

JavaScript

Introdução e sintaxe

```
if (top !== self) {  
  function calcWidth() {  
    var wW = 0;  
    if (typeof window.innerWidth == 'number') {  
      wW = window.innerWidth;  
    } else if (document.documentElement && documentElement.clientWidth) {  
      wW = document.documentElement.clientWidth;  
    } else if (document.body && document.body.clientWidth) {  
      wW = document.body.clientWidth;  
    }  
    if (eH = document.documentElement.scrollHeight) {  
      var wH = window.innerHeight || document.documentElement.clientHeight;  
      var eW = !document.all && (eH > wH) ? wH : wW;  
      // ... ('menu', 'width', eW);  
    }  
  }  
}
```

Para que serve o JS?

Como é uma linguagem de programação, os navegadores trazem um mecanismo de JS que permite executar JS no cliente, tornando as páginas interativas.

Além disso, o JavaScript pode servir para gerenciar a lógica do back-end com o node.js.

JS pode servir para desenvolver...

Jogo:

[Cookie Clicker](#)

[Sonic](#)

Relógio:

[Relógio - CSS + JS](#)

Calculadora:

[Calculator JS](#)

[Calculator JS 2](#)

Efeitos especiais:

[Efeito 1](#)

[Efeito 2](#)

[Efeito 3](#)

E muito mais

E-mails, bate-papos, gráficos de dados, relógios, calculadoras, formulários, etc...

Variáveis

Como declarar uma nova variável:

```
var nomeDaVariavel = valor
```

```
let nomeDaVariavel = valor
```

Alterar valor de uma variável

```
nomeDaVariavel = valor
```

Tipos básicos

"pepe"	string
10.7	number
true	boolean
{ }	object
[]	array
function() {}	function
null	null
undefined	undefined

Operadores aritméticos

<code>var num1 = 5</code>	
<code>var num2 = 3</code>	
<code>num1 + num2</code> <code>num1 - num2</code>	adição, subtração
<code>num1 * num2</code> <code>num1 / num2</code>	multiplicação, divisão
<code>num1 % 5</code>	módulo
<code>num1++</code>	incremento
<code>num1--</code>	decremento

Operadores de atribuição

<code>var numero = 10</code>	
<code>numero += 2</code>	<code>numero = numero + 2</code>
<code>numero -= 2</code>	<code>numero = numero - 2</code>
<code>numero *= 2</code>	<code>numero = numero * 2</code>
<code>numero /= 2</code>	<code>numero = numero / 2</code>

