

## Objetivo:

- I. Operadores aritméticos;
- II. Operadores de comparação;
- III. Operadores lógicos;
- IV. Estruturas de decisão if else;
- V. Estrutura condicional switch.

### I. Operadores aritméticos

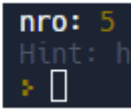
Na programação existem basicamente os mesmos operadores matemáticos, destacando-se:

- `%` é o operador de módulo (resto da divisão). Por exemplo: `11 % 3` irá resultar em 2;
- `**` é o operador de exponenciação (potência). Por exemplo: `2 ** 3` irá resultar em 8.

No JavaScript os operadores matemáticos podem ser utilizados na notação condensada:


- `nro += 2` é equivalente a instrução `nro = nro + 2`. A forma condensada só poderá ser utilizada quando a mesma variável é usada para leitura (à direita da atribuição) e para escrita (à esquerda da atribuição). Por exemplo:

```
nro = 3
nro += 2
console.log("nro:", nro)
```




- `nro -= 1` é equivalente a instrução `nro = nro - 1`. Por exemplo:

```
nro = 3
nro -= 1
console.log("nro:", nro)
```



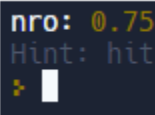
- `nro *= 4` é equivalente a instrução `nro = nro * 4`. Por exemplo:

```
nro = 3
nro *= 4
console.log("nro:", nro)
```



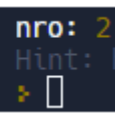
- `nro /= 4` é equivalente a instrução `nro = nro / 4`. Por exemplo:

```
nro = 3
nro /= 4
console.log("nro:", nro)
```



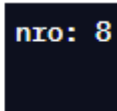
- `nro %= 11` é equivalente a instrução `nro = nro % 3`. Por exemplo:

```
nro = 11
nro %= 3
console.log("nro:", nro)
```



- `nro **= 3` é equivalente a instrução `nro = nro ** 3`. Por exemplo:

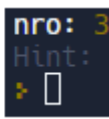
```
nro = 2
nro *= 3
print("nro:", nro)
```



Existem ainda os operadores de incremento e decremento:

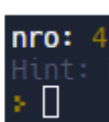
- O operador de incremento é usado para somar 1 no valor da variável. No exemplo a seguir, a instrução `n++` é uma forma condensada de escrever `nro = nro + 1`:

```
nro = 2
nro++
console.log("nro:", nro)
```



- O operador de decremento é usado para subtrair 1 no valor da variável. No exemplo a seguir, a instrução `n--` é uma forma condensada de escrever `nro = nro - 1`:

```
nro = 5
nro--
console.log("nro:", nro)
```



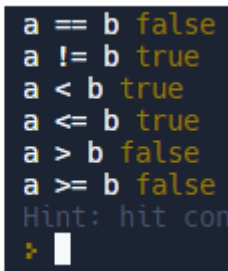
## II. Operadores de comparação

Um operador de comparação sempre compara dois lados e a resposta sempre será true ou false. Por exemplo:

- `a == b` o resultado será true se `a` e `b` forem iguais e false caso contrário;
- `a != b` o resultado será true se `a` e `b` forem diferentes e false caso contrário;
- `a < b` o resultado será true se `a` for menor que `b` e false caso contrário;
- `a <= b` o resultado será true se `a` for menor ou igual `b` e false caso contrário;
- `a > b` o resultado será true se `a` for maior que `b` e false caso contrário;
- `a >= b` o resultado será true se `a` for maior ou igual `b` e false caso contrário.

