INCIANDO A JAVASCRIPT







QUE A FORÇA ESTEJA COM VOCÊ. PROF. FERNANDO LUCAS

PROF. FERNANDO LUCAS

INTRODUÇÃO AO JAVASCRIPT

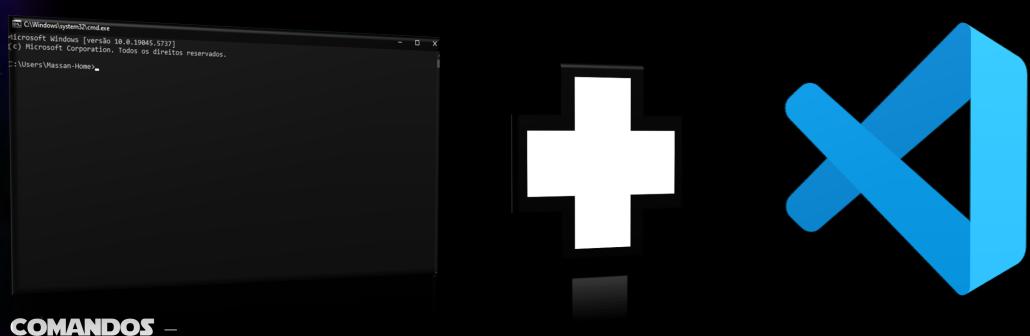
JAVASCRIPT É UMA LINGUAGEM DE PROGRAMAÇÃO USADA PARA ADICIONAR LÓGICA, COMPORTAMENTO E INTERATIVIDADE.

ELA É LEVE, INTERPRETADA, MULTIPARADIGMA E MUITO USADA TANTO NO NAVEGADOR QUANTO NO BACKEND (COM NODE.JS).





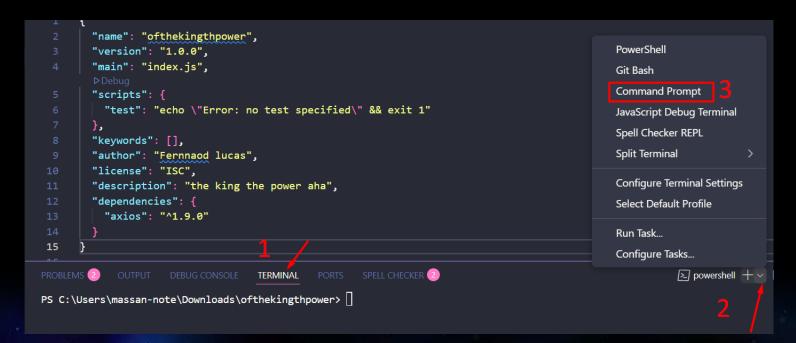
VAMOS UTILIZAR O CMD PARA CRIAR UMA NOVA PASTA E ABRIR A MESMA NO VSCODE.



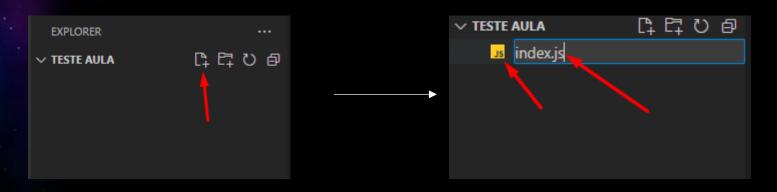
DIR – LISTA OS ARQUIVOS CD 'NOME DA PASTA' – NAVEGA NAS PASTAS MKDIR – CRIA UMA PASTA NOVA CODE . – ABRE O VSCODE NA PASTA QUE VOCÊ ESTA.

SEMPRE UTILIZAMOS O CMD PARA EXECUTAR NOSSOS PROJETOS. VAMOS CONTINUAR UTILIZANDO ELE MAS AGORA DENTRO DO VSCODE.

NO VSCODE APERTE CTRL J PARA ABRIR O TERMINAL. CASO ELE VENHA COM O POWERSHELL VOCÊ DEVE ALTERAR PARA O COMAND PROMPT (QUE É O CMD).



CRIANDO ARQUIVOS EM JAVASCRIPT NO VSCODE.



CRIE UM NOVO ARQUIVO 5.
ESCOLHA UM NOME QUE POR PADRÃO É INDEX, E REPARE QUE O QUE VOCÊ
COLOCA DEPOIS DO '.' É QUE DEFINE O TIPO DO ARQUIVO.
PODERIA SER:

INDEX HTML

INDEX.CSS

INDEX. JS

INDEX.PKG

ENTRE OUTRAS 1000 LINGUAGENS QUE EXISTEM.

CRIANDO UMA CONSTANTE E MOSTRANDO ELA NO CONSOLE.

```
teste.js > ...

const menssage = "Hello World"
console.log(menssage)
```

COPIE NO SEU NOVO ARQUIVO.JS

CTRL J
PARA ABRIR O CONSOLE

VERIFIQUE SE VOCÊ ESTA NA PASTA CORRETA.

