UNINASSAU

Variáveis e Operadores

Prof. Matheus Garrido

O que vamos estudar hoje?

- Javascript
- O que são Variáveis
- Tipos de Variáveis
- Como usar operadores com diferentes tipos de variáveis
  - Numbers
  - Booleans
  - Arrays
  - Strings

Javascript

Vamos codificar? 😍

Javascript

- Javascript é uma das **linguagens** de **programação** mais utilizadas atualmente.

- Uma linguagem de programação é um **conjunto** de **normas** (sintaxe) que permite criar comandos para o computador.

Javascript

- Fracamente tipada;
- Linguagem de scripts orientada a objetos;
- Interpretado pelo navegador;
- Não faz parte da plataforma Java;
- Não cria aplicações independentes;
- Se encontra dentro do documento HTML.

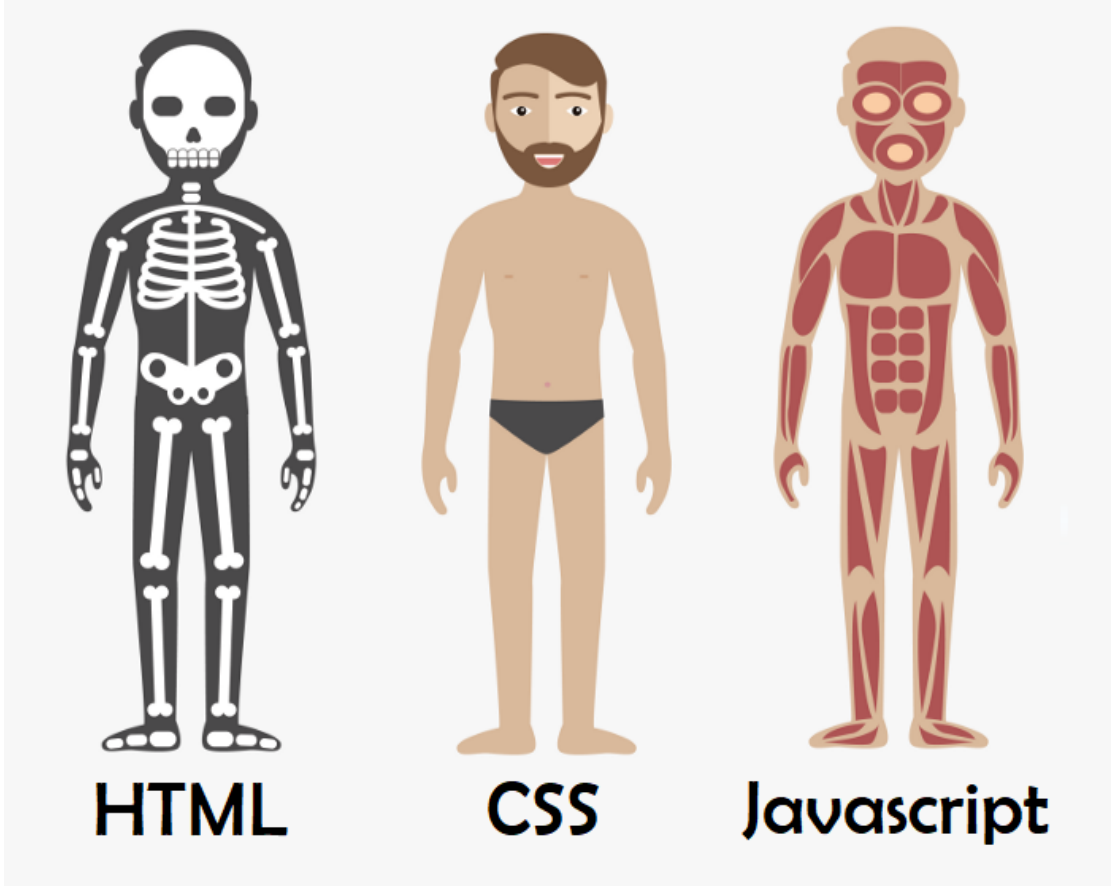
- Fortemente tipada;
- Linguagem de programação orientada a objetos;
- Necessita ser copilado;
- Cria aplicações independentes;
- Executa suas aplicações em uma máquina virtual ou em um browser.

