

# O GUIA QUE VOCÊ DEVE TER PARA PROGRAMAR

Exemplos em JavaScript pt-BR e JavaScript

## OPERADORES

### Atribuição

= (a = 2)

### Atribuição Composta

+= (a += n → a = a + n)

-= (a -= n → a = a - n)

\*= (a \*= n → a = a \* n)

/= (a /= n → a = a / n)

### Comparação

< (menor que)

> (maior que)

<= (menor ou igual a)

>= (maior ou igual a)

### Aritméticos

Adição: +

Subtração: -

Multiplicação: \*

Divisão: /

### Igualdade

== (1 == '1' verdadeiro)

=== (1 === '1' falso)

!= (1 != '1' falso)

!== (1 !== '1' verdadeiro)

### Incremento ou Decremento

++ (a++ como a = a + 1)

-- (a-- como a = a - 1)

### Acesso à Propriedade do Objeto

. (como em console.log)

### Lógicos

&& (e)

|| (ou)

## PRINCIPAIS TIPOS DE VALORES

Número (Number)

Texto (String)

Objeto (Object)

Lógico (Boolean) { verdadeiro ou falso  
(sim ou não)  
true or false

## COMENTÁRIOS

// Esse é um comentário de uma linha

/\* Esse comentário  
pode ter  
várias linhas \*/

## COMANDOS PARA CONTROLE DA EXECUÇÃO

### se

```
se (condição) {  
  ações se a condição for satisfeita  
} senão {  
  ações se a condição não for satisfeita  
}
```

### if

```
if (condition) {  
  actions if the condition is satisfied  
} else {  
  actions if the condition is not satisfied  
}
```

### quebrar (break)

Encerra o comando de repetição ou o caso do comando *escolher* (*switch*).

### continuar (continue)

Provoca o término da iteração corrente.

### escolher

```
escolher (expressão) {  
  caso opção1:  
    ações se a expressão coincidir com opção1  
    quebrar  
  caso opção2:  
    ações se a expressão coincidir com opção2  
    quebrar  
}
```

### switch

```
switch (expression) {  
  case option1:  
    actions if expression matches option1  
    break  
  case option2:  
    actions if expression matches option2  
    break  
}
```

### tentar...capturar

```
tentar {  
  ações  
} capturar (expressão) {  
  ações  
}
```

### try...catch

```
try {  
  actions  
} catch (expression) {  
  actions  
}
```

### lançar (throw)

Lança uma exceção que pode ser capturada em um comando *capturar* (*catch*).

## COMANDOS PARA REPETIÇÃO

### enquanto

```
enquanto (condição) {  
  ações se a condição for satisfeita  
}
```

### while

```
while (condition) {  
  actions if the condition is satisfied  
}
```

### fazer...enquanto

```
fazer {  
  ações  
} enquanto (condição)
```

repete as ações se a condição for satisfeita

### do...while

```
do {  
  actions  
} while (condition)
```

repeat actions if the condition is satisfied

### para

```
para (i = 0; i < 10; i++) {  
  ações se a condição for satisfeita  
}
```

### for

```
for (i = 0; i < 10; i++) {  
  actions if the condition is satisfied  
}
```

### para...em

```
para (x em y) {  
  ações por propriedade  
}
```

### for...in

```
for (x in y) {  
  actions per property  
}
```

### para...de

```
para (x de y) {  
  ações por elemento  
}
```

### for...of

```
for (x of y) {  
  actions per element  
}
```