INCIANDO JAVASCRIPT

CONHECENDO O JAVASCRIPT
COMENTÁRIOS, STRINGS E NUMBERS,
FAZENDO CÁLCULOS COM JAVASCRIPT,
OPERADORES RELACIONAIS E COMPARATIVOS
CONDICIONAIS,
OBJETOS E
ARAYS.





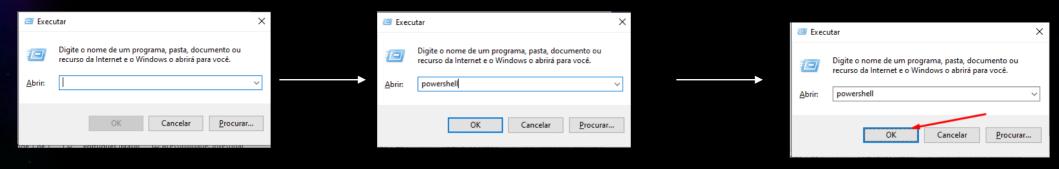
PowerShell

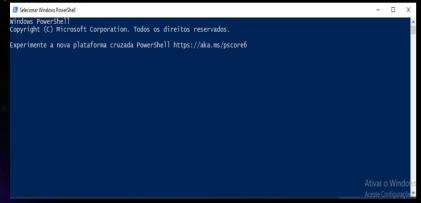


QUE A FORÇA ESTEJA COM VOCÊ. PROF. FERNANDO LUCAS



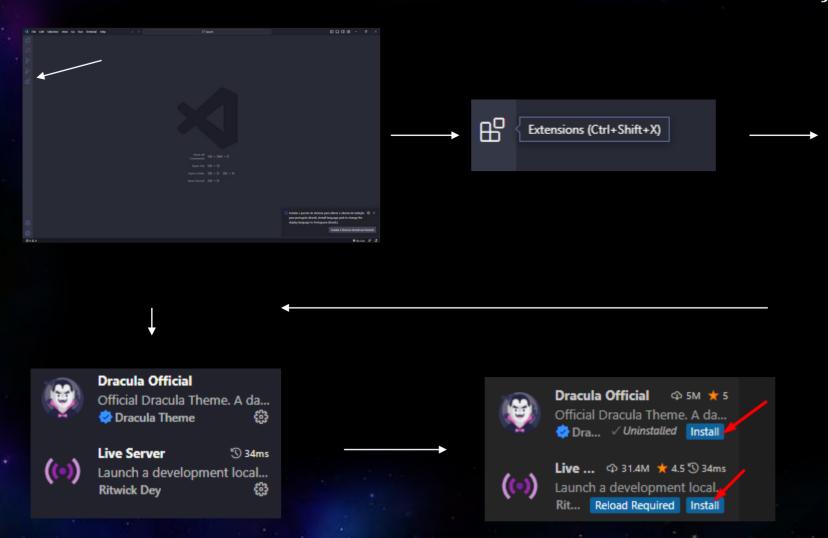
1 – VAMOS UTILIZAR O POWERSHELL PARA ABRIR O VSCODE E CRIAR UMA PASTA QUE SERÁ ONDE GUARDAREMOS NOSSO PROJETO.

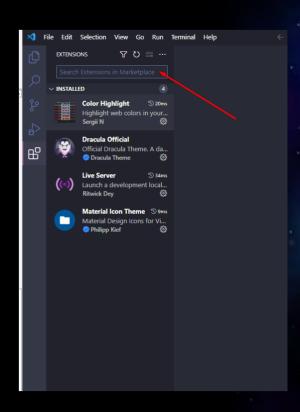




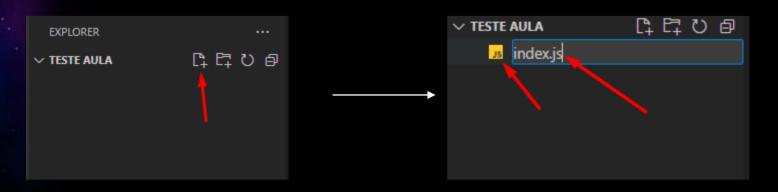
COMANDOS — DIR — LISTA OS ARQUIVOS CD 'NOME DA PASTA' — NAVEGA NAS PASTAS MKDIR — CRIA UMA PASTA NOVA CODE . — ABRE O VSCODE NA PASTA QUE VOCÊ ESTA.