AGORA NO CONSOLE ESCREVA.

O NOME DO MOTOR (espaço) O NOME DO ARQUIVO QUE VOCÊ QUER EXECUTAR.

Js teste.js

VARIÁVEIS E TIPOS DE DADOS

VARIÁVEIS SÃO ESPAÇOS NA MEMÓRIA PARA GUARDAR DADOS. EXEMPLOS:



= VALOR QUE PODE MUDAR.



= VALOR CONSTANTE (NÃO PODE SER ALTERADO DURANTE A EXECUÇÃO DO CODIGO).



= FORMA ANTIGA (EVITAR)

IMAGINE AS VARIÁVEIS COMO CAIXAS PARA GUARDAR DADOS!

JÁ OS TIPOS DE DADOS NÃO O QUE A VARIÁVEL GUARDA E SIM COMO SE DEFINE O QUE ELA ESTA GUARDANDO. EXEMPLO:

_let exemplo1 = "1

DECLARANDO QUE UM DADO SERÁ GUARDADO E QUE ELE PODE SER ALTERADO. DECLARANDO O
NOME DA VARIÁVEL
PARA QUE
POSSAMOS USA-LA
DEPOIS. (ESSE NOME
VOCÊ QUE
ESCOLHE)

let exemplo2 = 1

O TIPO DO DADO QUE ESTA SENDO GUARDADO. NESSE CASO UMA STRING OU SEJA UM TEXTO.

O TIPO DO DADO QUE ESTA SENDO GUARDADO. NESSE CASO UM INT OU SEJA UM VALOR INTEIRO.

E QUAL A DIFERENÇA?

EXEMPLO DE DIFERENÇA

EXERCICIO:

```
Crie um arquivo chamado exec1.js copie o seguinte código nele:
```

```
let exemplo1 = "1"
    let exemplo2 = 1
console.log(exemplo1+exemplo1)
Agora execute o código!
```

O NOME DO MOTOR (espaço) O NOME DO ARQUIVO QUE VOCÊ QUER EXECUTAR. O QUE ACONTECEU?

AGORA EXECUTE DESSA OUTRA FORMA:

console.log(exemplo2+exemplo2)

O QUE ACONTECEU?

VOCÊ NOTOU QUE DE ACORDO COM O TIPO DO DADO QUE A VARAIVEL GUARDA O RESULTADO PODE SER ALTERADO.

AGORA VAMOS FALAR DOS:

TIPOS PRINCIPAIS

TIPOS DE DADOS

STRING

= TEXTO - EXEMPLO: let exemplo1 = "1"

NUMBER

= VALOR INTERO OU DECIMAL - EXEMPLO: let exemplo2 = 1

BOOLEAN

= VERDADEIRO OU FALSO — EXEMPLO - let exemplo3 = true let exemplo4 = false

NULL

= NULO, VAZIO PROPOSITALMENTE - EXEMPLO - let exemplo5 = null

UNDEFINED

= NÃO DEFINIDO - EXEMPLO - let exemplo6 = undefined

OBJECT

ARRAY

FUNCTION

- VEREMOS POSTERIOMENTE

TEMPLATE STRINGS

JÁ ENTENDEMOS, QUE OS DADOS QUE ESTÃO ENTRE ASPAS SÃO STRINGS.

EXEMPLO: 'ISTO É UMA FRASE E NAO REPRESENTA UM COMANDO. '

QUER DIZER QUE NÃO PODEMOS CHAMAR COMANDOS OU VARIÁVEIS DENTRO DESSA STRING OU FRASE. VAMOS IMAGINAR QUE TEMOS A SEGUINTE SITUAÇÃO.

TEMOS ESSA A SEGUINTE VARIÁVEL:

let notaDoAluno1 = 7.4

IMAGINA QUE QUEREMOS UTILIZAR O VALOR DESSA VARIÁVEL DENTRO DE UMA FRASE. MAS ANTES VOCÊ CONSEGUE RESPONDER ESSAS DUAS QUESTÕES.

QUAL O VALOR QUE ESSA VARIÁVEL notaDoAluno1 ESTA GUARDANDO?
QUAL O TIPO DELA NO MOMENTO ATUAL?

TEMPLATESTRING

ISSO MESMO O VALOR QUE ELA GUARDA É 7.4 E O TIPO DELA É NUMBER.

AGORA CONTINUEMOS A SEGUINTE SITUAÇÃO.

QUEREMOS ESCREVER O VALOR DESSA VARIÁVEL DIRETO NA FRASE.

let notaDoAluno1 = 7.4
console.log("a nota do aluno foi notaDoAluno1")

EXECUTE O CÓDIGO VERIFIQUE O QUE ACONTECEU

TEMPLATESTRING

NOTE QUE FALHAMOS MISERAVELMENTE.

