

# Aulas teóricas e exercícios

## Créditos ao Curso Original:

Este material foi criado com base nas videoaulas do curso **CursoEmVídeo**, oferecido por **Prof. Gustavo Guanabara** na plataforma **CursoEmVídeo**. O conteúdo original foi utilizado como referência, sendo atualizado para refletir conceitos e práticas mais recentes no contexto de JavaScript.

Agradeço ao professor **Gustavo Guanabara** e à plataforma **CursoEmVídeo** por disponibilizarem as videoaulas e materiais, permitindo que o conteúdo seja utilizado para fins educacionais e adaptado, desde que a modificação seja feita com a devida referência ao material original.

O conteúdo original pode ser acessado no repositório público do Professor Gustavo Guanabara, disponível em:

<https://github.com/gustavoguanabara/javascript>

E as videoaulas podem ser acessadas na plataforma **CursoEmVídeo**:

<https://www.cursoemvideo.com/curso/javascript/>

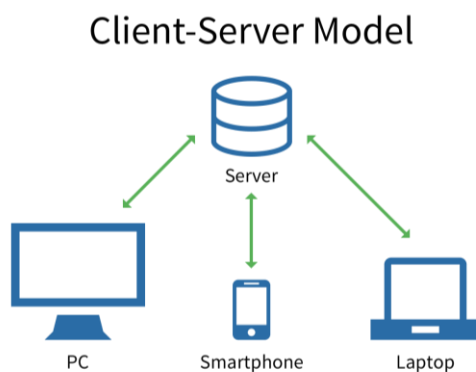
# Módulo A

## Aula 01

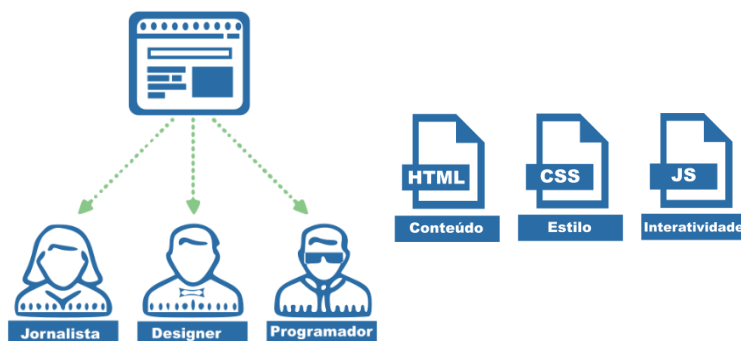
# O que JavaScript é capaz de fazer?

JavaScript serve para dar interatividade à página!

## O que é Client Server?



## Tecnologias



# Aula 02 – Como chegamos até aqui?

**/\* Q01**

Sabe a diferença entre um cliente e um servidor para a Internet? \*/

---

---

**/\* Q02**

Para que serve HTML, CSS e JavaScript? \*/

---

---

---

**/\* Q03**

Cite 4 sites que fazem o uso de JavaScript? \*/

---

---

---

---

# Aula 03

## Dando os primeiros passos

**/\*Q01**

Sabe qual empresa que criou o JavaScript?\*/

---

---

---

**/\*Q02**

Java e JavaScript possuem alguma relação?\*/

---

---

---

**/\*Q03**

O que ECMAScript tem a ver com JavaScript?\*/

---

---

---

**/\*Q04**

Sabia que o programa WhatsApp Desktop é criado em JavaScript?\*/

---

---

---

# **Bibliografias recomendadas**

## **Livros Pagos**

JAVASCRIPT o guia definitivo

Javascript o guia do programador

## **Guia online Grátis**

### **Guia Mozilla**

<https://developer.mozilla.org/pt-BR/>

### **ECMAScript**

<https://ecma-international.org/publications-and-standards/standards/ecma-262/>

# Aula 04

## Criando o seu primeiro script

**/\*Q01**

livros/ documentos nós recomendamos?\*/

---

---

---

---

**/\*Q02**

Para aprender JS, precisamos saber inglês?\*/

---

---

---

---

**/\*Q03**

Que dicas nós demos para quem quer estudar?\*/

---

---

---

---

**/\*Q04**

Quais programas nós instalamos no computador?\*/

---

---

---

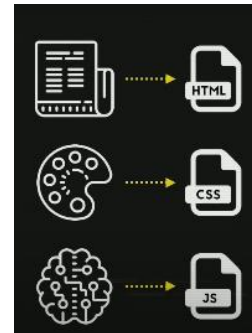
---

# Relembrando:

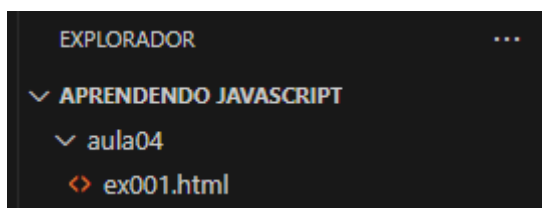
HTML – Conteúdo

CSS - Design

JavaScript – Interatividade/ Side Client



## Exercício em execução



```
ex001.html X
ex001.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>Meu primeiro programa...</title>
7      <style>
8          body {
9              background-color: rgb(0, 0, 255);
10             color: white;
11             font: normal 20pt Arial;
12         }
13
14         h1 {
15             color: yellow;
16         }
17     </style>
18 </head>
19 <body>
20     <h1>Olá, mundo!</h1>
21     <p>Já me livrei da maldição!</p>
22
23     <script>
24         window.alert('Minha primeira mensagem!')
25         window.confirm('Está gostando de JavaScript')
26         window.prompt('Qual é o seu nome? ') //Pergunta o nome
27     </script>
28 </body>
29 </html>
```

# Módulo B

## Comandos Básicos do JavaScript

**/\*Q01**

Já sabe gerenciar pastas no VSCode?\*/

---

---

---

---

**/\*Q02**

O **Node.js** está instalado no seu computador? Você sabe para que serve o **Node.js**?\*/