Exemplos dos operadores de comparação:

```
a = 2
b = 5
console.log("a == b", a == b)
console.log("a != b", a != b)
console.log("a < b", a < b)
console.log("a <= b", a <= b)
console.log("a > b", a > b)
console.log("a >= b", a >= b)
```



Observação: na linguagem JavaScript não podemos encadear operadores de comparação. A comparação `c < b < a` resulta em um valor errado, pois a comparação deve ser em pares. O correto é usar o operador lógico and (`&&`) para unir as expressões de comparação, por exemplo, `c < b && b < a`, ou seja, temos duas comparações independentes.

```
a = 2
b = 5
c = 9
console.log(c < b < a)
```

↑ ↑

É errado encadear operadores de comparação

### III. Operadores lógicos

Existem os operadores **and** (&&), **or** (||) e **not** (!). Esses operadores operam sempre com valores booleanos (true ou false). Os operadores && e || precisam de dois operandos booleanos e o operador not usa apenas um operando.

O operador && opera usando dois valores booleanos e o resultado é um valor booleano. Existem quatro possibilidades de operações usando o operador &&, veja que o resultado é true apenas quando ambos os operandos são true:

```
console.log("true && true", true && true)
console.log("true && false", true && false)
console.log("false && true", false && true)
console.log("false && false", false && false)
```

true && true	true
true && false	false
false && true	false
false && false	false

O operador || opera usando dois valores booleanos e o resultado é um valor booleano. Existem quatro possibilidades de operações usando o operador ||, veja que o resultado é true quando algum dos operandos é true:

```
console.log("true || true", true || true)
console.log("true || false", true || false)
console.log("false || true", false || true)
console.log("false || false", false || false)
```

true    true	true
true    false	true
false    true	true
false    false	false

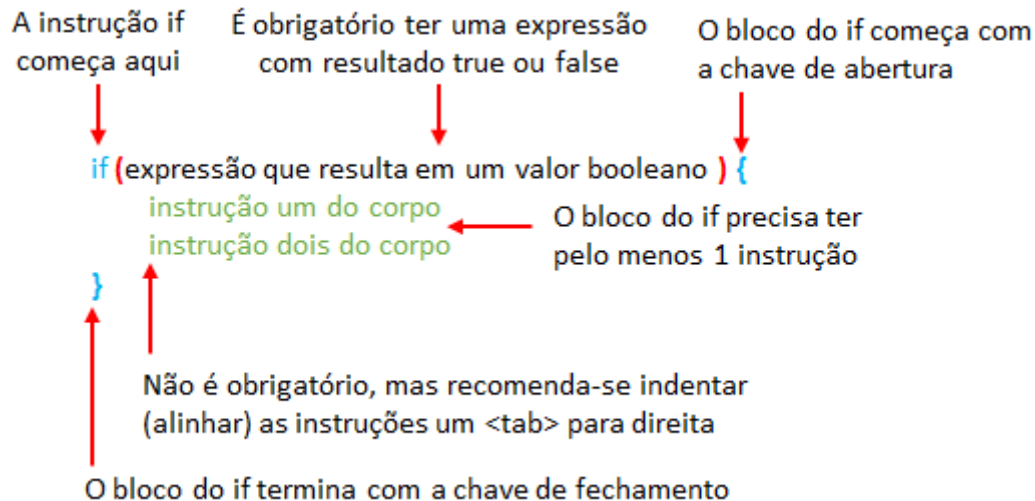
O operador ! inverte um valor booleano:

```
console.log("! true", ! true)
console.log("! false", ! false)
```

! true	false
! false	true

### IV. Estruturas de decisão if else

O **if** (se em português) é a principal estrutura de decisão. Ele possui a seguinte estrutura:



A estrutura de decisão if é usada para direcionar o fluxo de execução do programa. Em outras palavras, podemos decidir se um bloco de instruções será executado. Como exemplo:

```
a = 2
b = 5
if( a < b ){
    console.log("Dentro do corpo do if")
    console.log("a < b é true")
}
console.log("Instrução fora do bloco do if")
```

Dentro do corpo do if  
a < b é true  
Instrução fora do bloco do if

Podemos anexar o bloco **else** ao bloco **if**. O bloco **else** será executado somente se a expressão do **if** resultar em false. Desta forma temos que, somente um dos blocos será executado.

```
if (expressão que resulta em um valor booleano) {
    instrução um do corpo
    instrução dois do corpo
}
else {
    instrução um do corpo
    instrução dois do corpo
}
```

