

**Habilidades trabalhadas nesta aula:**

**(EMIFMAT07)** Identificar e explicar questões socioculturais e ambientais aplicando conhecimentos e habilidades matemáticas para avaliar e tomar decisões em relação ao que foi observado.

**(EM13CO01)** Explorar e construir a solução de problemas por meio da reutilização de partes de soluções existentes.

**(EM13CO02)** Explorar e construir a solução de problemas por meio de refinamentos, utilizando diversos níveis de abstração desde a especificação até a implementação.

**(EM13CO15)** Analisar a interação entre usuários e artefatos computacionais, abordando aspectos da experiência do usuário e promovendo reflexão sobre a qualidade do uso dos artefatos nas esferas do trabalho, do lazer e do estudo.

**(EMIFCG07)** Reconhecer e analisar questões sociais, culturais e ambientais diversas, identificando e incorporando valores importantes para si e para o coletivo que assegurem a tomada de decisões conscientes, consequentes, colaborativas e responsáveis.




**Aula 5**

# Criando botões de alternativa

**► Unidade**

**Funções: criando uma missão sobre Inteligência Artificial**

# O que vamos aprender?

-  Criar uma função para mostrar as alternativas.
-  Criar um loop para atravessar os elementos de uma lista.
-  Implementar os métodos **createElement** e **appendChild**.



ACOMPANHE O VÍDEO DA AULA



## Para cada opção, um botão

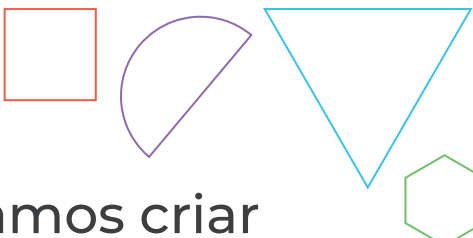
Na aula anterior, criamos uma função para mostrar as perguntas em nosso projeto utilizando JavaScript, fazendo com que elas apareçam na tela. Contudo, nossa página ainda não mostra os botões de alternativa. Nesta aula, utilizaremos uma função para exibir as alternativas na página.

### Você decide o futuro da IA

Assim que saiu da escola, você se depara com uma nova tecnologia: um chat que consegue responder a todas as dúvidas que uma pessoa pode ter. Além disso, o chat também gera imagens e áudios hiper-realistas. Qual o seu primeiro pensamento?

Isso é assustador! Isso é maravilhoso!

⚠ Para iniciar a aula, é importante que os estudantes estejam confortáveis com as estruturas básicas utilizadas até agora, principalmente a hierarquia das informações em listas e objetos, bem como o uso de funções. Além disso, os estudantes podem ser instigados a refletir sobre páginas da internet que já acessaram e que possuem um esquema de perguntas e respostas (como jogos, provas online, questionários etc.), bem como sobre quais eram as características principais dessas páginas. Esse exercício de reflexão ajudará na abstração das características que já conhecem para utilizar no projeto que estão construindo.



Observando nosso protótipo, podemos perceber que precisamos criar nossos botões de alternativas, sendo que esses botões aparecem de acordo com a quantidade de alternativas que temos em nosso projeto.


Para isso, o primeiro passo é fazer com que a nossa função de mostrar alternativas seja chamada juntamente com a função **mostraPergunta**. Desse modo, alteraremos o código, no arquivo *script.js*, para incluir a chamada dessa função que ainda vamos criar.

```
function mostraPergunta() {  
    perguntaAtual = perguntas[atual];  
    caixaPerguntas.textContent = perguntaAtual.enunciado  
    mostraAlternativas();  
}
```

⚠ Nessa etapa, é interessante reforçar com os estudantes que a função já foi chamada, mas ainda não foi criada. Sendo assim, se testarmos o código dessa maneira, veremos que nada foi alterado em nosso projeto.

