

FULL STACK WEB DEVELOPER

FERNANDO LIRA

FLAG

FUNDAMENTOS DE PROGRAMAÇÃO (COM JAVASCRIPT)

APRESENTAÇÃO – FERNANDO LIRA

3



it.fernandolira@gmail.com



<https://www.linkedin.com/in/fernandolira74/>



+351 93 317 99 21



@fernandolira74




A **linguagem de programação** é um método padronizado, formado por um conjunto de **regras sintáticas e semânticas**, de implementação de um código fonte - que pode ser **compilado** e transformado em um programa de computador, ou usado como **script interpretado** - que informará instruções de processamento ao computador. Permite que um programador especifique precisamente quais os dados que o computador irá atuar, como estes dados serão armazenados ou transmitidos e, quais ações devem ser tomadas de acordo com as circunstâncias. Linguagens de programação podem ser usadas para expressar **algoritmos** com precisão.

FUNDAMENTOS DE PROGRAMAÇÃO (COM JAVASCRIPT)

5



 Stackscale

Linguagens de Programação - Ranking

<https://pypl.github.io/PYPL.html>

<https://insights.stackoverflow.com/survey/2021#most-popular-technologies-language>

<https://spectrum.ieee.org/top-programming-languages/>

<https://www.geeksforgeeks.org/top-10-programming-languages-that-will-rule-in-2021/>

Porquê JSCRIPT para começar?

Popularidade – todos os browsers executam nativamente o JS
(mas nesta unidade não interessa a web!)

Linguagem C – das linguagens mais importantes e usadas desde sempre até no desenvolvimento de vários sistemas operativos =>

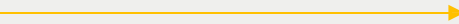
C-LIKE

Programa

- Conjunto de instruções que implementam uma tarefa específica
- É executado por um computador, sequencialmente, instrução a instrução
- É escrito por um **programador**, numa **linguagem de programação** específica, que é depois transformado em código máquina
 - Um computador apenas consegue “ler” instruções em código máquina
- Podem ser expressos sob a forma de um algoritmo

```
#include <stdio.h>

int main(void)
{
    printf("hello, world\n");
    return 0;
}
```



```
01010100000010100100101000101
10100001010100010010001010000
01000110001000001010000101001
00110100010101010110011000101
00000101000001001000100101010
10000101011000010011001011100
01101001001001001010010101010
```


Programar

☉ Arte (ou ciência?) de escrever programas de computador para resolver um determinado problema

-
- 1. Abstrair o problema da sua descrição
 - 2. Gerar sub-problemas
 - 3. Transformar sub-problemas em sub-soluções
 - 4. Integrar as sub-soluções num programa funcional
 - 5. Avaliar/Testar (e repetir)
- Perceber/Interpretar
- Planejar
- Implementar

Programar

- É uma sequência de passos para a realização de uma tarefa ou a resolução de um problema
- “Exemplos”
 - Receita num livro de culinária
 - Guia de reparação de uma oficina
 - ...
- Na escrita de um algoritmo é necessário identificar
 - Dados de entrada
 - Resultados de saída esperados
 - Processo pelo qual se obtém os resultados esperados

INGREDIENTES

1,5 kg amêijoas depuradas
1 cebola
6 dentes de alho
4 c. sopa azeite
1 molho de coentros
3 c. sopa vinho branco
1 limão



PREPARAÇÃO

1. Certifique-se de que as amêijoas estão bem depuradas, lave-as em água corrente e deixe-as escorrer dentro de um coador de rede.
2. Leve a cebola e os dentes de alho, finamente picados, ao lume com o azeite até começarem a alourar.
3. Adicione os pés dos coentros picados, incorpore as amêijoas e mexa.
4. Regue com o vinho branco, tape e cozinhe em lume moderado cerca de 5 minutos ou até as amêijoas abrirem, agitando o tacho de vez em quando para que todas recebam calor por igual.
5. Polvilhe com as folhas dos coentros e regue com o sumo do limão.