TENTE ASSIM AGORA:
TROQUE AS ASPAS ""
POR GRAVE ``

console.log(`a nota do aluno foi notaDoAluno1`)

EXECUTE O CÓDIGO E VEJA MAGICA

AGORA PRECISAMOS ENTENDER COMO FUNCIONA A MATEMÁTICA NO CÓDIGO.

AGORA VAMOS FALAR DE:

OPERADORES É EXPRESSÕES

OPERADORES SÃO SÍMBOLOS QUE REALIZAM OPERAÇÕES COM VARIÁVEIS E VALORES.

◆ PRINCIPAIS TIPOS DE OPERADORES EM JAVASCRIPT:

1. Operadores Aritméticos		
Operador	Significado	
•	Adição	
	Subtração	
•	Multiplicação	
	Divisão	
x	Resto da divisão	
**	Exponenciação	

EXEMPLO 1: QUESTÃO SIMPLES (VISUAL)

```
console.log(5 + 3); // 8
console.log(10 % 3); // 1
```

EXEMPLO 2: USO PRÁTICO COMUM

```
let preco = 20;
let quantidade = 3;
let total = preco * quantidade;
console.log(`Total a pagar: R$, ${total}`);
```



CRIE DUAS VARIÁVEIS: NOTA1 E NOTA2.

CALCULE A MÉDIA DAS NOTAS E MOSTRE NO CONSOLE:

"A MÉDIA DO ALUNO FOI: X"

OPERADORES DE ATRIBUIÇÃO

OPERADORES DE ATRIBUIÇÃO EM JAVASCRIPT:

Operador	Exemplo	Equivalente
	x = 5	atribui 5 a x
+=	x += 3	x = x + 3
=	x -= 2	x = x - 2
*=	x *= 4	x = x * 4
		x = x * 4

EXEMPLO 1: QUESTÃO SIMPLES (VISUAL)

```
let a = 10;
a += 5;
console.log(a);
```

EXEMPLO 2: USO PRÁTICO COMUM

```
let saldo = 100;
saldo -= 20;
console.log("Saldo atual:", saldo);
```



CRIE UMA VARIÁVEL ECONOMIAS COM VALOR 0. DEPOIS, ADICIONE 50 DUAS VEZES E SUBTRAIA 30. MOSTRE O VALOR FINAL.

OPERADORES DE COMPARAÇÃO

• OPERADORES DE COMPARAÇÃO EM JAVASCRIPT:

Operador	Significado
	lgual (com conversão)
	Igual (valor e tipo)
!=	Diferente (com conversão)
!==	Diferente (valor e tipo)
>	Maior que
<	Menor que
>=	Maior ou igual
<=	Menor ou igual
	D. H. St.

EXEMPLO 1: QUESTÃO SIMPLES (VISUAL)

```
console.log(5 == "5");
console.log(5 === "5");
```

EXEMPLO 2: USO PRÁTICO COMUM

```
let idade = 18;
console.log(idade >= 18);
```

EXEMPLO 3: USO PRÁTICO EM QUESTÕES

```
let senhaCorreta = "1234";
let senhaDigitada = "1234";
console.log(senhaCorreta === senhaDigitada);
```



CRIE UMA VARIÁVEL TEMPERATURA COM UM VALOR.
DEPOIS VERIFIQUE SE A TEMPERATURA ESTÁ ACIMA DE 30 GRAUS.
MOSTRE ESTÁ QUENTE? TRUE/FALSE.

OPERADORES DE COMPARAÇÃO

OPERADORES LÓGICOS:

Operador	Significado	Exemplo
&&	E (AND)	true && false → false
III	OU (OR)	true false → true
1	NÃO (NOT)	!true → false

EXEMPLO 1: QUESTÃO SIMPLES (VISUAL)

```
console.log(true && false);
console.log(true | false);
```

EXEMPLO 2: USO PRÁTICO COMUM

```
let idade = 20;
let temCarteira = true;
let podeDirigir = idade >= 18 && temCarteira;
console.log("Pode dirigir?", podeDirigir);
```

EXEMPLO 3: USO PRÁTICO EM QUESTÕES

```
let logado = true;
let admin = false;
console.log("Acesso permitido?", logado && admin);
```



CRIE DUAS VARIÁVEIS: TEMINGRESSO E MAIORDEHDADE. EXIBA SE A PESSOA PODE ENTRAR NO SHOW USANDO O OPERADOR &&.

