

Programador Web

Algoritmos e Lógica de Programação

Adriel Sales





O que é lógica?

“Parte da filosofia que trata das formas do pensamento em geral (dedução, indução, hipótese, inferência, etc.) e das operações intelectuais que visam à determinação do que é **verdadeiro** ou **falso**.”



Lógica



Tudo
começa
com um
PROBLEMA

Houston, we have a problem.
-Apollo 13 1995



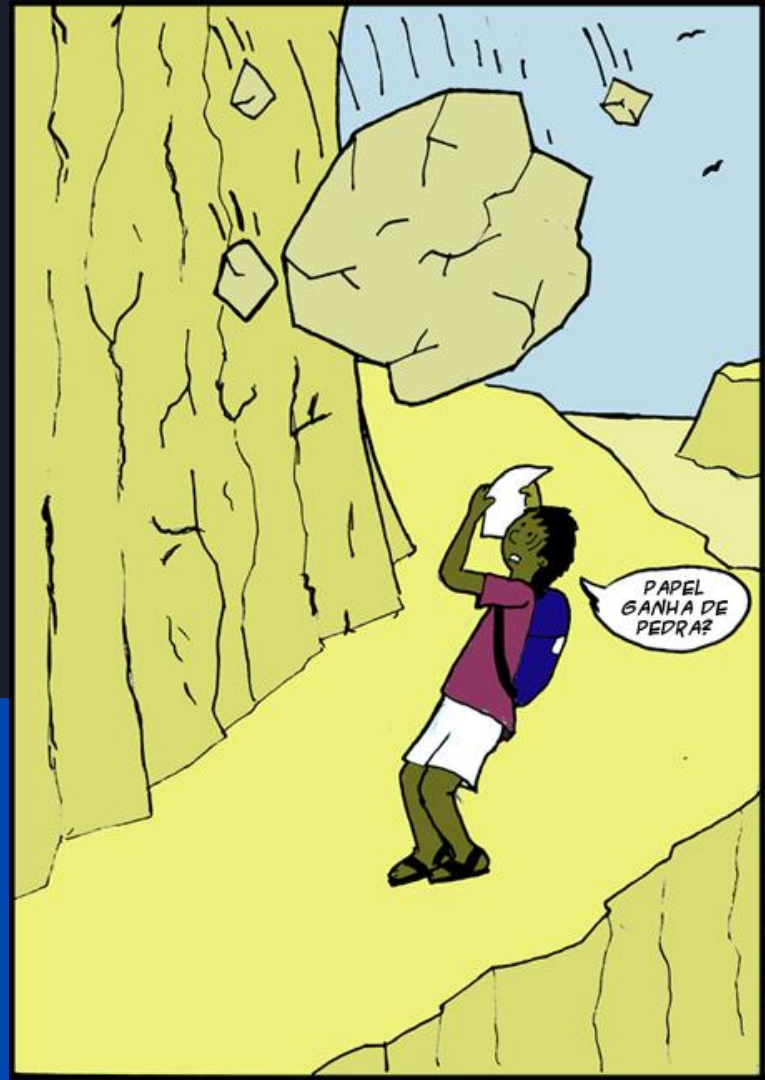
- Questão matemática proposta para ser resolvida.
- Questão difícil, delicada, suscetível de diversas soluções.
- Qualquer coisa de difícil explicação, mistério, enigma.
- Dúvida, questão.

PROBLEMA

Exemplos de problemas

- Trocar uma lâmpada
- Trocar o pneu de um carro
- Preparar o jantar

Porém, sempre que nos deparamos com um **problema**, buscamos sempre uma **SOLUÇÃO!**