Javascript

- Ele é usado para muitas coisas.

- Para nós, o que mais importa agora é que ele é o **responsável** pela **lógica** que o nosso **site** deve ter:
  - Lidar com interação (clicks, inputs) do usuário.
  - Determina o que deve ser mostrado no site.
  - Permite criar, modificar ou retirar elementos de layout
  - E mais...

Javascript



Javascript

- **Comentários**

- São estruturas que permitem escrevermos textos que serão ignorados para executar o programa.
- Eles devem começar com // ou estarem entre /* */

```
//Eu sou um comentário

/*
Eu sou ignorado na
execução do programa
*/
```

Javascript

- **Linkando arquivo JS no HTML**
- Podemos escrever scripts de código em nosso HTML de duas formas:
  - Externo ao elemento, no próprio HTML.
  - Externo ao arquivo, por meio de um arquivo.js

```
<head>
    <script>
        console.log("Olá, mundão")
    </script>
</head>
```

```
<script type="text/javascript" src="teste.js"></script>
```

Javascript

- **Imprimindo no Terminal**

- O JS possui uma sintaxe específica para imprimir informações no terminal.

```
console.log("Olá, mundão")
```

Javascript

- **Pedindo informações para o usuário**

- Em aplicações Web, conseguimos pedir que o usuário nos passe alguma informação, assim:

```
const nome = prompt("Informe seu nome");
        //A variável nome armazanará o nome
        //do usuário
```

Variáveis

Variáveis

- Variáveis são estruturas que permitem **guardar** e **acessar** quaisquer **informações** no nosso código.
- Elas funcionam como caixas ou gavetas.
- Antes de usarmos estas variáveis, nós precisamos declará-las (criá-las).

```
const nomeVariavel = 1
```

valor

declaração — nome — atribuição

Variáveis

- **const**: quando uma variável é declarada usando const, nós dizemos que ela é constante.
- O seu valor NÃO pode mudar ao longo do programa.

```
const nomeUsuario = "Maria"

nomeUsuario = "luiza"
```

Variáveis

- **let**: quando uma variável é declarada usando let, ela PODE ter seu valor alterado.

```
let nomeUsuario = "Maria"

nomeUsuario = "luiza"
```

Variáveis

- Devemos escolher nomes significativos.

- Nomes não podem começar com números ou caracteres especiais.

- Utilizamos o padrão CamelCase:
  - primeira letra minúscula
  - primeira letra entre uma palavra e outra maiúscula

```
const nomeUsuario
```

Tipos de Variáveis

Tipos de Variáveis

- Os valores que as **variáveis** do JS **possuem tipos**. Hoje, falaremos de 4 deles:
  - Strings;
  - Numbers;
  - Arrays;
  - Boolean.

Tipos de Variáveis

- **Strings**: são os tipos que representam textos.

```
const nome = "Matheus"
let idade = "15"
```

- **Numbers**: são os tipos que representam números.

```
const idade = 23
const altura = 1.86
const temperatura = -20
```

Tipos de Variáveis

- **Strings**: são os tipos que representam **textos**.

```
const nome = "Matheus"
let idade = "15"
```

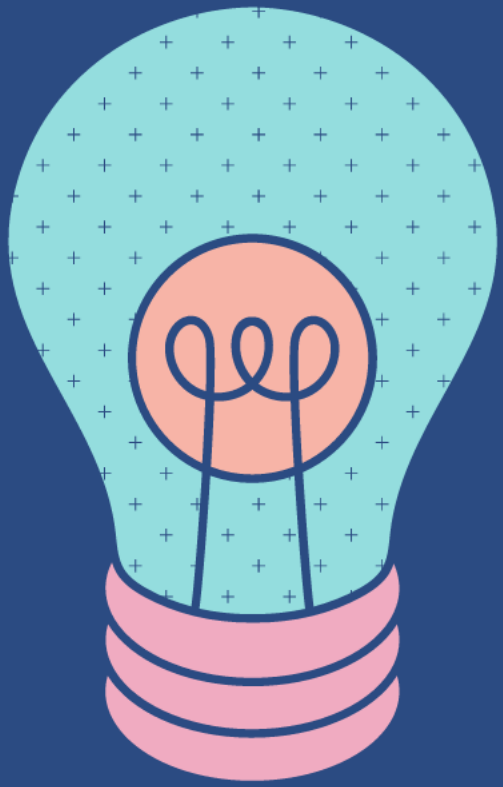
- **Numbers**: são os tipos que representam **números**.

```
const idade = 23
const altura = 1.86
const temperatura = -20
```