- Um algoritmo pode ser expresso de diferentes formas
 - Fluxograma
 - Pseudocódigo – linguagem informal, “inventada”, próxima da natural



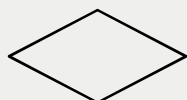
Utilizado no início e fim de um processo



Utilizado para processamento de um módulo



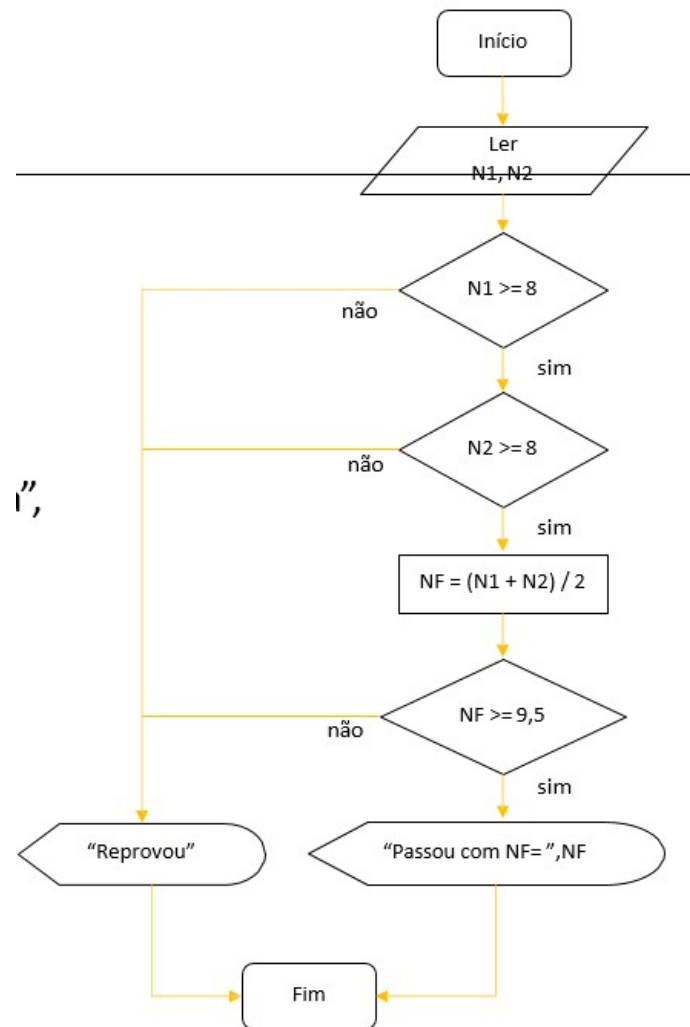
Utilizado para operações de entrada e saída de dados



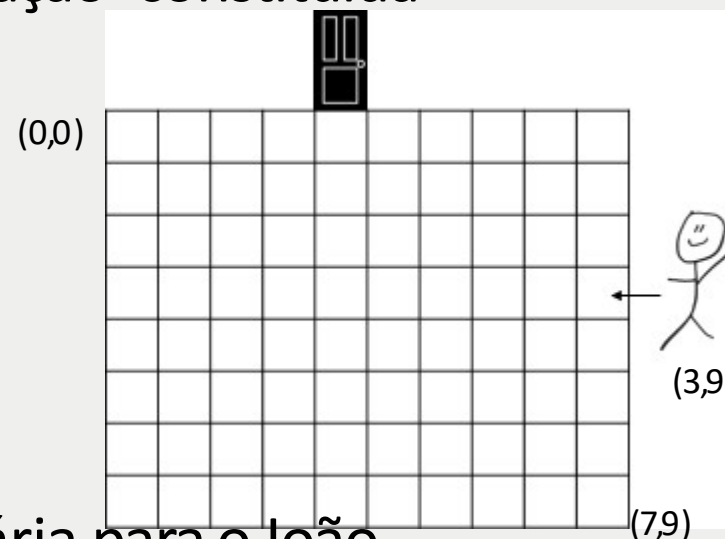
Utilizado para tomada de decisões



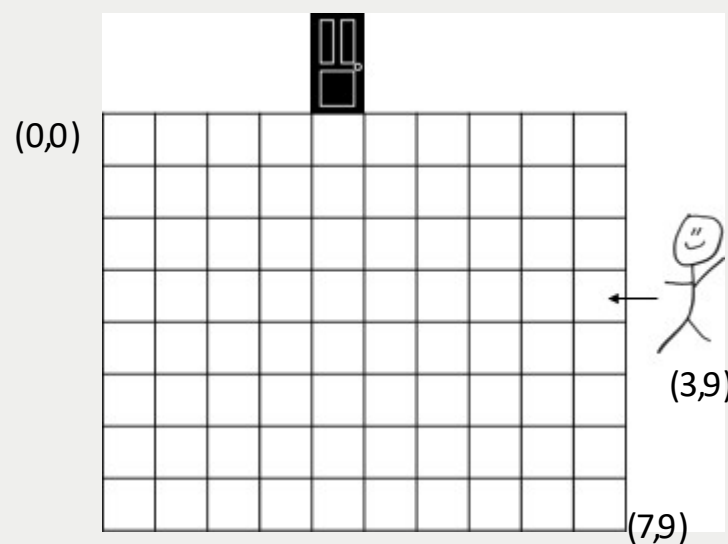
Utilizado para mostrar informações e resultados