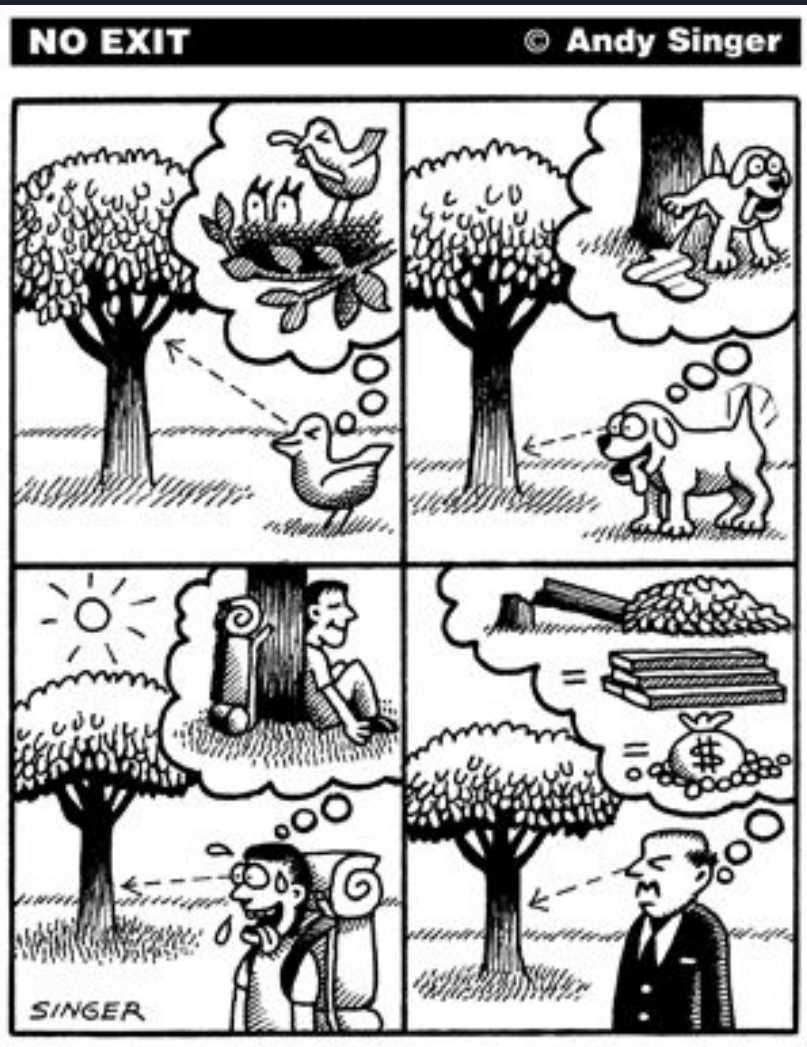


Lembre:

Um problema pode ter várias soluções!



Cada um
pensa de uma
maneira



Lógica de Programação



- **Lógica de Programação**
 - É a técnica de encadear pensamentos para atingir determinado objetivo.
- **Sequência Lógica**
 - São passos executados até atingir um objetivo ou solução de um problema.
- **Instruções**
 - Um conjunto de regras ou normas definidas para a realização de algo.

Ao utilizar lógica \Rightarrow Algoritmos



- **Algoritmo**

- É uma sequência finita de passos que levam a execução de uma tarefa ou solucionar um problema.

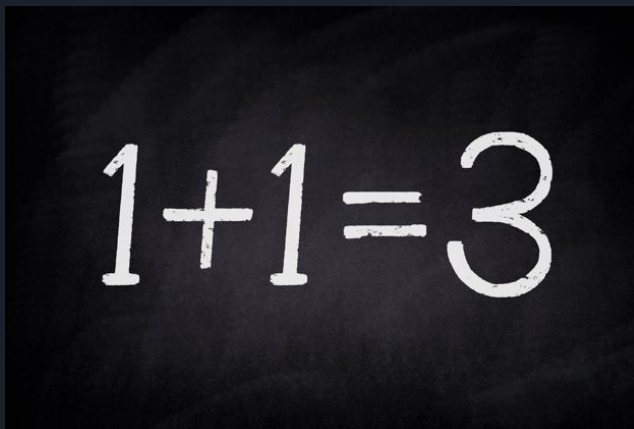
- **Programas**

- São algoritmos escritos em uma linguagem de programação e que são interpretados e executados por uma máquina.

Atenção!