AGORA VAMOS FALAR DA PARTE DE

CONTROLE DE FLUXO

VAMOS FOCAR EM ESTRUTURAS CONDICIONAIS (IF, ELSE, ELSE IF, SWITCH))

ESTRUTURAS CONDICIONAIS

◆ ESSA ESTRUTURA PERMITE EXECUTAR BLOCOS DIFERENTES DE CÓDIGO DEPENDENDO DE UMA CONDIÇÃO VERDADEIRA OU FALSA.

✓ SINTAXE BÁSICA:

```
if (condição) {
   // executa se condição for verdadeira
} else if (outraCondição) {
   // executa se a outra condição for verdadeira
} else {
   // executa se nenhuma das condições for verdadeira
}
```

EXEMPLO 1: QUESTÃO SIMPLES (VISUAL)

```
let idade = 17;
if (idade >= 18) {
console.log("Maior de idade");
} else {
  console.log("Menor de idade");
}
```

EXEMPLO 2: USO PRÁTICO

```
let nota = 7;
if (nota >= 7) {
  console.log("Aprovado");
else if (nota >= 5) {
  console.log("Recuperação");
else {
 console.log("Reprovado");
```

EXEMPLO 3: COMBINAÇÃO - EXERCÍCIO

```
let tempo = 60; // em minutos
if (tempo < 30) {</pre>
  console.log("Treino leve");
else if (tempo <= 60) {</pre>
console.log("Treino moderado");
else {
  console.log("Treino intenso");
```



RADAR DO MULTADOR DE RANDAMDAM

- CRIE UMA VARIÁVEL VELOCIDADE.
- SE FOR MENOR QUE 60 → "DENTRO DO LIMITE".
- DE 60 A 80 → "ATENÇÃO"ACIMA.
- DE 80 \rightarrow "MULTADO".

FORMA COMPACTA DO IF/ELSE, MUITO USADA EM INTERFACES COM REACT/REACT NATIVE PARA RENDERIZAR CONTEÚDO COM BASE EM CONDIÇÕES.

✓ IF TERNÁRIO (OPERADOR CONDICIONAL)

condição ? valor_se_verdadeiro : valor_se_falso

EXEMPLO 1: QUESTÃO SIMPLES (VISUAL)

```
let idade = 20;
console.log(idade >= 18 ? "Maior de idade" :
"Menor de idade");
```

EXEMPLO 2: QUESTÃO PRÁTICA (VISUAL)

```
const usuarioLogado = true;
const mensagem = usuarioLogado ? "Bem-vindo!" : "Faça login";
console.log(mensagem);
```

EXEMPLO 3: USO SIMPLES (VISUAL)

```
let nota = 8;
let resultado = nota >= 7 ? "Aprovado" : "Reprovado";
console.log("Resultado:", resultado);
```



CRIE UMA VARIÁVEL PONTUACAO. SE UM TERNÁRIO PARA IMPRIMIR: "VOCÊ GANHOU UM PRÊMIO!" SE FOR MAIOR OU IGUAL A 100"TENTE NOVAMENTE" SE FOR MENOR.

AGORA VAMOS VOLTAR A FALAR DA PARTE DE ARAMAZENAMENTO



◆ UM OBJETO EM JAVASCRIPT É UMA ESTRUTURA QUE ARMAZENA PARES DE CHAVE-VALOR, USADA PARA REPRESENTAR UM ELEMENTO COM PROPRIEDADES. MUITO USADO EM QUALQUER APLICAÇÃO, INCLUSIVE EM REACT/REACT NATIVE, PARA REPRESENTAR USUÁRIOS, PRODUTOS, ESTADOS, CONFIGURAÇÕES, ETC.

SINTAXE BÁSICA:

```
let aluno = {
  nome: "Ana",
  idade: 16,
  aprovado: true
};
console.log(aluno.nome); // "Ana"
```

AQUI TEMOS UM OBJETO ALUNO COM 3 PROPRIEDADES. PODEMOS ACESSAR OS VALORES USANDO O PONTO (.).

EXEMPLO 2: USO PRÁTICO

```
let produto = {
  nome: "Fone de ouvido",
  preco: 150,
  emEstoque: true
};

console.log("Produto:", produto.nome);
console.log("Preço: R$", produto.preco);

cousoje:jo8(_pusco: ya,' buogngo:busco);
```

OBJETOS PERMITEM ORGANIZAR DADOS DE FORMA CLARA E AGRUPADA, MUITO COMUM EM APHS, CADASTROS E CONFIGURAÇÕES.

