

```
if ($(window).scrollTop() > header1_initialDistance) {  
    if (parseInt(header1.css('padding-top'), 10) >= header1_initialPadding) {  
        header1.css('padding-top', '' + $(window).scrollTop() - header1_initialDistance + header1_initialPadding + 'px');  
    }  
} else {  
    header1.css('padding-top', '' + header1_initialPadding + 'px');  
}  
  
if ($(window).scrollTop() > header2_initialDistance) {  
    if (parseInt(header2.css('padding-top'), 10) >= header2_initialPadding) {  
        header2.css('padding-top', '' + $(window).scrollTop() - header2_initialDistance + header2_initialPadding + 'px');  
    }  
}
```

DESENVOLVIMENTO DE APLICAÇÕES PARA INTERNET

PROF. DR. WENDEL CASTRO

wendelcastro20@gmail.com



INTRODUÇÃO AO JAVASCRIPT

- Com a linguagem de programação JavaScript podemos construir páginas extremamente dinâmicas e interativas.
- O foco do JavaScript é implementar o comportamento ou a inteligência das páginas web.



Settings

Preferences

Workspace

Blackboxing

Devices

Throttling

Geolocations

Shortcuts

Preferences

☒ Bracket matching

☐ Code folding

Show whitespace characters: None

☒ Display variable values inline while debugging

☒ Enable CSS source maps

Default indentation: 4 spaces

Elements

☐ Show rulers

☐ Show user agent shadow DOM

☒ Word wrap

☒ Show HTML comments

☒ Reveal DOM node on hover

☒ Show detailed inspect tooltip

☒ Autocomplete from history

☒ Group similar

☒ Eager evaluation

☒ Evaluate triggers user activation

☐ Preserve log upon navigation

☐ Enable custom formatters

Extensions

Link handling: auto

Persistence

☐ Enable Local Overrides

Debugger

☐ Disable JavaScript

☐ Disable async stack traces

DevTools

☐ Auto-open DevTools for popups

☐ Emulate a focused page

TESTANDO O PODER DO JAVASCRIPT NO DEVTOOLS (SETTINGS)