Vamos praticar?

- Faça os seguintes itens:

    - 1. Crie uma variável e atribua seu primeiro nome.
    - 2. Crie uma variável e atribua seu sobrenome.
    - 3. Crie uma variável e atribua sua idade.
    - 4. Imprima o seu nome, sobrenome e idade.



Tipos de Variáveis

- **Arrays**: arrays são maneiras de **guardar** e **acessar mais** de uma **informação** ao mesmo tempo.

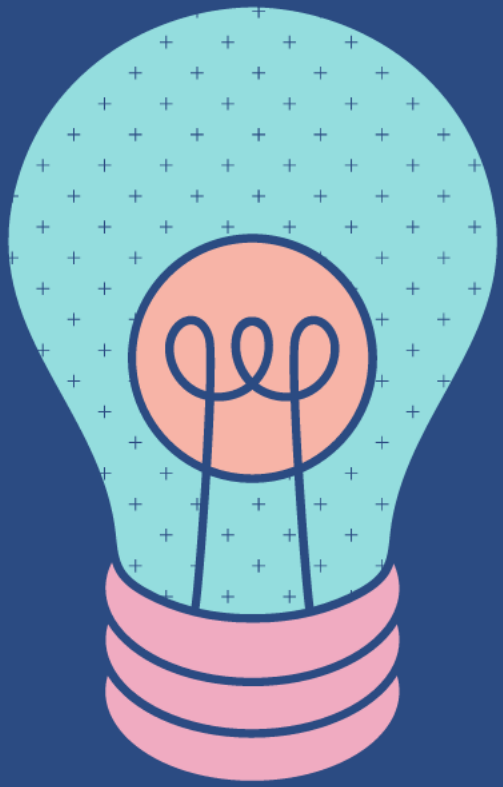
```
// Criamos um array assim:
let array = [1, 2, 3]
// Array pode receber informações
// com varíaveis distintas
array = ['a', 2, 'casa']
```

```
// Para acessar valores do array:
array[0]
// Começa a contar do 0.
// Para pegar o segundo elemento:
array[1]
```

Vamos praticar?

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

    1. Imprima no console o primeiro, o terceiro e o último elementos do array
    2. Altere o array de tal forma que os valores ímpares sejam substituídos pelos seus respectivos dobros, ou seja, o array deve conter, nesta ordem: 2, 2, 6, 4, 10 e 6



Tipos de Variáveis

- Variáveis Booleanas: são **variáveis** que só assumem os valores **true** ou **false**.

```
let variavelBoolean = true
variavelBoolean = false
```

Tipos de Variáveis

- **Descobrindo o tipo de uma variável**:

- **typeof**: comando que **permite ver** o **tipo** do valor da **variável.**

```
const got = 'Game of Thrones'
const temporadasGot = 8

typeof got //string
typeof temporadasGot //number
```

Tipos de Variáveis

- **undefined**: tipo que representa a **falta** de **valor** de uma **variável.**

```
let novaVariavel
typeof novaVariavel //undefined

novaVariavel = 2
typeof novaVariavel //number

novaVariavel = undefined
typeof novaVariavel //undefined
```

Tipos de Variáveis

- **null**: também representa a **falta** de **valor** da **variável.**

- Existem algumas **diferenças** entre **undefined** e **null**.
- Uma delas é que o **null** precisa ser **associado diretamente** a uma **variável.**

```
let minhaVariavel
typeof novaVariavel //undefined

minhaVariavel = null
typeof minhaVariavel //null
```