✓ EXEMPLO 3: COMBINAÇÃO — EXERCÍCIO

```
let livro = {
  titulo: "JavaScript Essencial",
  paginas: 240,
  autor: "João Silva"
};
console.log(`${livro.titulo} tem ${livro.paginas} páginas.`);
```



✓ EXEMPLO 3: COMBINAÇÃO — EXERCÍCIO

MODIFICAR OU ADICIONAR PROPRIEDADES:

VOCÊ PODE ALTERAR OU ADICIONAR PROPRIEDADES A UM OBJETO ASSIM:

```
livro.paginas = 300;
livro.genero = "Programação";
console.log(livro);
```

console.log(livro);



CRIE UM OBJETO CHAMADO CARRO COM AS PROPRIEDADES:
MODELO (STRING)
ANO (NÚMERO)
AUTOMATICO (BOOLEANO)
HMPRIMA UMA FRASE COMO:
"O CARRO MODELO CIVIC DO ANO 2020 É AUTOMÁTICO? TRUE"

- DICA DE USO EM REACT / REACT NATIVE
- OBJETOS SÃO FREQUENTEMENTE USADOS PARA REPRESENTAR O ESTADO DE UM COMPONENTE COM USESTATE:

```
const usuario = {
  nome: "Carlos",
  idade: 20
};

// JSX (React/React Native):
// <Text>{usuario.nome}</Text>
```

CASO VOCÊ QUEIRA SABER O TIPO DE VARIÁVEL QUE VOCÊ CRIOU EXISTE UM COMANDO PARA ISSO -



IMPRIMA O COMANDO CONSOLE.LOG TYPEOF ©©©©)

7 – PODEMOS ESCREVER COMENTÁRIOS, QUE SÃO PALAVRAS QUE NÃO SERÃO INTERPRETADAS COMO CÓDIGOS OU COMANDOS, SERVE PRINCIPALMENTE PARA INFORMAR ALGO.

```
// Criar um programa que calcula a média
// das notas entre os alunos e envia
// mensagem do cálculo da média.
```



```
const aluno01 = "Mayk"
const aluno02 = 'Diego'
const notaAluno01 = 9.8
```

9 - CRIANDO UMA CALCULADORA DE MEDIA DOS ALUNOS.

CRIE TRÊS VARIÁVEIS DO TIPO NUMBER PARA GUARDAR AS NOTAS.

CRIE UMA VARIÁVEL PARA GUARDAR O CÁLCULO DE MEDIA.

MOSTRE TUDO ISSO NO CONSOLE!

CRIANDO UMA CALCULADORA DE MÉDIA DOS ALUNOS.

```
const notaAluno01 = 9.8
const notaAluno02 = 10
const notaAluno03 = 2
```

CRIE TRÊS VARIÁVEIS DO TIPO NUMBER PARA GUARDAR AS NOTAS.

const media = (notaAluno01 + notaAluno02 + notaAluno03) / 3

CRIE UMA VARIÁVEL PARA GUARDAR O CÁLCULO DE MEDIA.

console.log(media)

MOSTRE TUDO ISSO NO CONSOLE!

node index.js

AGORA VAMOS PRO UPGRADE

UTILIZE UMA ESTRUTURA CONDICIONAL PARA PARABENIZAR A TURMA COM UMA MENSAGEM NO CONSOLE SE A MÉDIA FOR MAIOR QUE 5 E SE NÃO FOR JÁ SABE.

DICA

```
if () {
    // faz alguma coisa
} else {
    // faz outra coisa
}
```

```
IF (VARIAVEL > 5){
   CONSOLE.LOG ( ----- ) }
   else{
   Console.log ( ----- ) }
```

DESAFIO

UTILIZE UMA TEMPLATE STRING PARA MOSTRAR A VARIÁVEL MEDIA DENTRO DA MENSAGEM.

AGORA VAMOS PRO UPGRADE

UTILIZE UMA ESTRUTURA CONDICIONAL PARA PARABENIZAR A TURMA COM UMA MENSAGEM NO CONSOLE SE A MÉDIA FOR MAIOR QUE 5 E SE NÃO FOR JÁ SABE. DESAFIO

UTILIZE UMA TEMPLATE STRING PARA MOSTRAR A VARIÁVEL MEDIA DENTRO DA MENSAGEM.

```
if (media > 5) {
   console.log(`A nota foi de ${media}. Parabéns`)
} else {
    console.log('A média é menor que 5')
}
```

AGORA VAMOS PRO UPGRADE

DESAFIOS !

```
console.log( 5 > 4 ) // true
console.log( 5 < 4 ) // false
console.log( 5 >= 4 ) // true
console.log( 4 <= 4 ) // true</pre>
```

OPERADORES DE COMPARAÇÃO

```
> Maior
< Menor
>= Maior igual a
<= Menor igual a
== Igual a
=== Igual e do mesmo tipo
!= Diferente de
!== Diferente, inclusive do tipo</pre>
```

```
console.log( 4 == "4" )
console.log( 4 === "4" )
console.log( 4 !== "5" )
console.log( 4 !== "5" )
```

```
// DESAFIO 1
// dar bonificação de 1.000
// se o vendedor possuir mais que 100 pontos
```

```
// DESAFIO 2
// verificar se a pessoa é maior de 18 anos
// se sim, deixar entrar, se não, bloquear a entrada
// se a pessoa tiver 17 anos
// avisar para voltar quando fizer 18 anos
```

QUE A FORÇA ESTEJA COM VOCÊ. PROF. FERNANDO LUCAS

OPERADORES LÓGICOS

OPERADORES DE LÓGICOS

&& "E" As duas condições devem ser verdadeiras para que a condição final seja verdadeira.

|| "OU" Uma das condições deve ser verdadeira para que a condição final seja verdadeira.