---

---

---

---

**/\*Q03**

Sabe diferenciar **HTML**, **CSS** e **JS** dentro de um código?\*/

---

---

---

---

**/\*Q04**

Sabe disparar **alertas**, **confirmações** e **perguntas**?\*/

---

---

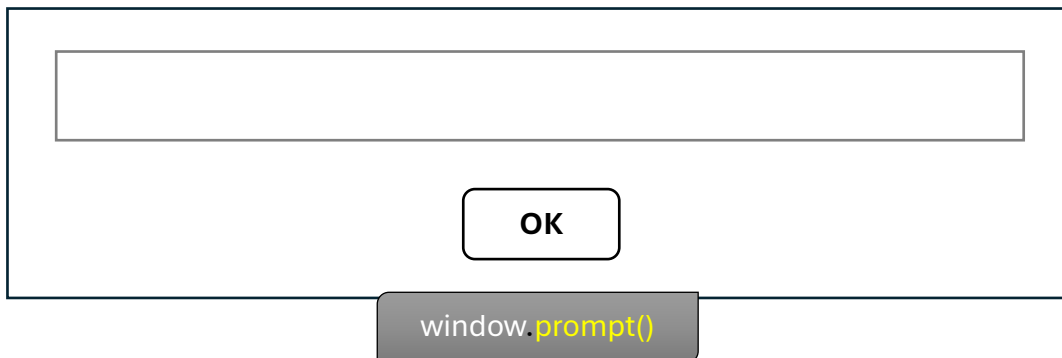
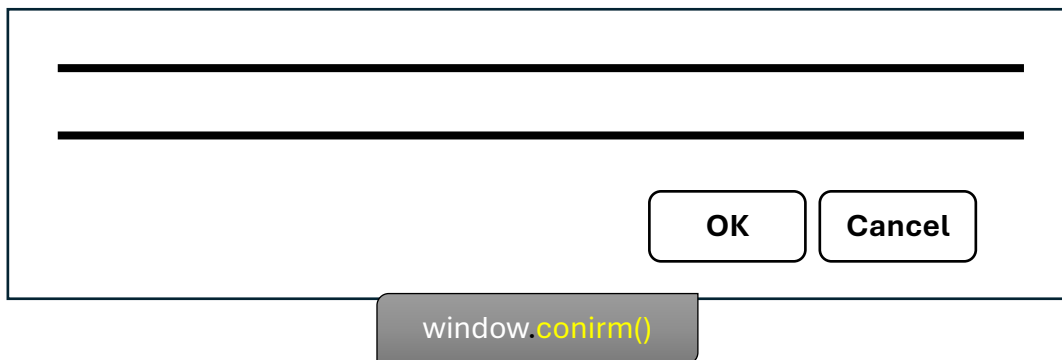
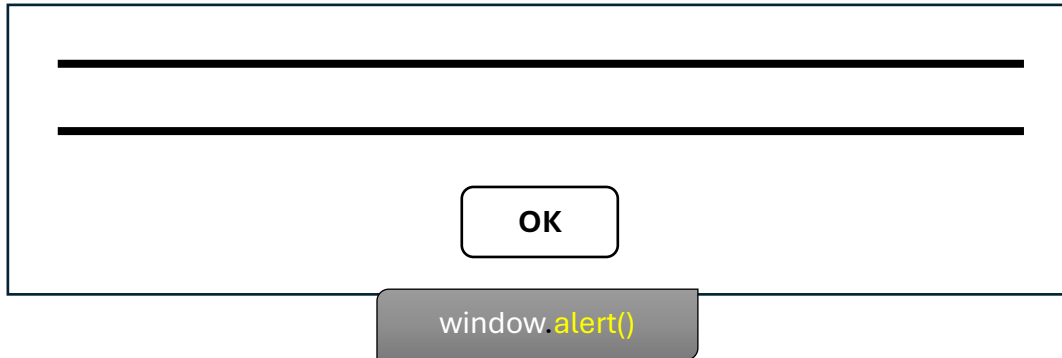
---

---



# Aula 05

## Variáveis e Tipos Primitivos



## Comentários em JS

// uma única linha

/\* mais de uma linha \*/

# Variáveis em JavaScript

No Javascript moderno além de utilizar a palavra **var**, também podemos usar a palavra **let**.

```
var n1 = 5
```

```
var n2 = 8.5
```

```
var n3 = 15
```

```
var s1 = "JavaScript"
```

```
var s2 = 'Curso em vídeo'
```

```
var s3 = `Guanabara`
```

## Identificadores

- Podem começar com **letra**, **\$** ou **\_**
- Não podem começar com **números**
- É possível usar **letras** ou **números**
- É possível usar **acentos** ou **símbolos**
- Não podem conter **espaços**
- Não podem ser palavras **reservadas**

## Dicas para nomes de Identificadores

- **Maiúsculas** e **minúsculas** fazem diferença
- Tente escolher **nomes coerentes** para as variáveis
- Evite se tornar um **‘programador alfabeto’** ou um **‘programador contador’**

# Data Types

typeof



# Aula 06

## Tratamentos de dados

**/\*Q01**

O que é uma variável?\*/

---

---

---

---

**/\*Q02**

Como declarar uma variável numérica Inteira ou Real usando JS?\*/

---

---

---

---

**/\*Q03**

Quais são os tipos primitivos do JS?\*/

---

---

---

---

**/\*Q04**

O que significa colocar o valor null dentro de uma variável?\*/

---

---

---

---

# Manipulação de Dados

## ▼ APRENDENDO JAVASCRIPT

> aula04

▼ aula06

<> ex002.html

```
aula06 > <> ex002.html > html > body > script
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>Meu primeiro programa...</title>
7      <style>
8          body {
9              background-color: rgb(0, 0, 255);
10             color: white;
11             font: normal 20pt Arial;
12         }
13
14         h1 {
15             color: yellow;
16         }
17     </style>
18 </head>
19 <body>
20     <h1>Olá, mundo!</h1>
21     <p>Já me livrei da maldição!</p>
22
23     <script>
24         var nome = window.prompt('Qual é o seu nome? ') //Vai perguntar o nome...
25         window.alert('É um grande prazer em ter conhecer, ' + nome + '!')//Concatenação!
26     </script>
27 </body>
28 </html>
```

## Código JavaScript:

▼ aula06

<> ex002.html

<> ex003.html

```
22
23     <script>
24         var n1 = Number(window.prompt('Digite um número: '))
25         var n2 = Number(window.prompt('Digite outro número: '))
26         var s = n1 + n2
27         window.alert(`A soma entre dois valores é ` + s)
28         //(number + number) para adição
29         //(string + string) para concatenar
30     </script>
```

### Precisamos converter:

String > Number

Number.parseInt(n) > Converte um número para um número Inteiro

Number.parseFloat(n) > Converte um número em Real

Number(n) > Converte os valores para Inteiro e Real sem precisar especificar o tipo de dados

### Numero > String

String(n)

n.toString

### Versão recente do ECMAScript apenas Number

```
23  <script>
24      var n1 = Number(window.prompt('Digite um número: '))
25      var n2 = Number(window.prompt('Digite outro número: '))
26      var s = n1 + n2
27      window.alert(`A soma entre ${n1} e ${n2} é igual a ${s}`)
28      //(number + number) para adição
29      //(string + string) para concatenar
30  </script>
```

# String > Number

String(n)

n.toString()

## Formatando String

Var s = 'JavaScript'

'Eu estou aprendendo s ' //não faz interpolação

'Eu estou aprendendo ' + s //usa concatenação

` Eu estou aprendendo \${s}` //usa string

Var s = 'JavaScript'

s.length // quantos caracteres a string tem

s.toUpperCase() // tudo para 'MAIÚSCULAS'

s.toLowerCase() // tudo para 'minúsculas'

```
ex004.html X
aula06 > ex004.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     body {
9       font: normal 20pt Arial;
10    }
11  </style>
12 </head>
13 <body>
14   <script>
15     var nome = window.prompt('Qual é o seu nome?')
16     document.write(`Olá, <strong>${nome}</strong>! Seu nome tem ${nome.length} letras.<br>`)
17     document.write(`Seu nome em maiúsculas é ${nome.toUpperCase()}<br>`)
18     document.write(`Seu nome em minúsculas é ${nome.toLowerCase()}`)
19   </script>
20 </body>
21 </html>
```

## Formatando números

n1.toFixed(numero de casa decimal)

n1.toFixed().replace(".", ",")

n1.toLocaleString('pt-BR', {style: 'currency', currency: 'BRL'})

# Aula 07

## Operadores (Parte 01)

### **/\*Q01**

Como é possível guardar o valor digitado em um **prompt()** dentro de uma variável?