O bloco if será executado se a expressão resultar em true

O bloco else será executado se a expressão resultar em false

Veja como exemplo o código a seguir. Nele foi executado o bloco do **if**, mas teria sido executado o bloco **else** se a variável **a** tivesse valor 6.

```
a = 2
b = 5
if( a < b ){
    console.log("a é menor que b")
}
else{
    console.log("a não é menor que b")
}
console.log("Fim do programa")
```

```
a é menor que b
Fim do programa
```

As instruções if e else podem ser aninhadas. No exemplo a seguir as estruturas estão aninhadas pelo fato delas estarem dentro do bloco de outra instrução. A instrução `idade < 18` só será executada se a instrução `idade < 10` for falsa:

```
idade = prompt("Entre com a idade:")
// converte de string para inteiro
idade = parseInt(idade)
if( idade < 10 ){
    console.log("criança")
}
else{
    if( idade < 18 ){
        console.log("adolescente")
    }
    else{
        console.log("adulto")
    }
}
console.log("Fim do programa")
```

Respostas possíveis para o programa

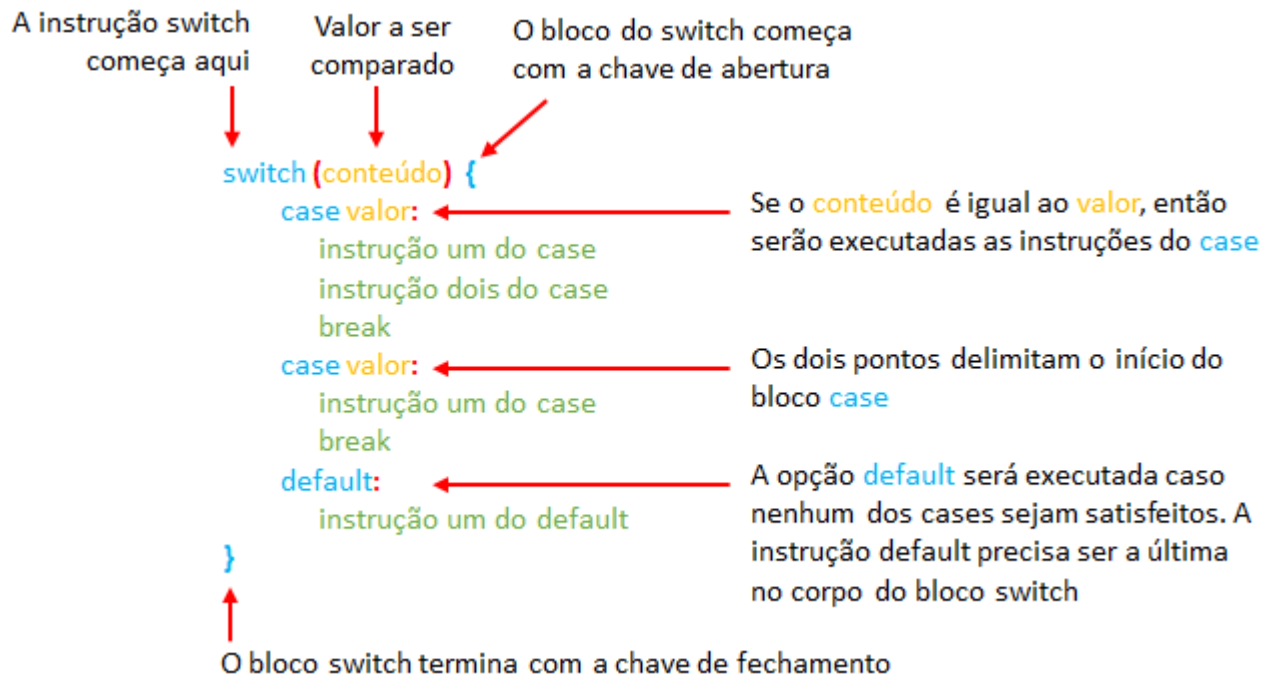
```
Entre com a idade:> 8
criança
Fim do programa
```

```
Entre com a idade:> 15
adolescente
Fim do programa
```

```
Entre com a idade:> 30
adulto
Fim do programa
```