! "NÃO" Nega uma condição

```
console.log(5 == 5 && 6 == 6) // true
console.log(5 == 5 && 6 != 6) // false

console.log(5 != 5 || 6 == 6) // true
console.log(5 == 5 || 6 != 6) // true

console.log(!(5 > 6)) // true
console.log(!(5 < 6)) // false</pre>
```

→ MELHORE O SEU CÓDIGO

```
// DESAFIO 1
// se sim, deixar entrar, se não, bloquear a entrada
// se a pessoa tiver 17 anos
// avisar para voltar quando fizer 18 anos
```

IMAGINE, SEU CADASTRO NA ESCOLA É ENCONTRADO NO SISTEMA PELO SEU NOME.

UMA ALUNA CHAMADA LAIS
NO SISTEMA DA ESCOLA TEM VÁRIAS INFORMAÇÕES COMO:
ENDEREÇO
TELEFONE
NOTA
SALA

AGORA IMAGINE SE O SISTEMA TIVESSE QUE CRIAR UMA VARIÁVEL
PARA CADA INFORMAÇÃO DA LAIS.
VAMOS CONVERTER ESSA SITUAÇÃO PRO JAVASCRIPT.

TURNO

ENDEREÇO TELEFONE NOTA SALA TURNO

AGORA IMAGINE SE O SISTEMA TIVESSE QUE CRIAR UMA VARIÁVEL
PARA CADA INFORMAÇÃO DA LAIS.
VAMOS CONVERTER ESSA SITUAÇÃO PRO JAVASCRIPT.

```
const nome = lais;
const enderecoDaLais = "rua kennedy n4"
const notaFinal = 9.8
const salaDaLais = 5
const turnoDaLais = "manha"
```

```
const nome = lais;
const enderecoDaLais = "rua kennedy n4"
const notaFinal = 9.8
const salaDaLais = 5
const turnoDaLais = "manha"
```

BOM, PERCEBEMOS QUE LOTAMOS NOSSO PROGRAMA DE VARIÁVEIS E ISSO CERTAMENTE NÃO É BOM. MAS SOLUCIONAMOS NOSSO PROBLEMA É CLARO.

E SE TIVESSE MAIS ALUNOS NA ESCOLA?

JÁ IMAGINOU QUANTAS VARIÁVEIS TERÍAMOS QUE CRIAR ?

```
const nome = lais:
                                          const nome = lais:
const enderecoDaLais const nome = lais;
                                                            )aLais = "rua kennedy n4"
cor const nome = lais const enderecoDaLais = "rua kennedy n4"
                                                            = 9.8
                                                                                      "rua kennedy n4"
cor const nome = lai const notaFinal = 9.8
                                                           lais;
cor const enderecoDa const salaDaLais = 5
                                                          coDaLais = "rua kennedy n4"
const notaFinal const turnoDaLais = "manha"
                                                            nal = 9.8
  c const salaDaLais = >
                                                 const salaDaLais = 5
    const turnoDaLais = "manha"
                                                 const turnoDaLais = "manha"
```

CERTAMENTE ISSO NÃO SERIA GOOD PRA GENTE.

PRA ISSO NOS USAMOS UM MÉTODO CHAMADO DE OBJETO.

E SE PUDÉSSEMOS GUARDAR TODOS OS DADOS DA LAIS DENTRO DE UMA VARIÁVEL SÓ? E MELHOR E SE PUDÉSSEMOS CHAMAR CADA INFORMAÇÃO SOMENTE CHAMANDO A VARIÁVEL LAIS.

ANTES NÓS PENSÁVAMOS EM VARIÁVEIS COMO UMA PEQUENA CAIXA. AGORA VAMOS PENSAR EM OBJETOS COMO UMA PEQUENA CÔMODA CHEIA DE GAVETAS.