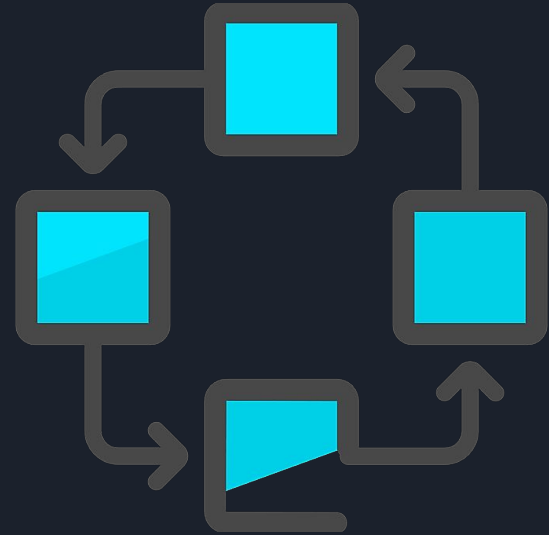
Cuidados devem ser tomados no desenvolvimento de um algoritmo:

Algoritmo Errado \Rightarrow Resultado Errado!


$$1+1=3$$

Métodos de Representação de Algoritmos

- Descrição Narrativa
- Pseudocódigo
- Fluxograma





Descrição Narrativa

Vantagem: simples de ser implementada.

Exemplo:

1. Adquira uma lâmpada nova.
2. Localize a lâmpada a ser trocada.
3. Em seguida, retire a lâmpada queimada e coloque a lâmpada nova.
4. Após a troca, descarte a lâmpada queimada.

Descrição Narrativa

Desvantagem:

pode haver erros ou equívocos na interpretação.





Pseudocódigo

- Utiliza-se de uma linguagem intermediária entre **linguagem falada** e a **linguagem de programação**.
- Deve ser **independente de linguagem de programação** a ser posteriormente utilizada na codificação.

Exemplo de um Pseudocódigo

Algoritmo: Procurar uma foto em um Livro

1. Pegar o Livro;
2. Abrir o Livro;
3. Visualizar a Página;
4. **Se** a foto procurada estiver na página
 1. Anotar o número da página;
 2. Guardar o livro;
 3. Fim.
5. **Se não**
 1. Virar a página;
 2. Voltar ao passo 3;



Laço de Repetição

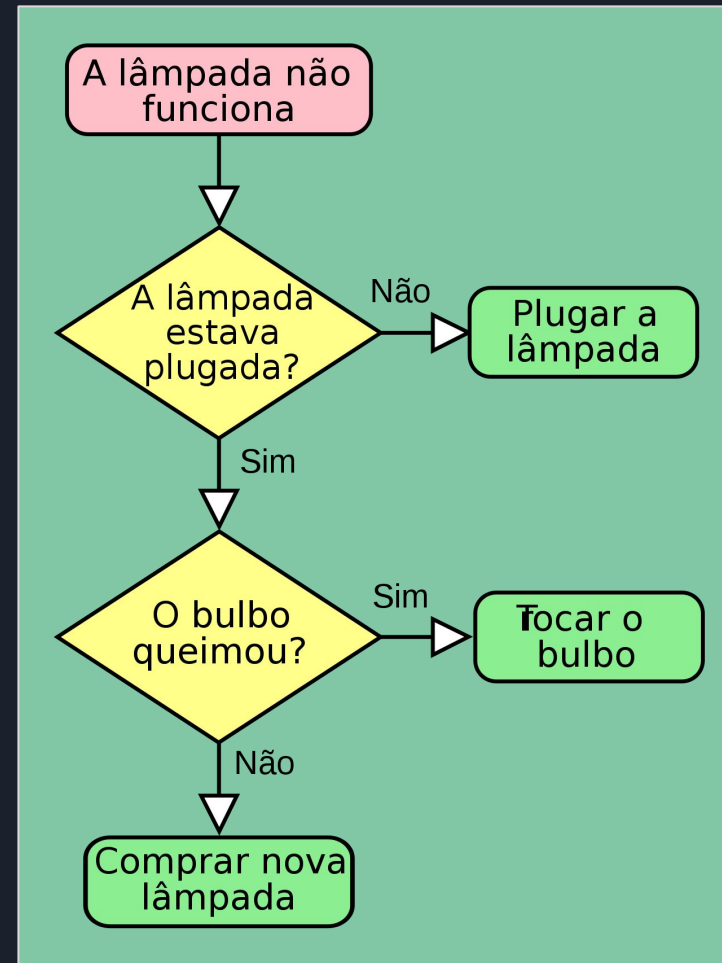
Instruções utilizadas:

Pegar, Abrir, Visualizar, Anotar, Guardar, Virar, Voltar.

ESTRUTURA CONDICIONAL (Se Verdadeiro, Senão)

Fluxograma

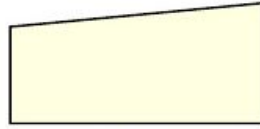
→ Modelo que utiliza figuras para representar o fluxo dos dados e os comandos do algoritmo. Cada operação a ser executada é representada por um símbolo cuja forma identifica o tipo de processo envolvido.



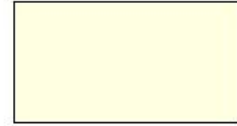
Fluxograma - símbolos básicos



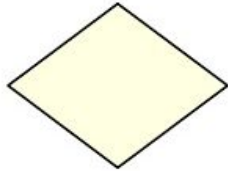
Início ou término
de um fluxograma



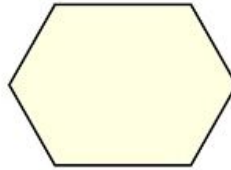
Entrada de dados
via teclado



Procedimento interno
e/ou
Mudança de conteúdo



Tomada de decisão

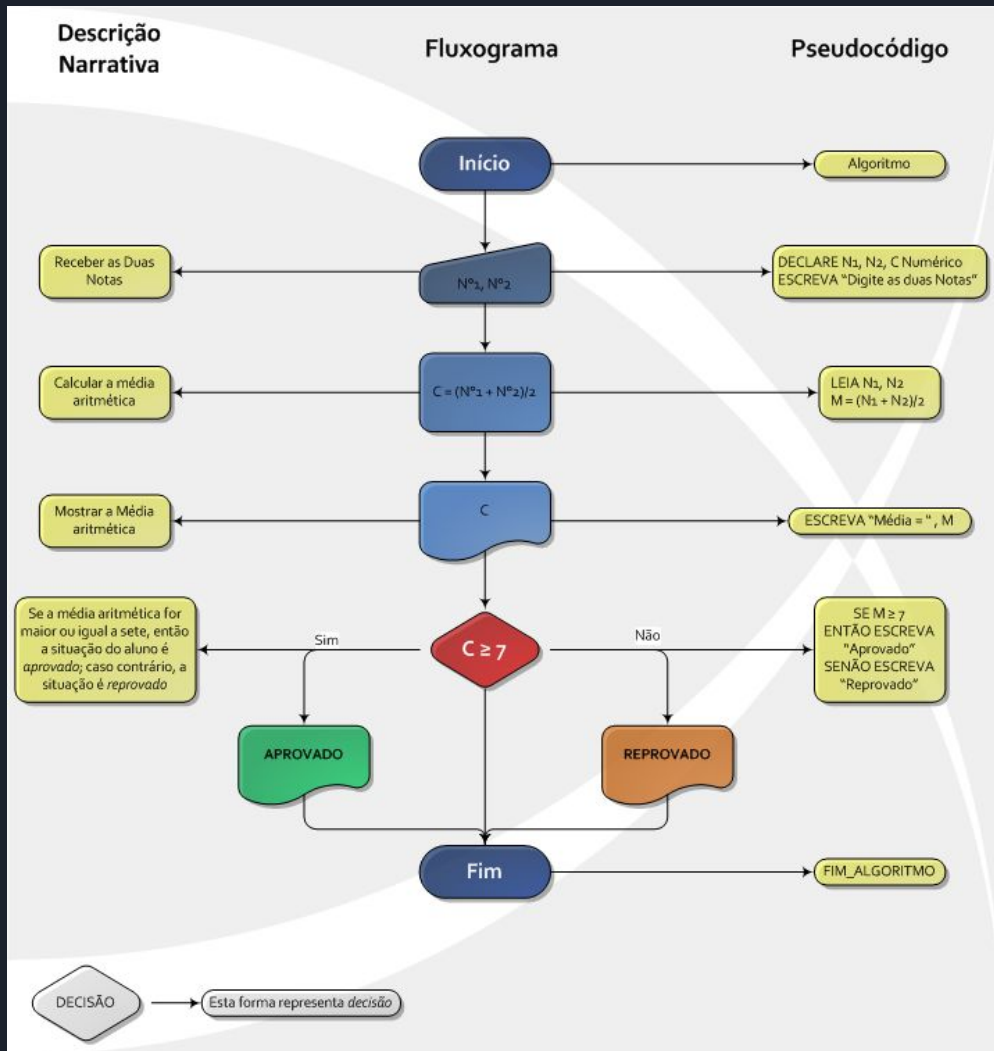


Repetição



Saída de dados
para monitor

Métodos de Representação de Algoritmos

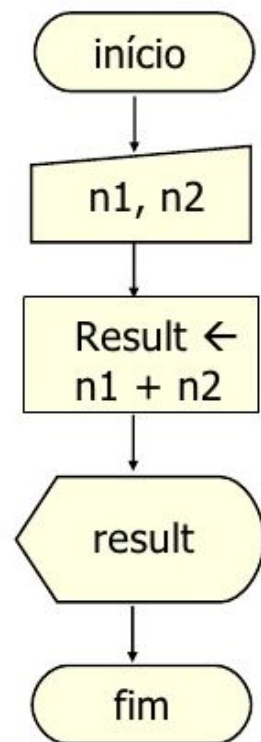


- × **Elaborar o algoritmo que faça a soma de dois números inteiros quaisquer:**

Pseudocódigo

```
algoritmo somadoisnum  
  var n1, n2, result: inteiro  
  
início  
  leia n1  
  leia n2  
  result  $\leftarrow$  n1 + n2  
  escreva result  
  