---

---

---

---

### **/\*Q02**

Como fazer com que um número digitado em um **prompt()** possa ser usado em cálculos?\*/

---

---

---

---

### **/\*Q03**

Como transformar um texto todo para letras **MAISÚSCULAS?\*/**

---

---

---

---

### **/\*Q04**

Como mostrar um número formatado como um valor monetário?\*/

---

---

---

---



## Operadores em JavaScript

Aritméticos

Atribuição

Relacionais

Lógicos

Ternário

### Aritméticos (+, -, \*, /, %, \*\* (ao quadrado))

5	+	2	→	7
5	-	2	→	3
5	*	2	→	10
5	/	2	→	2.5
5	%	2	→	1
5	**	2	→	25

### Ordem de Precedência

Precedência

()

\*\*

\*, /, %

+, -

### Atribuições Simples

Variáveis	Valores
var a = 5 + 3	8
var b = a % b	3
var c = 5 * b ** 2	45
var d = 10 - a / 2	6
var e = 6 * 2 / d	2
var f = b % e + 4 / e	3

### Auto-atribuições Simples

Variáveis	Valores
var n = 3	3
n = n + 4	7
n = n - 5	2
n = n * 4	8

<code>n = n / 2</code>	4
<code>n = n ** 2</code>	16
<code>n = n % 5</code>	1

### Simplificando

Variáveis	simplificado
<code>var n = 3</code>	
<code>n = n + 4</code>	<code>n += 4</code>
<code>n = n - 5</code>	<code>n -= 5</code>
<code>n = n * 4</code>	<code>n *= 4</code>
<code>n = n / 2</code>	<code>n /= 2</code>
<code>n = n ** 2</code>	<code>n **= 2</code>
<code>n = n % 5</code>	<code>n %= 5</code>

# Aula 08

## Operadores (Parte 02)

**/\*Q01**

Para que serve o operador % em JS? Seria para calcular porcentagem?

---

---

---

---

**/\*Q02**

**6 + 4 / 2** é igual a 5 ou 8?\*/

---

---

---

---

**/\*Q03**

Se uma variável **n** está valendo 10 e executamos um **n += 5**, qual será o novo valor?\*/

---

---

---

---

**/\*Q04**

Para que servem os operadores **++** e **--**?\*/

---

---

---

---

Relacionais				
5	>	2	→	True
7	<	4	→	False
8	>=	8	→	True
9	<=	7	→	False
5	==	5	→	True
4	!=	4	→	False

>	Maior que
<	Menor que
>=	Maior ou Igual
<=	Menor ou Igual
==	Igual
!=	Diferente

### Exemplos

preço >= 200.50      // O preço é maior ou igual a 200.50?

idade < 18              // A idade é menor do que 18?

curso = 'JavaScript'    // O curso é JavaScript?

n1 != n2                // O primeiro numero é diferente do segundo?

### Identidade

5 == 5 → true

5 == '5' → true

5 === '5' → false

### Operadores Lógicos

!    = negação

&& = conjunção (E logico)

||    = disjunção (Ou logico)

### Exemplos

idade >= 15 && idade <= 17      // a idade esta entre 15 e 17?

estado == 'RJ' || estado == 'SP' // o estado é RJ ou SP?

salário > 1500 && sexo != 'M'

// o salário é acima de 1500 e não é um homem?

Precedência	
	(), **, / ...
	>, <, >= ...
	!
	&&

Ternário

TESTE	?	TRUE	:	FALSE
Média >= 7.0	?	“Aprovado”	:	“REPROVADO”

# Módulo C: Entendendo o DOM

## Aula 09

### Introdução ao DOM

**/\*Q01**

Qual é a diferença entre usar `=`, `==` e `===` em códigos escritos em JS?

---

---

---

---

**/\*Q02**

Em uma mesma expressão, termos os operadores `||` e `&&`. Quem eu resolvo primeiro?\*/

---

---

---

---

**/\*Q03**

Sabe usar o operador ternário para colocar um entre dois valores em uma variável?\*/

---

---

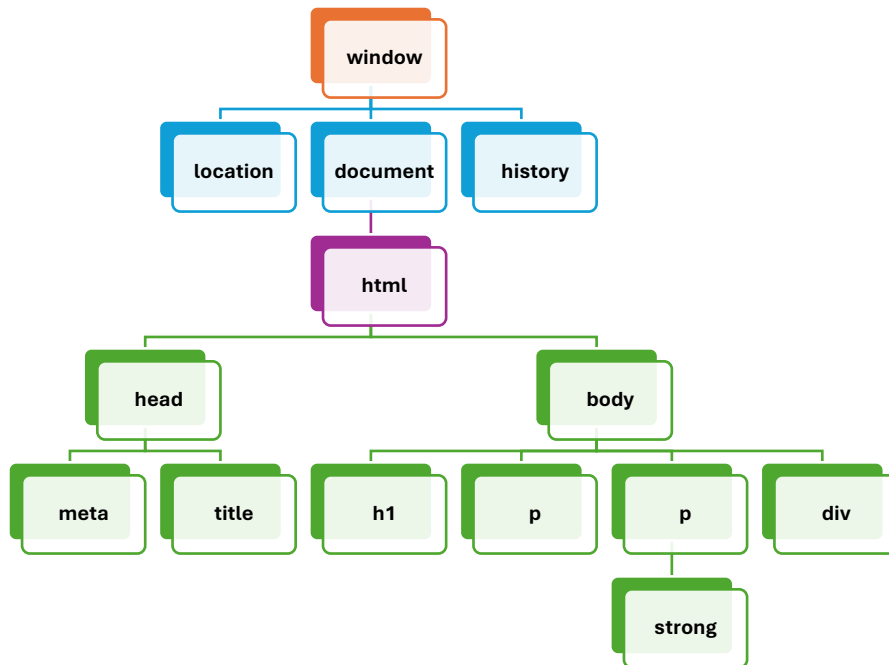
---

---

O que é?

**DOM** → **D**ocument **O**bject **M**odel

**Árvore DOM**



**Selecionando**

**por Marca** = `getElementsByTagName()`

**por ID** = `getElementById()`

**por Nome** = `getElementsByName()`

**por Classe** = `getElementsByClassName()`

**por querySelector** = `querySelector()` `querySelectorAll()`

# Aula 10

## Eventos DOM

**/\*Q01**

O que significa a sigla **DOM**?

---

---

---

**/\*Q02**

Sabe montar a **árvore DOM** de uma website simples?\*/

---

---

---

**/\*Q03**

O que são os elementos **Parent** e os elementos **Child** em uma árvore DOM?\*/

---

---

---

**/\*Q04**

Quais são os cinco **principais métodos** de selecionar elementos DOM dentro do JS?\*/

---

---

---



# Referência = Event Reference MDN

<https://developer.mozilla.org/en-US/docs/Web/Events>

Antes de falar sobre **Eventos** em JavaScript, precisamos falar entender o que são **Funções** em JS!

## Funções

```
function {
```

```
    bloco
```

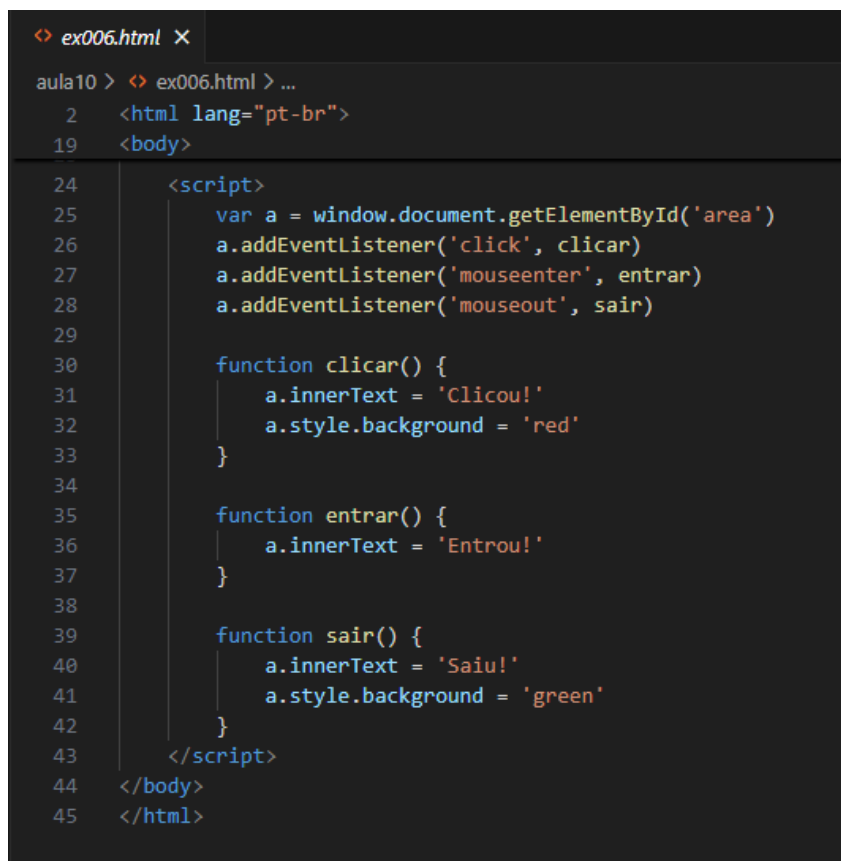
```
} //Função anonima
```

```
function ação() {
```

```
} //Funções de ação
```

```
function ação(param) {
```

```
} // Opcionalmente usar parametros para funções
```



```
ex006.html X
aula10 > ex006.html > ...
 2  <html lang="pt-br">
19  <body>
24  <script>
25      var a = window.document.getElementById('area')
26      a.addEventListener('click', clicar)
27      a.addEventListener('mouseenter', entrar)
28      a.addEventListener('mouseout', sair)
29
30      function clicar() {
31          a.innerText = 'Clicou!'
32          a.style.background = 'red'
33      }
34
35      function entrar() {
36          a.innerText = 'Entrou!'
37      }
38
39      function sair() {
40          a.innerText = 'Saiu!'
41          a.style.background = 'green'
42      }
43  </script>
44 </body>
45 </html>
```

