Questão única:

Acesse o site https://www.typescriptlang.org/, e utilizando a funcionalidade "Playground", como visto na aula, escreva um código TypeScript com as seguintes características:

- Crie uma função chamada "mostrar_soma" que recebe por parâmetros dois números (tipo "number").
- A função "mostrar_soma" não precisa retornar nada, mas deve mostrar utilizando o console.log a soma dos dois números passados por parâmetro.
- No mesmo código, abaixo da função "mostrar_soma", crie um código que utiliza essa função, passando os números 2 e 3 como parâmetro

Ao finalizar, copie e cole na sua resposta o seu código TypeScript e também o resultado da transpilação (código JavaScript) gerado.

Questão única:

Escreva um programa em TypeScript da seguinte forma:

- Crie uma classe chamada "ContaCorrente" com os seguintes atributos:
  - nome_correntista (um texto com o nome da pessoa que é o correntista)
  - cpf_correntista (O texto com o CPF do correntista)
  - saldo (um número que determina o saldo atual da conta)
- Adicione na classe "ContaCorrente" os seguintes métodos
  - setNome (esse métido deve receber um texto e alterar o nome do correntista para ele)
  - setCpf (deve receber um texto e alterar o cpf do correntista para ele)
  - setSaldo (deve receber um número e alterar o saldo da conta corrente para ele)
  - depositar (deve receber um número e adicionar esse valor ao saldo da conta corrente)
  - sacar (deve receber um número e deduzir esse valor do saldo da conta corrente)
  - mostrarInformacoes (deve exibir com o console.log o nome do correntista, seu cpf e saldo atual da conta corrente)
- Depois de definir a classe "ContaCorrente", no mesmo programa, crie uma nova conta corrente e utilize todos os métodos criados para setar o nome, cpf e saldo do correntista, depositar um valor, sacar um valor e ao final execute mostrarInformacoes() para exibir todos esses dados.

Aula 01 – Roteiro de Laboratório

Neste laboratório, iremos resgatar e revisar os conceitos básicos da linguagem Javascript que você já aprendeu em disciplinas anteriores. O objetivo é relembrar seus tipos de dados, declaração de variáveis, operadores e estruturas de controle.

Para desenvolver as atividades deste roteiro, utilize o editor de código de sua preferência. Para executa-lo, utilize o console das Ferramentas do Desenvolvedor do seu navegador ou qualquer outro recurso para execução de código Javascript (node.js, por exemplo).

Atividade 01)
Em um arquivo chamado *atividade01.js*, desenvolva um programa que dado um valor inteiro correspondente a idade de uma pessoa em dias (definido no código como uma variável) informe-a em anos, meses e dias. Teste o seu código com os valores 400, 800 e 30. A saída do seu programa, para a idade definida em 400 deve se similar a abaixo:

```
1 ano(s)
1 mes(es)
5 dia(s)
```

Obs.: apenas para facilitar o cálculo, considere todo ano com 365 dias e todo mês com 30 dias. Este é apenas um exercício com objetivo de testar raciocínio matemático simples e revisar os conceitos mais básicos da linguagem Javascript. Para truncar um valor de ponto flutuante para retornar um inteiro, utilize Math.trunc(x), por exemplo. Math.trunc(2.33) retornará 2.

Atividade 02)

Em um arquivo chamado *atividade02.js*, desenvolva um programa que dado 3 valores de ponto flutuante (definido no código como as variáveis *a, b* e *c*) efetue o cálculo das raízes da equação de segundo grau $a*x^2 + b*x + c$ utilizando a fórmula de Bhaskara. Se não for possível calcular as raízes, caso haja uma divisão por 0 ou raiz de número negativo, mostre a mensagem correspondente *"Impossivel calcular"*.

Teste o seu programa com os valores e verifique a saída esperada:

a	b	c	SAÍDA ESPERADA
1	-1	-2	X1 = 2 X2 = -1
10	3	5	Impossível calcular
10	20.1	5.1	x1 = -0.2978755413648882 x2 = -1.7121244586351119
0	20	5	Impossível calcular

Aula 02 – Roteiro de Laboratório

Neste laboratório iremos praticar uma importante funcionalidade das linguagens de programação: as funções, um conceito que você já viu em disciplinas anteriores, mas que agrega um princípio fundamental da programação: a modularidade, que permite que nosso código fique mais organizado, estruturado e de mais fácil manutenção.

Para desenvolver as atividades deste roteiro, utilize o editor de código de sua preferência. Para executa-lo, utilize o console das Ferramentas do Desenvolvedor do seu navegador ou qualquer outro recurso para execução de código Javascript (node.js, por exemplo).

Atividade 01)
Você já se perguntou quanto custa um “suprimento vitalício” do seu lanche favorito? Não se pergunte mais!

Escreva uma função chamada calcularSuprimento que:

Receba três argumentos: idade, quantidade de lanches por dia, preço do lanche;
Calcule a quantidade consumida para o resto da vida (com base em uma idade máxima constante de valor 85) convertida para o custo/preço necessário;
Imprima o resultado na tela dessa forma: “ Você precisará de R$XX para ter um suprimento de lanche até 85 anos.”

Atividade 02)
Chame duas vezes a função calcularArea abaixo. Uma com os argumentos necessários para que se consiga calcular corretamente a área de um quadrado de lado 5 e outra com os argumentos necessários para se calcular corretamente a área de um círculo de raio 2.

```
function calcularArea(propriedade, formula) {
    return formula(propriedade);
}
```

Aula 03 – Roteiro de Laboratório

Neste laboratório iremos explorar as principais operações envolvendo uma das estruturas de dados mais populares do JS: os *arrays*.

Para desenvolver as atividades deste roteiro, utilize o editor de código de sua preferência. Para executa-lo, utilize o console das Ferramentas do Desenvolvedor do seu navegador ou qualquer outro recurso para execução de código Javascript (node.js, por exemplo).

Atividade 01)
Escreva uma função chamada excluir que dado dois arrays como argumentos retorne o primeiro array sem os elementos contidos no segundo argumento.

A assinatura do método e alguns casos de testes (assim como as expectativas de saída) estão abaixo:

```
const excluir = (array, itensParaExcluir) => {
    // Implemente aqui a sua lógica
}

var exemplo = [1, 2, 3, 1, 2];
console.log(excluir(exemplo, [1, 2])); // [ 3 ]

exemplo = ['a', 'b', 'a', 'd', 'e', 'a'];
console.log(excluir(exemplo, ['a', 'd'])); // [ 'b', 'e' ]

exemplo = [true, false, false, true, true, false, false];
console.log(excluir(exemplo, [true])); // [ false, false, false, false ]
```

Atividade 02)
Escreva uma função que dado uma array passada como argumento retorne essa array sem nenhum valor repetido.

A assinatura do método e alguns casos de testes (assim como as expectativas de saída) estão abaixo:

```
const unico = (array) => {
    // Implemente aqui a sua lógica
}

var exemplo = [1, 2, 3, 1, 2];
console.log(unico(exemplo)); // [ 1, 2, 3 ]

exemplo = ['a', 'b', 'a', 'd', 'e', 'a'];
console.log(unico(exemplo)); // [ 'a', 'b', 'd', 'e' ]

exemplo = [true, false, false, true, true, false, false];
console.log(unico(exemplo)); // [ true, false ]
```