

## Aula 7

# Adicionando um ouvinte ao botão

► **Unidade**

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

# O que vamos aprender?

- Modificar a função **mostraAlternativa**.
- Modificar a estrutura das alternativas.
- Criar um **Event Listener** utilizando o método **addEventListener()**.



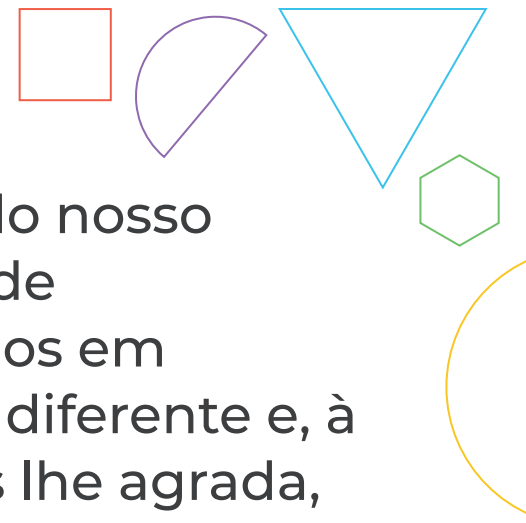
ACOMPANHE O VÍDEO DA AULA

# O método addEventListener()

Na aula anterior, utilizamos o CSS para modificar a parte de estilo da nossa página, deixando-o com um visual mais agradável. Nesta aula, começaremos a construir o resultado da nossa missão, conectando algumas afirmações em cada uma das alternativas das perguntas, fazendo com que o usuário possa visualizar essas afirmações à medida que for escolhendo cada uma das alternativas.

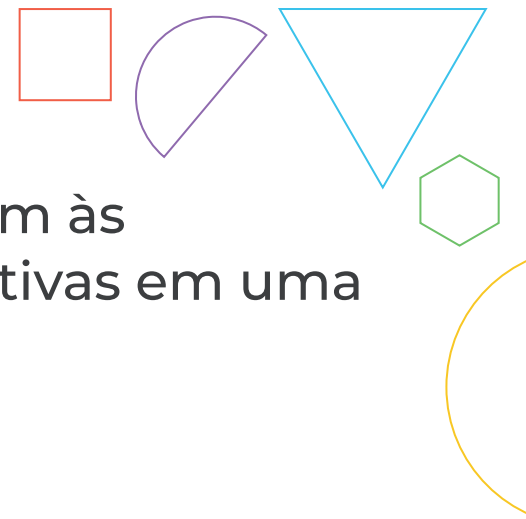
**Você decide o futuro da IA**

Você tem um trabalho de Biologia em grupo para entregar na próxima semana. O andamento do trabalho está um pouco atrasado, e uma pessoa do seu grupo decidiu fazer a parte dela com a ajuda de uma IA. O problema é que o trabalho está totalmente igual ao do chat. O que você faz?



Observando nosso Figma, podemos notar que o resultado do nosso projeto é um texto que será definido como uma sequência de afirmações que está conectada às alternativas que colocamos em nossos botões. Para cada alternativa, temos uma afirmação diferente e, à medida que o usuário for escolhendo a alternativa que mais lhe agrada, o texto vai sendo construído.

Assim, precisamos construir essa conexão entre alternativas e afirmações em nosso VS Code para que nosso projeto funcione corretamente.




Portanto, para fazermos com que as alternativas se conectem às afirmações, transformaremos nosso **array** (lista) de alternativas em uma lista de objetos.

Retornaremos ao código original no arquivo *script.js*:


```
const perguntas = [  
  {  
    enunciado: "Insira o enunciado da Pergunta 1",  
    alternativas: ["Alternativa 1", "Alternativa 2"],  
  },  
]
```

Temos uma lista **perguntas** que contém objetos com duas propriedades: o enunciado da pergunta e a lista de alternativas.