ex007.html X

aula10 > ex007.html > ...

```
2  <html lang="pt-br">
23 <body>
24   <h1>Somando valores</h1>
25   <input type="number" name="txtn1" id="txtn1"> +
26   <input type="number" name="txtn2" id="txtn2">
27   <input id="cor" type="button" value="Somar" onclick="somar()">
28
29   <div id="res">Resultado</div>
30   <script>
31     function somar() {
32       var tn1 = window.document.getElementById('txtn1')
33       var tn2 = window.document.querySelector('input#txtn2')
34       var res = window.document.getElementById('res')
35       var n1 = Number(tn1.value)
36       var n2 = Number(tn2.value)
37       var s = n1 + n2
38       res.innerHTML = `A soma entre um ${n1} + ${n2} = ${s}`
39     }
40   </script>
41 </body>
42 </html>
```

# Módulo D: Condições em JavaScript

**/\*Q01**

Consegue citar **três eventos** que pode, acontecer com elementos DOM de um site?\*/

---

---

---

---

**/\*Q02**

O que é uma **função**? Como criar funções em JS\*/

---

---

---

---

**/\*Q03**

Sabe como **pegar os valores** de caixas de texto e fazer cálculos com eles?\*/

---

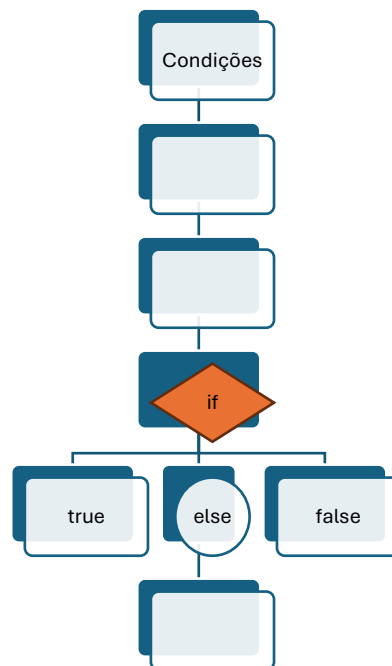
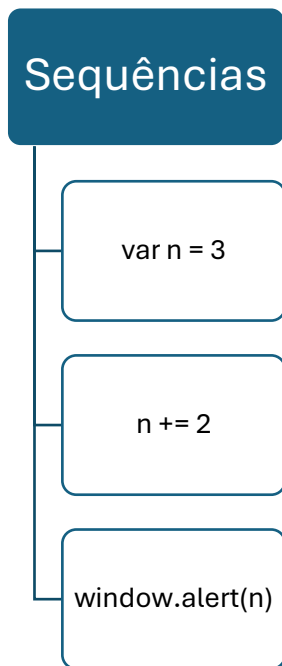
---

---

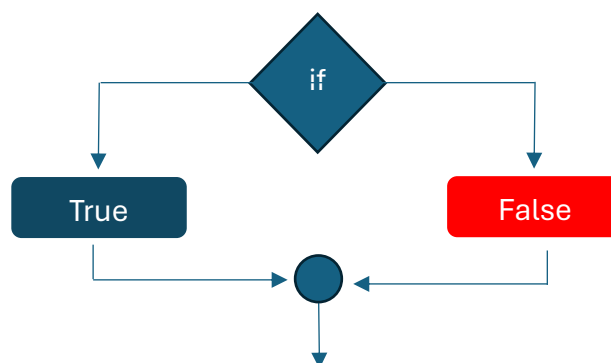
---

# Aula 11

## Condições (Parte 1)



```
if (condição) {  
    true  
} else {  
    false  
}
```

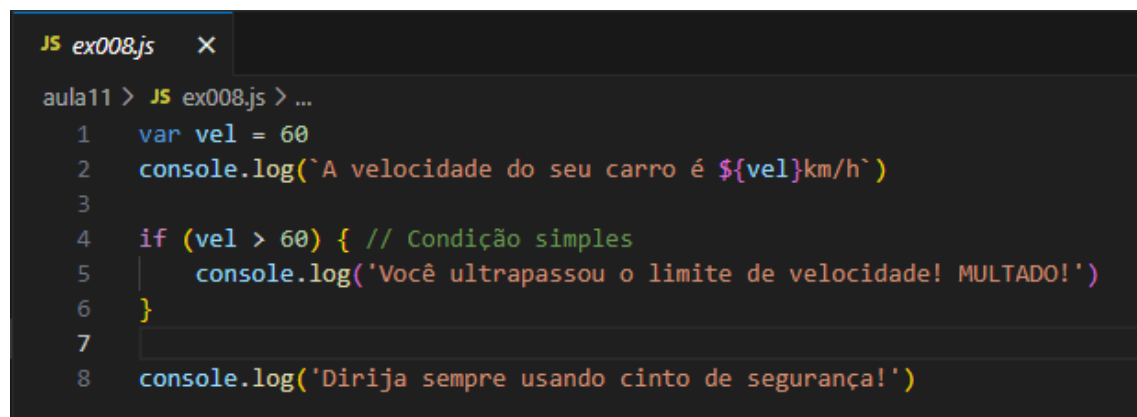


Obs: Crie um grafico para representar os blocos e as condições!

