeiros (truthy).

Veremos a aplicação deste comportamento mais adiante!

## Operadores lógicos





#### **OPERADORES LÓGICOS**

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

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

Operadores AND tem maior precedência do que operadores OR

#### **OPERADORES LÓGICOS**

O operador **OR** avalia cada um de seus operandos convertendo para booleano, e retorna o **valor** daquele operando que retornar **verdadeiro** (**true**) **primeiro**.

Já o operador **AND** avalia cada um de seus operandos e, se todos eles, após serem convertidos para booleano, forem **verdadeiros** (**true**), retorna o valor do **último** operando da cadeia.

#### Tabela verdade operador AND (&&)

Condição 1	Condição 2	Resultado
true	true	true
true	false	false
false	false	false

#### Tabela verdade operador OR (||)

Condição 1	Condição 2	Resultado
true	true	true
true	false	true
false	false	false

#### Tabela verdade operador NOT (!)

Condição	Resultado
true	false
false	true





```
let resultado_a = 10 > 7;
let resultado_b = 5 >= 6;
console.log(resultado_a && resultado_b)
```

```
let cor = "Azul";
let ano = 2010;
console.log(cor == "Azul" && ano >= 2008)
```

```
let resultado_a = 10 > 7;
let resultado_b = 5 >= 6;
console.log(resultado_a && !resultado_b)
```

```
let cor = "Vermelho";
let ano = 2010;
console.log(cor == "Azul" || ano >= 2008)
```

```
let cor = "Vermelho";
let ano = 2010;
console.log(cor == "Azul" || cor == "Branco")
```

```
let cor = "Vermelho";
let ano = 2010;
console.log(ano > 2008 && (cor == "Azul" || cor == "Branco"))
```



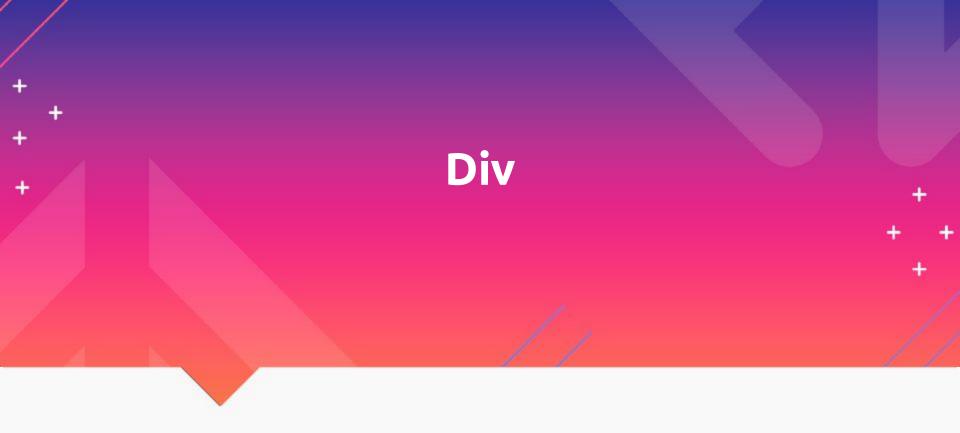


Vamos aprender como construir uma lista em HTML. Para isso precisamos primeiro definir o tipo de lista se será:

- ordenada
- não ordenada

#### Exemplo lista ordenada:

Exemplo lista não ordenada:







#### Div

A tag <div> define um divisão ou uma seção em uma página HTML. Ela é comumente usada para criar um container, isto é, uma demarcação de um espaço contendo vários elementos internos.

```
<div>
    <h2>Aluno: Bruno Costa</h2>

        <span>Curso: FMT </span>
        Email: bruno@aluno-ftm.com.br

</div>
```



Uma <div> cria um bloco de conteúdo sem significado semântico.

```
<div>
    <h2>Aluno: Bruno Costa</h2>

        <span>Curso: FMT </span>
        Email: bruno@aluno-ftm.com.br

</div>
```

## Formulários





#### **Formulários**

A tag <form> é utilizada para criar um formulário em HTML. Um formulário é utilizado para obter dados do usuário.

```
<form>
<label for="nome">Nome:</label><br>
<input type="text" id="nome" name="nome" value="Bruno"><br>
<label for="sobrenome">Sobrenome:</label><br>
<input type="text" id="sobrenome" name="sobrenome" value="Costa"><br>
<input type="submit" value="Enviar"></form>
</form>
```

#### **Formulários**

Para criar um formulário precisamos de vários elementos. São eles:

- form demarca onde inicia e termina o formulário
- label texto descritivo do campo de texto
- input campo de texto / entrada de dados
- button botão de ação

##