- Considere o seguinte problema: O João está na posição (3,9) e quer sair da sala
- Considere uma “linguagem de programação” constituída pelas seguintes instruções
 - Caminhar em frente (1quadrícula)
 - Caminhar para trás (1quadrícula)
 - Rodar 90º esquerda
 - Rodar 90º direita
 - Abrir porta
 - Fechar porta
- Quais os dados de entrada?
- Qual o resultado esperado?
- Qual a sequência de instruções necessária para o João sair da sala?

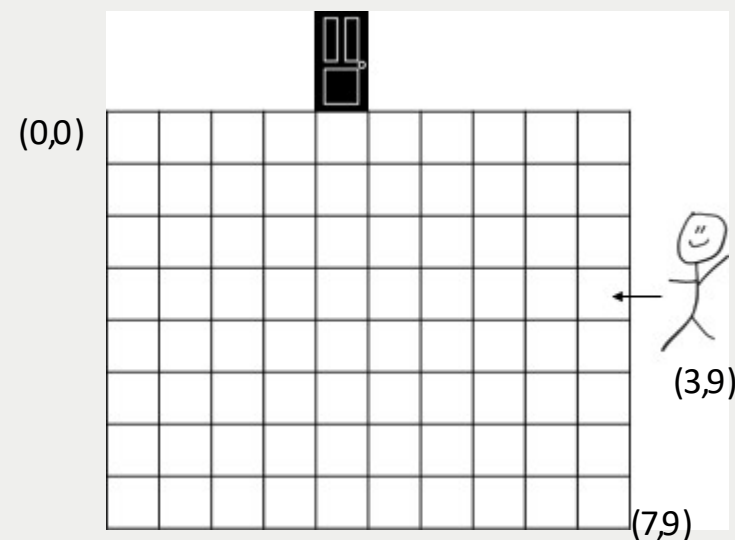


Caminhar em frente
Caminhar em frente
Caminhar em frente
Caminhar em frente
Caminhar em frente
Rodar 90° direita
Caminhar em frente
Caminhar em frente
Caminhar em frente
Abrir porta
Caminhar em frente
Fechar porta



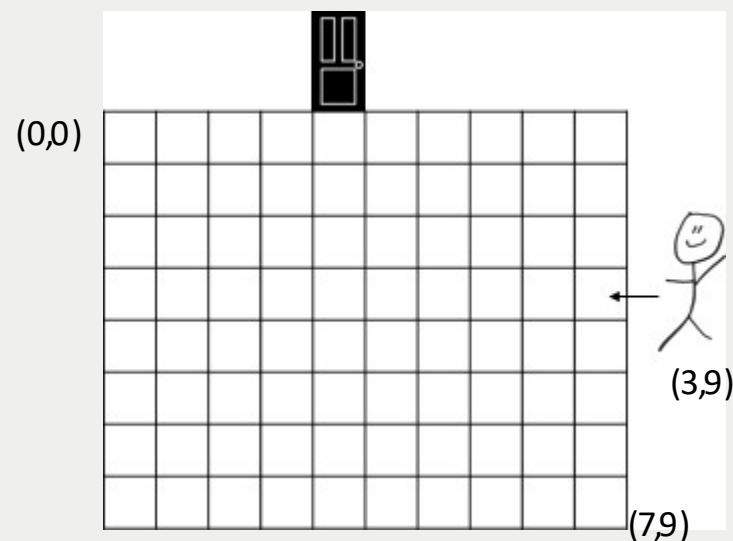
- O que acontece à medida que cada linha do algoritmo é executada?
- Qual o resultado final?
- Pode dizer-se que o algoritmo resolve o problema?

```
Caminhar em frente  
Caminhar em frente  
Caminhar em frente  
Caminhar em frente  
Caminhar em frente  
Rodar 90° direita  
Caminhar em frente  
Caminhar em frente  
Caminhar em frente  
Abrir porta  
Caminhar em frente  
Fechar porta
```



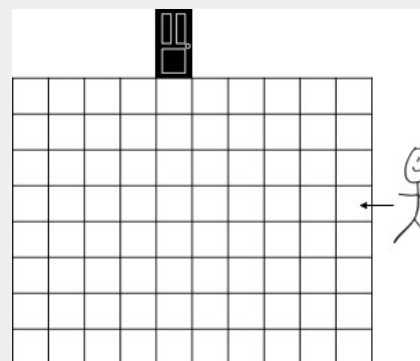
- Um algoritmo ou programa diz-se correto quando resolve o problema para o qual foi desenhado
- Um programa deve ser **testado** em diferentes cenários ou casos de uso
 - Há programas que resolvem apenas uma parte do problema ou que falham em determinadas condições
- O algoritmo à direita, resolve o problema?