Operadores de comparação simples

==	igualdade de valor
!=	valor diferente
<=	menor e igual
>=	maior ou igual

Operadores de comparação estrita

===	igualdade de valor e tipo
!==	valor ou tipo diferente

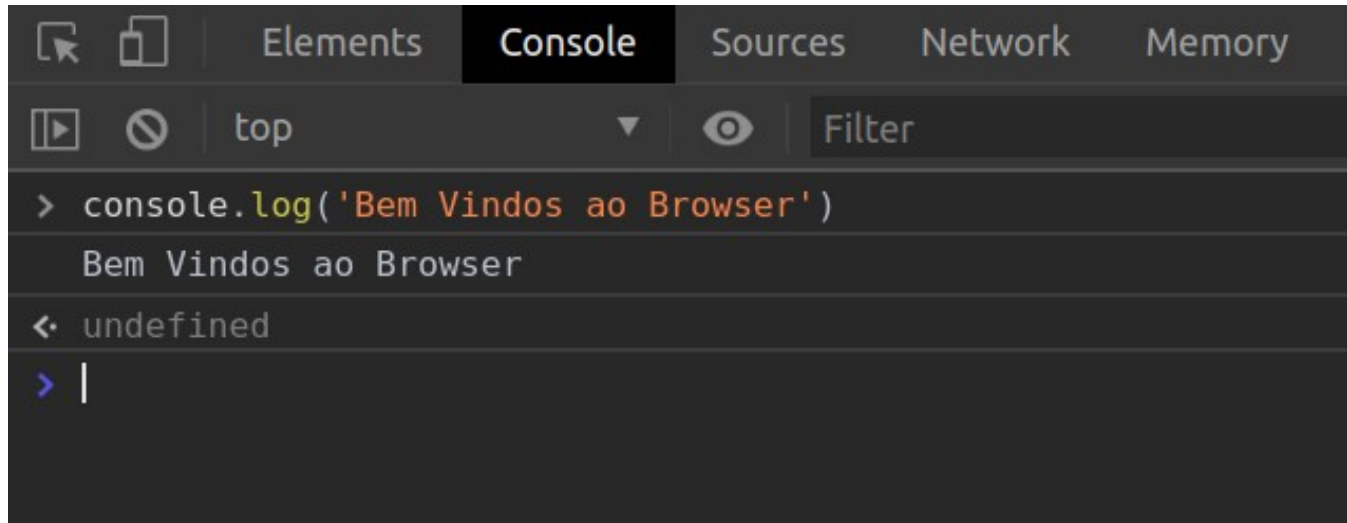
Operadores Lógicos

&&	Operador and (e)
	Operador or (ou)
!	Operador de negação

A linha de comando

Os navegadores trazem uma linha de comando integrada para interpretar JavaScript.

Normalmente usamos F12 para abri-la e começar a escrever o código.



The image shows a screenshot of a web browser's developer console. The 'Console' tab is selected, showing a single log entry. The entry is a message 'Bem Vindos ao Browser' in orange text, preceded by a green '>' symbol. Below the message, the text 'Bem Vindos ao Browser' is displayed in white. At the bottom of the console, there is a prompt '> |' in blue, indicating that the console is ready for the next command. The console is part of a larger interface with tabs for 'Elements', 'Console', 'Sources', 'Network', and 'Memory'. The 'Console' tab is currently active. The interface has a dark theme.

```
> console.log('Bem Vindos ao Browser')  
Bem Vindos ao Browser  
< undefined  
> |
```

“ JavaScript é a linguagem mais popular hoje em dia em Fullstack, de acordo com a StackOverflow [Stackoverflow 2018](#) “

IF - Sintaxe

```
if (condição) {  
    // fazer isto caso a condição seja  
    verdadeira  
} else {  
    // fazer isto caso a condição seja falsa  
}
```

IF - Exemplo

```
var diaDaSemana = 'Domingo'

if (diaDaSemana == 'Domingo') {
  console.log('Hoje é dia de churrasco!')
} else {
  console.log('Hoje não é dia de churrasco :(')
}
```

Resultado: Hoje é dia de churrasco!

IF ternário - Sintaxe

condição ? **expressão1** : **expressão2**

condição Qualquer expressão **booleana**

expressão1 Expressão retornada se **condição** for **true**

expressão2 Expressão retornada se **condição** for **false**

IF ternário - Exemplo



```
var diaDaSemana = 'Segunda-feira'
```

```
diaDaSemana == 'Domingo'
```

```
? console.log('hoje é dia de churrasco!')
```

```
: console.log('hoje não é dia de churrasco :(')
```

Resultado: Hoje não é dia de churrasco :(

Switch



```
var fruta = "mamão";
```

```
switch (fruta) {  
  case "abacaxi":  
    console.log("Me ajuda ona descascar esse  
    abacaxi?");  
    break;  
  case "mamão":  
    console.log("É mamão com açúcar!");  
    break;  
  default:  
    console.log("É outra fruta");  
}
```

For - Sintaxe

```
for (inicio; condicao; incremento) {  
    // fazer isso enquanto a condição for  
    verdadeira  
}
```

For - Exemplo

```
for (let i = 0; i < 3; i++) {  
  console.log('Olá ' + i)  
}
```

Resultado: Olá 0 Olá 1 Olá 2

For - Break

```
for (let i = 0; i < 4; i++) {  
  console.log('Olá ' + i);  
  if(i === 1){  
    break; // interrompe o loop FOR  
  }  
}
```

Resultado: Olá 0 Olá 1

While - Sintaxe

```
while (condição) {  
    // executar enquanto a condição  
    for verdadeira  
}
```

While - Exemplo

```
let numero = 0
while (numero < 3) {
  console.log('Olá ' + numero)
  numero++
}
```

Resultado: Olá 0 Olá 1 Olá 2

Do While - Sintaxe

```
do {  
    // código a ser executado  
} while (condição);
```

O do while irá executar o código que está entre o do {} antes de verificar a condição.