Strings e Arrays

Desenvolver Algoritmos

Msc. Lucas G. F. Alves

e-mail: lgfalves@senacrs.com.br





Planejamento de Aula

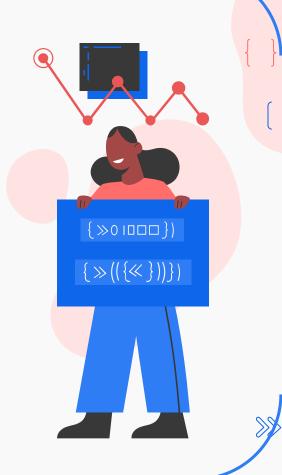
Revisão Operadores Lógicos

Strings

Protótipos de Strings

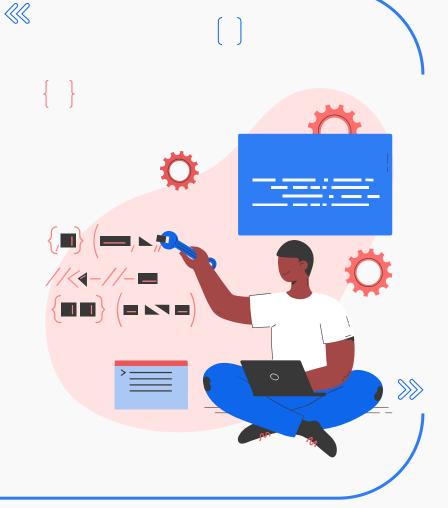
Arrays

Protótipos de Arrays





Revisão Operadores





Operadores Lógicos

>>>

- São operadores especiais usados entre booleanos
- Retornam um valor booleano
- Existem 3 importantes:

```
Operador E: &&
```

Operador Ou: ||

Operador Não/Negação: !









Operadores Lógicos

Operador E (&&)

Retorna **true** se, e somente se, os **todos booleanos** envolvidos também forem **true**.

а	b	a && b
V	V	V
V	F	F
F	V	F
F	F	F

{\{(\(\{\infty}\))\\)\\





Exercícios



Antes de começar, crie 3 variáveis: a, b e c.

Atribua os valores true, false e true, respectivamente a = true, b = false e c = true.

- 1. Realize a operação: a && b
- 2. Realize a operação: **b && c**
- 3. Realize a operação: a && c
- 4. Realize a operação: **a && b && c**









Operadores Lógicos

Operador OU (||)

Retorna false se, e somente se, todos booleanos envolvidos também forem false.

а	b	a b
V	V	V
V	F	V
F	V	V
F	F	F
16416 24		

({(({≪}}))) ≪ }





Exercícios



Antes de começar, crie 3 variáveis: a, b e c.

Atribua os valores true, false e true, respectivamente a= true, b = false, c = true.

- 1. Realize a operação: a || b.
- 2. Realize a operação: **b** || **c**.
- 3. Realize a operação: **a** || **c**.
- 4. Realize a operação: **a || b || c**









Operadores Lógicos

Operador NÃO (!)

Sempre retorna o booleano oposto.

а	!a
V	F
F	V

```
!true  // resultado = false
!false  // resultado = true
```







>>

Resumo

Operadores Lógicos

- &&
- ||







Exercícios

Faça um programa que receba o nome, ano de nascimento de uma pessoa e o ano atual e mostre:

- O nome da pessoa
- A idade dessa pessoa
- Um true ou false que diz se ela é maior de idade

Nome: Laís Idade: 23

É maior de idade? true

Idade em 2050: 52

• Quantos anos ela terá em 2050











Numbers

• Operadores Aritméticos:

Comparadores:





Resumo

>>

Operador E (&&): Só devolve true se todos os booleanos forem true.

а	b	a && b
V	V	V
V	F	F
F	V	F
F	F	F







Resumo

>>>

Operador Ou (||): Só devolve false se todos forem false

а	b	a b
V	V	V
V	F	V
F	V	V
F	F	F







Resumo

>>

Operador!: Devolve o oposto

а	!a
V	F
F	V











Exercícios de interpretação de código.