Settings

Preferences

Workspace

Blackboxing

Devices

Throttling

Geolocations

Shortcuts

Preferences

☒ Bracket matching

☐ Code folding

Show whitespace characters: None

☒ Display variable values inline while debugging

☒ Enable CSS source maps

Default indentation: 4 spaces

Elements

☐ Show rulers

☐ Show user agent shadow DOM

☒ Word wrap

☒ Show HTML comments

☒ Reveal DOM node on hover

☒ Show detailed inspect tooltip

☒ Autocomplete from history

☒ Group similar

☒ Eager evaluation

☒ Evaluate triggers user activation

☐ Preserve log upon navigation

☐ Enable custom formatters

Extensions

Link handling: auto

Persistence

☐ Enable Local Overrides

Debugger

☐ Disable JavaScript

☐ Disable async stack traces

DevTools

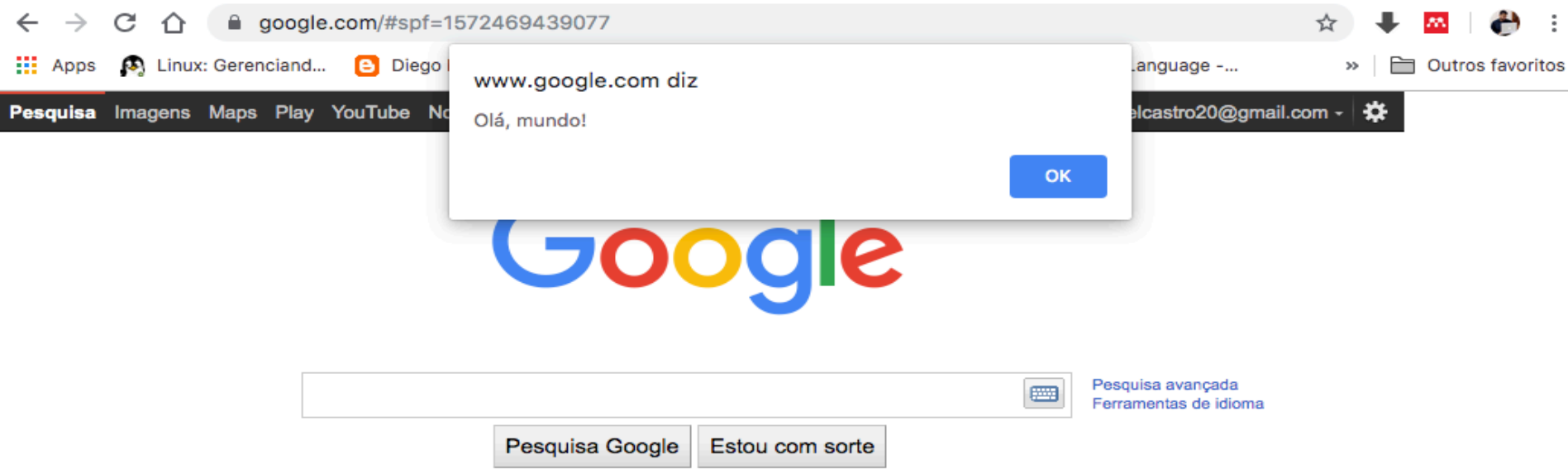
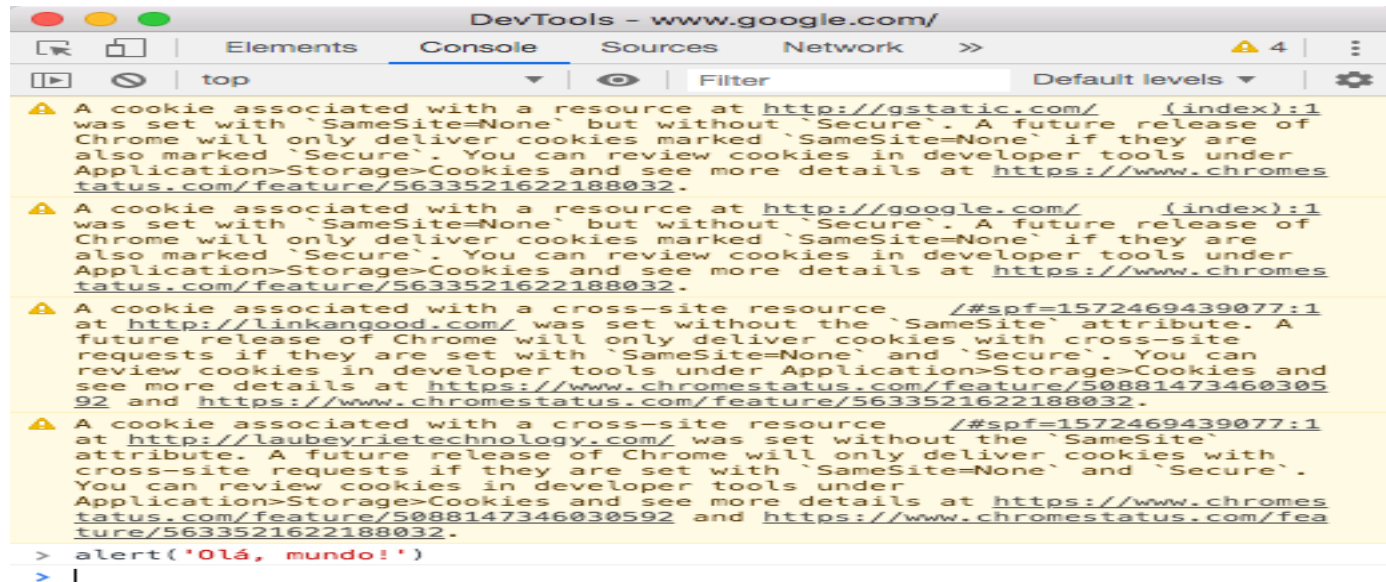
☐ Auto-open DevTools for popups

