

# IW-II

# INTERFACE WEB II

---

Prof. Anderson Vanin

# **Estruturas Condicionais**

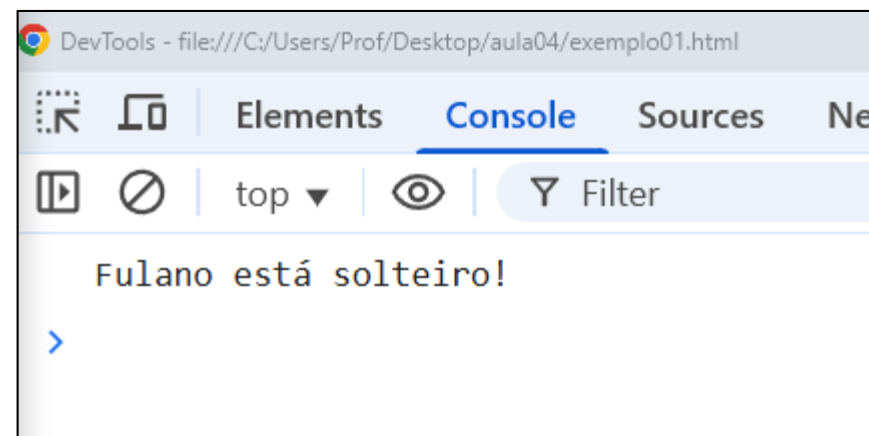
# Expressões IF e ELSE

Agora que já vimos a tabela verdade, neste tópico vamos tratar sobre uma das questões mais importantes que está presente em todas as linguagens de programação, chamamos de expressões if e else que são o que nos habilita a tomarmos decisões dentro de nosso código.

Estas expressões também são conhecidas como expressões de controle de estrutura, isso significa que podemos controlar o que será executado por nosso código ou não.

# Expressões IF e ELSE

```
<> exemplo01.html > html
1  <!DOCTYPE html>
2  <html lang="pt-BR">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Exemplo 01</title>
7  </head>
8  <body>
9      <h1>Estruturas de Controle - IF e ELSE</h1>
10     <script>
11
12         var nome = "Fulano";
13         var idade = 35;
14         var estaSolteiro = true;
15
16         // if (estaSolteiro) { // retorno da decisão true
17         if (estaSolteiro === true) { // retorno da decisão true
18             console.log(nome + " está solteiro!");
19         } else { // se retorno da decisão for false
20             console.log(nome + " está casado!");
21         }
22
23     </script>
24 </body>
25 </html>
```



# Expressões IF e ELSE

Utilizando os operadores lógicos descritos nas tabelas verdade, é possível criar uma gama de decisões lógicas, vamos ver exemplos:

# Expressões IF e ELSE

```
<script>

var variavel1 = 53;
var variavel2 = true;
var variavelTexto = "zero";
var variavelNumero = "24";
var variavelLogica = false;
var variavelCurta = "texto curto";

if (variavel2 && variavelNumero == 24) { // retorno da decisão true
    console.log("Retorno verdadeira");
} else { // se retorno da decisão for false
    console.log("Retorno falso");
}

if (variavel1 > 50) { console.log(variavel1 + " é maior que 50"); }
if (variavelTexto != "um") { console.log(variavelTexto + " é diferente de um"); }
if (!variavelLogica) { console.log("Negação da variável lógica é verdadeira"); }
if (variavelTexto != variavel1) { console.log("Valor das variáveis é diferente"); }

</script>
```

Retorno verdadeira

53 é maior que 50

zero é diferente de um

Negação da variável lógica é verdadeira

Valor das variáveis é diferente

# Else If

Além de decisões lógicas simples e compostas podemos criar também decisões lógicas caso uma decisão lógica retorne negativo, parece estranho, porém veja o exemplo:

```
<script>

var numeroPassageiros = 23;

if (numeroPassageiros === 0) {
    console.log("O transporte está vazio.");
} else if (numeroPassageiros > 0 && numeroPassageiros < 50) {
    console.log("O transporte tem passageiros.");
} else {
    console.log("O transporte está lotado.");
}

</script>
```

O transporte tem passageiros.