Operadores - Numbers

Operadores - Numbers

- **Soma:**

```
const primeiroValor = 12
const segundoValor = 45

const resultadoSoma = primeiroValor + segundoValor + 6
```

Operadores - Numbers

- **Subtração:**

```
const primeiroValor = 12
const segundoValor = 45

const resultadoSubtracao = segundoValor - primeiroValor
```

Operadores - Numbers

- **Multiplicação:**

```
const primeiroValor = 12
const segundoValor = 45

const resultadoMultiplicacao = segundoValor * primeiroValor
```

Operadores - Numbers

- **Divisão:**

```
const primeiroValor = 50
const segundoValor = 5

const resultadoDivisao = primeiroValor / segundoValor
```

Operadores - Numbers

- **Resto da Divisão:**

- Existem casos em que a divisão **não** dá um **número inteiro** (sem vírgula).
- Quando acontece isso, dizemos que há um resto na divisão.
- Por exemplo: dividir 20 por 3:
  - Dá o resultado 6, com resto 2;
  - 20 = 3 x 6 + 2.

Operadores - Numbers

- **Resto da Divisão:**

```
let restoDaDivisao = 20 % 3

console.log(restoDaDivisao) //2
```

Operadores - Numbers

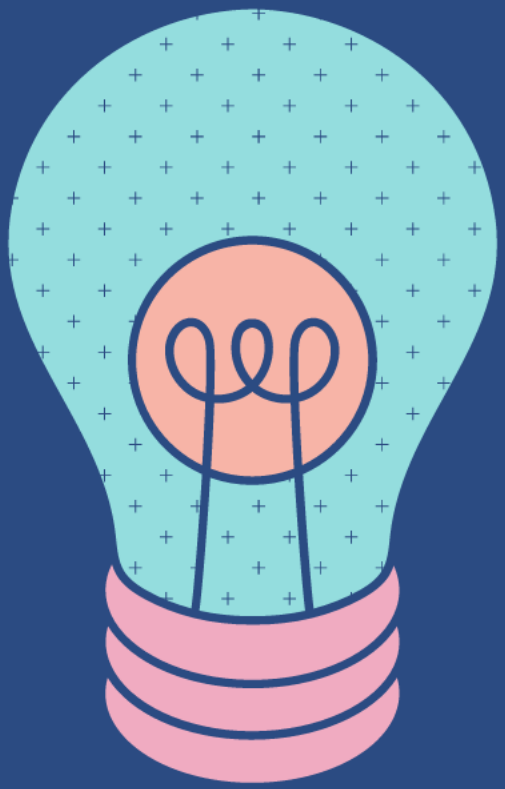
- **Simplificação:**

```
let primeiroValor = 20

primeiroValor = primeiroValor + 10
//Equivale a:
primeiroValor += 10

primeiroValor = primeiroValor - 10
//Equivale a:
primeiroValor -= 10

primeiroValor = primeiroValor * 10
//Equivale a:
primeiroValor *= 10

primeiroValor = primeiroValor / 10
//Equivale a:
primeiroValor /= 10
```

Vamos praticar?

- Faça as seguintes operações usando o computador:

1. Somar 3 com 4;
2. Multiplicar 3 com 5 e dividir o resultado por 2;
3. Subtrair 5 de 4 e multiplicar o resultado por -1;
4. Determinar o resto da divisão de 234 por 5.



Operadores - Numbers

- **Comparadores:** são operadores que permitem comparar variáveis entre si.

- São eles:
  - ===
  - !==
  - > e >=
  - < e <=

Operadores - Numbers

- **Comparadores:** são operadores que permitem comparar variáveis entre si.

- O resultado destes operadores é sempre um booleano.
- Quando a comparação for correta, o resultado é true. Caso contrário, é false.

