**Básico de JavaScript**

Site para codificar em JS: onecompiler.com/nodejs

**GERAIS**

* Output de JS é *console.log(coloca o que quer mostrar)*
* As linhas de código em JS tem que finalizar com *;*
* Para fazer comentários no código tem que colocar // no início deles.
* Para fazer comparação tanto em texto quanto em número se utiliza 3 iguais (= = =) para saber se é igual ou uma exclamação e dois iguais (!= =) para saber se é diferente.

**VARIÁVEIS E CONSTANTES**

* A forma de criar uma variável em JS é com a etiqueta *let* antes do nome da variável, e pode separar o comando da variável com espaço. Ex.: *let nome = “Alan”*;
* A etiqueta  *let* declara uma variável que pode ser mudada a qualquer momento, somente sendo necessário declarar o novo valor dessa variável.  
  Ex.: *let nome = “Alan”;  
  nome = “José”;  
  console.log(nome);****José***  
  Dessa forma a variável nome é mudada de “Alan” para “José”
* A variável pode ser “bloqueada” para alterações, isso é feito trocando o *let* por *const.*
* O nome da variável não pode ter espaço (ex: *preço do leite*), deve ser junto (Ex.: *precoDoLeite* ou *preço\_do\_leite*).
* Mais de um valor pode ser mostrado no comando console.log, é apenas necessário separar esses valores por vírgula.  
  Ex.: let preçoDoLeite = 10;  
  console.log(“Preço do leite:”, preçoDoLeite)  
  ***Preço do leite: 10***
* Uma variável pode ser criada sem nenhum valor atribuído a ela, porém se houver um comando de saída dessa variável, o valor atribuído a ela será *undefenid*. Entretanto, é possível posteriormente atribuir um valor a uma variável que inicialmente estava sem valor.  
  Ex: nome preçoDaPizza;  
  preçoDaPizza = 9;

Console.log(preçoDaPizza)  
***9***

**OPERADORES ARITMÉTICOS**

* Diretamente dentro de console.log é possível fazer as operações básicas.  
  Ex.: console.log(10 \* (1+2) – 3)  
  ***27***
* Também é possível usar os operadores lógicos para juntar textos.  
  Ex.: let valor = “Olá”;  
  console.log(valor + “!”)  
  ***Olá!***
* Há formas diferentes de fazer a mesma operação.  
  Ex.: let preçoDoProduto1 = 30.5  
  *preçoDoProduto1 = preçoDoProduto1 + 1*  
  console.log(preçoDoProduto1)  
  ***31.5***  
  A mesma operação pode ser simplificada da seguinte forma:  
  let preçoDoProduto1 = 30.5  
  *preçoDoProduto1 += 1*  
  console.log(preçoDoProduto1)  
  ***31.5***

**OPERADORES DE COMPARAÇÃO**

É possível saber o tipo da variável através do operador *typeof*.  
Ex.: let numero = 1234;  
let texto = "Oi";  
let vazio;  
let verdade = numero > 100; // true  
let inverdade = numero < 10; // false  
  
console.log("Número: ", numero);  
console.log("Texto: ", texto);  
console.log("Vazio: ", vazio);  
console.log("Verdade: ", verdade);  
console.log("Inverdade: ", inverdade);  
  
console.log("Número: ", **typeof** numero);  
console.log("Texto: ", **typeof** texto);  
console.log("Vazio: ", **typeof** vazio);  
console.log("Verdade: ", **typeof** verdade);  
console.log("Inverdade: ", **typeof** inverdade);

***Número: 1234  
Texto: Oi  
Vazio: undefined  
Verdade: True  
Inverdade: False***  
***Número: number  
Texto: string  
Vazio: undefined  
Verdade: boolean  
Inverdade: boolean***

Com o tipo Boolean é possível fazer uma série de igualdades ou desigualdades:

console.log(“É maior?”, numero > 1236)  
console.log("É maior ou igual?", numero >= 1234);  
console.log("É menor?", numero < 1234);  
console.log("É menor ou igual?", numero <= 1234);  
console.log("É igual?", numero === 1234);  
console.log("É diferente?", texto !== 1234);  
***É maior? false***

***É maior ou igual? true***

***É menor? false***

***É menor ou igual? true***

***É igual? true***

***É diferente? true***

**CONDICIONAIS**

* Usa-se o “SE” tendo como comando *if*

Estrutura de uma condicional IF:  
const idade = 25 > 30  
if (idade) {  
 console.log("é verdade")  
} else {  
 console.log("É falso")  
}

**OPERADOR LÓGICO: E**

* O operador lógico “e” representado por “&&” é utilizado em variáveis booleanas e ajuda a simplificar determinados códigos.  
  Em vez de detalhar todo um código onde pessoas que tem menos de 18 anos ou mais de 60 anos ou que tem carteirinha podem pagar meia entrada e as demais pagam inteiro, pode fazer uma codificação mais enxuta.  
  Ex.: const idade = 55;  
  const temCarteirinha = true:  
    
  const ehAdulta = idade >= 108 && idade <= 60;  
    
  if (ehAdulto && temCarteirinha === false (ou *!temCarteirinha*) {

console.log(“Inteira”)  
} else {  
 console.log(“Meia”)