## V. Estrutura condicional switch

A estrutura condicional switch é usada para decidir entre as opções disponíveis nos cases (casos em português). Na estrutura a seguir o `conteúdo` é comparado com cada `valor` de case disponível no corpo do switch e será executado o bloco `default` caso nenhum dos cases tenha sido satisfeito:



Nos exemplos a seguir será executado o case que possui o mesmo valor da variável **dia**.

Resultado do código:  
final de semana

```
dia = "domingo"
switch (dia) {
  case "domingo":
    console.log("final de semana")
    break
  case "segunda":
    console.log("trabalho")
    break
  case "terça":
    console.log("trabalho")
    break
  case "quarta":
    console.log("trabalho")
    break
  default:
    console.log("dia desconhecido")
}
```

Resultado do código:  
trabalho

```
dia = "terça"
switch (dia) {
  case "domingo":
    console.log("final de semana")
    break
  case "segunda":
    console.log("trabalho")
    break
  case "terça":
    console.log("trabalho")
    break
  case "quarta":
    console.log("trabalho")
    break
  default:
    console.log("dia desconhecido")
}
```

A instrução **break** interrompe a execução do switch, caso contrário todos os cases abaixo serão executados. No exemplo a seguir o case **segunda** será executado, como não existe um **break**, então os todos demais **case** e **default** serão executados.

Resultado do código:  
segunda-feira

trabalho  
trabalho  
dia desconhecido

```
dia = "segunda"
switch (dia) {
  case "domingo":
    console.log("final de semana")
  case "segunda":
    console.log("segunda-feira")
  case "terça":
    console.log("trabalho")
  case "quarta":
    console.log("trabalho")
  default:
    console.log("dia desconhecido")
}
```

É possível agrupar diferentes cases no mesmo código deixando o corpo do case vazio. No exemplo a seguir se o conteúdo da variável `dia` for segunda, terça ou quarta será executada a mesma instrução.

Resultado do código:

trabalho

```
dia = "segunda"
switch (dia) {
  case "domingo":
    console.log("final de semana")
    break
  case "segunda":
  case "terça":
  case "quarta":
    console.log("trabalho")
    break
  default:
    console.log("dia desconhecido")
}
```

## Exercícios

Veja o vídeo se tiver dúvidas nos exercícios: <https://youtu.be/62e9bLE59WM>

Para fazer os exercícios recomenda-se o uso da interface de programação online <https://replit.com/> por eles requerem entrada do usuário.

Nos exercícios 1 e 2 trabalharemos o operador de módulo (resto da divisão).

**Exercício 1:** Fazer um programa que pede para o usuário

Exemplo de saída:

entrar com um número inteiro e na sequência o programa imprime o resto da divisão desse número por 2.

Dica: use a função `prompt` para ler o teclado, a função `parseInt` para converter de string para inteiro, e a função `console.log` para imprimir o resultado.

**Exercício 2:** Fazer um programa que pede para o usuário entrar com dois números inteiros, um de cada vez, e na sequência o programa imprime o resto da divisão do primeiro número pelo segundo.

Dica: use a função `prompt` para ler o teclado, a função `parseInt` para converter de string para inteiro, e a função `console.log` para imprimir o resultado.

Nos exercícios 3 a 6 trabalharemos a estrutura de decisão `if - else`.

**Exercício 3:** Fazer um programa que pede para o usuário entrar um número inteiro e na sequência o programa imprime na tela se o número é par ou ímpar.

Dica: um número é par se o resto da divisão dele por 2 é zero.