2 - COM VSCODE ABERTO VAMOS ADICIONAR NOSSAS EXTENÇÕES.





3 - CRIANDO ARQUIVO JAVASCRIPT NO VSCODE.



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

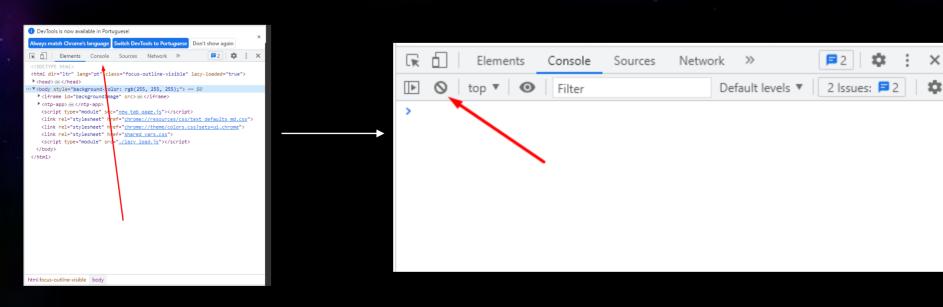
INDEX C22

INDEX_JS

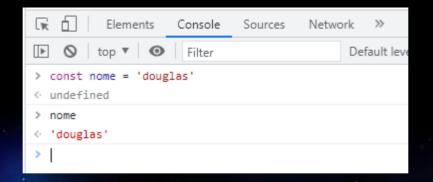
INDEX PKG

ENTRE OUTRAS 1000 LINGUAGENS QUE EXISTEM.

4 - ACESSE O CONSOLE DO CHROME APERTANDO F12



CRIE UMA CONSTANTE E EXECUTE ELA EM JAVASCRIPT NO MOTOR DO CHROME.



5 - CRIANDO UMA CONSTANTE E MOSTRANDO ELA NO CONSOLE.



CONSOLE LOG(NOME)

CTRL J
PARA ABRIR O CONSOLE

PS D:\teste aula> node index.js juliana PS D:\teste aula>

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

D powershell + ∨ ∏ ⋒ ··· ∧ ×

6 – 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 nome = "Mayk"
const nome2 = 'Diego'
```

7 - CRIANDO E ENTENDENDO A DIFERENÇA ENTRE UMA VARIÁVEL DO TIPO 'STRING' E 'TEMPLATE STRING'.

const nome2 = 'Diego \${nome}'

- DESSA FORMA TUDO DENTRO DA VARIÁVEL SERÁ RECONHECIDO COMO UM TEXTO ' ------ '.



const nome3 = `Valeska e \${nome}`

- E UTILIZANDO \${nome} PODEMOS CHAMAR OUTRA VARIÁVEL DENTRO DELA.

CRIE UMA DE CADA VARIÁVEL E IMPRIMA ELAS COM O COMANDO CONSOLE.LOG()

8 - AGORA VAMOS CRIAR UMA VARIÁVEL DO TIPO NUMBER.

const notaAluno01 = 9.8

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



IMPRIMA O COMANDO CONSOLE.LOGI TYPEOF COCO

9 - CRIANDO UMA CALCULADORA DE MEDIA DOS ALUNOS.

CRIE TRÊS VARIÁVEIS DO TIPO NUMBER PARA GUARDAR AS NOTAS.

CRIE UMA VARIÁVEL PARA GUARDAR O CALCULO DE MEDIA.

MOSTRE TUDO ISSO NO CONSOLE!

CRIANDO UMA CALCULADORA DE MEDIA 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 CALCULO 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 MEDIA FOR MAIOR QUE 5 E SE NÃO FOR JÁ SABE.

DICA