fim
```

Fluxograma



Linguagem de Programação

```
program somadoisnum;  
  var n1, n2, result: integer;  
  
begin  
  readln(n1);  
  readln(n2);  
  result := n1 + n2;  
  writeln (result);  
  
end.
```

**Primeiro Problema:
algoritmo para trocar
uma lâmpada.**



Primeiro algoritmo: trocando uma lâmpada.

1. Do que precisamos?
2. Como iremos proceder?
3. Qual foi o resultado final?

O que orienta a obtenção dos
procedimentos para a solução é a

Lógica.

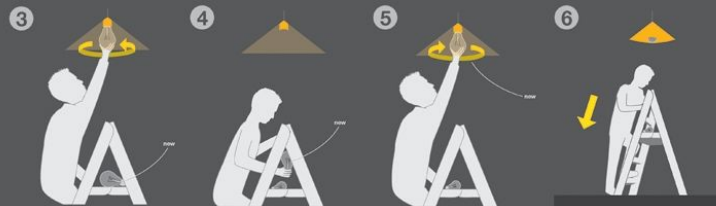
How to change a light bulb

things you need
x1 x1



First, please make sure the switch is off, the platform of the ladder is locked, and both hands are dry.

Place the new bulb on the platform of the ladder, climb up the ladder.



Rotate the old bulb in clockwise direction, take it off, and place it on the platform of the ladder.

Take the new bulb

Fix the new bulb by rotating it to the socket in anti-clockwise direction.

Climb down the ladder.



Switch on the light, it should light up if working properly.



Dispose the old light bulb wrapped by paper.

