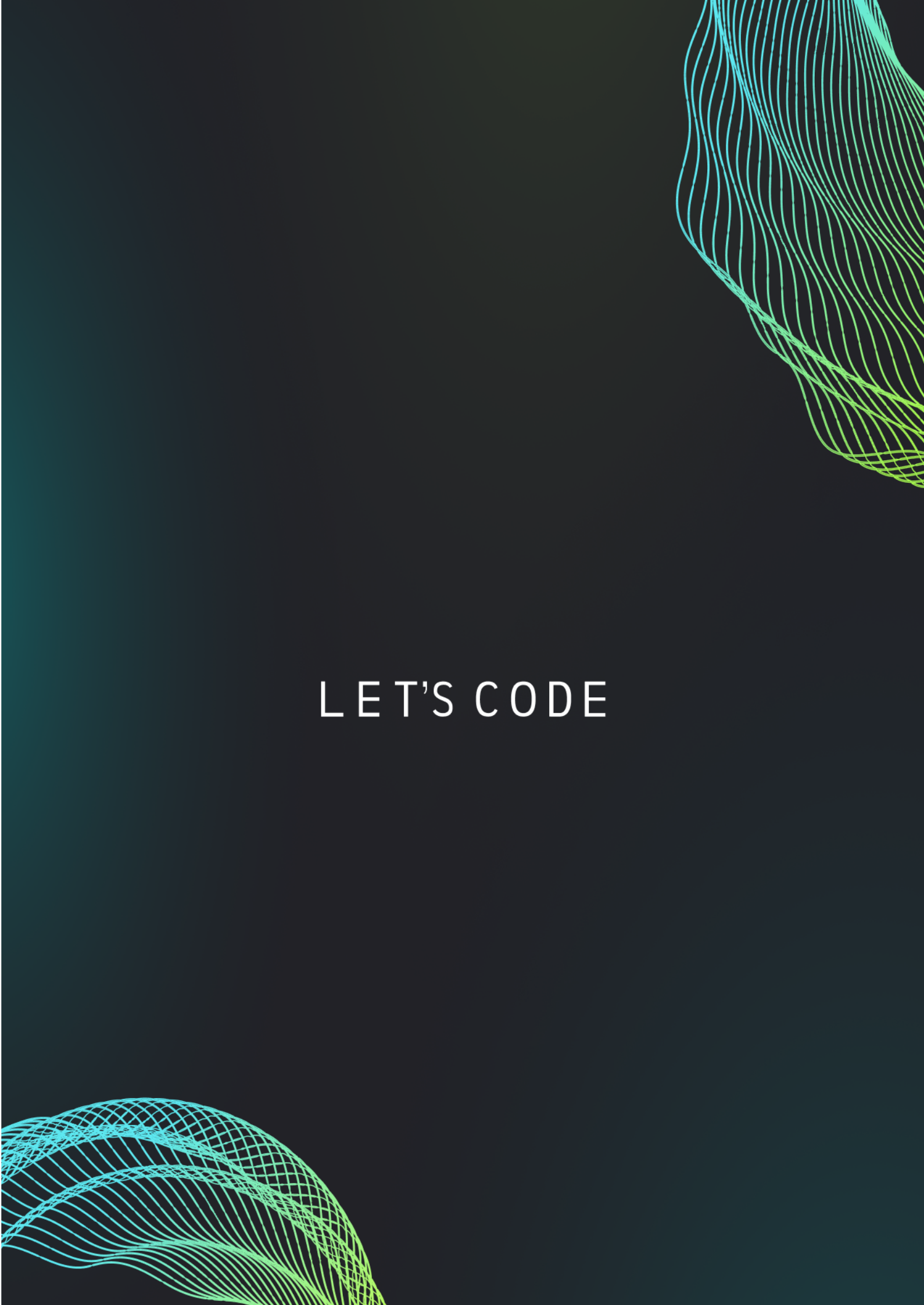
LET'S CODE

Lista de exercícios - JS

Questão 1.
Faça um programa que peça a temperatura em graus Fahrenheit (°F), transforme e mostre a temperatura em graus Celsius (°C).

$$T(^\circ C) = \frac{5 \times (T(^\circ F) - 32)}{9}$$

Questão 2.
Faça um programa que leia as coordenadas de 2 (dois) pontos em um plano cartesiano 2D: a coordenada x do primeiro ponto (x_1), a coordenada y do primeiro ponto (y_1), a coordenada x do segundo ponto (x_2) e a coordenada y do segundo ponto (y_2). Em seguida, calcule a distância euclidiana entre os pontos, utilizando a equação abaixo:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Para exemplificar, se o usuário informar os valores 1, 7, 5 e 9 para as coordenadas dos pontos, respectivamente, o resultado deve ser `4.4721`.