```
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 MEDIA 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

NA CASINHA!

```
OPERADORES DE COMPARAÇÃO

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

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

```
// DESAFIO 2
// dar bonificação de 1.000
// se o vendedor possuir mais que 100 pontos
```

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

MELHORE O SEU CÓDIGO

```
// DESAFIO 1
// se sim, deixar entrar, se não, bloquear a entrada
// se a pessoa tiver 17 anos
// puisar para voltar guanda fizar 19 anos
```

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

UMA ALUNA CHAMADA LAIS
NO SISTEMA DA ESCOLA TEM VARIAS INFORMAÇÕES COMO:
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.

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"
cccconst notaFinal const turnoDaLais = "manha"
                                                        nal = 9.8
 c const salaDaLais = 5
                                                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 NOS 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

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 NUMERO DE VARIÁVEIS TEMOS OS



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 O.

```
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",
                                                   console.log(escola[1].endereco,escola[1].nome)
    notaFinal: 9.8.
    sala: 5.
   turno: "manha"}.
   nome: "pedro teles",
    endereco: "rua amanha tem copa".
    notaFinal: 9.8.
    sala: 5,
                                                        E ASSIM CONSEGUIMOS CHAMAR AS
    turno: "manha"},
                                                         INFORMAÇÕES QUE QUEREMOS.
    nome: "savio",
    endereco: "rua keny west ",
    notaFinal: 9.8,
    sala: 5,
    turno: "manha"},
```

AGORA É SUA VEZ !!!!!

CRIE UM ARRAY COM AS SEGUINTES DIRETRIZES:

- 1 5 POSIÇÕES.
- 2 CADA POSIÇÃO UM OBJETO COM 3 PROPRIEDADES.
- 3 IMPRIMA NA TELA TUDO QUE ESTIVER NA TERCEIRA POSIÇÃO DO ARRAY.

PENSANDO EM REAPROVEITAR NOSSOS CÓDIGOS

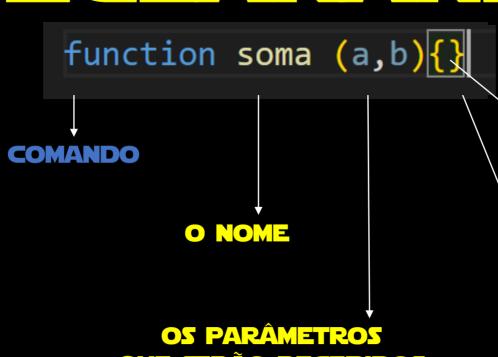


ELA PERMITE QUE POSSAMOS REAPROVEITAR UM BLOCO DE CÓDIGO VARIAS VEZES, SEM PRECISAR ESCREVE-LO NOVAMENTE.

ELA POSSUI DOIS MOMENTOS

1 - DECLARAR 2 - CHAMAR

DECLARANDO



OS PARÂMETROS
QUE SERÃO RECEBIDOS
DENTRO DA FUNÇÃO
E EXECUTADOS.

return

O RETORNO
DA FUNÇÃO.

OS COMANDOS QUE ELA DEVE EXECUTAR.

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 NOS CHAMAMOS NOSSA FUNÇÃO PRECISAMOS ENTREGAR OS PARÂMETROS PARA SEREM SUBSTITUÍDOS DENTRO DELA, NESTE CASO É O (5,6).

TRY AI MAN!

ESTRUTURA DE REPETIÇÃO O/

PENSANDO EM REPETIR COMANDOS

ESTRUTURA DE REPETIÇÃO - FOR -

ELA PERMITE QUE POSSAMOS REPETIR O MESMO CÓDIGO VARIAS VEZES.

ESTRUTURA DE REPETIÇÃO O/

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

ESTRUTURA DE REPETIÇÃO O/

EXEMPLO

O COMANDO

O FOR FUNCIONA COMO UM CONTADOR, ENQUANTO A CONDIÇÃO NÃO 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!