Operadores - Numbers

- ===: verifica se o valor e o tipo são iguais.

```
"1" === "2" //false
"3" === "3" //true

//O valor da comparacao
//pode ser armazenado
//em uma variável:
const comparacao = 1 === 2
```

Operadores - Numbers

- **> e >=:**
- Pode ser **usado** com **numbers**.
- **>=**: retorna true se os números envolvidos forem iguais ou se o primeiro for maior que o segundo.
- **>**: retorna true só se o primeiro for maior que o segundo.

Operadores - Numbers

- > e >=:

```
5 > 4 //true, 5 é maior que 4
5 > 5 //false, 5 é igual a 5
5 > 6 //false, 5 é menor que 6

5 >= 4 //true, 5 é maior que 4
5 >= 5 //true, 5 é igual a 5
5 >= 6 //false, 5 é menor que 6
```

Operadores - Numbers

- **< e <=:**
- Também pode ser usado com numbers.
- **<=**: retorna true se os números envolvidos forem iguais ou se o primeiro é menor que o segundo.
- **<**: retorna true só se o primeiro é menor que o segundo.

Operadores - Numbers

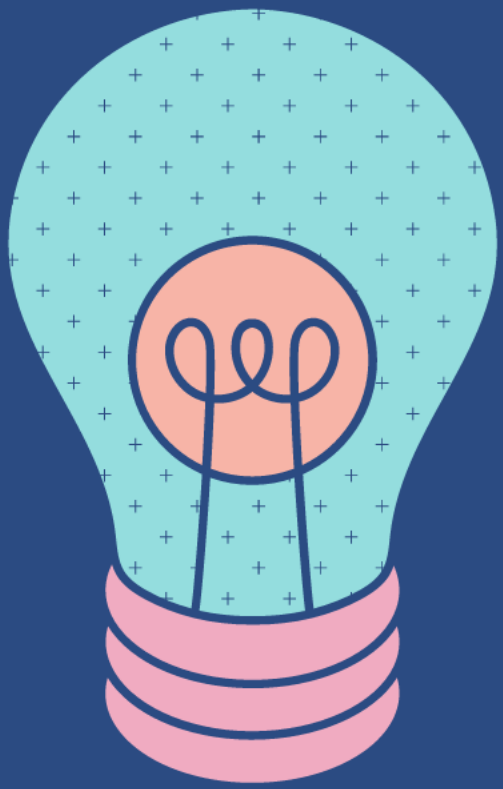
- < e <=:

```
5 < 4 //false, 5 é maior que 4
5 < 5 //false, 5 é igual a 5
5 < 6 //true, 5 é menor que 6

5 <= 4 //false, 5 é maior que 4
5 <= 5 //true, 5 é igual a 5
5 <= 6 //true, 5 é menor que 6
```

Vamos praticar?

- Crie duas variáveis que guardem dois números. Imprima na tela as seguintes mensagens:

1. O primeiro número é igual ao segundo? True/False;
2. O primeiro número é diferente do segundo? True/False;
3. O primeiro número é maior que o segundo? True/False;
4. O primeiro número é menor que o segundo? True/False;



Operadores - Booleans

Operadores - Booleans

- São operadores especiais **usados** entre **booleanos**.

- **Retornam** um valor **booleano**.

- Existem 3 importantes:
  - Operador E: &&
  - Operador Ou: ||
  - Operador Não/Negação: !

Operadores - Booleans

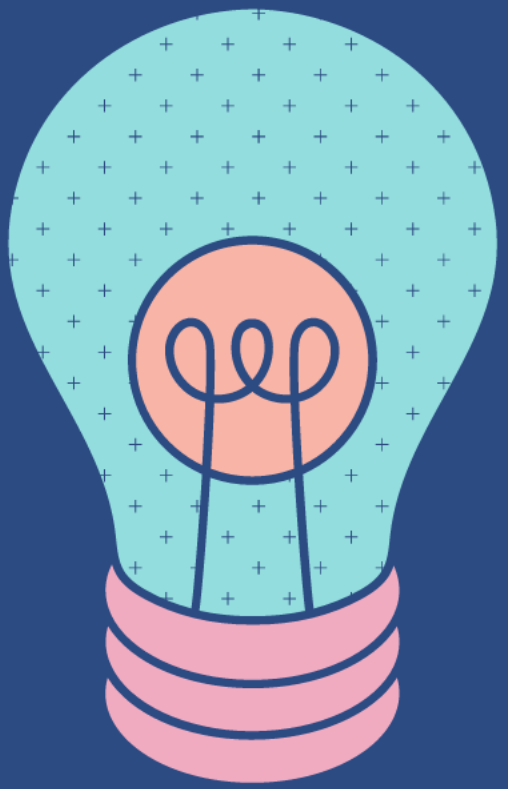
- **Operador E (&&):** retorna **true** se, e somente se, os **todos booleanos** envolvidos também **forem true**.

```
true && true //true
true && false //false
false && true //false
false && false //false
```