Questão 3.
Crie um programa que leia um valor qualquer e apresente uma mensagem dizendo em qual dos seguintes intervalos ([0,25], (25,50], (50,75], (75,100]) este valor se encontra. Caso o valor não esteja em nenhum destes intervalos, deverá ser impressa a mensagem "Fora de intervalo". Veja alguns exemplos abaixo:

💡 Entrada: 25.01 | Saída: (25,50]
Entrada: 25.00 | Saída: [0,25]
Entrada: 100.00 | Saída: (75,100]
Entrada: -25.02 | Saída: Fora de intervalo

📌 Lembrando que o `[` ou `]` representa que o valor está contido no intervalo, enquanto o `(` ou `)` representa que o valor associado não está contido no intervalo. Em outras palavras, `(75, 100]` representa o intervalo que vai de maior que 75 (não incluindo o 75) até menor ou igual 100.

Questão 4.
Crie o jogo "Pedra, Papel, Tesoura" por meio de um código em JavaScript. Para isso, solicite que o primeiro jogador informe a sua escolha e depois o mesmo para o segundo jogador. Por fim, utilize os `if's` para saber quem seria o vencedor.

Questão 5.
Faça um programa, usando loops, que peça para o usuário digitar vários números, um após outro, e que só finaliza quando o usuário digitar 0. Ao final imprima a soma de todos os números digitados.

Questão 6.
Faça um programa que imprima a tabuada do 9 (de 9 x 1 a 9 x 10) usando loops.

Questão 7.
Crie uma função que recebe o valor do raio de um círculo como parâmetro e retorna o valor da área desse círculo. Lembrando que a área de círculo é dada pela equação: $A = \pi \times r^2$.

> 💡 Dica: Para utilizar um valor mais preciso do pi (π), você pode utilizar o `Math.PI`, que é um recurso nativo do JavaScript.

Questão 8.
Faça um programa que dados dois arrays de mesmo tamanho, imprima o produto escalar entre eles.

> ❓ Produto escalar é a soma do resultado da multiplicação entre o número na posição i do array1 pelo número na posição i do array2, com i variando de 0 ao tamanho do array.
>
> Exemplo: `[3, 2, 1]` e `[5, 3, 2]` ⇒ Resultado: (3 * 5) + (2 * 3) + (1 * 2) = 15 + 6 + 2 = `23`

Questão 9.

Vamos fazer um programa para verificar quem é o assassino de um crime. Para descobrir o assassino, a polícia faz um pequeno questionário com 5 perguntas onde a resposta só pode ser sim ou não:

1. *Mora perto da vítima?*
2. *Já trabalhou com a vítima?*
3. *Telefonou para a vítima?*
4. *Esteve no local do crime?*
5. *Devia para a vítima?*

Cada resposta "sim" dá um ponto para o suspeito. A polícia considera que os suspeitos com 5 pontos são os assassinos, com 4 a 3 pontos são cúmplices e 2 pontos são apenas suspeitos, necessitando de outras investigações. Valores abaixo de 2 são liberados.
No seu programa, você deve fazer essas perguntas e, de acordo com as respostas do usuário, informar como a polícia o considera.

Questão 10.

Faça um programa que fique pedindo uma resposta do usuário, entre 1, 2 e 3. Se o usuário digitar 1, o programa deve cadastrar um novo usuário, solicitando nome, idade, e-mail e CPF, guardando esse cadastro em um objeto, e cada objeto devo ser adicionado em um array. Quando o usuário digitar 2, o programa deve imprimir os usuários cadastrados; e se o usuário digitar 3, o programa deve encerrar.

Exemplo da estrutura do array de objetos:

```
[
  {nome: 'Maria', idade: 20, email: 'maria@email.com', cpf: '987.654.321-00'},
  ...
]
```

Desafio final 🚀 - JS

Você já deve ter jogado o Jogo da Forca, certo? O que você acha de desenvolver o seu próprio Jogo da Forca em JavaScript? Esse será o seu desafio!

Para te ajudar com isso, vamos te passar algumas diretrizes para que você tenha uma noção clara de como o jogo deve funcionar e de quais etapas você deve seguir para atingir esse resultado. Vamos lá!

