

AULA 1 – IF/ELSE E OPERADOR TERNÁRIO

OBJETIVO DA AULA

Conhecer, diferenciar e aplicar as estruturas condicionais.

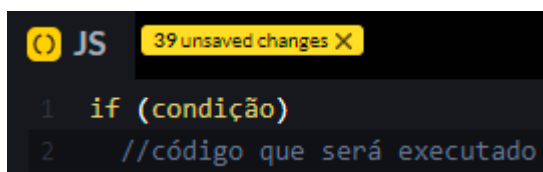
APRESENTAÇÃO

Nesta aula aprenderemos o que é uma estrutura condicional e como podemos aplicá-la ao nosso código. Basicamente, fazemos uso desta estrutura para verificarmos uma condição e definirmos o que será feito caso a condição seja verdadeira ou falsa. Você sabia que também recorremos a estruturas condicionais para nos comunicarmos no nosso dia a dia? Observe a frase: “Se fizer sol, vou à praia”, neste exemplo temos uma condição (se fizer sol) para executar uma ação (vou à praia). Resumindo, se a condição for verdadeira, a ação será executada. Na programação seguimos a mesma lógica. Aqueles que já conhecem outra linguagem de programação vão conseguir perceber uma certa semelhança de sintaxe.

1. IF/ELSE

A estrutura condicional mais utilizada é o *if* (se); tal estrutura permite ao JavaScript testar uma ou mais condições pré-estabelecidas para daí executar o trecho de código referente aquele teste. Para criar uma condição é necessário utilizar os operadores (aritméticos, relacionais e lógicos). Podemos utilizar o *if* de 3 formas diferentes. Vejamos:

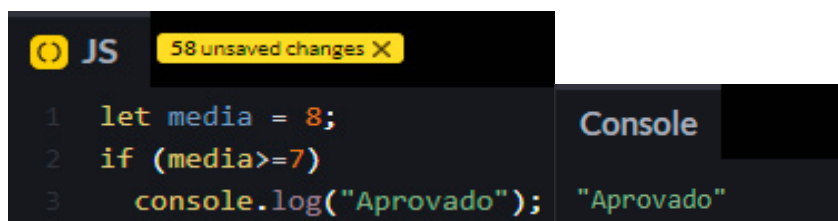
- **if**: executa o código caso a condição seja verdadeira.



```

JS 39 unsaved changes X
1  if (condição)
2    //código que será executado
  
```

Como ficaria na prática?

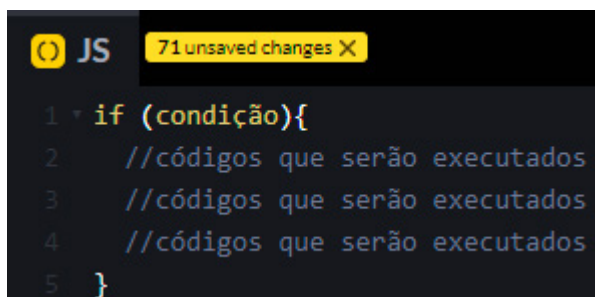


```

JS 58 unsaved changes X
1  let media = 8;
2  if (media >= 7)
3    console.log("Aprovado");
  
```

Console: "Aprovado"

No exemplo acima criamos uma variável *media* (linha 1) e atribuímos um valor a ela. Em seguida, utilizamos a palavra *if* (linha 2) seguido da condição *media >= 7*, caso a variável *media* tenha uma nota maior ou igual a 7 (verdadeiro) a linha abaixo (linha 3) será executada, ou seja, o comando *console.log* exibirá a mensagem “Aprovado” no console. Quando temos mais de uma linha de código a ser executada devemos usar *{ }* para delimitar o trecho de código.