☐ Emulate a focused page

Desabilitar



TESTANDO O PODER DO JAVASCRIPT NO GOOGLE DEVTOOLS (CONSOLE)



TESTANDO O PODER DO JAVASCRIPT NO GOOGLE DEVTOOLS (CONSOLE)

Alguns comando:

- `alert('Olá, mundo!')`
- `document.body.style.background = 'red'`
- `document.querySelector('img#hplogo').style.visibility = 'hidden'`



INTRODUÇÃO AO JAVASCRIPT

- **Aplicando JavaScript ao HTML**
- Há duas formas de associar código JavaScript em documentos HTML.
- **JavaScript interno**

```
<script >  
    alert ('Desenvolvimento para Internet');  
</ script >
```

- Nessa abordagem, o código JavaScript fica “amarrado” a um único documento HTML.



INTRODUÇÃO AO JAVASCRIPT

```
<!DOCTYPE html>
<html>
  <head>
    <title>Documento</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <script>
      alert('Olá mundo')
    </script>
  </body>
</html>
```



INTRODUÇÃO AO JAVASCRIPT

- **JavaScript externo**
- O código JavaScript pode ser definido em arquivos separados e depois associados aos documentos HTML através do elemento **script**.

```
alert ("Desenvolvimento para Internet ");
```

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

- Nessa outra abordagem, podemos reutilizar o mesmo código JavaScript em vários documentos HTML.



INTRODUÇÃO AO JAVASCRIPT

○ Carregamento

- Esse elemento pode ser adicionado dentro do corpo dos elementos **head** e **body**. A localização do elemento **script** afeta o momento no qual o código JavaScript será carregado pelos navegadores.
- No exemplos iremos ver a diferença: o elemento **script** foi colocado dentro do corpo do **head** e posteriormente no corpo do **Body**.



```
<!DOCTYPE html>
<html>
  <head>
    <title>Documento</title>
    <meta charset="UTF-8">
    <script>
      alert('Olá mundo2')
      confirm('Está gostando de JavaScript?2')
      prompt('Quantos anos você tem?2')
    </script>
  </head>
  <body>
    <script>
      alert('Olá mundo')
      confirm('Está gostando de JavaScript?')
      prompt('Quantos anos você tem?')
    </script>
  </body>
</html>
```

INTRODUÇÃO AO JAVASCRIPT

○ Variáveis

- A criação de variáveis é feita através da palavra chave **var**.
- Toda variável deve ter um **nome** (identificador).
- Tipos de Variáveis:
 - **booleano, constante, null, undefined, object e function.**
- Sintaxe:

```
var nome-da-variável = valor-da-variável;
```



INTRODUÇÃO AO JAVASCRIPT

○ Exemplo de atribuição de variável (var):

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Variáveis</title>
```

```
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  </head>
```

```
  <body>
```

```
    <script>
```

```
      var valor1 = 10;
```

```
      var valor2 = 3.14;
```

```
      document.write(valor1 + " é um valor inteiro, já " + valor2 + " é um ponto flutuante!\n");
```

```
    </script>
```

```
    <br>
```

```
    <script>
```

```
      var nome = "André";
```

```
      var cidade = "Rio Claro";
```

```
      document.write("Meu nome é " + nome + " e sou de " + cidade);
```

```
    </script>
```

```
  </body>
```

```
</html>
```

INTRODUÇÃO AO JAVASCRIPT

- Exemplo de atribuição de variável com prompt e alert):

```
<!DOCTYPE html>

<html>
  <head>
    <title>Variáveis</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <script>
      var nome = prompt('Digite seu nome: ')
      var idade = prompt('Digite a sua idade: ')
      alert('Olá! seu nome é: ' + nome + ' você tem ' + idade + ' anos')
    </script>
  </body>
</html>
```

INTRODUÇÃO AO JAVASCRIPT

○ TRATAMENTO DE DADOS

- O comando `prompt()`, nos retorna `String`, por isso é importante tratar os dados quando não forem `String`. (`parseInt`, `parseFloat`, etc..)

