



# JAVASCRIPT

Prof. Marcelo

## Sumário

1. COMPILADOR E INTERPRETADORES.....	3
1.1 Interpretador.....	3
1.2 Compilador.....	3
2. A LINGUAGEM DE PROGRAMAÇÃO JAVASCRIPT.....	4
3. OPERADORES ARITMÉTICOS JS.....	5
3.1 Aritméticos.....	5
3.2 Comparação.....	6
4. FUNÇÕES INTERNAS DE ENTRADA E SAÍDA DE DADOS.....	7
5. O CONSOLE.LOG JS.....	7

## 1. COMPILADOR E INTERPRETADORES

O computador deve converter os comandos dados em linguagem de **alto nível** para linguagem de máquina (códigos binários). Esta tarefa de conversão é feita por um programa especial de computador, isto é, um programa que recebe as instruções em linguagem de alto nível e dá como saída outro programa constituído de instruções binárias.

1. **Linguagens de programação de baixo nível** são aquelas cujos símbolos são uma representação direta do código de máquina. Ex: **Assembly**.
2. **Linguagem de programação de alto nível** são aquelas composta de símbolos inteligível pelo ser humano e não-executável diretamente pela máquina. Exemplo: Pascal, Java, Python.

### 1.1 Interpretador

O programa conversor recebe a primeira instrução do programa fonte, confere para ver se está escrita corretamente, converte-a em linguagem de máquina e então ordena ao computador que execute esta instrução.

Depois repete o processo para a segunda instrução, e assim sucessivamente, até a última instrução do programa fonte. Quando a segunda instrução é trabalhada, a primeira é perdida, isto é, apenas uma instrução fica na memória em cada instante.

Se este programa fonte for executado uma segunda vez, novamente haverá uma nova tradução, comando por comando, pois os comandos em linguagem de máquina não ficam armazenados para futuras execuções. Neste método, o programa conversor recebe o nome de **interpretador**.

### 1.2 Compilador

O programa conversor recebe a instrução do programa fonte, verifica se está escrita corretamente, converte-a para **linguagem de máquina** obtendo sucesso passa para a próxima instrução, repetindo o processo sucessivamente até a última instrução do

programa fonte. Caso tenha terminado a transformação da última instrução do programa fonte e nenhum erro tenha sido detectado, o computador volta à primeira instrução, já transformada para linguagem de máquina e executa-a. Passa à instrução seguinte, executa-a, etc., até a última.

Se este programa for executado uma segunda vez, não haverá necessidade de uma nova tradução, uma vez que todos os comandos em linguagem binária foram memorizados em um novo programa completo. Neste método, o programa conversor recebe o nome de **compilador**.

## 2. A LINGUAGEM DE PROGRAMAÇÃO JAVASCRIPT

O JavaScript outrora chamado de LiveScript foi criado pela Netscape (Brendan Eich ) nos anos 90, época em que as páginas web eram totalmente estáticas. Foi por este motivo que o JavaScript foi criado, para dar interação e vida às páginas html.



Figura 1: Brendan Eich

Segundo o site da Developer Mozilla o JavaScript é uma linguagem de programação versátil e interpreta. Sua sintaxe é semelhante ao Java com algumas derivação do Python. É muito comum imaginarem que o JS tem algum parentesco direto com o Java, o que não é verdade.

### 3. OPERADORES ARITMÉTICOS JS

Os operadores são símbolos especiais que tem um significado próprio para a linguagem e estão associados a determinadas operações.

Existem **operadores aritméticos** como a soma (+) ou a subtração (-), **operadores relacionais** como a comparação de igualdade (==) e operadores de cadeias de caracteres como a concatenação (+). A sintaxe dos operadores em Javascript é baseada nas linguagens C, C++ e Java.

#### 3.1 Aritméticos

Operadores	Significado	Exemplos	Explicação
+, -	Soma / Subtração	a+b	Soma o valor da variável a com o valor da variável b.
*, /	Multiplicação / Divisão	x * 2 Soma / 3	Multiplica o valor da variável x por 2. Divide o valor da variável soma por 3.
%	Resto da divisão	x % 2	Calcula o resto da divisão do valor de x por 2. Ou seja, retorna 0 se x for um número par ou 1 se for ímpar.
++	Incremento	i++ ++i	Incrementa o valor de i. Ou seja i passa a valer i+1.
--	Decremento	i-- --i	Decrementa o valor de i. Ou seja i passa a valer i-1.
=	Atribuição Simples	Nome = "Mauricio"; Media = (a+b+c) / 3;	Armazena a cadeia de caracteres "Mauricio" dentro da variável Nome. Calcula a soma a+b+c, divide-a por 3 e armazena o resultado na variável Media.
+=	Atribuição Composta	S+=3;	Soma o valor de S com 3 e armazena o resultado na própria variável S. É equivalente a S=S+3;
-=	Atribuição Composta	A -= 1;	Subtrai o valor de A de 1 e armazena o resultado na própria variável A. É equivalente a A=A-1;
*=	Atribuição Composta	P*=10;	Multiplica o valor de P por 10 e armazena o resultado na própria variável P. É equivalente a P=P*10;
/=	Atribuição Composta	X /= 2;	Divide o valor de X por 2 e armazena o resultado na própria variável X. É equivalente a X = X / 2;
%=	Atribuição Composta	Y %=2;	Divide o valor de Y por 2 e armazena o resto desta divisão na própria variável Y. É equivalente a Y = Y % 2;