Vamos praticar?

- Antes de começar, crie 3 variáveis: a, b e c. Atribua os valores true, false e true, respectivamente.

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 - Booleans

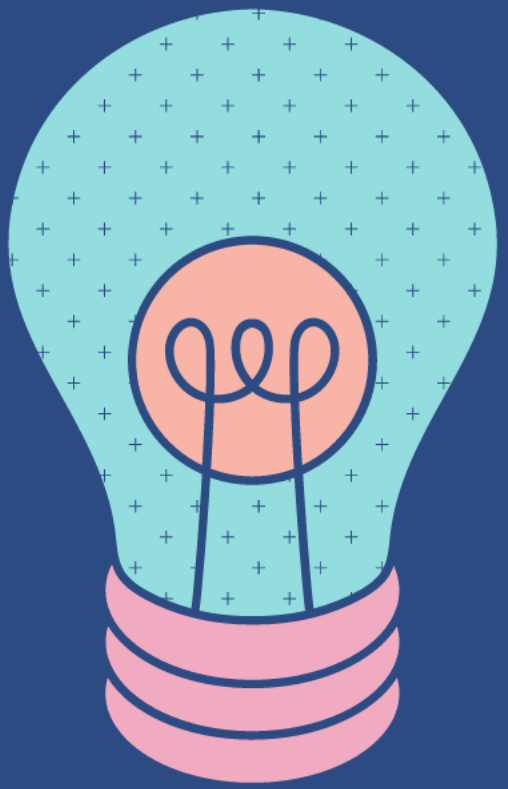
- **Operador OU (||)**: retorna **false** se, e somente se, **todos booleanos** envolvidos também forem **false**.

```
true || true //true
true || false //true
false || true //true
false || false //false
```

Vamos praticar?

- Antes de começar, crie 3 variáveis: a, b e c. Atribua os valores true, false e true, respectivamente.

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 - Booleans

- **Operador NÃO (!)**: sempre retorna o **booleano oposto.**

```
!true //retorna false
!false //retorna true
```

Operadores - Arrays

Operadores - Arrays

- **.length**: Determina o **tamanho** do array.

- O tamanho representa a quantidade de elementos neste array.

```
const array = [1, 2, 'a', 'maria', true, -9]

console.log(array.length) //6
```

Operadores - Arrays

- **.push(elemento)**: **Adiciona** um **elemento** no fim do array.

```
let array = [1, 2, 'a', 'maria', true, -9]

array.push('fui add')

console.log(array) //[1, 2, 'a', 'maria', true, -9, 'fui add']
```