## Tipos de condição

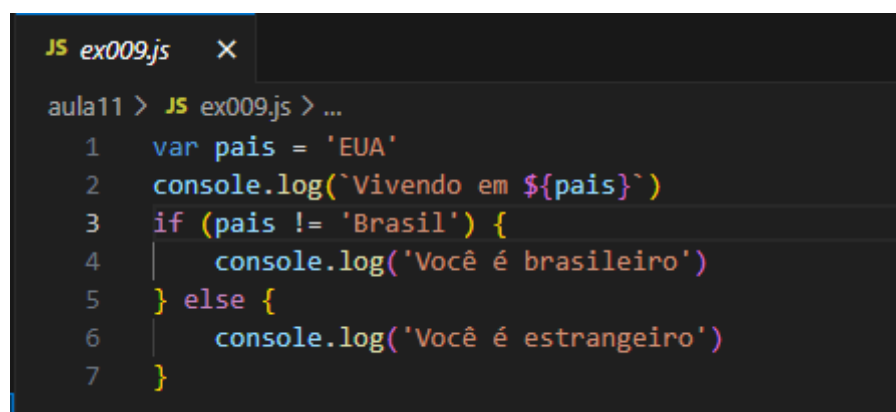
```
If (condição) {  
    true  
} //condição simples
```

```
If (condição) {  
    true  
} else {  
    false  
} //condição composta
```



The screenshot shows a terminal window with a tab labeled 'JS ex008.js'. The prompt is 'aula11 > JS ex008.js > ...'. The code being executed is as follows:

```
1 var vel = 60  
2 console.log(`A velocidade do seu carro é ${vel}km/h`)  
3  
4 if (vel > 60) { // Condição simples  
5     console.log('Você ultrapassou o limite de velocidade! MULTADO!')  
6 }  
7  
8 console.log('Dirija sempre usando cinto de segurança!')
```



The screenshot shows a terminal window with a tab labeled 'JS ex009.js'. The prompt is 'aula11 > JS ex009.js > ...'. The code being executed is as follows:

```
1 var país = 'EUA'  
2 console.log(`Vivendo em ${país}`)  
3 if (país != 'Brasil') {  
4     console.log('Você é brasileiro')  
5 } else {  
6     console.log('Você é estrangeiro')  
7 }
```

```
ex010.html X
aula11 > ex010.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>DETRAN</title>
7 </head>
8 <body>
9   <h1>Sistema de Multas</h1>
10  Velocidade do carro: <input type="number" name="txtvel" id="txtvel"> km/h
11  <input type="button" value="Verificar" onclick="calcular()">
12  <div id="res"></div>
13  <script>
14    function calcular() {
15      var txtv = window.document.querySelector('input#txtvel')
16      var res = window.document.querySelector('div#res')
17      var vel = Number(txtv.value)
18      res.innerHTML = `<p>Sua velocidade atual é de <strong>${vel}</strong> km/h</strong></p>`
19      if (vel > 60) {
20        res.innerHTML += `<p>Você está <strong>MULTADO</strong> por excesso de velocidade!</p>`
21      }
22      res.innerHTML += `<p>Dirija sempre com cinto de segurança</p>`
23    }
24  </script>
25 </body>
26 </html>
```

# Aula 11

## Condições (Parte 1)

**/\*Q01**

Já sabe como usar a extensão **Node Exec** no VSCode?\*/

---

---

---

---

**/\*Q02**

Sabe criar uma **condição** em seu programa JS?\*/

---

---

---

---

**/\*Q03**

Consegue diferenciar **condições simples** e as **condições compostas**?\*/

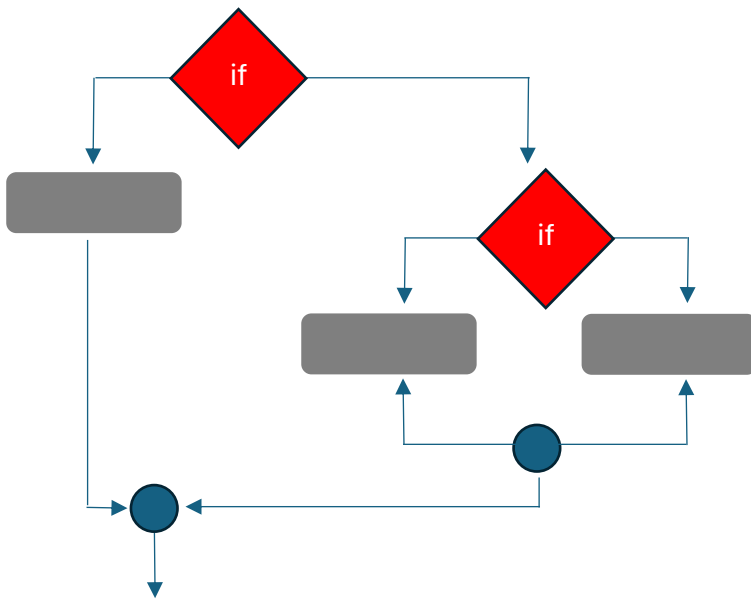
---

---

---

---

# Condições Aninhadas



```
if (cond1){
```

```
    bloco1
```

```
} else{
```

```
    if (cond2){
```

```
        bloco2
```

```
    } else{
```

```
        bloco3
```

```
    }
```

```
}
```

```
JS ex011.js X
aula12 > JS ex011.js > ...
1  var idade = 60
2  console.log(`Você tem ${idade} anos.`)
3  if (idade < 16) {
4      console.log('Não vota')
5  } else if (idade < 18 || idade > 65) {
6      console.log('Voto opcional')
7  } else {
8      console.log('Voto obrigatório')
9  }
```



JS ex012.js X

aula12 > JS ex012.js > ...

```
1  var agora = new Date()
2  var hora = agora.getHours()
3  console.log(`Agora são exatamente ${hora} horas`)
4  if (hora < 12) {
5      console.log('Bom Dia')
6  } else if (hora <= 18) {
7      console.log('Boa Tarde')
8  } else {
9      console.log('Boa Noite')
10 }
```

JS ex013.js X

aula12 > JS ex013.js > ...

```
1  var now = new Date()
2  var dayW = now.getDay()
3
4  //console.log(dayW)
5  switch (dayW) {
6      case 0:
7          console.log('Domingo')
8          break
9      case 1:
10         console.log('Segunda')
11         break
12     case 2:
13         console.log('Terça')
14         break
15     case 3:
16         console.log('Quarta')
17         break
18     case 4:
19         console.log('Quinta')
20         break
21     case 5:
22         console.log('Sexta')
23         break
24     case 6:
25         console.log('Sabado')
26         break
27     default:
28         console.log('[ERRO] Dia Inválido')
29         break
30 }
```

# Módulo D: Condições em JavaScript

## Exercícios JavaScript (Parte 1)

### Modelo do Site

```
aula12ex > ex015 > modelo.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>Modelo de Exercício</title>
7   <link rel="stylesheet" href="estilo.css">
8 </head>
9 <body>
10   <header>
11     <h1>Título</h1>
12   </header>
13   <section>
14     <div>
15       Testando...
16     </div>
17     <div>
18       Olá...
19     </div>
20   </section>
21   <footer>
22     <p>&copy; Danilo Menon</p>
23   </footer>
24   <script src="script.js"></script>
25 </body>
26 </html>
```

```
# estilo.css
aula12ex > ex015 > # estilo.css > header
1 body {
2   background: rgb(107, 107, 241);
3   font: normal 15pt Arial;
4 }
5
6 header {
7   color: white;
8   text-align: center;
9 }
10
11 section {
12   background: white;
13   border-radius: 10px;
14   padding: 15px;
15   width: 500px;
16   margin: auto;
17   box-shadow: 3px 3px 10px rgba(0, 0, 0, 0.267);
18 }
19
20 footer {
21   color: white;
22   text-align: center;
23   font-style: italic;
24 }
```