```
<!DOCTYPE html>

<html>
  <head>
    <title>Variáveis</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <script>
      var nome = prompt('Digite seu nome: ');
      var idade = parseInt(prompt('Digite a sua idade: '));
      alert('Olá! seu nome é: ' + nome + ' você tem ' + idade + ' anos');
    </script>
  </body>
</html>
```

TRATAMENTO DE DADOS

- Na versão atual do JavaScript não há necessidade de utilizar `parseInt` ou `parseFloat`, hoje basta utilizar o `number()`, e ele já converte para qualquer número, independente de ser inteiro ou fracionado.
- Conversão para String
 - `→ String(n)`
 - `→ n.toString()`



INTRODUÇÃO AO JAVASCRIPT

- Booleanos: Retorna verdadeiro ou falso (true/false).

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Tipos booleanos</title>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
</head>
```

```
<body>
```

```
<script>
```

```
var nome = false;
```

```
if (nome == false) {
```

```
    document.write("A variável booleana está <b>FALSA</b>.")
```

```
} else {
```

```
    document.write("A variável booleana está <b>VERDADEIRA</b>.")
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

Fazer um exemplo de booleanos com prompt

INTRODUÇÃO AO JAVASCRIPT

- **Null**: A variável é vazia.
- **Object**: São conhecidas como Object Array, que são uma lista de elementos.
- **Function**: Valor retornado de uma expressão.



INTRODUÇÃO AO JAVASCRIPT

- As variáveis são classificadas em dois tipos, em qualquer linguagem de programação, que são as globais e as locais.
 - **Variável Global**: Criada ou declarada fora de uma função, portanto podem ser utilizadas a qualquer momento no seu script.
 - **Variável Local**: Criada ou declarada dentro de uma função, portanto só podem ser utilizadas dentro da função criada.
- Obs: **Variáveis Locais** precisam da instrução **var**.



INTRODUÇÃO AO JAVASCRIPT

○ Exemplo:

Global

Local

```
<script>
```

```
var nome = "José";  
document.write("Nome é " + nome + "");
```

```
function exemplo1() {
```

```
var nome = "Luiz";  
document.write("Nome é " + nome + "");  
}
```

```
</script>
```

```
<body>
```

```
  <script>
```

```
    exemplo1();
```

```
    document.write(nome);
```

```
  </script>
```

```
</body>
```

INTRODUÇÃO AO JAVASCRIPT

○ Operadores

Aritmético	Atribuição	Relacional	Lógico
+	=	==	&&
-	+=	!=	
*	-=	<	!
/	*=	<=	
%	/=	>	
**	%=	>=	
	++		
	--		

INTRODUÇÃO AO JAVASCRIPT

○ Aritmético

Símbolo	Operação
+	Soma
-	Subtração
*	Multiplicação
/	Divisão
%	Resto da Divisão
**	potência



INTRODUÇÃO AO JAVASCRIPT

- Exemplo:

$$4 + 2 = 6$$

$$4 - 2 = 2$$

$$2 * 2 = 4$$

$$10 / 2 = 5$$

$$10 \% 3 = 1$$

Precedência (Hierarquia nas operações)


Hierarquia	Operação
1	Parênteses
2	Função
3	-, + (unários)
4	^
5	*, /, \, %
6	+, -

EXERCÍCIO:

- Criar um script que faça a média entre três números e mostre em um alerta no browser, esses valores devem ser passados pelo usuário.



INTRODUÇÃO AO JAVASCRIPT

- Seguindo a precedência, realize a seguinte operação:
 - $X = 7$
 - $Y = x / (6 - 2 - (3 * 5) / (16 - 1));$
 - Obs: Existe também os operadores de **potenciação** e **raiz**, contudo podemos representar pelos métodos:
 - **Math.pow** e **Math.sqrt**
- 

INTRODUÇÃO AO JAVASCRIPT

- Concatenação de Strings
- Sintaxe