Caminhar em frente
Rodar 90° direita
Caminhar em frente
Rodar 90° direita
Caminhar em frente
Caminhar em frente
Caminhar em frente
Abrir porta
Caminhar em frente
Fechar porta



- Os dois algoritmos à direita resolvem o problema, no entanto são diferentes
- Existem muitas soluções diferentes para o mesmo problema
 - Cada programador desenvolve mecanismos mentais e técnicas de resolução de problemas próprios
 - Algumas soluções são melhores que outras

Caminhar em frente	Caminhar em frente
Caminhar em frente	Caminhar em frente
Caminhar em frente	Caminhar em frente
Caminhar em frente	Caminhar em frente
Caminhar em frente	Caminhar em frente
Rodar 90° direita	Caminhar em frente
Caminhar em frente	Caminhar em frente
Caminhar em frente	Caminhar para trás
Caminhar em frente	Caminhar para trás
Abrir porta	Rodar 90° direita
Caminhar em frente	Caminhar em frente
Fechar porta	Caminhar em frente
	Caminhar em frente
	Abrir porta
	Caminhar em frente
	Fechar porta

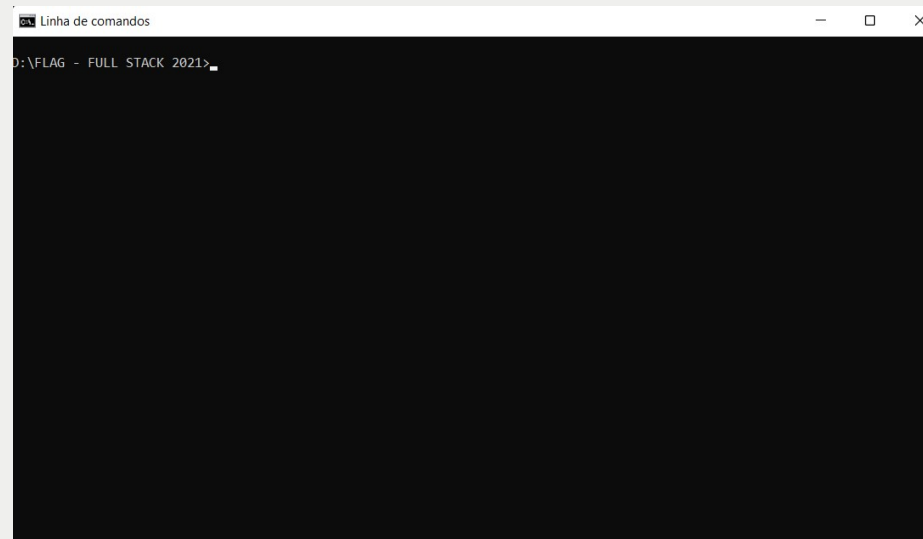


- Os problemas (e a sua resolução) podem ser classificados de acordo com a sua dificuldade/complexidade
- Um problema (ou um programa) mais complexo vai requerer mais recursos para ser resolvido
 - No exemplo anterior, a solução da direita tem 4 instruções a mais
- Um programador deve ter como preocupação implementar programas que façam uma gestão adequada da memória e do processamento necessário para executar o programa

Ambiente de desenvolvimento

1. Online
2. Browser
3. Node + Visual Studio Code

A linha de comandos – comandos básicos



Visual Studio Code

- Code Runner (.run)
- Material Icon Theme
- Bracket Pair Colorizer 2

Visual Studio Code

Add cursors to all matching selections

Mac: CMD + SHIFT + L

Windows/Linux: CTRL + SHIFT + L

Add cursor to next matching selection

Mac: CMD + D

Windows/Linux: CTRL + D

Undo last cursor operation

Mac: CMD + U

Windows/Linux: CTRL + U

Go to a specific line

Mac: CTRL + G

Windows/Linux: CTRL + G

Close all open editor tabs

Mac: CMD + K + W

Windows/Linux: CTRL + K + W

Toggle block comment

Mac: SHIFT + OPTION + A

Windows: SHIFT + ALT + A, Linux: CTRL+SHIFT+A

Format document

Mac: SHIFT + OPTION + F

Windows: SHIFT+ALT+F, Linux: CTRL+SHIFT+I

Sentenças e blocos de código

```
console.log("Etapa #1");  
console.log("Etapa #2");  
  
console.log("Etapa #3")  
  
{  
    console.log("Etapa #4");  
    console.log("Etapa #5");  
    console.log("Etapa #6");  
}
```