1) Leia o código abaixo. Indique todas as mensagens impressas no console, SEM EXECUTAR o programa.

```
//bool1 recebe verdadeiro
//bool2 recebe falso
//bool3 recebe o inverso de bool2
//imprime bool 1 & & bool 2 falso
//imprime bool1 && bool2 && bool3 falso
//imprime o resultado anterior invertido &&
(bool1 | bool2) true
//imprime o tipo do resultado anterior bool
```

```
const bool1 = true
const bool2 = false
const bool3 = !bool2
let resultado = bool1 && bool2
console.log("a. ", resultado)
resultado = bool1 && bool2 && bool3
console.log("b. ", resultado)
resultado = !resultado && (bool1 || bool2)
console.log("c. ", resultado)
console.log("d. ", typeof resultado)
```



>>>

Exercícios de interpretação de código.

2) Seu colega se aproxima de você falando que o código dele não funciona como devia. Vamos ajudá-lo: consegue perceber algum problema? O que será impresso

no console?

```
let primeiroNumero = prompt("Digite um numero!")
let segundoNumero = prompt("Digite outro numero!")
const soma = primeiroNumero + segundoNumero
console.log(soma)
```

Sugira ao seu colega uma solução para que o valor impresso no console seja, de fato, a soma dos dois números.



Será impresso a junção do primeiro número com o segundo. Para ajustar devemos converter para Number os prompts







Exercícios de escrita de código

- 1) Faça um programa que:
- a) Pergunte a idade do usuário; const idadeUsuario = Number(prompt("Sua Idade?"))
 - b) Pergunte a idade do seu melhor amigo ou da sua melhor amiga; const idadeAmigo = Number(prompt("Idade do melhor amigo?"))
- c) **Imprima no console** a seguinte mensagem: "Sua idade é maior do que a do seu melhor amigo?", seguido pela resposta (true ou false)

let resultado = idadeUsuario > idadeAmigo console.log("Sua idade é maior do que a do seu melhor amigo?", resultado)

d) Imprima no console a diferença de idade.



resultado = idadeUsuario-idadeAmigo console.log(idadeUsuario-idadeAmigo)







Exercícios de escrita de código

- 2) Faça um programa que:
 - a) Peça ao usuário que insira um número par;

```
const nPar = Number(prompt("Digite um número par"))
```

- b) Imprima no console **o resto da divisão** desse número por 2; console.log(nPar%2)
- c) Teste o programa com diversos números pares. Você notou um padrão? Escreva em um comentário de código; Sim sempre o número O.
- d) O que acontece se o usuário inserir um número ímpar? Escreva em um comentário de código;

O resto da divisão é diferente de 0.









Exercícios de escrita de código

3) Faça um programa que pergunte ao usuário sua idade em anos. Depois, imprima no console.

```
const idade = Number(prompt("Sua idade?"))
```

- a) A idade do usuário em meses; const resultado = idade * 12 console.log("Sua idade em meses é:", resultado)
- b) A idade do usuário em dias; let resultado1 = idade * 365 console.log("Sua idade em dias é:", resultado1)
- c) A idade do usuário em horas; resultado1 *= 24 console.log("Sua idade em horas é:", resultado1)







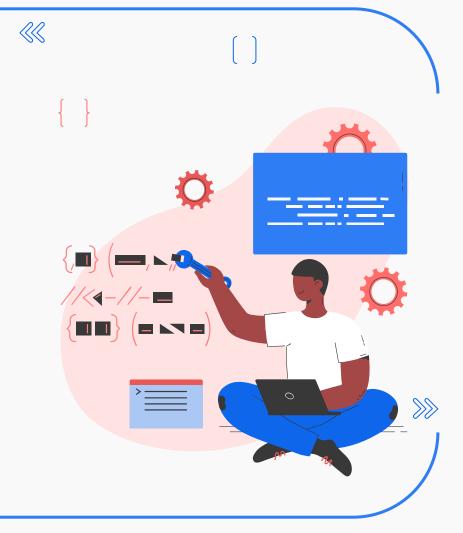


Exercícios de escrita de código

- 4) Faça um programa que pergunte ao usuário dois números. Em seguida, faça as operações e imprima no console as seguintes mensagens seguidas pelo true ou false:
 - O primeiro numero é maior que segundo? true
 - O primeiro numero é igual ao segundo? false
 - O primeiro numero é divisível pelo segundo? true
 - O segundo numero é divisível pelo primeiro? true
- obs: O true ou false vai depender dos números inseridos e do resultado das operações.