# Exercícios JavaScript (Parte 2)

```
aula12ex > ex015 > modelo.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>Modelo de Exercício</title>
7   <link rel="stylesheet" href="estilo.css">
8 </head>
9 <body>
10   <header>
11     <h1>Título</h1>
12   </header>
13   <section>
14     <div>
15       Testando...
16     </div>
17     <div>
18       Olá...
19     </div>
20   </section>
21   <footer>
22     <p>&copy; Danilo Menon</p>
23   </footer>
24   <script src="script.js"></script>
25 </body>
26 </html>
```

```
# estilo.css > # estilo.css > body
1 body {
2   background: rgb(107, 107, 241);
3   font: normal 15pt Arial;
4 }
5
6 header {
7   color: white;
8   text-align: center;
9 }
10
11 section {
12   background: white;
13   border-radius: 10px;
14   padding: 15px;
15   width: 500px;
16   margin: auto;
17   box-shadow: 3px 3px 10px rgba(0, 0, 0, 0.267);
18 }
19
20 section img {
21   width: 350px;
22   height: 350px;
23 }
24
25 div {
26   text-align: center;
27   padding: 8px;
28 }
29
30 footer {
31   color: white;
32   text-align: center;
33   font-style: italic;
34 }
```

```
JS script.js > JS script.js > carregar
1 function carregar() {
2   var msg = window.document.getElementById('msg')
3   var img = window.document.getElementById('imagem')
4   var data = new Date()
5   var hora = data.getHours()
6   var min = data.getMinutes()
7   var sec = data.getSeconds()
8   //var hora = 20
9   msg.innerHTML = `Agora são ${hora}:${min}:${sec}`
10   if (hora >= 0 && hora < 12) {
11     document.body.style.background = '#ffe36e'
12     //BOA DIA
13     img.src = 'fotomanha.png'
14   } else if (hora >= 12 && hora <= 18) {
15     //BOA TARDE
16     img.src = 'fototarde.png'
17     document.body.style.background = '#e9b976'
18   } else {
19     //BOA NOITE
20     img.src = 'fotonoite.png'
21     document.body.style.background = '#372c34'
22   }
23 }
```

# Módulo E: Repetições e JavaScript

## Aula 13

### Repetições (Parte 1)

**/\*Q01**

Sabe mudar a cor de fundo de um site **dinamicamente**, usando JS?\*/

---

---

---

---

**/\*Q02**

Consegue inserir uma imagem dentro de um site sem ter a tag **<img>** previamente criada?\*/

---

---

---

---

**/\*Q03**

Já treinou bastante o uso de **condições simples, compostas, aninhadas e múltiplas**?\*/

---

---

---

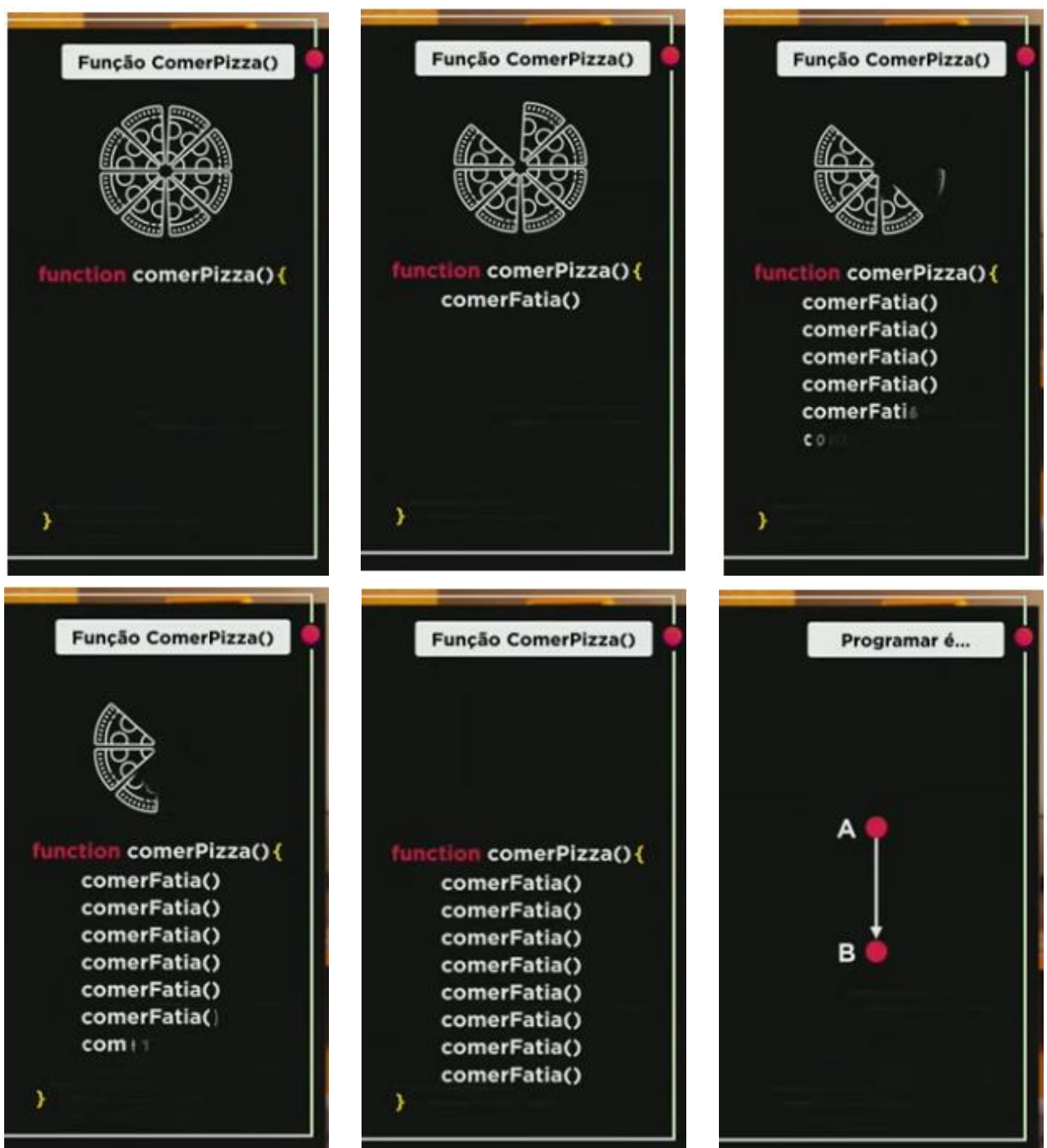
---

## RASCUNHO |

Essas estruturas de repetição podem ser chamadas de **laços** ou **iterações** em JavaScript.

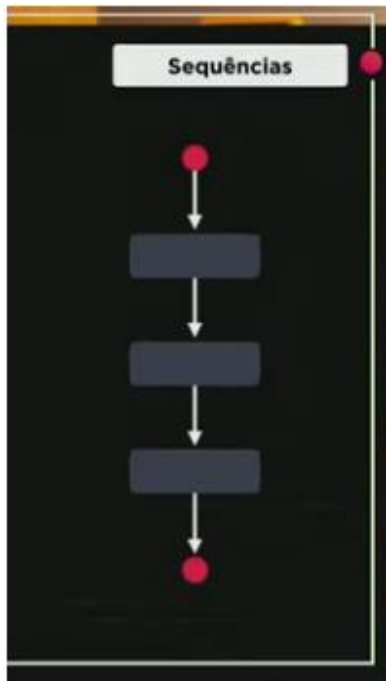
Repetições é tão simples que usaremos como exemplo comer uma pizza 🍕😊

A minha pizza tinha 8 pedaços, então abaixo eu tenho 8 comandos!

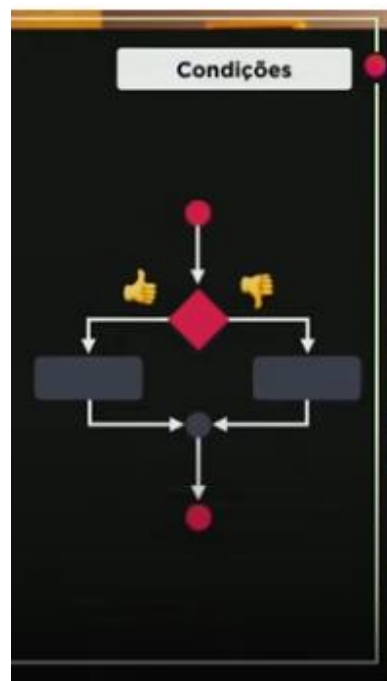


A minha atividade é partir do primeiro comando até o último comando até cumprir a minha missão

## Estrutura de controle



## Teste lógico



Às vezes nem tudo são sequências, temos desvios/possibilidades.