O que precisamos alterar em nosso código atual é que cada alternativa, em vez de ser apenas o texto, seja um objeto com duas propriedades: o texto da alternativa e o texto da afirmação. Para criar objetos no JS, precisamos fazer uso de chaves e separar os atributos por vírgulas. Feitas essas alterações, seu código ficará da seguinte forma:

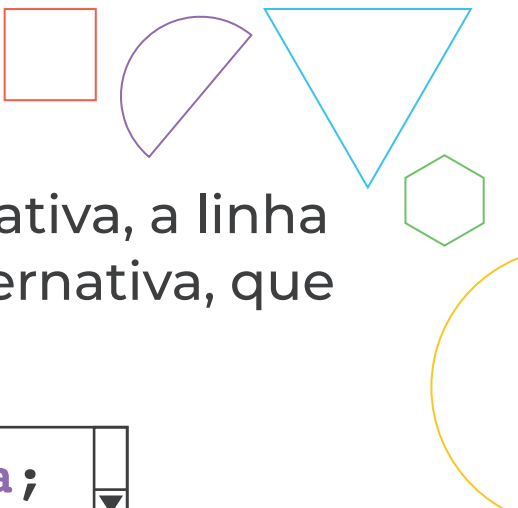
```
const perguntas = [  
  {  
    enunciado: "Insira o enunciado da Pergunta 1",  
    alternativas: [  
      {  
        texto: "Alternativa 1",  
        afirmação: "Afirmação da alternativa 1"  
      },  
      {  
        texto: "Alternativa 2",  
        afirmação: "Afirmação da alternativa 2"  
      }  
    ]  
  },  
  (...continuam as próximas perguntas)
```



Faça essa alteração para todas as perguntas que você criou, transformando a lista de alternativas em uma lista de objetos de alternativa. Agora, precisamos fazer com que nossos botões funcionem e, para isso, voltaremos na função que criamos nas aulas anteriores, a função **mostraAlternativa**.

O código atual dessa função é:

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



Como anteriormente tínhamos uma lista de textos de alternativa, a linha de código que remete ao conteúdo do texto do botão de alternativa, que integra a propriedade **textContent**, é a própria alternativa.

```
botaoAlternativas.textContent = alternativa;
```

Mas, agora, temos um objeto na lista de alternativas, de modo que precisamos passar para o **textContent** apenas a propriedade texto da alternativa. Sendo assim, colocaremos um **.texto** após a alternativa. Observe:

```
botaoAlternativas.textContent = alternativa.texto;
```





Feita essa alteração, seguimos com um problema: nosso botão ainda não funciona quando clicamos nele.

Para fazermos essa correção, utilizaremos um método do JavaScript chamado **addEventListener()**, que adiciona um ouvinte ao elemento, ou seja, adiciona um ouvinte aos nossos botões. Você pode imaginar que esse método é algo que faz com que a nossa aplicação fique escutando se o botão foi apertado ou não.


Esse método pode ser invocado no próprio elemento **botaoAlternativas**

```
botaoAlternativas.addEventListener();
```

Esse método funciona com parênteses e espera dois parâmetros: precisamos dizer para ele qual o tipo de evento ele precisa ouvir e o que ele deve fazer quando esse evento for ouvido.

O primeiro parâmetro é simples: queremos que o botão “ouça” se ele foi ou não clicado. Então, adicionamos o evento “**click**”.

```
botaoAlternativas.addEventListener("click");
```



Agora, para dizer o que o botão deve fazer quando for clicado, utilizaremos como o segundo parâmetro uma função anônima, **function()**, que é invocada quando o **EventListener** ouve o evento do clique.