>

# SWITCH

Ainda falando sobre decisões lógicas, existe uma forma mais elegante de criar diversas decisões lógicas para uma variável, ao invés de criar vários comandos ***if else*** e assim por diante. Esta forma se chama **switch**.

Imagine que você precisa criar um código para verificar qual é a profissão de alguém, no que esta pessoa trabalha. Existem diversos tipos de trabalhos e poderíamos criar esta verificação utilizando comandos ***if else***, mas para deixar nosso código mais apresentável, vamos usar o comando **switch**, vejamos o exemplo:



# SWITCH

```
<script>

var nome = "Michele";
var emprego = "desenvolvedor";
switch (emprego) {
  case "policial":
    console.log(nome + " é uma policial.");
    break ;
  case "cozinheiro":
    console.log(nome + " é uma cozinheira.");
    break ;
  case "desenvolvedor":
    console.log(nome + " é uma desenvolvedora");
    break ;
  case "lutador":
    console.log(nome + " é uma lutadora.");
    break ;
  default :
    console.log(nome + " trabalha em outra coisa.");
}

</script>
```

Michele é uma desenvolvedora

> |

# **Laços de Repetição**

# FOR

O laço de repetição **for** é provavelmente o mais utilizado dentro da linguagem JavaScript e dentro de muitas linguagens de programação. A estrutura deste laço de repetição é:

```
for (contador; condição; incremento) {  
    ...lógica...  
}
```

# FOR

```
<script>
// for (contador; condição; incremento) {
for (var i = 0; i < 10; i++) {
|   console.log(i);
}
</script>
```

0

1

2

3

4

5

6

7

8

9

>

# WHILE

Diferente do laço de repetição for , o laço de repetição while tem uma estrutura diferente onde não há um local definido para um contador ou incremento , vejamos:

```
while (condição) {  
    ...lógica...  
}
```

# WHILE

```
<script>
  var i = 0;
  while (i<10){
    console.log(i);
    i++;
  }
</script>
```

0
1
2
3
4
5
6
7
8
9

# EXERCÍCIOS

## 1. Par ou Ímpar

Peça um número ao usuário e exiba se ele é par ou ímpar.

## 2. Verificação de Maioridade

Solicite a idade do usuário e informe se ele é maior de idade (18 anos ou mais) ou menor de idade.

## 3. Contagem de 1 a 10

Exiba os números de 1 a 10 no console usando um laço for.

## 4. Contagem Regressiva

Peça um número ao usuário e exiba uma contagem regressiva até o usando um while.

## 5. Tabuada

Peça um número e exiba a tabuada de 1 a 10 usando um laço for.

# EXERCÍCIOS

## 6. Somatório

Solicite um número e **some todos os números de 1 até ele**.

## 7. Número Primo

Peça um número ao usuário e informe se ele é **primo** ou **não**.

## 8. Login Simples

Peça ao usuário um **nome de usuário** e uma **senha**. Se ele errar, peça novamente até acertar.

## 9. Soma de Números Positivos

Peça ao usuário vários números e some-os até que ele **digite um número negativo**, então exiba o total.

## 10. Adivinhação

Escolha um número aleatório de **1 a 10** e peça ao usuário para adivinhar. Dê dicas até ele acertar.



# EXERCÍCIOS

## 11. Cálculo de Fatorial

Peça um número e exiba seu fatorial (n!).

## 12. Validação de Nota

Peça uma nota de 0 a 10 e continue pedindo até que o usuário insira um valor válido.

## 13. Média de Notas

Peça 3 notas ao usuário, calcule e exiba sua média. Informe se ele foi aprovado (média  $\geq 7$ ) ou reprovado

## 14. Caixa Eletrônico

Peça um valor ao usuário e informe as possíveis cédulas para saque (100, 50, 20, 10, 5, 2, 1).

## 15. Números Ímpares

Peça um número ao usuário e exiba todos os números ímpares de 1 até esse número.