```
<!DOCTYPE html>

<html>
  <head>
    <title>Concatenação de String</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <script>
      var nome = 'Wendel';
      var espaco = " ";
      var sobrenome = 'Castro';
      var concatenacao = nome + espaco + sobrenome;
      document.write(concatenacao);
    </script>
  </body>
</html>
```

INTRODUÇÃO AO JAVASCRIPT

○ Atribuição

Símbolos	Operações
=	Simples
+=	Incremental
-=	Decremental
*=	Multiplicativa
/=	Divisória
%=	Modular
++	Incremento
--	Decremento



INTRODUÇÃO AO JAVASCRIPT

- **Pré e Pós Incremento ou Pré e Pós Decremento**
- Os operadores “++” e “--” podem ser utilizados de duas formas diferentes, antes ou depois de uma variável numérica.



```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Atribuição</title>
```

```
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  </head>
```

```
  <body>
```

```
    <script>
```

```
      var i = 10;
```

```
      var j = ++i ;
```

```
      alert(j);
```

```
    </script>
```

```
    <script>
```

```
      var i = 10;
```

```
      var j = i++;
```

```
      alert(j);
```

```
    </script>
```

```
  </body>
```

INTRODUÇÃO AO JAVASCRIPT

- O que seria exibido nos exemplos abaixo?

```
var i = 10;  
var j = ++i + i--;  
alert (j);
```



INTRODUÇÃO AO JAVASCRIPT

○ Relacionais

Símbolos	Operações
<code>==</code>	Igualdade
<code>!=</code>	Diferença
<code><</code>	Menor
<code><=</code>	Menor ou igual
<code>></code>	Maior
<code>>=</code>	Maior ou igual



INTRODUÇÃO AO JAVASCRIPT

○ Lógicos

Símbolos	Operações
&&	E
	OU
!	Não

$a > 0.2$	$b < 0.8$	$a > 0.2 \ \&\& \ b < 0.8$
V	V	V
V	F	F
F	V	F
F	F	F

$a > 0.2$	$b < 0.8$	$a > 0.2 \ \ b < 0.8$
V	V	V
V	F	V
F	V	V
F	F	F



DOM (DOCUMENT OBJECT MODEL)

○ Exemplo 1

```
<!DOCTYPE html>

<html>
  <head>
    <title>Eventos - Document Object Model (DOM)</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
      div#area{
        font: normal 20pt Arial;
        background: blue;
        color: white;
        width: 400px;
        height: 400px;
        line-height: 400px; /*controlar o espaçamento entre as linhas de um texto
        text-align: center;
      }
    </style>
  </head>
  <body>
    <div id="area">Eventos de Interação</div>
  </body>
</html>
```

```

<script>
//***** por marca *****
//-- [0] quer dizer a primeira tag <p> que ele ver
var p1 = window.document.getElementsByTagName('p')[0]
//escrevendo no doc html o que o p1 tem
window.document.write("<-->" + p1.innerHTML)
p1.style.color = 'black'
//com alert
window.alert(p1.innerHTML)

//***** por ID *****
|
//por Nome
//por Classe
//por Seletor - mais novo e indicado
</script>

```

```

<div class="msg">Iniciando</div>

```

```

<script>

```

```

//*****por Classe *****
var classe = window.document.getElementsByClassName('msg')[0]
classe.style.color = 'black'

//*****por Seletor - mais novo e indicado*****
var seletor = window.document.querySelector('div.msg')
seletor.style.color = 'blue'

/*

```

```

<div id="msg">Iniciando</div>

```

```

<script>

```

```

/*
//***** por marca *****
//-- [0] quer dizer a primeira tag <p> que ele ver
var p1 = window.document.getElementsByTagName('p')[0]
//escrevendo no doc html o que o p1 tem
window.document.write("<-->" + p1.innerHTML)
//passando um style para p1
p1.style.color = 'black'
//com alert
window.alert(p1.innerHTML)
*/