Operadores - Arrays

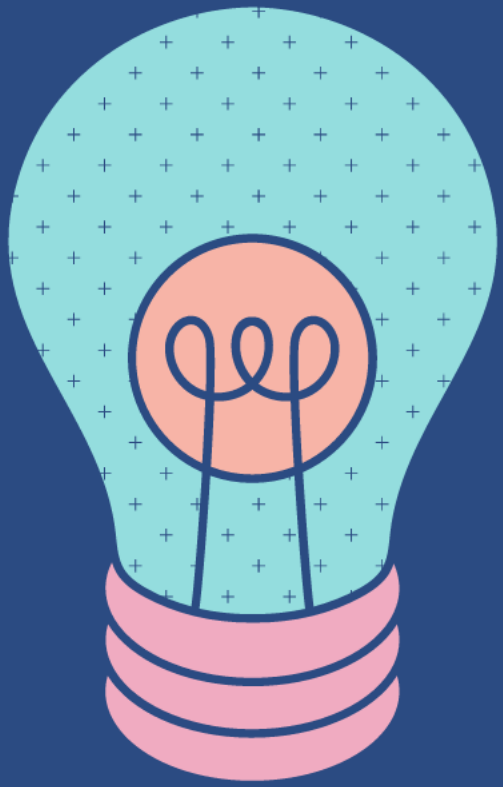
- **.splice(i, n)**: **Remove** n **elementos** a partir do índice i.

```
let array = [1, 2, 'a', 'maria', true, -9]

array.splice(2, 2)

console.log(array) //[1, 2, true, -9]

array = [1, 2, 'a', 'maria', true, -9]

array.splice(3, 2)

console.log(array) //[1, 2, 'a', -9]
```

Vamos praticar?

- 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.



Operadores - Strings

Operadores - Strings

- **Concatenação de Strings**:

- Concatenar significa unir.
- Conseguimos concatenar as Strings, usando o operador **+**

```
const nome = "Paola"
const sobrenome = "Bracho"
const idade = "25"

const minhaBio =
"Olá, sou a " + nome + " " + sobrenome + ", e tenho " + idade + " anos."

console.log(minhaBio) //Olá, sou a Paola Bracho, e tenho 25 anos.
```

Operadores - Strings

- **.length**:

- Também pode ser utilizado para se determinar o tamanho de uma string, a quantidade de caracteres nela.

```
const nome = "Paola"
const sobrenome = "Bracho"
const idade = "25"

const minhaBio =
"Olá, sou a " + nome + " " + sobrenome + ", e tenho " + idade + " anos."

console.log(minhaBio.length) //41
```

Operadores - Strings

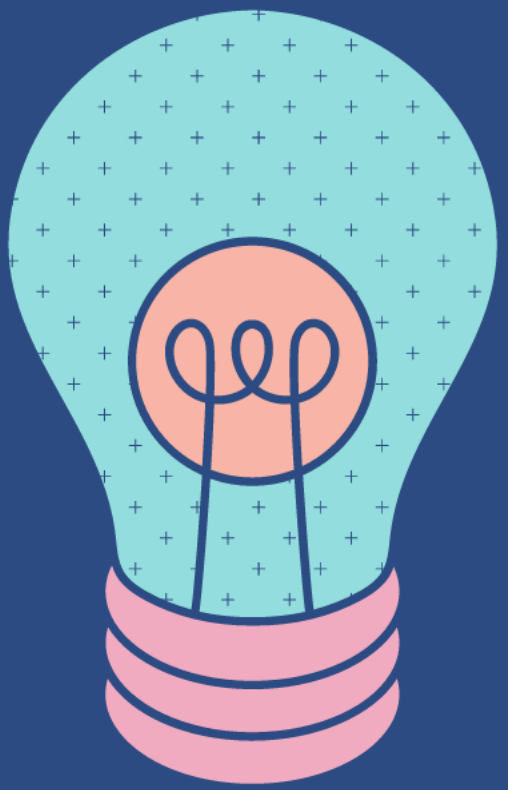
- **Cast**:

- **Transformar** o tipo de uma variável.
- Passando String para Number:

```
const idade = prompt("Qual a sua idade?")
const idadeDaMae = prompt("Qual a idade de sua mãe?")

const diferencaIdade = Number(idadeDaMae) - Number(idade)
```

Vamos praticar?

1. Crie uma variável e atribua seu primeiro nome;
2. Determine o tamanho do seu primeiro nome;
3. Crie uma variável e atribua seu sobrenome;
4. Determine o tamanho do seu sobrenome;
5. Crie uma variável que una seus dois nomes (com um espaço no meio);
6. Determine o tamanho do seu nome com seu sobrenome.



Dúvidas?

Obrigado!

Prof. Matheus Garrido