Sentenças e blocos de código

```
console.log(  
    "Etapa #7"  
);  
  
console.log(1);  
console.log(4+5);  
  
console.log("Idade: "+45);  
console.log("4"+8);
```


Comentários

```
//Comentário de uma única linha
/*
Comentário em várias linhas
*/

/*
* Mais visível assim
* desta forma
* fica mais destacado
*/

console.log("Etapa #8"); //Comentário de linha
```

Dados

```
console.log("Computador pessoal");  
console.log(2);  
console.log(750.99);  
console.log(45.5);  
  
var nome = "Computador pessoal"  
var quantidade = 2;  
var preco = 750.99;  
var imposto = 45.5;  
var precoFinal = preco + imposto;  
  
console.log(nome);  
console.log(quantidade);  
console.log(preco);  
console.log(imposto);  
console.log(precoFinal);  
  
nome = "Computador pessoal ASUS";  
console.log(nome);
```

Dados

```
var preco = 19.90;  
var desconto= 0.4;  
  
console.log(19.90 * 0.6);  
  
console.log(preco * (1 - desconto));  
  
var precoComDesconto = preco * (1 - desconto);  
console.log(precoComDesconto);
```

Dados

```
var nome = "Caderno";  
var categoria = "Papeleria";  
console.log(nome + categoria);  
  
console.log(  
    "Produto:" + nome,  
    "Categoria: " + categoria,  
    "Preço: " + preco,  
    "Desconto: " + desconto  
);
```

Tipos Básicos de Dados

```
var idade = 47;  
console.log(typeof 47);  
console.log(typeof idade);  
  
var salario = 2345.67;  
console.log(typeof salario);  
  
var reformado = false; // ou true  
console.log(typeof reformado);  
  
console.log(typeof "Teste");  
console.log(typeof 'teste');
```

Constantes

```
var a = 3;  
  
// ...  
  
a = 10;  
  
// ...  
  
a = a + 5;  
  
console.log(a);  
  
const B = 7;  
  
console.log(B);
```

EXERCÍCIOS

Desafios

Área da circunferência

```
// PI * raio * raio  
// PI = 3.141592 ou Math.PI
```

Troca de Valores

```
// a = 7  
// b = 94  
// ... depois da troca a = 94 e b = 7
```