ESTRUTURA DO OBJETO EM JAVASCRIPT:

```
const lais = {
    endereco: "rua kennedy n4",
    notaFinal: 9.8,
    sala: 5,
    turno: "manha"
}
console.log (lais.notaFinal)
```

AGORA VAI FICAR GOOD, POSSO CHAMAR SOMENTE A INFORMAÇÃO QUE EU QUERO. CHAME NO CONSOLE SOMENTE A INFORMAÇÃO QUE VOCÊ QUER IMPRIMIR.

```
const lais = {
endereco: "rua kennedy n4",
notaFinal: 9.8,
sala: 5,
turno: "manha"
}
console.log (lais.endereco)
```

```
const lais = {
    endereco: "rua kennedy n4",
    notaFinal: 9.8,
    sala: 5,
    turno: "manha"
}
console.log (lais.notaFinal)
```

TUDO QUE ESTÁ DENTRO DO OBJETO CHAMAMOS DE PROPRIEDADES.

USAMOS '{}' PARA INICIAR O OBJETO
USAMOS A ',' PARA SEPARAR AS PROPRIEDADES
USAMOS O '.' PARA ACESSAR AS PROPRIEDADES

NA SIMPLICIDADE, AGORA CRIE VÁRIOS OBJETOS ENVOLVENDO AS MESMAS PROPRIEDADES PARA TODOS.

```
const lais = {
endereco: "rua kennedy n4",
notaFinal: 9.8,
sala: 5,
turno: "manha"
}
```

UMA CURIOSIDADE, ESTAMOS USANDO O COMANDO CONSOLE.LOG ()

E ESSE DANADO É LITERALMENTE UM OBJETO.



OBJETO JAVASCRIPT

. PARA CHAMAR UMA PROPRIEDADE.

A PROPRIEDADE ESCOLHIDA.

AINDA NO SENTIDO DE DIMINUIR NOSSO NÚMERO DE VARIÁVEIS TEMOS OS

ARRAYS

(VETORES)

QUE SÃO UMA FORMA DE ARMAZENAR TODOS OS OBJETOS EM UMA VARIÁVEL.

DECLARANDO DESSA FORMA....

VAMOS CRIAR UM PROGRAMA QUE GUARDA AS INFORMAÇÕES DOS ALUNOS DE UMA DETERMINADA ESCOLA.

PRIMEIRO CRIAMOS O ARRAY E DAMOS UM NOME A ELE,
SÃO OS [] QUE VÃO FAZER COM QUE NOSSA VARIÁVEL SE
TORNE UM ARRAY.

const escola = [[{},{},{},{}]] ——— DENTRO DE CADA {} HAVERÁ UM OBJETO COM SUAS PROPRIEDADES.

AGORA TEMOS QUE ENTENDER QUE CADA OBJETO GANHOU UMA POSIÇÃO NO NOSSO ARRAY. COMEÇANDO A CONTAR DO NUMERO 0.

```
const escola = [{},{},{},{}]
```

```
const escola = [
    nome:"lais",
    endereco: "rua kennedy n4",
    notaFinal: 9.8,
    sala: 5,
    turno: "manha"}.
    nome:"pedro teles",
    endereco: "rua amanha tem copa",
    notaFinal: 9.8,
    sala: 5,
    turno: "manha"},
    nome: "savio",
    endereco: "rua keny west ",
    notaFinal: 9.8,
    sala: 5,
    turno: "manha"},
```

AGORA COM O NOSSO COMANDO GOD DOS TESTES O CONSOLE.LOG ().

VAMOS A PARTIR DA POSIÇÃO DO ARRAY IMPRIMIR AS INFORMAÇÕES DO PEDRO.

```
const escola = [
    nome:"lais",
    endereco: "rua kennedy n4",
    notaFinal: 9.8,
    sala: 5,
    turno: "manha"}.
    nome: "pedro teles",
    endereco: "rua amanha tem copa",
    notaFinal: 9.8,
    sala: 5.
    turno: "manha"},
    nome: "savio",
    endereco: "rua keny west ",
    notaFinal: 9.8,
    sala: 5,
    turno: "manha"},
```

console.log(escola[1].endereco,escola[1].nome)

E ASSIM CONSEGUIMOS CHAMAR AS INFORMAÇÕES QUE QUEREMOS.

AGORA É SUA VEZ !!!!!

CRIE 2 ARRAYS CHAMADOS DE ALUNOSDATURMAA ALUNOSDATURMAB

2 – CADA POSIÇÃO UM OBJETO COM 3 PROPRIEDADES.
NOME
NOTA
ENDEREÇO

3 – IMPRIMA NA TELA TUDO QUE ESTIVER NA TERCEIRA POSIÇÃO DO ARRAY.

PENSANDO EM REAPROVEITAR NOSSOS CÓDIGOS

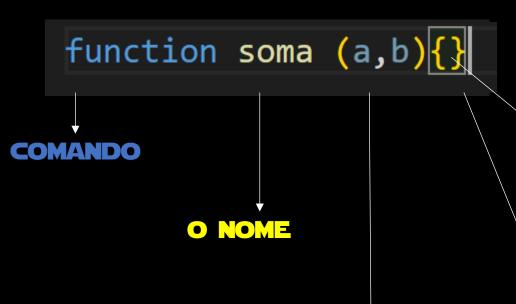


ELA PERMITE QUE POSSAMOS REAPROVEITAR UM BLOCO DE CÓDIGO VÁRIAS VEZES, SEM PRECISAR ESCREVE-LO NOVAMENTE.