Para isso temos as condições, por exemplo:

Comprei uma pizza de 8 oito pedaços, mas é dividido em 2 sabores diferentes, pizza de calabresa e pizza de margarida, mas eu não gosto de marguerita

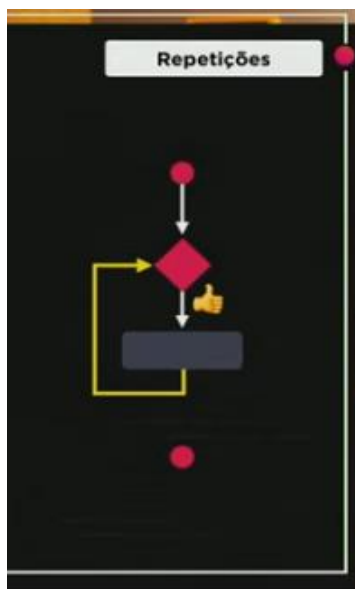
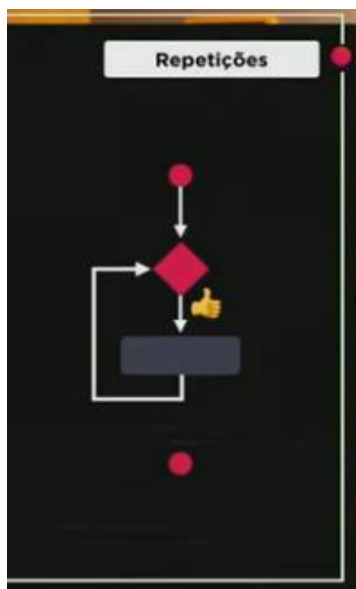
Se (if) a pizza não for de marguerita eu como

Se não (else if) eu não como

# Agora, veremos uma nova estrutura de controle!

## Repetições| Anteriormente citado Laço ou Iterações.

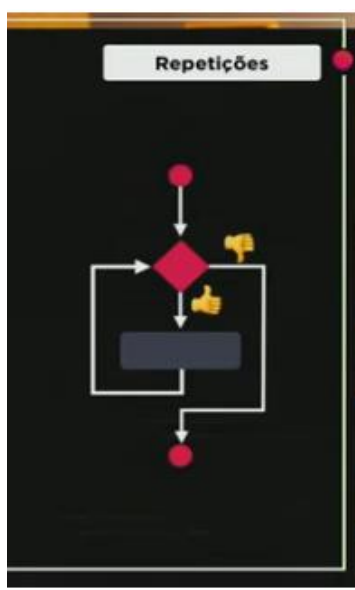
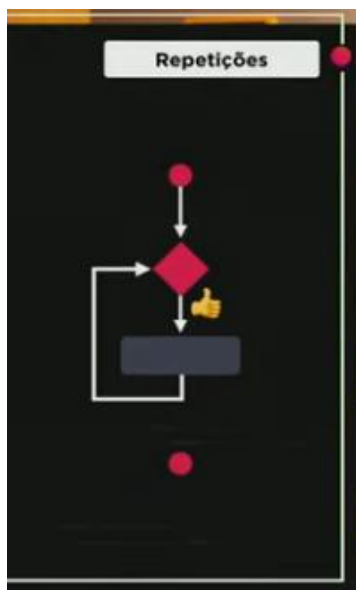
Um laço começa exatamente como uma condição, ele vai testar uma expressão.



### Observação:

Ele vai fazer o loop enquanto essa condição for verdadeira

Se o losango/condição for falsa o laço é quebrado!



E como eu escrevo essa estrutura no JavaScript?

```
while(condição) {  
  }  
}
```

Antigamente era comer fatia, comer fatia... 8 vezes!

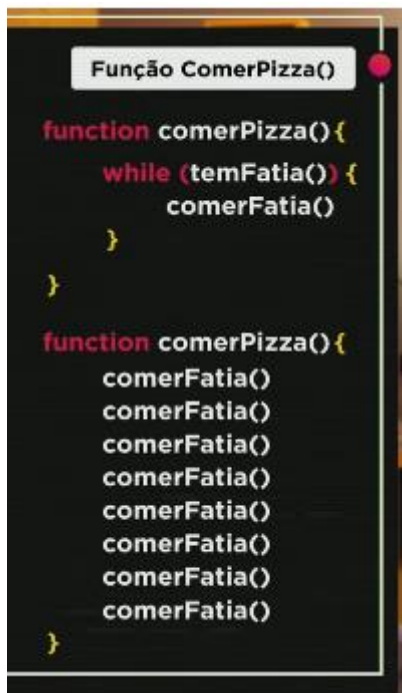
E se eu dividir essa pizza em 16, 20, 60 vezes? Ficaria cansativo demais!

Com a estrutura **while()** tudo ficara mais fácil!





Repare a diferença entre o primeiro programa e o segundo com a estrutura `while()`. O programa atual é muito mais eficiente que o primeiro, pois, o código anterior eu podia comer oito pedaços de pizza e o segundo programa eu consigo comer quantos pedaços eu quiser de pizza 🍕!



## Repetições (Parte 1)

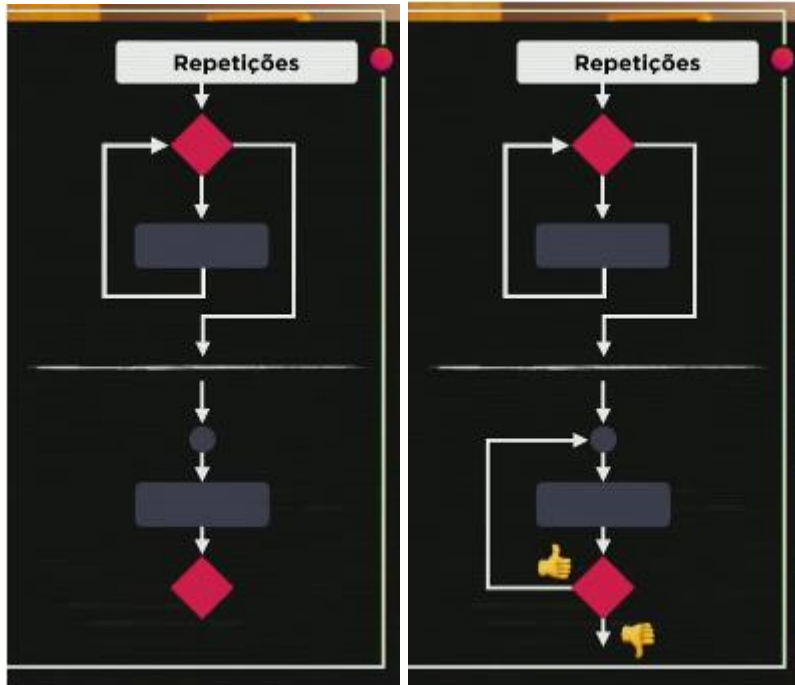
`while()` é classificada como estrutura de repetição com teste lógico no início!

```
JS ambiente.js M X
aula013 > JS ambiente.js
1 //Método menos inteligente
2
3 console.log('Tudo bem?')
4 console.log('Tudo bem?')
5 console.log('Tudo bem?')
6 console.log('Tudo bem?')
7 console.log('Tudo bem?')
8 console.log('Tudo bem?')
```

```
JS ambiente.js M X
aula013 > JS ambiente.js > ...
1 //Método inteligentes e eficaz
2
3 var c = 1
4
5 while (c <= 6) {
6     console.log(`Passo ${c}`)
7     c++ //c++ = c = c + 1
8 }
```