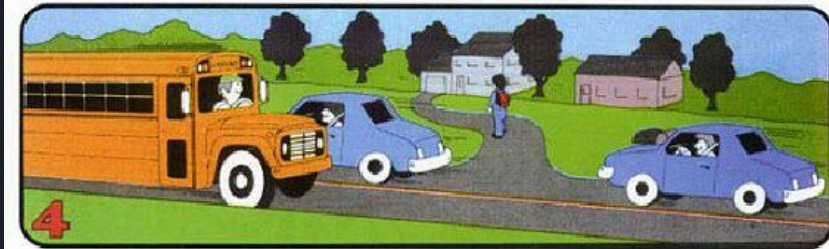
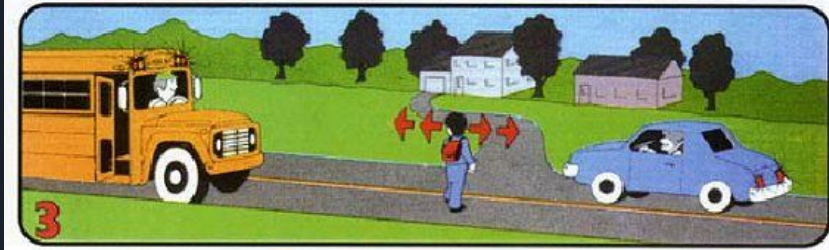
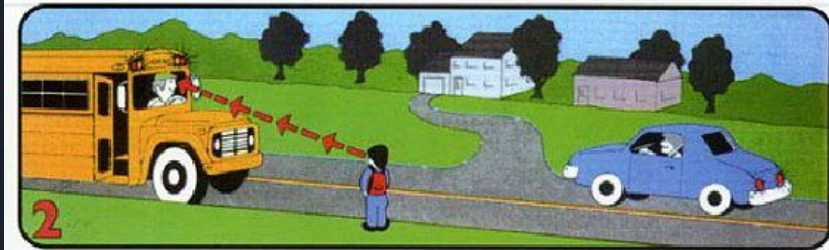
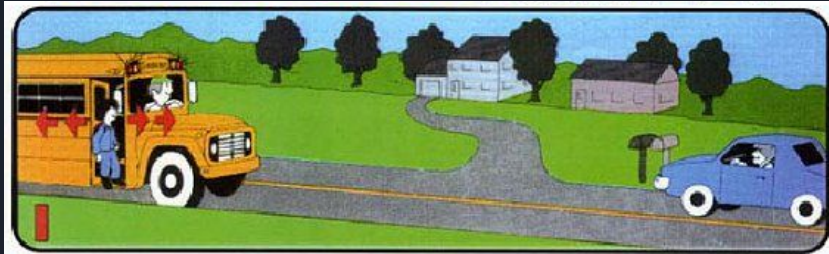
or



Dispose the old light bulb to light bulb recycling bin.

Segundo Problema: algoritmo para cruzar a rua.

1. Do que precisamos?
2. Como iremos proceder?
3. Qual foi o resultado final?





Algoritmos: Conceitos Básicos

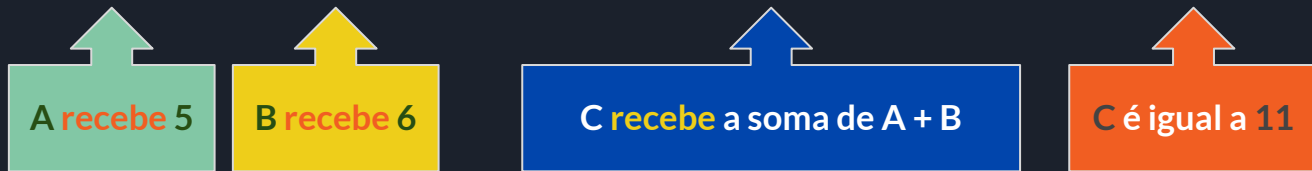
É preciso ter compreensão de alguns conceitos como:

- Constante;
- Variável;
- Identificador;
- Palavra Reservada;
- Entrada;
- Saída.