//***** por ID *****
//USAREMOS A DIV QUE SÓ TEMOS UMA COLOCANDO UM ID NELA
var id = window.document.getElementById('msg')
//passando um style
id.style.background = 'black'
//mudando o conteúdo da mensagem
id.innerHTML = 'Iniciando com o DOM'

```

EXEMPLO 2

Eventos de Interação

```
<body>
  <!--chamando o evento na div area - onclick="clicar()" -->
  <div id="area" onclick="clicar()">Eventos de Interação</div>

  <script>
    function clicar(){
      var area = window.document.getElementById('area')
      //evento para mudar o texto
      area.innerHTML = "Você clicou..."
    }
  </script>
</body>
```

```
<body>
  <!--chamando o evento na div area - onclick="clicar()" -->
  <div id="area" onclick="clicar()" onmousemove="mover()" onmouseout="sair()">Eventos de Interação</div>

  <script>
    //tornando a variável global
    var area = window.document.getElementById('area')

    function clicar(){
      area.innerHTML = "Você clicou..."
    }

    function mover(){
      area.innerHTML = "Você moveu o mouse..."
    }

    function sair(){
      area.innerHTML = "Você saiu..."
    }
  </script>
</body>
```

```
<div id="area">Eventos de Interação</div>
```

```
<script>
```

```
//Chamando eventos com "ouvidores - Listener" - tirando os Eventos do HTML  
var area = window.document.getElementById('area')  
area.addEventListener('click',clicar) //evento e função como parâmetro  
area.addEventListener('mousemove',mover)  
area.addEventListener('mouseout',sair)
```

```
function clicar(){  
    area.innerHTML = "Você clicou..."  
    area.style.background = 'red'  
}
```

```
function mover(){  
    area.innerHTML = "Você moveu o mouse..."  
}
```

```
function sair(){  
    area.innerHTML = "Você saiu..."  
    area.style.background = 'black'  
}
```

```
</script>
```

EXEMPLO 03

Somando dois Valores:

+

Resultado

```
<head>
<title>Trabalhando com soma</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
  body{
    font: normal 20pt Arial;
  }
  input{
    font: normal 20pt Arial;
    width: 100px;
  }
</style>
</head>
<body>
<h1>Somando dois Valores:</h1>
<input type="number" name="txtn1" id="txtn1"> +
<input type="number" name="txtn2" id="txtn2">
<input type="button" value="Somar" onclick="somar()">
<div id="result">Resultado</div>
<script>
  function somar(){
    var n1 = window.document.getElementById('txtn1')
    var n2 = window.document.getElementById('txtn2')
    var num1 = Number(n1.value) //convertendo o campo de texto para número
    var num2 = Number(n2.value)
    var resultado = num1 + num2
    var r = window.document.getElementById('result')//pegar o valor da DIV (Resultado)
    r.innerHTML = resultado
  }
</script>
```

INTRODUÇÃO AO JAVASCRIPT

- **Controle de fluxo**
- **if e else**
- **Estrutura de condição simples:**
- **Sintaxe:**

```
if (condição) {  
    ação  
}
```

- **Sintaxe em pseudocódigo:**

```
se (condição) for  
verdadeira {  
    execute as ações  
}
```



INTRODUÇÃO AO JAVASCRIPT

- Exemplos práticos:



INTRODUÇÃO AO JAVASCRIPT

- **Estrutura de condição composta:**

```
if (condição) {  
    ação 1  
} else {  
    ação 2  
}
```

- **Exemplo prático.**



INTRODUÇÃO AO JAVASCRIPT

- **While**
- **Sintaxe:**

```
while (comparação) {  
    código que será executado  
}
```



INTRODUÇÃO AO JAVASCRIPT

○ For

```
for (declaração 1; declaração 2; declaração 3) {  
    Bloco de ação a ser executado  
}
```

```
for (i = 0; i <= 10; i++) {  
    document.write(i)  
}
```



INTRODUÇÃO AO JAVASCRIPT

○ Função

- `function Nome_da_Função () {
 <Comandos>
}`
- `function Nome_da_Função(parâmetro) {
 <Comandos>
}`