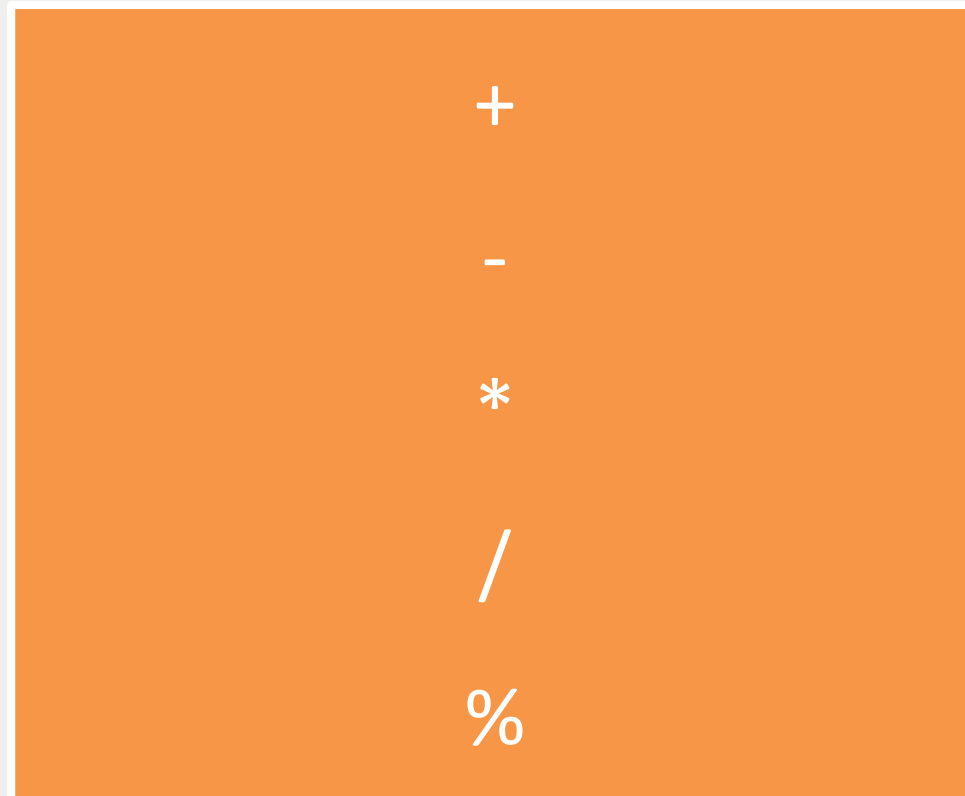

Classificação dos operadores

- Binários
- Unários
- Ternários

Tipos de operadores

- Aritméticos
- Relacionais
- Atribuição
- Lógicos
- Unários
- Ternários

Operadores Aritméticos



EXERCÍCIOS

Desafios

Converter milhas em quilómetros

// 1 milha = 1.609344

Converter Fahrenheit em celsius

// Celsius = (fahrenheit – 32) / 1.8

Operadores Relacionais

>

>=

<

<=

==

!=

Tipagem forte vs Tipagem fraca

```
var a = 1;  
var b = "1";  
  
console.log(a == b);  
console.log(a === b);  
console.log(a != b);  
console.log(a !== b);
```

Operadores de atribuição

```
var x = 0;  
  
x = x + 3;  
  
console.log(x);  
  
x += 5;  
  
console.log(x);  
  
var a = "Qualquer";  
  
a += " coisa";  
  
console.log(a);
```


Operadores lógicos – E - &&

```
var temDinheiro = true;  
var estaSol = true;  
  
var resultadoE = "#1 - Vai para o Shopping? ";  
  
resultadoE += temDinheiro && estaSol;  
  
console.log(resultadoE);
```

Operadores lógicos – OU - ||

```
var temDinheiro = true;
var estaSol = true;
var temCarro = true;

var resultadoE = "#1 - Vai para o Shopping? ";

resultadoE += temDinheiro && estaSol;

console.log(resultadoE);

var resultadoOU = "#2 - Vai para o Shopping? ";

resultadoOU += estaSol || temCarro;

console.log(resultadoOU);
```

Operadores lógicos – XOR – Ou Exclusivo

```
console.log(true ^ false);  
console.log(true ^ true);  
console.log(false ^ false);  
console.log(false ^ true);  
  
console.log(true != false);  
console.log(true != true);  
console.log(false != false);  
console.log(false != true);
```

Operadores lógicos – NOT - Negação

```
console.log(!true);  
console.log(!false);
```

Operadores unários

```
var a = 1;  
  
a = a + 1;  
a += 1;  
a++;  
++a;  
  
a--;  
--a;  
  
console.log(a);
```

Operadores unários – True ou False?!?

```
var a=1;  
var b=2;  
  
console.log( ++a == b++ );
```

Operadores unários – !!

```
console.log(!!true);  
console.log(!!false);  
  
console.log(!!"Texto");  
console.log(!!"");  
  
console.log(!!1);  
console.log(!!0);  
console.log(!!-1);  
console.log(!!-0.00001);
```

Operadores ternários

```
const hora = 14;  
const saudacao = hora <= 11 ? "Bom dia!" : "Boa tarde";  
  
console.log(saudacao);
```


EXERCÍCIOS

Desafios

Indicar se um número é par ou ímpar

```
const numero = 25;
```

Indicar se um número é positivo, negativo ou igual a zero

```
const numero = -4;
```

Realizar uma das 4 operações aritméticas básicas (+ - * /) entre dois números de acordo com o operador indicado

```
const a = 3;
```

```
const b = 4;
```

```
const operador = "+";
```

Estruturas de controlo - IF

```
if (true)
    console.log("Vai ser executado");

if (false)
    console.log("Não vai ser executado");

console.log("Fim da execução");
```

Estruturas de controlo - IF

```
const nota = 12;  
const bomComportamento = false;  
  
if (nota >= 9.5 && bomComportamento)  
  console.log("Parabéns!");  
  
if (nota < 9.5 || !bomComportamento)  
  console.log("Uma pena!");  
  
console.log("Fim!");
```

