



Ponto e vírgula

Na linguagem JavaScript não requer ponto e vírgula(;) ao final da linha, caso você tenha apenas um comando por linha.

```
x += 10
```

Portanto, se você tem mais de um comando por linha, você deverá colocar o ponto e vírgula (;)

```
x += 10; y -= 5;
```



Variáveis

Regras de nomes para as variáveis:

A variável pode conter letras, números e os símbolos (\$ e _);

Espaço e pontuação não é permitido

O primeiro caractere não pode ser número

Os nomes são case sensitive



Variáveis do tipo string

Variáveis do tipo string recebem o conteúdo entre aspas simples(') ou duplas("")

```
saudacao = "Olá";
```

```
alerta = 'cuidado';
```

Quando você precisar passar uma informação com aspas simples, você pode usar a dupla para abrir/fechar a string, ou vice-versa.

Também pode ser usada uma contrabarra(\) antes das aspas que estão no meio da informação.

```
saudacao = "Olá, \“seja bem-vindo\”!";
```



Variáveis numéricas

Para criar a variável numérica é preciso apenas atribuir um valor a mesma, sem o uso de aspas.

```
cont = 40;
```

```
temperatura = 38.5;
```

Para especificar um número decimal é necessário utilizar o ponto(.)



Vetores

Assim como em outras linguagens de programação, o JavaScript implementa vetores.

```
brinquedos = ['bola', 'quebra-cabeça', 'boneca'];
```

O array pode ser multidimensional.

```
face = [ ['R', 'G', 'Y'],  
         ['W', 'R', 'O'],  
         ['Y', 'W', 'G']];
```

```
document.write(face[1][2]); //Saída O
```



Vetor associativo

Também comumente chamado de array associativo, nesse caso o elemento é acessado através do seu nome e não pelo índice numérico.

```
balls = {"golf": "Golf balls, 6",  
"tennis": "Tennis balls, 3",  
"soccer": "Soccer ball, 1",  
"ping": "Ping Pong balls, 1 doz"}
```

```
document.write(balls['soccer']) //Soccer ball, 1
```



Operações com vetores

Concatenação

```
pets = ["Cat", "Dog", "Fish"];  
more_pets = pets.concat("Rabbit", "Hamster");
```

Iteração

```
pets = ["Cat", "Dog", "Rabbit", "Hamster"];  
pets.forEach(output);
```

```
function output(element, index, array){  
    document.write("Element at index " + index + " has the value " + element + "<br>");  
}
```



Operações com vetores

Inserindo elementos

```
sports = ["Football", "Tennis", "Baseball"];  
sports.push("Hockey");
```

Removendo elementos

```
sports = ["Football", "Tennis", "Baseball"];  
removed = sports.pop(); //Remove o último elemento do array
```

Invertendo o vetor

```
sports = ["Football", "Tennis", "Baseball", "Hockey"];  
sports.reverse();
```




Operações com vetores

Ordenando o vetor - Ordem alfabética crescente

```
sports.sort();
```

Ordenando o vetor - Ordem alfabética decrescente

```
sports.sort().reverse();
```



Operadores matemáticos

Operador	Descrição
+	Soma
-	Subtração
*	Multiplicação
/	Divisão
%	Resto da divisão
++	Incremento
--	Decremento



Operadores de atribuição

Operador	Exemplo
=	j = 99;
+=	j += 2;
-=	j -= 12;
*=	j *= 2;
/=	j /= 6;
%=	j %= 7;

Operadores de comparação

Operador	Descrição
==	É igual
!=	Não é igual
>	Maior que
<	Menor que
>=	Maior ou igual
<=	Menor ou igual
===	É exatamente igual
!==	Não é exatamente igual



Operadores lógicos

Operador	Descrição
&&	.E.
	.OU.
!	.NÃO.



Manipulação de string

Concatenação

```
nome = "Maria";  
nome += " da Silva";
```

Sequência de escape

```
cabecalho = "Nome \t Idade";
```

\t - tabulação

\n - Quebra de linha

\b - Backspace

\' - Aspas simples

\\" data-bbox="188 787 200 819"> - Aspas duplas



Tipos de variáveis

Obter o tipo da variável

```
document.write(typeof n)
```

Conversão

```
n= "123"; //String
```

```
n=n*1; //Converte para inteiro ou parseInt()
```

```
n=123 //Inteiro
```

```
n = n+ "" //Converte para string ou toString()
```



Funções

O JavaScript permite que o desenvolvedor crie funções específicas.

```
function produto(a, b){  
    return a*b;  
}
```