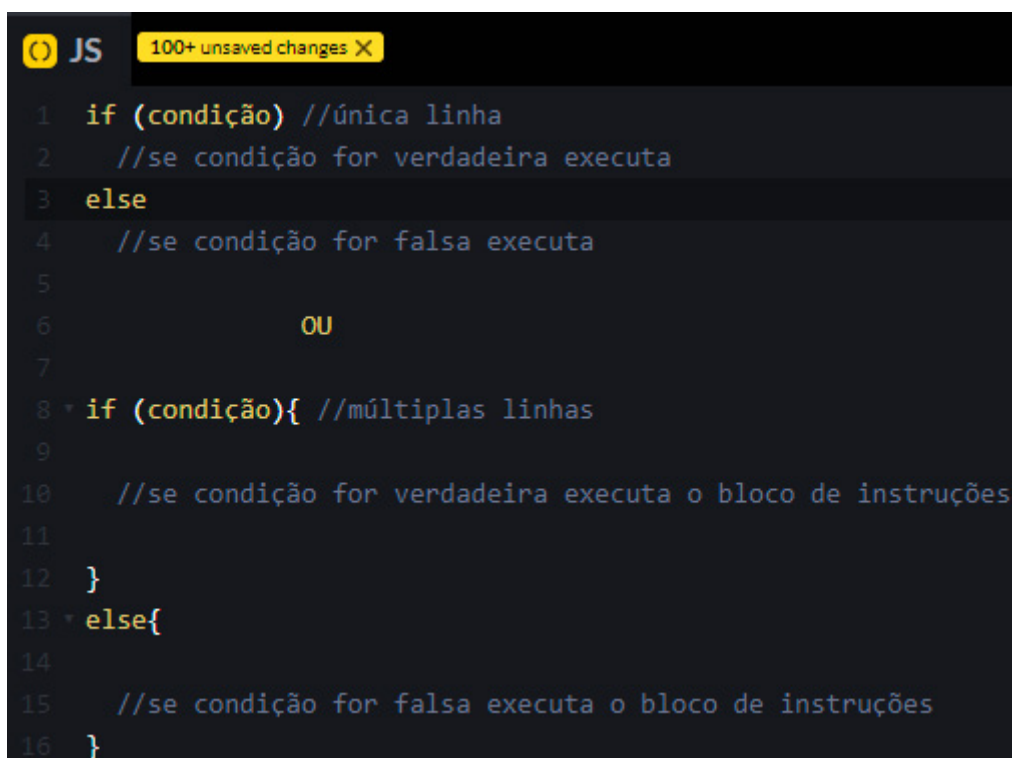
```

1 * if (condição){
2   //códigos que serão executados
3   //códigos que serão executados
4   //códigos que serão executados
5 }

```

Neste caso, caso a condição seja verdadeira, o JavaScript executará o bloco de instruções que está dentro das chaves *{ }*.

- **if-else:** executa o código caso a condição seja verdadeira ou falsa. Para isso utilizamos a palavra-chave *else*.

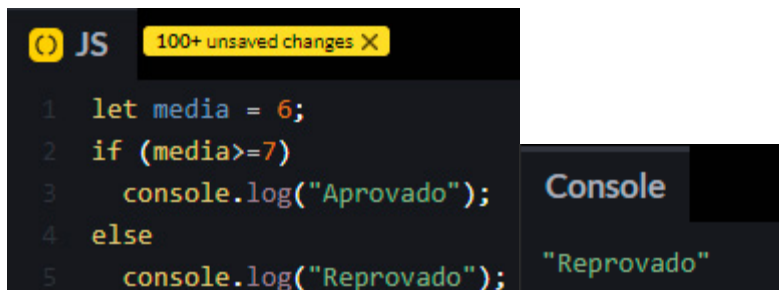


```

1 if (condição) //única linha
2   //se condição for verdadeira executa
3 else
4   //se condição for falsa executa
5
6   OU
7
8 * if (condição){ //múltiplas linhas
9
10  //se condição for verdadeira executa o bloco de instruções
11
12  }
13 * else{
14
15  //se condição for falsa executa o bloco de instruções
16  }