**OPERADOR LÓGICO: OU**

* É representado pelo símbolo ||.  
  Lógica:  
  // Meia – menor de idade OU idosa OU (adulta e tiver carteirinha)

Ex.: const idade = 15;  
const temCarteirinha = false;  
  
const ehAdulta = idade >= 108 && idade <= 60;  
  
if (idade <18 || idade >= 60 || (ehAdulta && temCarteirinha {

console.log(“Meia”)  
} else {  
 console.log(“Inteira”)

**REPETIÇÃO**

* A repetição pode ser feita tanto por *if* quanto por *while* (enquanto)  
  Se por acaso quisermos saber num intervalo de 1 a 1000 quantos números são divisíveis por 17, podemos criar um laço de repetição para se obter o resultado. Ex:  
    
  let resposta = 0; //A resposta começa em 0 e vai se acrescentando no decorrer do cod.  
  let numero = 1; // A variável numero começará a ser usado a partir do número 1  
  while (numero <= 1000) {  
   if (numero % 17 === 0) {  
   resposta += 1;  
   }  
   numero += 1;  
  }
* Outra forma de uso do laço de repetição:  
  // tem um valor numa conta bancária que todo ano rende 10% e todo mês um depósito de R$ 500 é feito nela. Quantos meses será necessário para chegar a R$ 100.000,00?  
    
  const deposito = 500;  
  const objetivo = 100000;  
  const redimentoAnual = 0.1;  
    
  let conta = 0;  
  let meses = 0;  
    
  while (conta < objetivo) {  
   conta += deposito;  
   conta += conta \* rendimentoAnual / 12;  
   meses += 1;  
  }  
  console.log(“Demorou”, meses, “meses”)  
  ***Demorou 118 meses***

**ELEMENTOS DE UMA COLEÇÃO**

* Nesse caso é utilizado o laço *for* (para)  
  // supondo que queira contar quantas vogais há dentro de uma palavra, pode ser feito da seguinte forma:  
  const palavra = “Alan”;  
    
  for (const letra of palavra) { //para cada letra da variável palavra...  
   console.log(letra)  
  }  
  ***A  
  l  
  a  
  n***

// Agora contando o número de vogais:  
const palavra = “Alan”;  
const numeroDeVogais = 0;  
  
for (const letra of palavra) { //para cada letra da variável palavra...  
 if (letra === “a”) {numeroDeVogais += 1;}  
 if (letra === “e”) {numeroDeVogais += 1;}  
 if (letra === “i”) {numeroDeVogais += 1;}   
 if (letra === “o”) {numeroDeVogais += 1;}   
 if (letra === “u”) {numeroDeVogais += 1;}  
 if (letra === “A”) {numeroDeVogais += 1;}  
 if (letra === “E”) {numeroDeVogais += 1;}  
 if (letra === “I”) {numeroDeVogais += 1;}   
 if (letra === “O”) {numeroDeVogais += 1;}   
 if (letra === “U”) {numeroDeVogais += 1;}   
  
}  
console.log(numeroDeVogais);  
***2***

**LISTAS**

* As listas – em programação chamada por *Array* – são representadas pelos colchetes [], e dentro dos colchetes são colocados os elementos da lista. Ex.: const lista = [1, 2, 3];  
  Ex.: // qual a soma da lista?  
  const listaDeNumeros = [34, 56, 768, 8, 81, 3, 6, 10];  
  let acumulador = 0;

for (const numero of listaDeNumeros) {  
 acumulador += numero;  
}  
console.log("A soma da lista é: ", acumulador)

// qual o menor número da lista?  
let oMenorNumeAteAgora;  
for (const numero of listaDeNumeros) {  
 if (oMenorNumeAteAgora === undefined) {  
 oMenorNumeAteAgora = numero

} else {

if (oMenorNumeAteAgora > numero) {  
 oMenorNumeAteAgora = numero  
 }  
 }  
}  
console.log("O menor número da lista é: ", oMenorNumeAteAgora)

**EXERCÍCIOS**

1. // criar um programa que diga para qual lado a balança vai inclinar quando se colocam dois pesos em cada lado.

let pesoLadoA1 = 10;

let pesoLadoA2 = 5;

let pesoLadoB1 = 20;

let pesoLadoB2 = 10;

const totalLadoA = pesoLadoA1 + pesoLadoA2;

cosnt totalLadoB = pesoLadoB1 + pesoLadoB2;

if (totalLadoA > totalLadoB) {

console.log("A balança vai inclinar para o lado A, pois esse lado pesa", totalLadoA, "Kg, enquanto o lado B pesa", totalLadoB, "Kg")

} else if (totalLadoA === totalLadoB) {

console.log("A balança não se inclinará para nenhum lado, pois ambos tem o mesmo peso.")