- 1. Para este exercício, será necessário o conhecimento das fórmulas para mudar a unidade de temperatura entre Graus Celsius(°C), Graus Fahrenheit(°F) e Kelvin(K). Abaixo estão duas delas:
 - Graus Fahrenheit(°F) para Kelvin(K) (KELVIN) = (GRAUS_FAHRENHEIT - 32)*(5/9) + 273.15
 - Graus Celsius(°C) para Graus Fahrenheit (°C) (GRAUS_FAHRENHEIT) = (GRAUS_CELSIUS)*(9/5) + 32
 - a) Calcule e mostre o valor de 77°F em K, mostrando a unidade no console também;
 - b) Calcule e mostre o valor de 80°C em °F, mostrando a unidade no console também;
- c) Calcule e mostre o valor de 30°C em °F e K, mostrando as unidades no console também;
 - d) Altere o último item para que o usuário insira o valor em graus Celsius que ele $(\{((\{ \gg \})) \ll \})$ deseja converter;







- 2. Quilowatt-hora é uma unidade de energia; e é muito utilizada para se determinar o consumo de energia elétrica em residências. Sabe-se que o quilowatt-hora de energia custa R\$0.05. Faça um programa que receba a quantidade de quilowatts consumida por uma residência.
- a) Calcule e mostre o valor a ser pago por uma residência que consuma 280 quilowatt-hora
- b) Altere o programa para receber mais um valor: a porcentagem de desconto. Calcule e mostre o valor a ser pago pela mesma residência acima considerando 15% de desconto









- 2. Quilowatt-hora é uma unidade de energia; e é muito utilizada para se determinar o consumo de energia elétrica em residências. Sabe-se que o quilowatt-hora de energia custa R\$0.05. Faça um programa que receba a quantidade de quilowatts consumida por uma residência.
- a) Calcule e mostre o valor a ser pago por uma residência que consuma 280 quilowatt-hora
- b) Altere o programa para receber mais um valor: a porcentagem de desconto. Calcule e mostre o valor a ser pago pela mesma residência acima considerando 15% de desconto







>>

- 3. Um grande problema que o mundo tem atualmente é a quantidade de unidades que existem para representar a mesma coisa. Por exemplo, para representar a Massa de um corpo, podemos usar quilograma (kg), onça (oz) e até libra (lb). Para representar Distâncias, existem metro (m), pés (ft), milha (mi). Até para volumes, há várias opções: litro (l), galão (gal), xícaras (xic). Dada essa introdução, faça o que se pede:
- a) Procure uma forma de converter libra (lb) para quilograma (kg) e escreva um programa que converta 20lb para kg. Imprima a resposta no console da seguinte forma: 20lb equivalem a X kg
- b) Procure uma forma de converter onça (oz) para quilograma (kg) e escreva um programa que converta 10.5oz para kg. Imprima a resposta no console da seguinte forma: 10.5oz equivalem a X kg
 - c) Procure uma forma de converter milha (mi) para metro (m) e escreva um programa ({(({\$>>}))}) «} que converta 100mi para m. Imprima a resposta no console da seguinte forma: 100mi equivalem a X m





d) Procure uma forma de converter pés (ft) para metro (m) e escreva um programa que converta 50ft para m. Imprima a resposta no console da seguinte forma:

50ft equivalem a X m

e) Procure uma forma de converter galão (gal) para litro (l) e escreva um programa que converta 103.56gal para litro. Imprima a resposta no console da seguinte forma:

103.56gal equivalem a X I

f) Procure uma forma de converter xícara (xic) para litro (l) e escreva um programa que converta 450xic para litro. Imprima a resposta no console da seguinte forma:

450 xic equivalem a X I

g) Escolha ao menos **um** dos itens anteriores e modifique o programa para que ele peça ao usuário o valor da unidade original antes de converter











O que são Strings?

Como vimos anteriormente, Strings são os tipos referentes à textos.

Temos 3 maneiras de escrever uma string:

- Aspas Duplas: "Olá Mundo"
- Aspas Simples: 'Olá Mundo'
- Crase (Template String ou Template Literals):

'Olá Mundo'









Concatenação.

Também podemos juntar várias strings para formar uma nova.

Chamamos esse processo de **concatenação** e utilizamos o sinal de + para fazê-lo.

```
const nome = "Lucas"
const idade = 32
const frase = "Meu nome é " + nome + " e tenho " + idade + " anos"
```









Template.

Não há diferença entre usar aspas simples ou duplas!