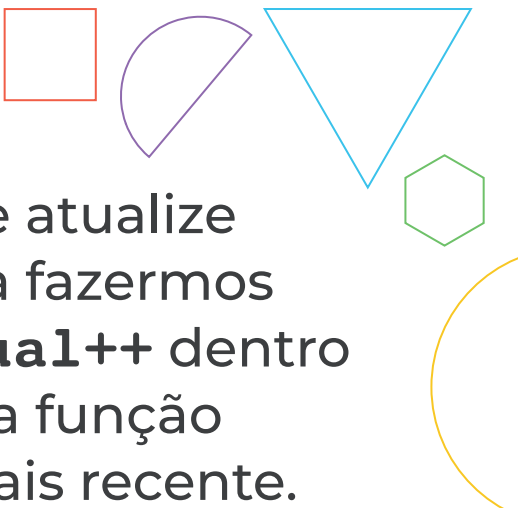
```
botaoAlternativas.addEventListener("click", function () {  
  
});
```

Contudo, o que precisamos colocar nessa função anônima? Você deve se lembrar de que estamos usando um contador para saber qual é a pergunta atual:

```
let atual = 0;
```

Esse contador é iniciado em zero, de modo que a função **mostraPergunta()** que chamamos no início do código mostre apenas a primeira pergunta

```
function mostraPergunta() {  
    perguntaAtual = perguntas[atual];
```




Desse modo, quando o botão for clicado, precisamos que ele atualize o número do contador “atual” para a próxima pergunta. Para fazermos isso, mudaremos a nossa pergunta atual, acrescentando **atual++** dentro da nossa função anônima, além de chamarmos novamente a função **mostraPergunta()** com o número atual alterado para o mais recente.

```
botaoAlternativas.addEventListener("click", function () {  
    atual++;  
    mostraPergunta();  
});
```



O código completo ficará da seguinte forma:

```
function mostraAlternativa() {  
    for (const alternativa of perguntaAtual.alternativas) {  
        const botaoAlternativas = document.  
createElement("button");  
        botaoAlternativas.textContent = alternativa.texto;  
        botaoAlternativas.addEventListener("click",  
function () {  
            atual++;  
            mostraPergunta();  
        });  
        caixaAlternativas.appendChild(botaoAlternativas);  
    }  
}
```




Vamos testar como está ficando o nosso projeto? Ele deve estar semelhante à imagem a seguir:

## 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!



Podemos perceber que os botões aparecem e conseguimos clicar neles. Mas, toda vez que clicamos sobre eles, novos botões surgem, fazendo com que fiquem todos acumulados na tela.

## Você decide o futuro da IA

Após a elaboração do trabalho escrito, a professora realizou um debate entre a turma para entender como foi realizada a pesquisa e escrita. Nessa conversa também foi levantado um ponto muito importante: como a IA impacta o trabalho do futuro. Nesse debate, como você se posiciona?

Isso é assustador!

Isso é maravilhoso!

Utiliza uma ferramenta de busca na internet que utiliza IA para que ela ajude a encontrar informações relevantes para o trabalho e explique numa linguagem que facilite o entendimento.

Escreve o trabalho com base nas conversas que teve com colegas, algumas pesquisas na internet e conhecimentos próprios sobre o tema.

Defende a ideia de que a IA pode criar novas oportunidades de emprego e melhorar habilidades humanas.

Me preocupo com as pessoas que perderão seus empregos para máquinas e defendem a importância de proteger os trabalhadores.

Na próxima aula, veremos como corrigir esse problema.

## ► Desafio

Nesta aula, utilizamos JavaScript para dar funcionalidade aos botões, além de transformamos a lista de alternativas em uma lista de objetos de alternativa. Porém, você percebeu que terminamos com um problema: os botões de alternativa estão se acumulando na tela.

Seu desafio é transformar novas listas em objetos, aumentando a complexidade do seu projeto e tornando sua história mais instigante (ou mesmo criando uma “side story”, se desejar)! Deixe todos os seus botões devidamente configurados para que possamos fazê-los funcionar na próxima aula.



CLIQUE AQUI PARA AVALIAR ESTE MATERIAL