Estruturas de controlo - IF

```
if (nota >=14) {  
    console.log("Quadro de Honra");  
    console.log("Parabéns!");  
}
```

Estruturas de controlo – IF – Problema...

```
if (nota >=14); {  
    console.log("Quadro de Honra");  
    console.log("Parabéns!");  
}
```

EXERCÍCIOS

Desafios

De acordo com a classificação, indique o escalão

Excelente – entre 18,0 e 20,0

Bom – 14,0 e 17,9

Suficiente – entre 9,5 e 13,9

Insuficiente – entre 4,0 e 9,4

Mau – entre 0,0 e 3,9

Estruturas de controlo – IF / ELSE

```
if (true)
  console.log("Vai ser executado");
else
  console.log("Não vai ser executado");

if (false)
  console.log("Não vai ser executado");
else
  console.log("Vai ser executado");
```

Estruturas de controlo – IF / ELSE

```
const hora = 14;  
  
var saudacao;  
  
if (hora<12)  
    saudacao = "Bom dia!";  
else  
    saudacao = "Boa tarde!";  
  
console.log(saudacao);
```

Estruturas de controlo – IF / ELSE

```
const hora = 15;

var saudacao;

if (hora<12)
    saudacao = "Bom dia!";
else
    if (hora<19)
        saudacao = "Boa tarde!";
    else
        saudacao = "Boa noite!";

console.log(saudacao);
```

EXERCÍCIOS

Desafios

Indique o maior de 2 números

```
const numA = 15
```

```
const numB = 12
```

Indique o maior de 3 números

```
const numA = 15
```

```
const numB = 12
```

```
const numC = 17
```

Desafios

Indique o número de dias existentes num mês (Fevereiro=28)

```
const mes = 4
```

Verifique se um ano é bissexto

```
const ano = 2018
```

[//https://docs.microsoft.com/pt-pt/office/troubleshoot/excel/determine-a-leap-year](https://docs.microsoft.com/pt-pt/office/troubleshoot/excel/determine-a-leap-year)

Verifique se uma data é válida

```
const dia = 15
```

```
const mes = 12
```

```
const ano = 2021
```

Estruturas de controlo – SWITCH

```
const mes = 2;
var mesExtenso;

switch (mes) {
  case 1: mesExtenso = "Janeiro"; break;
  case 2: mesExtenso = "Fevereiro"; break;
  case 3: mesExtenso = "Março"; break;
  case 4: mesExtenso = "Abril"; break;
  case 5: mesExtenso = "Maio"; break;
  case 6: mesExtenso = "Junho"; break;
  case 7: mesExtenso = "Julho"; break;
  case 8: mesExtenso = "Agosto"; break;
  case 9: mesExtenso = "Setembro"; break;
  case 10: mesExtenso = "Outubro"; break;
  case 11: mesExtenso = "Novembro"; break;
  case 12: mesExtenso = "Dezembro"; break;
  default: mesExtenso = "Mês inválido!";
}

console.log(mesExtenso);
```

Estruturas de controlo – SWITCH

```
const nota = 6.0;
var avaliacao;

switch (Math.ceil(nota)) {
  case 1:
  case 2:
  case 3:
  case 4: avaliacao = "Insuficiente"; break;
  case 5:
  case 6: avaliacao = "Suficiente"; break;
  case 7:
  case 8:
  case 9: avaliacao = "Bom"; break;
  case 10: avaliacao = "Excelente"; break;
  default: avaliacao = "Nota inválida!";
}

console.log(avaliacao);
```


Estruturas de Repetição – FOR

```
for(var contador = 1; contador <= 10; contador++)  
    console.log(contador);
```

Estruturas de Repetição – Template String

```
var i=2;  
var j=4;  
  
console.log("i = " + i + " j = " + j);  
  
//template string  
console.log(`i = ${i} j = ${j}`);
```

EXERCÍCIOS

Desafios

Apresente a tabuada de um número

const tabuada = 8

Apresente todas as tabuadas de 1 a 10

Apresente metade de uma árvore de natal com # para 6 linhas

#

##

###

####

#####

#####

Desafios

Apresente metade de uma árvore de natal com # para 6 linhas sem usar números

```
#  
##  
###  
####  
#####  
#####
```

Desafios

Apresente metade de uma árvore de natal com # indicando a base

Const base = 7 (número ímpar)

#

###

#####

#####

```
//var arredondadoParaBaixo = Math.floor(3/2);
```

```
//var arredondadoParaProximo = Math.round(3/2);
```

```
//var arredondadoParaCima = Math.ceil(3/2);
```