Agora, podemos começar a desenvolver a função de mostrar alternativas, logo abaixo da função **mostraPergunta**, mas fora de suas chaves.


```
function mostraPergunta() {  
    perguntaAtual = perguntas[atual];  
    caixaPerguntas.textContent = perguntaAtual.enunciado  
    mostraAlternativas();  
}  
function mostraAlternativas() {
```



Nessa função, queremos que sejam criados os botões e que, nesses botões, apareçam os textos das nossas alternativas. Além disso, precisamos fazer com que esses botões apareçam na tela. Lembre-se de que nossas alternativas pertencem a uma lista de perguntas e estão salvas como uma lista dentro da propriedade **alternativas**, no arquivo *script.js*. Observe:

```
{  
  enunciado: "Assim que saiu da escola, você se  
depara com uma nova tecnologia: um chat que consegue  
responder a todas as dúvidas que uma pessoa pode ter.  
Além disso, o chat também gera imagens e áudios  
hiper-realistas. Qual o seu primeiro pensamento?",  
  alternativas: [  
    "Isso é assustador!",  
    "Isso é maravilhoso!"  
  ],  
},
```

⚠ É fundamental garantir que a hierarquia das informações esteja clara. Sendo assim, você pode aproveitar esse momento para retomar com a sala sobre essa hierarquia e garantir que não haja dúvidas de como acessar as informações nas listas e objetos.



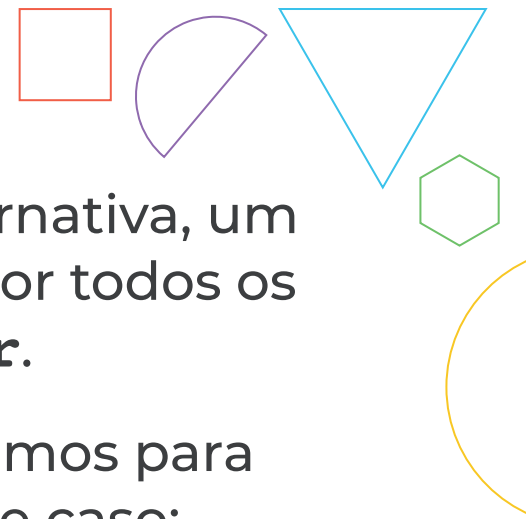
Além disso, precisamos lembrar de que estamos utilizando um contador na variável atual para salvar a pergunta, sendo chamada pela função **mostraPergunta()**, na variável **perguntaAtual**. Observe:

```
function mostraPergunta() {  
    perguntaAtual = perguntas[atual];  
    caixaPerguntas.textContent = perguntaAtual.enunciado;  
}
```

Desse modo, na função **mostraAlternativas()** podemos acessar a lista de alternativas utilizando o código a seguir:

```
perguntaAtual.alternativas
```

Vamos observar como ela deve ser inserido dentro da função de modo que funcione corretamente.



Para isso, precisamos de um código que crie, para cada alternativa, um botão diferente. Você deve se lembrar de que, para passar por todos os elementos de uma lista, podemos utilizar um loop (laço) **for**.

No loop **for**, precisamos fornecer uma variável que utilizaremos para fazer referência aos elementos que estamos passando, nesse caso:

```
const alternativa
```


Seu código completo ficará da seguinte forma:

```
function mostraAlternativas() {  
  for (const alternativa of perguntaAtual.alternativas) {
```

Esse código pode ser lido como “para cada alternativa de **perguntaAtual.alternativas**”, execute as instruções dentro das chaves.

⚠️ Caso existam dúvidas sobre o uso de *loops for*, recomenda-se sempre estimular a pesquisa da documentação ou vídeos e sites. A documentação da linguagem é algo comumente visitado por desenvolvedores com o objetivo de tirar dúvidas do dia a dia da programação. Um exemplo de documentação pode ser encontrado no link: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>.





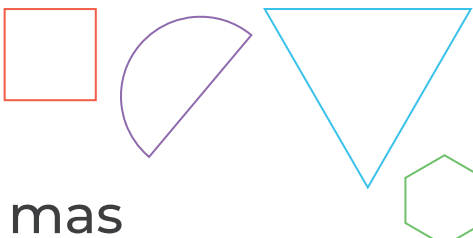
Agora que já temos nossa função e nosso loop para passar por todas as alternativas, começaremos a criar os botões.

O primeiro passo é incluir o código para criar um elemento HTML de botão. Para isso, usaremos o método **createElement** do objeto **document**.

```
function mostraAlternativas() {  
  for (const alternativa of perguntaAtual.alternativas) {  
    const botaoAlternativas =  
    document.createElement("button");
```

No código, estamos salvando esse elemento na variável **botaoAlternativas** e passando, como parâmetro do método **createElement**, o texto “button”.