Algoritmos: Conceitos Básicos

- **Constante:** São endereços de memória destinados a armazenar informações fixas, inalteráveis durante a execução do programa. **Exemplo:** $\pi = 3.1416$.
- **Variável:** São endereços de memória destinados a armazenar informações temporariamente.

◆ Se $A = 5$ e $B = 6$, então, se $C = A + B$, logo $C == 11$.





Algoritmos: Conceitos Básicos

- **Identificadores:** São os nomes que damos as variáveis, constantes e programas: **Exemplo: `var idade`.**
- **Palavras Reservadas:** São identificadores predefinidos que possuem significados especiais para o interpretador do algoritmo.
 - ◆ **`var, if, else, function, round`, entre outras.**

Neste caso, estamos tratando da linguagem JavaScript

Algoritmos: Conceitos Básicos

- **Entrada:** São os dados informados no início ou durante a execução do programa, vindos ou não de um teclado.
- **Saída:** São os resultados do processo da computação, impressos ou exibidos em tela.





Operadores de Atribuição

→ Nome, operador e significado:

◆ Atribuição: $X = Y$ (x recebe y)



Operadores Aritméticos

→ Operação e Símbolo:

◆ Adição +

◆ Subtração -

◆ Multiplicação *

◆ Divisão /



Operadores Relacionais

→ Operação e Símbolo:

- ◆ Igual a ==
- ◆ Diferente de !=, <> ou # (no JS usaremos !=)
- ◆ Maior que >
- ◆ Menor que <
- ◆ Maior ou igual a >=
- ◆ Menor ou igual a <=



Tipos de dados

- Numérico: inteiro (0, 2, 11, -5) ou real (0.10, -5.25, 30.78).
- Literal (**string**): “Somos programadores web!”, “andar”.
- Lógico (**booleano**): true ou false (verdadeiro ou falso)



Operadores Lógicos

→ Operação e descrição:

- ◆ **AND** lógico (&&): Retorna verdadeiro caso ambos operandos sejam verdadeiros; caso contrário, retorna falso.

```
var a1 = true && true;    // t && t retorna true  
var a2 = true && false;   // t && f retorna false
```

Operadores Lógicos

→ Operação e descrição:

- ◆ **OU** lógico (`||`): se utilizado com valores booleanos, retorna verdadeiro caso ambos os operandos sejam verdadeiro; se ambos forem falsos, retorna falso..

```
var o1 = true || true;    // t || t retorna true
var o2 = false || true;   // f || t retorna true
var o3 = true || false;   // t || f retorna true
var o4 = false || (3 == 4); // f || f retorna false
```



Operadores Lógicos

→ Operação e descrição:

- ◆ **NOT** lógico (!): Retorna falso caso o único operando possa ser convertido para verdadeiro; senão, retorna verdadeiro.

```
var n1 = !true;    // !t retorna false  
var n2 = !false;   // !f retorna true  
var n3 = !"Gato";  // !t retorna false
```

Precedência Geral dos Operadores Aritméticos



Quando uma **expressão aritmética** precisa ser **avaliada num algoritmo**, o analisador processa a expressão **dando prioridade para certos operadores**. Essa ordem de prioridade na avaliação dos operadores numa expressão aritmética é chamada de **precedência de operadores**.

Precedência Geral dos Operadores Aritméticos



Ordem	Operação	Símbolo
1ª	Parênteses	()
2ª	Potenciação	**
3ª	Multiplicação, Divisão, Resto e Divisão Inteira	*, /, mod, div
4ª	Adição, Subtração	+, -

$$(5+3)**2 * (5-2) + 8$$

$$8**2 * 3 + 8$$

$$64 * 3 + 8$$

$$192 + 8$$

$$200$$



Mais um exemplo

Escreva um algoritmo que armazene o valor 20 em uma variável A e o valor 5 em uma variável B. em seguida, some as variáveis A com B e exiba o resultado (faça a descrição narrativa e o pseudocódigo).

Solução

Algoritmo: somaDoisNumeros

1. **A** recebe **20**.
2. **B** recebe **5**.
3. **Soma** recebe o valor de A somado ao valor de B.
4. Escreva o valor de **Soma**.