```

Seguindo o nosso exemplo, como ficaria na prática?



```

1 let media = 6;
2 if (media >= 7)
3   console.log("Aprovado");
4 else
5   console.log("Reprovado");

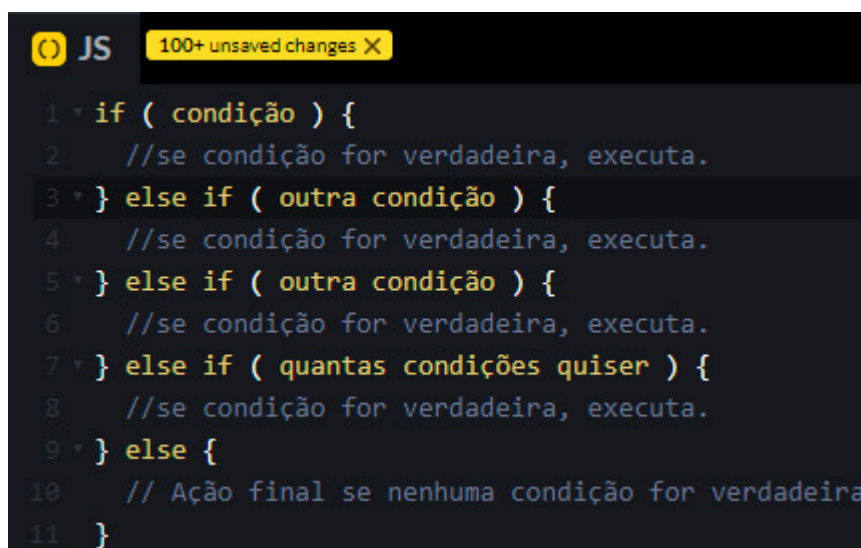
```

Console

"Reprovado"

Na nossa versão melhorada, caso a condição seja verdadeira, a *linha 3* será executada e exibirá a mensagem "Aprovado" no console, caso contrário, a *linha 4* será executada e exibirá a mensagem "Reprovado" no console.

- **else if:** quando há casos em que duas opções (verdadeiro/falso) não são o suficiente, devemos utilizar o **else if** para testar uma nova condição.

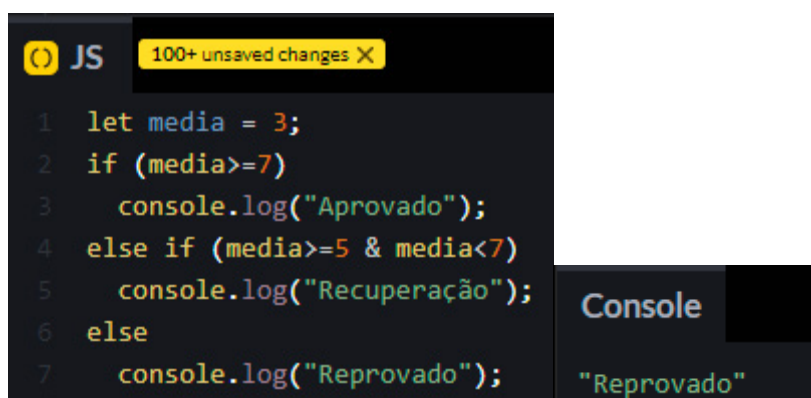


```

1 * if ( condição ) {
2   //se condição for verdadeira, executa.
3 * } else if ( outra condição ) {
4   //se condição for verdadeira, executa.
5 * } else if ( outra condição ) {
6   //se condição for verdadeira, executa.
7 * } else if ( quantas condições quiser ) {
8   //se condição for verdadeira, executa.
9 * } else {
10  // Ação final se nenhuma condição for verdadeira
11 }

```

Seguindo o nosso exemplo, como ficaria na prática?



```

1 let media = 3;
2 if (media >= 7)
3   console.log("Aprovado");
4 else if (media >= 5 & media < 7)
5   console.log("Recuperação");
6 else
7   console.log("Reprovado");

```

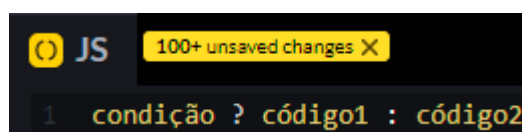
Console

"Reprovado"

Na nossa versão melhorada, **se** a condição for verdadeira, a *linha 3* será executada e exibirá a mensagem “Aprovado” no console, **senão se** a nova condição *for* verdadeira a *linha 5* será executada e exibirá a mensagem “Recuperação” no console, **senão** a *linha 7* será executada e exibirá a mensagem “Reprovado” no console. Chamamos esta estrutura de **se encadeado** ou **se aninhado**.