⚠️ Caso existam dúvidas sobre o uso do método `createElement`, recomenda-se sempre estimular a pesquisa da documentação ou vídeos e sites. Esse tipo de função é muito comum e acaba sendo decorada por conta de seu uso recorrente. Contudo, é comum desenvolvedores e desenvolvedoras que estão aprendendo recorrerem à documentação constantemente.



Observe que o código anterior apenas criou um botão vazio, mas queremos que o texto do nosso botão seja o texto da alternativa.

Nesse momento, precisamos lembrar que estamos fazendo um *loop for*, então, o texto da alternativa está na variável **alternativa** do loop. Desse modo, passaremos o valor de “alternativa” para a propriedade **textContent** do botão.

```
function mostraAlternativas() {  
    for (const alternativa of perguntaAtual.alternativas) {  
        const botaoAlternativas = document.  
createElement("button");  
        botaoAlternativas.textContent = alternativa;  
    }  
}
```



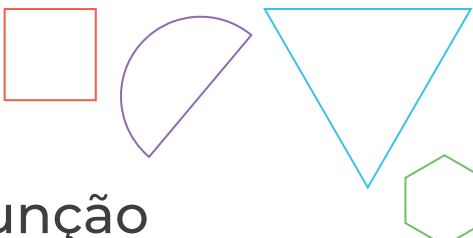
Por fim, precisamos fazer com que esses botões de alternativa apareçam na tela. Lembre-se de que, no arquivo `index.html`, temos uma `div` com a classe **`caixa-alternativas`**.

```
<div class="caixa-alternativas"></div>
```

Também salvamos essa `div` em uma variável, no começo do arquivo `script.js`:

```
const caixaAlternativas = document.querySelector  
(".caixa-alternativas");
```

Porém, ainda não inserimos o botão que criamos nessa `div`.



Para fazermos isso, adicionaremos mais uma linha à nossa função **mostraAlternativa**, utilizando o método **appendChild** na variável **caixaAlternativas**. Passaremos como parâmetro desse método o **botaoAlternativa** que criamos. Esse método serve para adicionarmos um elemento HTML como filho de outro elemento; nesse caso, o botão será adicionado como filho da div. Observe:

```
function mostraAlternativas() {  
    for (const alternativa of perguntaAtual.alternativas) {  
        const botaoAlternativas = document.  
createElement("button");  
        botaoAlternativas.textContent = alternativa;  
        caixaAlternativas.appendChild(botaoAlternativas);  
    }  
}
```



Agora, temos nossa função completa e podemos observar tanto o enunciado quanto os botões de alternativas aparecendo na tela.

## Você decide o futuro da IA

Assim que saiu da escola, você se depara com uma nova tecnologia: um chat que consegue responder a todas as dúvidas que uma pessoa pode ter. Além disso, o chat também gera imagens e áudios hiper-realistas. Qual o seu primeiro pensamento?

Isso é assustador!

Isso é maravilhoso!

Contudo, ao clicarmos nos botões de alternativas, podemos perceber que nada acontece. Resolveremos esse problema nas próximas aulas.

## ► Desafio

Nesta aula, utilizamos uma função para fazer os botões das alternativas aparecerem na tela. Criamos a função **mostraAlternativa** e utilizamos um loop para passar por todos os elementos da lista de alternativas, além de utilizar os métodos **createElement** e **appendChild** para transformar o HTML a partir do JavaScript.

Seu desafio agora é pensar em como criar um novo botão que adicione uma nova funcionalidade ao jogo, como iniciar, jogar novamente, game over ou qualquer outra que você deseja criar. O botão não precisa funcionar efetivamente, mas você deve pensar em um texto para ele e utilizar os conhecimentos da aula de hoje para fazer com que ele apareça na página HTML.

⚠ Espera-se que os estudantes apliquem os conhecimentos adquiridos ao longo desta unidade e consigam criar um novo botão para o projeto, tornando-o mais dinâmico e praticando o uso de funções e loop no JavaScript.



CLIQUE AQUI PARA AVALIAR ESTE MATERIAL