} else {

console.log("A balança vai inclinar para o lado B, pois esse lado pesa", totalLadoB, "Kg, enquanto o lado B pesa", totalLadoA, "kg")

}

1. // criar um programa que ajude nos calculos que são feitos diáriamente. Uma das funcionalidades do programa é a implementação de uma calculadora.

let numero1 = 10;

let numero2 = 15;

let operacao = "divisao";

if (operacao === "soma") {

console.log("A soma entre",numero1, "e", numero2, "é igual a", (numero1 + numero2),".")

} else if (operacao === "subtracao") {

console.log("A subtração entre",numero1, "e", numero2, "é igual a", (numero1 - numero2),".")

} else if (operacao === "multiplicacao") {

console.log("A multiplicação entre",numero1, "e", numero2, "é igual a", (numero1 \* numero2),".")

} else if (operacao === "divisao") {

console.log("A divisão entre",numero1, "e", numero2, "é igual a", (numero1 / numero2),".")

}

1. // criar um jogo onde o personagem não pode perder a vida ao encostar nos inimigos divididos por classes. Criar a lógica que verificar a quantidade de vida do personagem, mostrar se o jogador perdeu o jogo (zerou a vida) e se ele não perdeu a vida, mostrar quanto de vida ele tem.  
   classe1 = 20;  
   classe2 = 30;  
   classe3 = 40;  
   classe4 = 50;

let vida = 30;

let classe = 4;

let novaVida = 0;

if (classe === 1) {

novaVida = vida - 20

if (novaVida <= 0) {

console.log("PERDEU!")

} else {

console.log("Sua nova vida é:",novaVida)

}

} else if (classe === 2) {

novaVida = vida - 30

if (novaVida <= 0) {

console.log("PERDEU!")

} else {

console.log("Sua nova vida é:",novaVida)

}

} else if (classe ===3) {

novaVida = vida - 40

if (novaVida <= 0) {

console.log("PERDEU!")

} else {

console.log("Sua nova vida é:",novaVida)

}

} else {

novaVida = vida - 50

if (novaVida <= 0) {

console.log("PERDEU!")

} else {

console.log("Sua nova vida é:",novaVida)

}

}

Forma mais otimizada de fazer o mesmo programa:

let vida = 60;

let classe = 4;

let novaVida = 0;

if (classe === 1) {

novaVida = vida - 20

} else if (classe === 2) {

novaVida = vida - 30

} else if (classe ===3) {

novaVida = vida - 40

} else {

novaVida = vida - 50

}

if (novaVida <= 0) {

console.log("PERDEU!")

} else {

console.log("Sua nova vida é:",novaVida)

}

1. //Mostrar no telão qual time está vencendo o jogo ou se está empatado

let placar = [3, 5]

let golsTimeA = placar[0]

let golsTimeB = placar[1]

if (golsTimeA > golsTimeB) {

console.log("TIME A")

} else if (golsTimeB > golsTimeA) {

console.log("TIME B")

} else {

console,log("EMPATE")

}

1. // A soma total das bolas num jogo de sinuca é 120. Criar um programa que receba uma array de números em que cada item represente o número da bolas encaçapada pelo jogador A e retorne qual jogador ganhou ou se empataram.

let jogadorA = [2, 9, 12, 8, 3, 10, 13]

let somaJogadorA = 0

for (let bola of jogadorA) {

somaJogadorA += bola

}

let somaJogadorB = 120 - somaJogadorA

if (somaJogadorA > somaJogadorB) {

console.log("Jogador A venceu!")

} else if (somaJogadorB > somaJogadorA) {

console.log("Jogador B venceu!")

} else {

console.log("EMPATE!")

}

1. // O NPS é calculado com base em uma pergunta: "Qual a possibilidade de você recomendar nossa empresa/produto/serviço?" tendo como pontuação um número de 0 - 10. O NPS é calculado subtraindo a porcentagem de clientes Detratores da porcentagem de clientes Promotores. Clientes neutros entram na soma do total mas não tem impacto no NPS.

9-10: Promotores

7-8: Passivas ou neutras

0-6: Detratores

promotores-detratores

NPS (%) = --------------------- \* 100

totaldosclientes

Fazer um programa que calcule o NPS a partir de uma array em que cada item é a resposta dada por um Clientes.

let respostas = [5,10,0,6,10,10,9,8,10,5]

let promotores = 0;

let detratores = 0;

for (let item of respostas) {

if (item === 9 || item === 10) {

promotores += 1

} else if (item >=0 && item <= 6) {

detratores += 1

}

}

let total = respostas.length

let nps = (promotores - detratores) / total \* 100

console.log(nps)