A única diferente é a Template String, pois ela nos permite colocar variáveis

javascript no meio da string.

```
const nome = "Lucas"
const idade = 32

const frase = `Meu nome é ${nome} e tenho ${idade} anos`

// Meu nome é Lucas e tenho 32 anos

({(({>>>}))
```

Para sinalizar que é uma



Exercícios



1. Crie um programa que peça ao usuário para inserir o seu nome e sua cor favorita e imprima a mensagem:

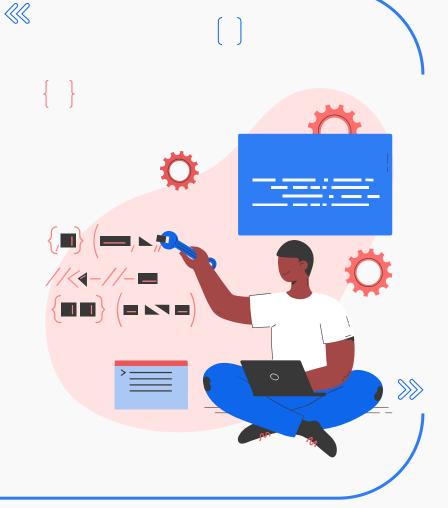
"A cor favorita de FULANO é COR: "

2. Realize o exercícios duas vezes. Utilizando template strings e concatenação.





Protótipo de Strings





Protótipo de Strings



O javascript nos fornece algumas informações (**propriedades**) e ações (**métodos**) que podemos realizar sobre uma string.

Será falado algumas delas na aula, mas se quiser conhecer mais, pode visitar o site do W3Schools que é referência em desenvolvimento web.

https://www.w3schools.com/Jsref/jsref obj string.asp









Protótipo de Strings

>>>

Propriedade length

A propriedade length nos diz qual é o tamanho de uma string, incluindo espaços.

```
const nome = "Prof Lucas Alves"
```

console.log(nome.length) // 16









>>>

Método toLowerCase()

O método toLowerCase() transforma todas as letras da sua string em minúsculas

```
const frase = "OieEeEee!"
const fraseMaiuscula = frase.toLowerCase()
// fraseMaiuscula = oieeeeee!
```











Método to Upper Case()

O método to Upper Case () transforma todas as letras da sua string em maiúsculas.

```
const frase = "OieEeEee!"
const fraseMinuscula = frase.toLowerCase()
// fraseMinuscula = oieeeeee!
```









>>>

Método trim()

O método trim() retira os espaços que existem antes e depois da sua string.

Útil em formulários como por exemplo de login!

```
const email = " LGFALVES@senacrs.com.br "
console.log(email.trim())
// "LGFALVES@senacrs.com.br"
```









>>>

Método includes (caracteres)

O método **includes(caracteres)** determina se um conjunto de caracteres pode ser encontrado dentro de outra string, retornando **true** ou **false**.

```
const frase = "Hoje comi cenoura"
frase.includes("cenoura") // true
frase.includes("batata") // false
```









>>>

Método replaceAll(chars1, chars2)

O método **replaceAll(chars1, chars2)** troca todas as ocorrências de um conjunto de caracteres (chars1) por alguma outra coisa (chars2)

```
const frase = "Hoje comi cenoura, adoro cenoura"
const novaFrase = frase.replaceAll("cenoura","batata")
// novaFrase = Hoje comi batata, adoro batata
```







Exercícios



Peça para o usuário escrever uma frase e imprima no console a frase alterada, com:

- Todas as letras maiúsculas;
- Na língua do i (substituindo a vogal "o" por "i");
- O tamanho da frase.







Resumo



Existem 3 Maneiras de escrever Strings:

Aspas duplas, Aspas simples e Crase;

Formando novas Strings:

Concatenação e Template Strings;

Protótipo de Strings:

length, toLowerCase(), toUpperCase(), trim(), includes(caracteres) e replaceAll(chars1, chars2).











O que são arrays?

Arrays nada mais são do que listas de elementos.

Será armazenado mais de uma coisa dentro de uma variável.

Ex: lista de compras, lista de alunos, lista de números da loteria, lista telefônica...

No javascript, usamos colchetes [] para agrupar os itens de uma lista:

const listaDeCompras = ["batata", "alface", "queijo"] const listaDeNumerosMega = [2, 13, 26, 35, 41, 60]









O que são arrays?

Podemos colocar elementos de **qualquer tipo** que vimos até agora dentro de um array!

Números, strings e booleanos.

Também podemos ter elementos de tipos diferentes dentro de um mesmo array

const meuArray = ["banana", 15, true]







O que são arrays?