**Exercício 4:** Fazer um programa que pede para o usuário entrar com dois números inteiros, um de cada vez, e na sequência o programa imprime na tela se o 1º número é múltiplo do 2º número.

Dica: um número é múltiplo do outro se o resto da divisão deles é zero.

**Exercício 5:** Fazer um programa que pede para o usuário entrar com dois números inteiros, um de cada vez, e na sequência o programa imprime na tela o maior valor.

```
Entre com um número inteiro:> 11
Resto: 1
```

Exemplo de saída:

```
Entre com um número inteiro:> 12
Entre com um número inteiro:> 5
Resto: 2
```

Exemplo de saída:

```
Entre com um número inteiro:> 11
11 é ímpar
Fim do programa
```

Exemplo de saída:

```
Entre com um número inteiro:> 12
Entre com um número inteiro:> 5
12 não é múltiplo de 5
Fim do programa
```

Exemplo de saída:

```
Entre com um número inteiro:> 2
Entre com um número inteiro:> 5
2 é menor que 5
Fim do programa
```



**Exercício 6:** Fazer um programa que pede para o usuário entrar com a idade dele e, na sequência, o programa indica se ele é de menor ou adulto.

Exemplo de saída:

```
Entre com a idade:> 15
De menor
Fim do programa
```

Outra saída:

```
Entre com a idade:> 30
Adulto
Fim do programa
```

Nos exercícios 7 a 9 trabalharemos a estrutura de decisão aninhada.

**Exercício 7:** Fazer um programa que pede para o usuário entrar com três números, um de cada vez, e na sequência o programa imprime o maior valor fornecido pelo usuário.

Exemplo de saída:

```
Entre com um número:> 1
Entre com um número:> 3
Entre com um número:> 2
Maior: 3
Fim do programa
```

**Exercício 8:** Fazer um programa que pede para o usuário fornecer a temperatura corporal e, na sequência, o programa imprime a classificação da temperatura de acordo com a tabela a seguir.

Exemplo de saída:

```
Entre com a temperatura corporal:> 37.2
Normal
Fim do programa
```



**Exercício 9:** O IMC (Índice de Massa Corporal) é calculado usando peso/(altura\*altura). Fazer um programa que pede para o usuário entrar com o peso (em Kg) e altura (em metros) e, na sequência, o

Exemplo de saída:

programa imprime na tela a classificação do IMC de acordo com a classificação a seguir.

IMC	Classificação
abaixo de 18,5	abaixo do peso
entre 18,6 e 24,9	Peso ideal (parabéns)
entre 25,0 e 29,9	Levemente acima do peso
entre 30,0 e 34,9	Obesidade grau I
entre 35,0 e 39,9	Obesidade grau II (severa)
acima de 40	Obesidade III (mórbida)

```
Entre com o peso (Kg):> 71.5
Entre com a altura (m):> 1.7
Peso ideal
Fim do programa
```

No exercício 10 trabalharemos a estrutura condicional switch.

**Exercício 10:** Fazer um programa que pede para o usuário fornecer o nome da fruta (todas as letras minúsculas) e, na sequência, o programa informa o valor do suco de acordo com a tabela a seguir.

O programa deverá exibir “Não temos esse suco”, caso o usuário forneça um nome de fruta inexistente na tabela.

Exemplos de saída:

```
Entre com o nome da fruta:> manga
R$4,00
Fim do programa
```

```
Entre com o nome da fruta:> abacaxi
Não temos esse suco
Fim do programa
```

LANCHONETE			
LANCHES		SUCOS	
X_TUDO	R\$10,00	LARANJA	R\$3,50
X_SALADA	R\$7,00	LIMÃO	R\$3,00
X_BACON	R\$8,50	ACEROLA	R\$3,50
X_EGG	R\$8,00	MANGA	R\$4,00
X_CALABRESA	R\$6,00	MELANCIA	R\$4,00
MISTO QUENTE	R\$4,50	MORANGO	R\$5,00
		MARACUJA	R\$4,50
		AÇAÍ	R\$6,00