

Estruturas de Controle if ()...else

Quando queremos tomar uma decisão em nosso código, ou seja, definir outras opções em JavaScript utilizamos o comando if (se).

A sintaxe do if no JavaScript é a seguinte

```
if (condicaoBooleana)
{ codigo; }
```

Uma condição booleana é qualquer expressão que retorne true ou false. Para isso, você pode usar os operadores <, >, <=, >= e outros.

Além da utilização do if, nós poderemos utilizar a cláusula else (senão) para indicar o comportamento que deve ser executado no caso da expressão booleana ser falsa.

Vejamos o exemplo abaixo para identificar essas duas cláusulas:

```
string nome = prompt();
console.log("Nome: ");
var nome = alert();
console.log("Peso: ");
var peso = prompt();
console.log("Altura: ");
var altura = alert();
var imc = peso / (altura * altura);
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>

  <script src="./script.js"></script>
</body>
</html>
```

Em relação a este exemplo caso o seu IMC for maior ou igual a 40 irá aparecer duas mensagens ("*Obesidade grau III*" e "*Muito cuidado com o seu peso*") atendendo a nossa necessidade.

Mas caso o seu IMC for 34 além de apresentar a mensagem "*Obesidade grau I*" irá aparecer "*Muito cuidado com o seu peso*" não atendendo a nossa necessidade.

Isso se deve ao fato de termos uma regra em relação as chaves de demarcação de cada estrutura if(). Só se torna obrigatório o uso das chaves { } quando ver mais de uma linha de execução no bloco, quando ver apenas uma única linha de execução não se torna necessário.

Nesse mesmo exemplo se quisermos colocar mais de uma opção dentro da condição iremos utilizar os operadores lógicos, como no exemplo abaixo:

```
if (sexo == 'F' && imc < 19.1) {  
  alert("Abaixo do peso.");  
}  
else if (sexo == 'F' && imc <= 25.8) {  
  alert("Peso ideal.");  
}  
else if (sexo == 'F' && imc <= 27.3) {  
  alert("Um pouco acima do peso.");  
}  
else if (sexo == 'F' && imc <= 32.3) {  
  alert("Acima do peso ideal.");  
}  
else if (sexo == 'F') {  
  alert("Obeso.");  
}  
else if (sexo == 'M' && imc < 20.7) {  
  alert("Abaixo do peso.");  
}  
else if (sexo == 'M' && imc <= 26.4) {  
  alert("Peso ideal.");  
}  
else if (sexo == 'M' && imc <= 27.8) {  
  alert("Um pouco acima do peso.");  
}  
else if (sexo == 'M' && imc <= 31.1) {  
  alert("Acima do peso ideal.");  
}  
}
```

```
if (imc < 18.5)  
  alert("Abaixo do peso ideal.");  
else if (imc < 25)  
  alert("Peso ideal.");  
else if (imc < 30)  
  alert("Acima do peso.");  
else if (imc < 35)  
  alert("Obesidade grau I.");  
else if (imc < 40)  
  alert("Obesidade grau II.");  
else  
  alert("Obesidade grau III.");  
alert("Muito cuidado com seu peso.");
```

```
else if (sexo == 'M') {  
  alert("Obeso.");  
}
```

```
teste = valor < 40 && valor > 20; // teste = true  
teste = valor < 40 && valor > 30; // teste = false  
teste = valor > 30 || valor > 20; // teste = true  
teste = valor > 30 || valor < 20; // teste = false  
teste = valor < 50 && valor == 30; // teste = true
```

	<p>Alerta de BSM: Sempre importante utilizar atenção aos detalhes nas diferença dos operadores lógicos maior e menor ;</p>
<p>O operador lógico && avalia as expressões do lado esquerdo e direito e retorna apenas</p>	
<p>um resultado booleano. Para a expressão completa ser verdadeira, tanto o lado direito</p>	
<p>como o lado esquerdo devem ser verdadeiras, mas para que a expressão completa seja</p>	
<p>falsa, pelo menos um lado deve ser falso.</p>	
<p>Ainda existe o operador lógico “OU”, representado por</p>	
<p>código-fonte de nosso exemplo para usá-lo.</p>	
<p>Veja que, usando o operador</p>	
<p>de programação, porém deixamos as expressões booleanas dos ifs mais complexas, pois</p>	
<p>agrupamos duas expressões que usam o operador && dentro de outra que usa o</p>	
<p>operador</p>	
<p>O operador lógico</p>	
<p>booleano. Para que a expressão completa seja verdadeira, pelo menos um lado deve ser</p>	
<p>verdadeiro.</p>	

Estrutura de Controle switch()

Sintaxe:

```
switch (Expressao){  
  case valor1: conjuntoDeSentencas;  
  break;  
  case valor2: SentencasAlternativas;  
  break;  
  case valor3: SentencasAlternativas2;  
  break;  
  case valor4: SentencasAlternativas3;  
  break;  
}
```

O comando switch pode ter vários possíveis caminhos de decisão (casos). O primeiro caso que estiver de acordo com o valor passado para o switch inicia a execução das instruções do caso.

A sentença 'break' atrás de cada opção de case serve para que não avalie o resto de opções e sim que saia diretamente do 'Switch', por isso, dependendo do que quiser fazer, você colocará ou não.

Vamos verificar um exemplo para demonstrar a execução desse laço:

```
switch (i) {  
  case '1': alert( "i contem um 1");  
  case '2': alert( "i contem um 2");  
  case '3': alert( "i contem um 3");  
}
```

Exemplo usando o break:

```
var dia;  
alert("Digite um numero para encontrar o seu dia: ");  
dia = pront();  
switch (dia) {  
  case 1:  
    alert("Domingo");  
    break;  
  case 2:  
    alert("Segunda");  
    break;  
  case 3:  
    alert("Terça");  
    break;  
  case 4:  
    alert("Quarta");  
    break;  
  case 5:  
    alert("Quinta");  
    break;  
  case 6:  
    alert("Sexta");  
    break;  
}
```

```
    alert("Sabado");  
    break;  
    default:  
        alert("Número digitado é inválido!!!");  
    break;  
    }  
}
```