Atenção, o operador de soma (+) também é usado para concatenar (unir) strings. Alguns inconvenientes podem ocorrer com uma simples soma entre valores. Para evitar algum problema de interpretação utilize as funções **parseInt()** e **parseFloat()**.

A função `parseInt` converte seu primeiro argumento para uma string, analisa, e retorna um inteiro. Já a `parseFloat` analisa um argumento string, e retorna um número de ponto flutuante. Ex.:

```
var i = 1;
var n = 2;
var r = parseInt(i) + parseFloat(n);
```

Exemplo de **concatenação** de strings no JavaScript:

```
var s1 = "Bom";
var s2 = "Dia";
var bd = s1 + " " + s2; // Bom dia;
```

## 3.2 Comparação

Operadores	Significado
==	Igualdade
===	Idêntico
!=	Desigualdade
!==	Não idêntico
<	Menor que
>	Maior que
<=	Menor ou igual
>=	Maior ou igual

## 3.3 Lógico

Operadores	Significado
AND lógico (&&)	(E lógico) - Retorna <code>expr1</code> caso possa ser convertido para falso; senão, retorna <code>expr2</code> . Assim, quando utilizado com valores booleanos, <code>&amp;&amp;</code> retorna verdadeiro caso ambos operandos sejam verdadeiros; caso contrário, retorna falso.
OU lógico (  )	(OU lógico) - Retorna <code>expr1</code> caso possa ser convertido para verdadeiro; senão, retorna <code>expr2</code> . Assim, quando utilizado com valores booleanos, <code>  </code> retorna verdadeiro caso ambos os operandos sejam verdadeiros; se ambos forem falsos, retorna falso.
NOT lógico (!)	(Negação lógica) Retorna falso caso o único operando possa ser convertido para verdadeiro; senão, retorna verdadeiro.

## 4. FUNÇÕES INTERNAS DE ENTRADA E SAÍDA DE DADOS

O JavaScript possui algumas funções “métodos” internas para entrada e saída de dados, são elas:

**alert( )**: Emite uma caixa de diálogo com uma mensagem;

**confirm()**: Emite uma mensagem com dois botões o cancelar e o ok, retorna um valor booleano;

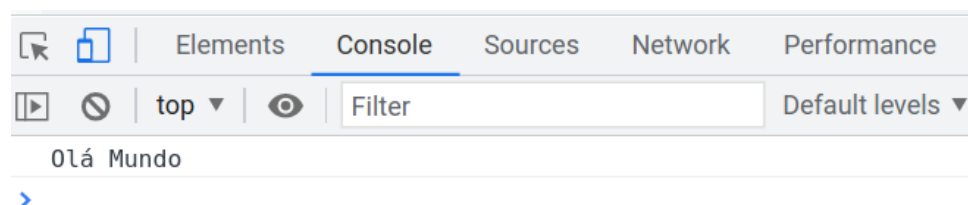
**prompt()**: Captura uma entrada de dados em uma caixa de diálogo

**document.write()**: Escreve em um documento html

## 5. O CONSOLE.LOG JS

O console.log é um método interno que permite depurar comandos em um console presente nos navegadores na ferramenta DevTools. Este método deve ser “chamado” dentro de um arquivo JS para que a saída de dados seja informada no console. Ex.:

**console.log(“Olá Mundo”);**



*Figura 2: Console do Navegador*

Para abrir o console do navegador, na maioria dos *browsers* a tecla de atalhos é a F12.

## REFERÊNCIAS

[Gerra dos Navegadores]

[https://pt.wikipedia.org/wiki/Guerra\\_dos\\_navegadores](https://pt.wikipedia.org/wiki/Guerra_dos_navegadores)

[Compilador e Interpretador]

<http://www.inf.ufsc.br/~j.barreto/cca/arquitet/arq4.htm>

[JavaScript]

<http://shipit.resultadosdigitais.com.br/blog/javascript-1-uma-breve-historia-da-linguagem/>

[Operadores]

<http://www.nce.ufrj.br/ginape/js/conteudo/variaveis/operadores.htm>

[https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Expressions_and_Operators#operadores_logicos)

[Expressions\\_and\\_Operators#operadores\\_logicos](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Expressions_and_Operators#operadores_logicos)