ELA POSSUI DOIS MOMENTOS

1 - DECLARAR 2 - CHAMAR

DECLARANDO



OS PARÂMETROS
QUE SERÃO RECEBIDOS
DENTRO DA FUNÇÃO
E EXECUTADOS.

OS COMANDOS QUE

ELA DEVE EXECUTAR.

return

CHAMANDO

soma (5,6)

NOME

OS PARÂMETROS QUE SERÃO ENVIADOS PARA DENTRO DA FUNÇÃO.

EXEMPLO

```
function soma (a,b) {
return a+b }
}

console.log(soma (5,6))
```

QS PARÂMETROS DA NOSSA FUNÇÃO SÃO A E B (a,b), QUANDO NÓS CHAMAMOS NOSSA FUNÇÃO PRECISAMOS ENTREGAR OS PARÂMETROS PARA SEREM SUBSTITUÍDOS DENTRO DELA, NESTE CASO É O (5,6).

TRY AI MAN!

EXEMPLO

```
function calculaMedia(alunos) {
    return (alunos[0].nota + alunos[1].nota + alunos[2].nota) / 3
const media1 = calculaMedia(alunosDaTurmaA)
const media2 = calculaMedia(alunosDaTurmaB)
function enviaMensagem(media, turma) {
   // Se a média for maior que 5, parabenizar a turma
    if (media > 5) {
        console.log(`A media da turma ${turma} foi de ${media}. Parabéns`)
    } else {
        console.log(`A média da turma ${turma} é menor que 5`)
enviaMensagem(media1, 'turmaA')
enviaMensagem(media2, 'turmaB')
```

PENSANDO EM REPETIR COMANDOS

ESTRUTURA DE REPETIÇÃO - FOR -

ELA PERMITE QUE POSSAMOS REPETIR O MESMO CÓDIGO VÁRIAS VEZES.

NOVA CONFIGURAÇÃO DE VARIÁVEL DESBLOQUEADA



DIFERENTE DA CONFIGURAÇÃO CONST, O LET CONSEGUE ALTERAR SEU VALOR A QUALQUER MOMENTO DO PROGRAMA.

const qualquerNome = 5



let qualquerNome = 5

EXEMPLO

```
for (let i = 0; i< 9; i++){
    console.log(i)
}</pre>
```

O COMANDO

i++

É A MESMA

COISA DE i + 1.

O FOR FUNCIONA COMO UM CONTADOR, ENQUANTO A CONDIÇÃO FOR VERDADEIRA ELE VAI CONTINUAR RODANDO O CÓDIGO.

E SEMPRE QUE ELE EXECUTAR O CÓDIGO ELE VAI ALTERAR O VALOR DA VARIÁVEL DE CONTROLE, NO NOSSO CASO (1).

ENTÃO SEMPRE QUE O CÓDIGO É EXECUTADO É ADICIONADO +1 A VARIÁVEL LET (1).

TRY AI MAN!

EXEMPLO

COM ARRAYS

NO LUGAR DE ESCREVERMOS UMA QUANTIDADE ESPECIFICA PARA O CONTADOR, PODEMOS PEDIR QUE ELE CONTE AS POSIÇÕES DO ARRAY DE FORMA AUTOMÁTICA COM A PROPRIEDADE LENGTH.

```
for (let i = 0; i < alunos.length; i++) {
  console.log(i);
}</pre>
```

TRY AI MAN!

EXEMPLO

COM ARRAYS

CONSTRUA UMA FUNÇÃO COM UM "FOR" QUE PERCORRE E CALCULA A MÉDIA DE TODOS OS ALUNOS DAS TURMAS QUE VOCÊ CRIOU.

DEPOIS IMPRIMA ESSES VALORES NO CONSOLE.

DESAFIO — TENTE UTILIZAR TEMPLATES STRINGS DENTRO DAS IMPRESSÕES.

```
function calculaMedia(alunos) {
    let soma = 0;
    for (let i = 0; i < alunos.length; <math>i++) {
        soma = soma + alunos[i].nota
    const media = soma / alunos.length
    return media
const media1 = calculaMedia(alunosDaTurmaA)
const media2 = calculaMedia(alunosDaTurmaB)
function enviaMensagem(media, turma) {
    if (media > 5) {
        console.log(`A media da turma ${turma} foi de ${media}. Parabéns`)
    } else {
        console.log(`A média da turma ${turma} é menor que 5`)
```

```
enviaMensagem(media1, 'turmaA')
enviaMensagem(media2, 'turmaB')
```