Descrição Narrativa

Algoritmo: somaDoisNumeros

var a, b, soma: **inteiro**

a = 20;

b = 5;

soma = a + b;

resultado = soma;

Escreva **resultado**;

Pseudocódigo

Hora de rodar codar JavaScript!

- É a linguagem do browser
- A linguagem mais popular do mundo
- Feita em 1995 para o Netscape 2.0 sob o nome de Mocha depois LiveScript e por fim JavaScript (ECMA Script).



O trio de sucesso do Front End



JavaScript

JS

```
<script type="text/javascript">
```

Tag HTML para JavaScript

Atribuição de valores

Declaração de variáveis

```
var num1 = 10;
```

```
var num2 = parseInt(prompt("Informe um número"));
```

Expressão aritmética

Concatenação de Strings

```
var soma = num1 + num2;
```

```
document.write("<h1>A soma é " + soma + "</h1>");
```

```
</script>
```

Parâmetro da função `write()`;

Obs.: funções podem receber 0 ou N parâmetros

Instruções acabam com ponto e vírgula (;)

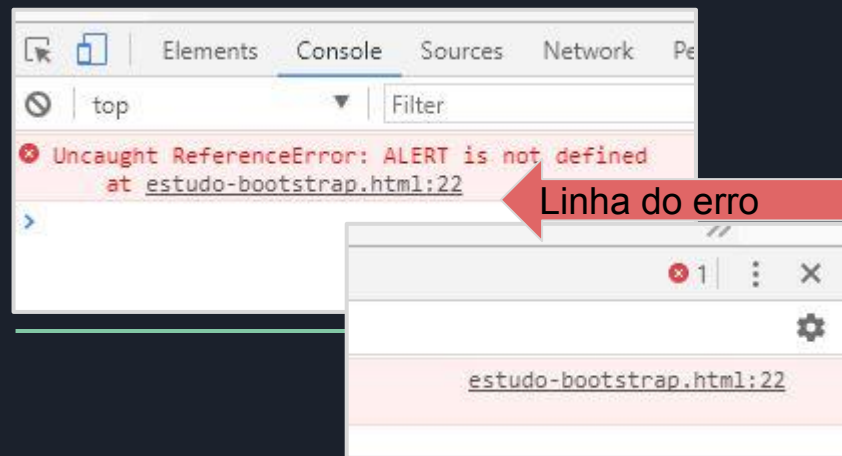
JavaScript é Case Sensitive

JS

`alert("hellow word!");`

FUNCIONA

`ALERT("hellow word!");`



NÃO FUNCIONA

JavaScript: declaração Constantes e Variáveis

JS

```
const HORAS_DO_DIA = 24;
```

```
var semCaracteresEspeciais;
```

```
var sem_Espaços-no-meio;
```

```
var sem_numeros_no_inicio_01;
```

```
const horas_do_dia = 24;
```

```
var @&* &* &"" %;
```

```
var minha variável;
```

```
var 123;
```

É aceitável usar \$ ou _ no início do nome. Ex: \$idade, _idade;

CORRETO

ERRADO





REFERÊNCIAS:

LACERDA, Ivan Max Freire de, OLIVEIRA, Ana Liz Souto.

Programador Web: um guia para programação e manipulação de banco de dados.

Rio de Janeiro: Senac Nacional, 2013.

W3schools: <http://www.w3schools.com>

Slide: Algoritmo e lógica de programação:

https://pt.slideshare.net/engenhariadecomputacao/algoritmo-e-logica-de-programacao-aula-1?qid=4b7e1bbc-c2c4-4e0e-b267-e50d0867c402&v=&b=&from_search=2

Slide: Material de apoio de algoritmo e lógica de programação:

https://pt.slideshare.net/rodfernandes/material-de-apoio-de-algoritmo-e-logica-de-programacao?qid=d0e7d40-1d80-4a6b-a730-d5da859fe9f1&v=&b=&from_search=1

Apontamentos da Aula:

<https://github.com/adrielsales/senac/wiki/Aulas-Senac>