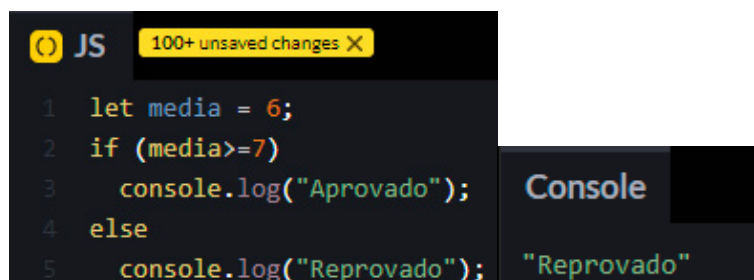
2. OPERADOR TERNÁRIO

O operador ternário nada mais é do que uma versão mais compacta do **if-else**, com ele podemos deixar nosso código mais limpo, pois um **if** de 4 linhas é substituído por apenas 1 linha. É possível encadear diversas verificações de condição, porém não é recomendado, pois a leitura do código fica mais difícil do que um **if** convencional. Vamos conhecer a sua sintaxe:



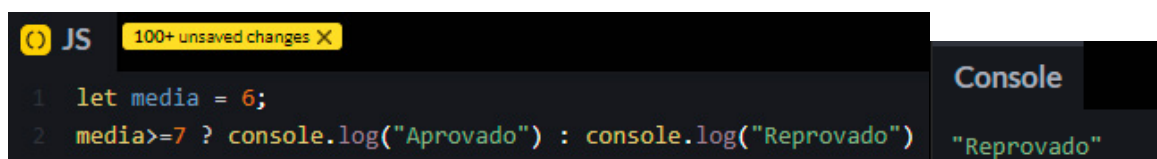
```
1 condição ? código1 : código2
```

Caso a condição seja verdadeira, o código1 é retornado, caso contrário, o código2 é retornado. O retorno pode ser atribuído a uma variável ou ao retorno de uma função. Vamos comparar o uso do **operador ternário** com o uso do **if** para que você possa perceber a diferença entre eles.



```
1 let media = 6;
2 if (media >= 7)
3   console.log("Aprovado");
4 else
5   console.log("Reprovado");
```

Console
"Reprovado"



```
1 let media = 6;
2 media >= 7 ? console.log("Aprovado") : console.log("Reprovado")
```

Console
"Reprovado"

Repare que no primeiro exemplo escrevemos o código em 5 linhas e no segundo exemplo escrevemos o mesmo código em apenas duas linhas. Logo após o ponto de interrogação (?) temos a instrução que representa a verdade e depois dos dois pontos (:) temos a instrução que representa a falsidade.

CONSIDERAÇÕES FINAIS

Nesta aula aprendemos a utilizar as estruturas condicionais **if**, **if-else** e **else if** e também o **operador ternário**. Você deve estar se perguntando: “São tantas variações da estrutura condicional **if**, qual delas devo utilizar?” Você decidirá qual utilizar segundo o contexto da aplicação. Além do **if**, vimos também o **operador ternário**, utilizado como atalho para o **if**, que pode deixar o código mais enxuto e mais limpo. Os conceitos abordados nesta aula são fundamentais para o seu desenvolvimento em qualquer linguagem de programação.

MATERIAIS COMPLEMENTARES

Operador ternário:

<https://youtu.be/uAyfj46vOJE>

If/Else:

<https://youtu.be/8UXQ6S0KURk>

REFERÊNCIAS

FLANAGAN, David. *JavaScript: O guia definitivo*. Porto Alegre, RS: Bookman, 2013.