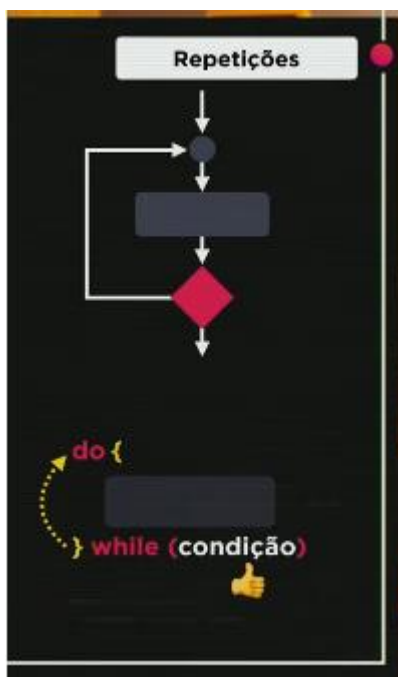
```
JS ambiente.js M X
aula013 > JS ambiente.js > ...
1 var c = 1
2
3 do {
4     console.log(`Passo ${c}`)
5     c++ //c++ = c = c + 1
6 } while (c <= 6)
7
```

Em vez de fazermos o teste logico no início e depois executar o bloco no final, temos a possibilidade de fazer a estrutura ao contrário, primeiro executa o bloco e depois faz o teste.



Ambos são estruturas de repetições, mas como comportamentos diferentes!

O bloco de cima primeiro testa e depois executa bloco, o de baixo primeiro executa bloco e depois faz o teste lógico.



Estrutura de repetição com teste lógico no início!

Estrutura de repetição com teste lógico no final!

# Aula 14

## Repetições (Parte 2)

**/\*Q01**

O que faço se quiser fazer uma determinada atividade acontecer **várias vezes**?\*/

---

---

---

---

**/\*Q02**

Sabe a diferenciar as estruturas de **condição** e as estruturas de **repetição**?\*/

---

---

---

---

**/\*Q03**

Consegue dizer a diferença entre estruturas **while** e a **do...while**?\*/

---

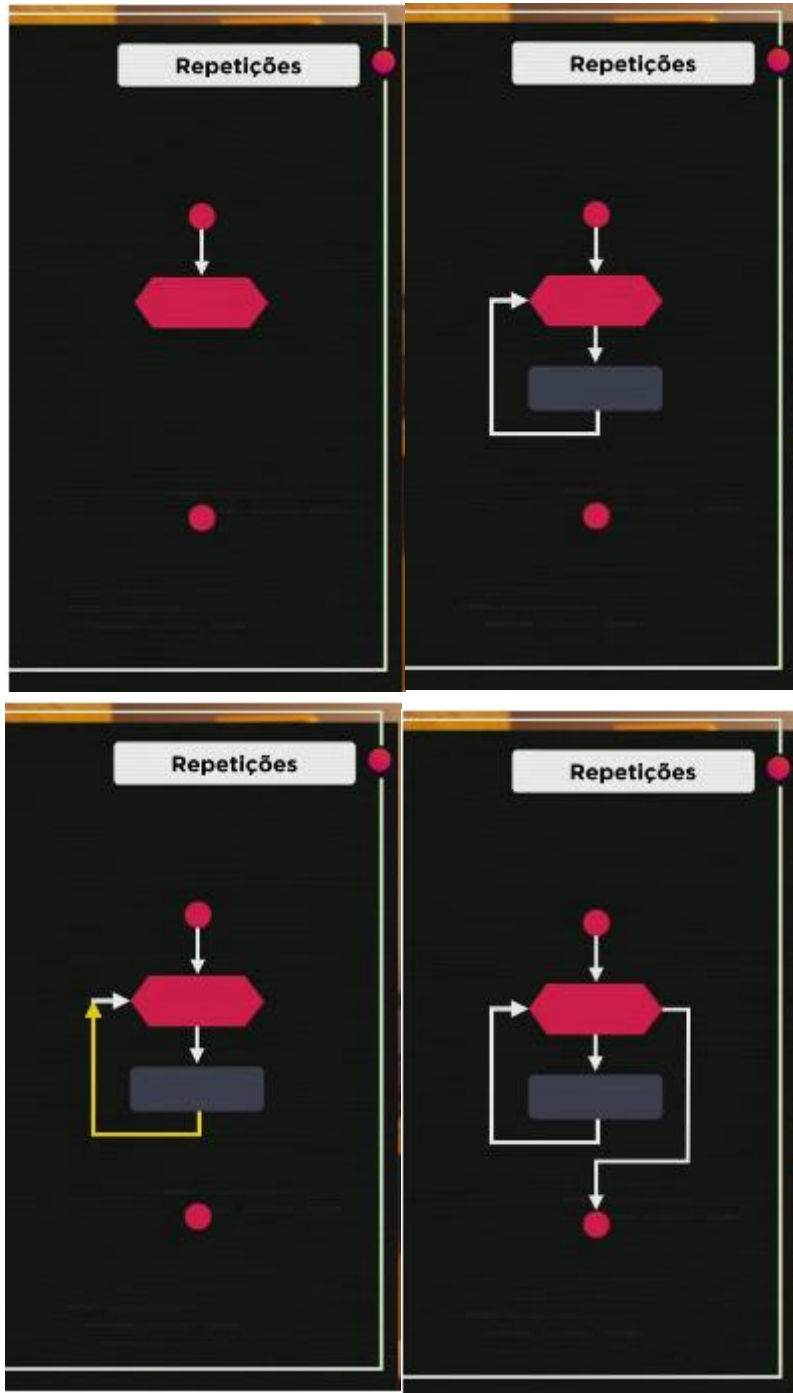
---

---

---

**Isso nos trás a nova estrutura de repetição que iremos ver, agora!**

Estrutura de repetição com variavel de controle



Inicialização;

Teste logico;

Incremento;

## Como se escreve o comando?

Utilizaremos o comando `for(início;teste;incr)`

Uma vez executado o bloco ele volta para cima e só que enquanto ele volta ele faz o incremento, isso acontecerá enquanto o teste lógico for verdadeiro! Se o teste lógico for falso ele segue o fluxo normal de execução do nosso script.

```
JS ambiente.js M X
aula014 > JS ambiente.js > ...
1  | for(var c = 1; c <= 10; c++){
2  |     console.log(c)
3  | }
4  |
5  |
```

# Aula 14

## Exercícios (Parte 4)

**/\*Q01**

Já aprendeu a usar direito a extensão Watch in Chrome no VSCode?\*/

---

---

---

---

**/\*Q02**

Já aprendeu a usar o Modo de Depuração do VSCode?\*/

---

---

---

---

**/\*Q03**

Já fez todos os testes com as estruturas **while**, **do...while** e **for**?\*/

---

---

---

---

**/\*Q04**

Quer aprender um pouco mais como manipular dados em **controles HTML** com JS?\*/

---

---

---

---

# Aula 15

## Variáveis Compostas

**/\*Q01**

Já treinou bastante o uso das estruturas de repetição **while**, **do...while** e **for**?\*/

---

---

---

---

**/\*Q02**

Apreendeu como manipular objetos de formulários **HTML** usando **JS**?\*/

---

---

---

---

**/\*Q03**

Seria bom tornar seus códigos menores e mais simplificados?\*/

---

---

---

---

**/\*Q04**

Você é daquele tipo de pessoa que ouviu alguém dizer que **“usar array é difícil”**?\*/

---

---

---

---

# Módulo F: Avançando os estudos em JavaScript

## Aula 15 | Variáveis Compostas

### Variáveis

- **Variáveis simples** só conseguem armazenar **um valor** por vez.
- **Variáveis compostas** deve ser capaz de armazenar **vários valores** em uma mesma estrutura.