Acessando um elemento

Em um array, acessamos os elementos através da posição (índice) deles na lista!

Funciona como se fosse uma lista numerada:

Lista de Compras

- 1. Abacate
- 2. Banana
- 3. Tomate



Qual é o **item na posição 2**?

Resposta: Banana







Acessando um elemento

Mas no caso dos arrays, a numeração não começa no 1, mas sim no 0!

Para acessar um item, colocamos a sua posição (índice) entre colchetes após o nome do array

const listaDeCompras = ["Abacate", "Banana", "Tomate"]
const segundoItem = listaDeCompras[2] // "Tomate"

Lista de Compras

- 0. Abacate
- 1. Banana
- 2. Tomate





Exercícios



- 1. Crie um array com pelo menos 5 raças de cachorro.
- 2. Peça para o usuário inserir um número de 0 a 4 e Imprima no console a raça correspondente à posição escolhida pelo usuário.







Resumo

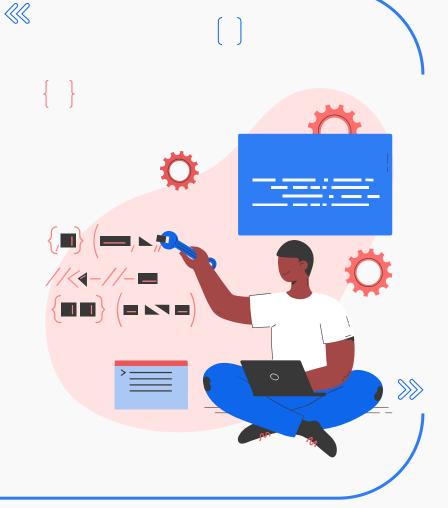
>>

Arrays são listas que podem conter elementos de qualquer tipo (strings, números, etc).

Para acessar um elemento de um array, utilizamos a sua posição (ou **índice**).











O javascript nos fornece algumas informações (**propriedades**) e ações (**métodos**) que podemos realizar sobre uma lista (array)

Também será falado algumas delas na aula, mas se quiser conhecer mais, pode visitar o site do W3Schools que é referência em desenvolvimento web.

https://www.w3schools.com/Jsref/jsref_obj_array.asp









Propriedade length

A propriedade length nos diz qual é a quantidade de itens de um array.

```
const pokemon = ["bulbasauro", "squirtle", "charmander"]
console.log(pokemon.length) // 3
```











Método includes (elemento)

O método includes (elemento) determina se um array contém um determinado elemento, retornando true ou false

```
const seriesBoas = ["Vikings", "The Big Bang Theory"]
```

```
seriesBoas.includes("Vikings") // true
seriesBoas.includes("Game of Thrones") // false
```









Método push(elemento)

O método push(elemento) adiciona um ou mais elementos ao final de um array

```
const numeros = [1, 2, 3]

numeros.push(4)

console.log(numeros) // [1, 2, 3, 4]

numeros.push(5, 6, 7)

console.log(numeros) // [1, 2, 3, 4, 5, 6, 7]
```







Método pop()

O método pop() remove o último elemento de um array

```
const meusPeixes = ["palhaço", "mandarim", "esturjão"]
```

```
meusPeixes.pop()
console.log(meusPeixes) // ["palhaço", "mandarim"]
```









>>>

Método splice(i, n)

O método splice(i, n) remove n (n representa a quantidade) elementos à partir da posição i do array.

 $(\{((\{ \gg \})) \ll \}) | \text{letras.splice}(3, 2) // \text{letras} = ["A", "B", "D", "G", "H"]$





Exercícios



Para este exercício, comece criando um array com os valores: 1, 2, 3, 4, 5 e 6.

- 1. Determine o tamanho do array.
- 2. Adicione o número 7.
- 3. Remova os números 4 e 5.
- 4. Determine o novo tamanho do array.







Resumo

>>

Arrays são listas de elementos que podem ter qualquer tipo.

Agrupamos esses itens usando colchetes [].

Acessamos um item pelo índice (ou seja, sua posição na lista)

Protótipo de Arrays:

length, includes(), push(elemento), pop() e splice(i, n);





Obrigado!

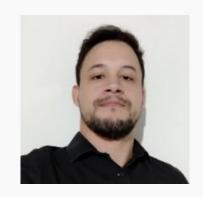
E-mail: lgfalves@senacrs.com.br







Professor



Lucas G. F. Alves







