

Aula 4

Função mostra pergunta


► **Unidade**

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

O que vamos aprender?

 Compreender os atributos **textContent** do DOM e o enunciado da pergunta.

 Compreender funções.

 Criar uma função que manipule esses atributos.



 ACOMPANHE O VÍDEO DA AULA



Exibindo a pergunta

Na aula anterior, preparamos nosso código de JavaScript. Também fizemos referências às classes do HTML e salvamos seu conteúdo em variáveis constantes, além de criarmos uma lista de perguntas contendo objetos de pergunta, com os atributos enunciado e alternativas.

Contudo, a página ainda não exibe nenhuma pergunta. Nesta aula, vamos utilizar uma função para exibir a pergunta na página.

Você decide o futuro da IA

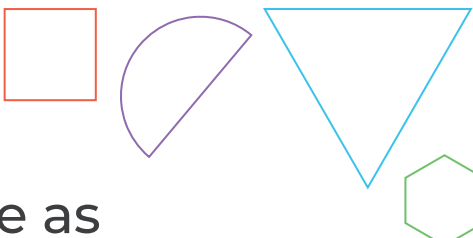
Insira o enunciado da Pergunta 1



De volta ao VSCode, a primeira coisa que faremos será inserir o texto dos enunciados e das alternativas.

Para isso, copie o objeto da pergunta o número de vezes que for necessário e insira, no atributo enunciado, as perguntas, bem como as alternativas nas listas de alternativas. Não se esqueça de separar as perguntas por vírgula e de inserir o número correto de perguntas que tiver elaborado. Observe o exemplo:

```
const perguntas = [  
  {  
    enunciado: "Insira o enunciado da Pergunta 1",  
    alternativas: ["Alternativa 1", "Alternativa 2"],  
  },  
  {  
    enunciado: "Insira o enunciado da Pergunta 2",  
    alternativas: ["Alternativa 1", "Alternativa 2"],  
  },  
  {  
    enunciado: "Insira o enunciado da Pergunta 3",  
    alternativas: ["Alternativa 1", "Alternativa 2"],  
  }  
];
```

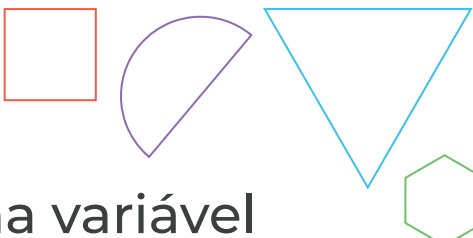


Ao abrirmos nosso arquivo HTML no navegador, veremos que as perguntas ainda não aparecem:

Você decide o futuro da IA

Para resolver isso, a primeira coisa que criaremos é uma variável para marcar a posição na lista que queremos mostrar naquele momento. Você deve se lembrar que listas começam sempre na posição zero, então, o valor inicial dessa variável deve ser 0. Contudo, o valor dela mudará quando percorremos a lista, de modo que essa variável precisa ser declarada usando **let**, logo ao final do código. Observe:

```
let atual = 0;
```




Tendo um marcador de posição, também precisamos de uma variável que receberá o texto da pergunta. Essa variável também mudará com o tempo, recebendo o texto de diferentes perguntas. Desse modo, podemos criar uma variável vazia, também utilizando **let**. Observe:

```
let atual = 0;  
let perguntaAtual;
```

Agora, para que possamos efetivamente mostrar a pergunta, vamos usar uma função. A primeira coisa que nossa função precisa fazer é colocar dentro da variável **perguntaAtual** o texto da pergunta. Então, vamos primeiro declarar a função:

```
function mostraPergunta() {  
}
```



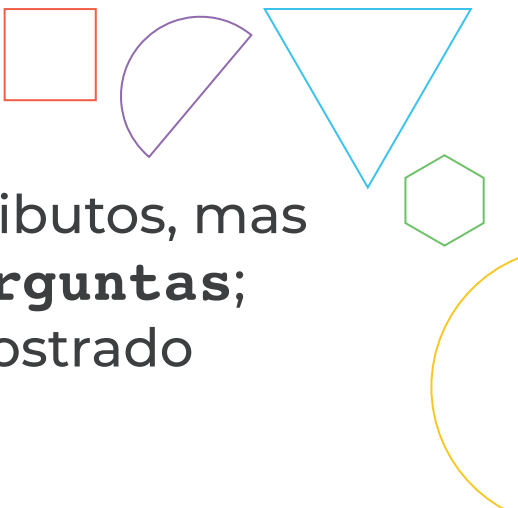
Em seguida, vamos passar a instrução para receber o texto da pergunta atual. A posição da lista que queremos utilizar é o marcador variável **atual**. Por isso, devemos adicioná-la à função:

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

Contudo, a função criada ainda não mostra a pergunta no navegador, pois ela está apenas salvando o objeto da pergunta dentro da variável **perguntaAtual**. Para fazer com que a pergunta apareça no navegador, precisamos manipular o DOM. Você deve se lembrar de que, na aula anterior, salvamos a classe **caixa-perguntas** em uma variável **caixaPerguntas** com o seguinte código:

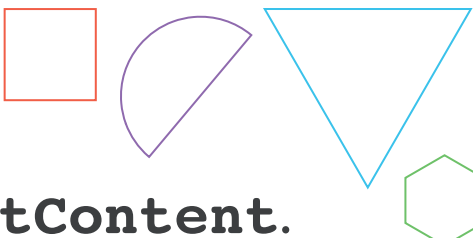
```
const caixaPerguntas = document.querySelector  
(".caixa-perguntas");
```

Ou seja, dentro dessa variável **caixaPerguntas** temos um objeto do HTML que queremos manipular.



Contudo, observe que não queremos manipular todos os atributos, mas apenas o texto. Ou seja, não alteraremos a variável **caixaPerguntas**; modificaremos apenas seu atributo **textContent**, como mostrado a seguir:

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


Entretanto, não queremos colocar a pergunta inteira no **textContent**. Lembre-se de que a variável **perguntas** é uma lista de objetos, ou seja, o valor de **perguntas[atual]** que inserimos na variável **perguntaAtual** é um objeto. Observe:

```
perguntas[atual]
perguntaAtual = perguntas[atual];
```

Assim, a única coisa que queremos inserir no **textContent** é o atributo **enunciado**.

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

Mesmo fazendo isso, ao atualizar o navegador, veremos que nossas perguntas ainda não são exibidas.



Isso ocorre porque somente declaramos a função **mostraPergunta**, mas ainda não chamamos por ela. Assim, vamos chamar a função:

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

Agora, a pergunta aparecerá no navegador:

Você decide o futuro da IA

Insira o enunciado da Pergunta 1

► Desafio

Nesta aula, utilizamos uma função para fazer a pergunta aparecer na tela. Essa função manipulou o DOM, alterando o atributo **textContent** da classe que havíamos salvado na variável **caixaPergunta**, inserindo apenas o atributo enunciado do objeto **pergunta** nela.

Seu desafio agora é pensar como você faria para inserir as alternativas e fazer com que elas apareçam na página. Não se preocupe em acertar tudo, faremos isso juntos na próxima aula. Comece analisando documentação e o código atual, e pensando em como você faria para mostrar as alternativas. Copie a pasta com os arquivos HTML, CSS e JS para modificá-los à vontade, sem bagunçar o código atual, e teste seu trabalho abrindo o arquivo HTML.



CLIQUE AQUI PARA AVALIAR ESTE MATERIAL