Estruturas de Repetição – WHILE

```
var contador = 1;  
  
while (contador<=10)  
    console.log(contador++);
```

Estruturas de Repetição – DO WHILE

```
var contador = 1;  
  
do {  
    console.log(contador++);  
} while (contador <=10)
```


Estruturas de Repetição – BREAK e CONTINUE

```
for (var contador = 1; contador <= 10; contador++) {  
    if (contador === 3)  
        break;  
    console.log(contador);  
}  
  
for (var contador = 1; contador <= 10; contador++) {  
    if (contador === 3)  
        continue;  
    console.log(contador);  
}
```

Extra – o undefined, o null e o NaN

```
var a;  
var b = 7;  
var c = null;  
  
console.log(a);  
console.log(b);  
console.log(c);  
console.log(a + b);  
console.log(b + c);
```

Estruturas de Repetição – número aleatório

```
const min = 1;  
const max = 20;  
  
var aleatorio = Math.floor(Math.random() * (max - min + 1)) + min;  
  
console.log(aleatorio);
```

EXERCÍCIOS

Desafios

Implemente um programa que um inteiro n e n inteiros (gerados aleatoriamente entre 1 e 100), determinando o maior e o menor desses n inteiros.

Implemente o seu programa usando (i) uma instrução de ciclo `for` e (ii) uma instrução de ciclo `while`.

Exemplo de dados: 4 6 8 1 3

Resultado: 8 1

Implemente um programa que em 100 inteiros (gerados aleatoriamente entre 1 e 100), determine e indique a maior sequência crescente.

Exemplo de dados: 4 6 8 1 3 5 7 3 4 3 6 7 1

Resultado:

Nº Elementos: 4

Sequência: 1 3 5 7

Desafios

Implemente um programa que imprima o número de divisores de um inteiro dado.
Exemplo de dados: 6
Resultado: 4 (pois 6 tem 4 divisores, 1, 2, 3 e 6).

Implemente um programa que determine se um inteiro dado é primo.
Exemplo de dados: 23
Resultado: Sim

Determinar quantos são os números primos existentes entre os valores 1 e 1000 (excluindo os limites do intervalo).

Determinar o número primo mais alto anterior a um número N

Desafios

O resultado da divisão inteira de um número inteiro por outro número inteiro pode sempre ser obtido utilizando-se apenas o operador de subtração. Assim, se quisermos calcular $(7/2)$, basta subtrair o dividendo (2) ao divisor (7), sucessivamente, até que o resultado seja menor do que o dividendo.

O número de subtrações realizadas corresponde ao quociente inteiro, conforme o exemplo seguinte:

$$7 - 2 = 5$$

$$5 - 2 = 3$$

$$3 - 2 = 1$$

Descrever um programa para o cálculo da divisão de um inteiro pelo outro. Note que se o dividendo for zero, esta é uma operação matematicamente indefinida.

Desafios

Um número n é perfeito se a soma dos divisores inteiros de n (excepto o próprio n) é igual ao valor de n . Por exemplo, o número 28 tem os seguintes divisores: 1, 2, 4, 7, 14, cuja soma é exactamente 28. (Os seguintes números são perfeitos: 6, 28, 496, 8128.) Escreva um programa que verifique se um número é perfeito.

Escreva um programa para extrair os algarismos que compõem um número e os visualize individualmente.

Escreva um programa que calcule a soma dos algarismos que compõem um número. Por exemplo: $7258 = 7+2+5+8 = 22$

Escreva um programa que dado um inteiro positivo N e verifica se ele contém o dígito 3 em qualquer posição.

O número 3025 possui a seguinte característica: $30 + 25 = 55 \rightarrow 55*55 = 3025$. Fazer um programa para obter todos os números de 4 algarismos com a mesma característica do número 3025

Desafios

De 1000 numeros gerados aleatoriamente entre 1 e 100 quantas vezes se repete o número N

Faça um programa que receba um montante em euros (com cêntimos), e que determina o menor número de moedas de cada tipo necessário para perfazer esse montante. Pode utilizar moedas de euros de todos os valores disponíveis (2€, 1€, ..)

Calculo do fatorial de N

Escrever numero N (entre 0 e 9999) por extenso

Dado um país A, com 5000000 de habitantes e uma taxa de natalidade de 3% ao ano, e um país B com 7000000 de habitantes e uma taxa de natalidade de 2% ao ano, escrever um programa que seja capaz de calcular e mostrar o tempo necessário para que a população do país A ultrapasse a população do país B.