1. *No início do código, você pode solicitar o nome do jogador. Assim, você pode imprimir uma mensagem de boas-vindas. Talvez até imprimir uma mensagem explicando como o jogo funciona. Porém, uma sugestão é deixar tudo isso para o final, porque assim você pode focar, inicialmente, no funcionamento do jogo, em si.*
2. *Você vai precisar definir uma palavra para o jogador descobrir, certo? Na primeira versão do seu código, coloque uma palavra fixa, como "banana", por exemplo. Em uma segunda versão, você pode criar um array com várias palavras e, no início do programa, sortear uma delas. Veja a seção Dicas, ao final deste material, para saber como você poderia fazer esse sorteio.*
3. *Escolhida uma palavra para o jogador descobrir, você já pode mostrar para ele quantas letras a palavra tem. Também vai ser importante ter uma outra variável que consiste na palavra que o jogador está tentando acertar.*
    1. *Para isso, nossa sugestão é que você crie um array que inicia com vários `'_'`, sendo o número de `_` igual ao número de caracteres da palavra que ele precisa descobrir.*
    2. *Por exemplo, se a palavra for "banana", você deve um array com o seguinte conteúdo: `['_', '_', '_', '_', '_', '_']`. Observe que temos 6 caracteres.*
    3. *Esse array vai ser importante para que você possa mostrar ao usuário o "formato" da palavra e, a medida que ele for acertando as letras, você coloca a letra dentro do array, na posição correta, o que vai facilitar a visualização/identificação da palavra.*
4. *A partir daí o jogo começa, ou seja, você irá pedir que o usuário informe uma letra repetidas vezes, até que ele erre 6 vezes (pela regra tradicional do jogo da forca) ou acerte todas as letras da palavra. Observe que isso corresponde a uma estrutura de repetição.*
5. *Lembre-se que, ao término dessa repetição, você deve mostrar que o usuário perdeu, caso ele tenha errado 6 vezes; ou que ele acertou a palavra, caso ele a tenha completado. Além disso, é importante que você informe, em todo caso, qual era a palavra a ser descoberta.*

Pronto! Essas são algumas diretrizes para que você possa se guiar no desenvolvimento do seu desafio.

Dicas

É possível que você queira utilizar alguns recursos que não foram passados durante o curso. Por exemplo, você pode querer dar boas-vindas ao jogador e deixar o seu código "parado" até que ele pressione a tecla Enter para iniciar. Ou você pode querer obter uma palavra aleatória de um array de palavras.

Com base nisso, já preparamos algumas dicas que podem te ajudar nesse sentido.

1. Como avançar apenas quando o usuário pressionar a tecla Enter?

Para isso, você pode utilizar a biblioteca readline-sync *que vimos durante o curso e, em seguida, utiliza o método question para solicitar que o usuário pressione Enter para continuar.*

Como não precisamos salvar o que o usuário digitar, já que não esperamos que ele informe algum dado, não precisamos guardar o retorno em uma variável.

O código abaixo ilustra o funcionamento descrito acima:

```
const input = require('readline-sync');

input.question('Pressione Enter para continuar...')

console.log('Essa linha será executada somente quando o usuário pressionar Enter.')
```

2. Como posso sortear um elemento de um array no JavaScript?

Você pode utilizar o método random da biblioteca Math. Esse método te retorna um número aleatório entre 0 e 1. No entanto, precisamos obter um número entre 0 e o tamanho do array (exclusivo), ou seja, desejamos gerar um índice do array, de forma aleatória.

Por exemplo: se o nosso array contém 8 elementos, queremos um número aleatório entre 0 e 7. Assim, poderíamos acessar um elemento do nosso array, utilizando esse índice aleatório.

Para gerar, então, um índice aleatório, poderíamos utilizar o código abaixo:

```
const meuArray = [5, 9, 4, 39, 23, 56, 1, 7]

const indiceAleatorio = Math.floor(Math.random() * meuArray.length)
```

Observando o código acima, é possível notar que estamos multiplicando o `Math.random()` *pelo tamanho do array. Dessa forma, teremos um número aleatório entre 0 e o tamanho do array (não incluindo este último). Porém, esse número ainda é decimal, por isso, utilizamos o* `Math.floor()` *para arredondar este número para baixo (remover a parte decimal).*

Pronto! Tendo este `indiceAleatorio`*, podemos apenas acessar o elemento do array. O código abaixo representa a versão completa do que você precisa para obter um valor aleatório de um array no JavaScript:*

```
const meuArray = [5, 9, 4, 39, 23, 56, 1, 7]

const indiceAleatorio = Math.floor(Math.random() * meuArray.length)

const elementoAleatorio = meuArray[indiceAleatorio]
```