

## Aula 9

# Adicionando condição de parada


► **Unidade**

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

# O que vamos aprender?

 Criar uma concatenação de texto.

 Criar um código para “limpar” o conteúdo e alterá-lo.

 Criar a lógica de uma condição de parada.



ACOMPANHE O VÍDEO DA AULA



# Quando as perguntas acabam


Na aula anterior, utilizamos JavaScript para dar mais funcionalidade aos botões do nosso projeto e começamos a salvar os dados das afirmações para compor a história final. Porém, ainda terminamos com as alternativas se empilhando na tela e ainda não havíamos criado uma condição de parada que determine o final do jogo. Nesta aula, resolveremos esses problemas.

## Você decide o futuro da IA

Com a descoberta desta tecnologia, chamada Inteligência Artificial, uma professora de tecnologia da escola decidiu fazer uma sequência de aulas sobre esta tecnologia. No fim de uma aula, ela pede que você escreva um trabalho sobre o uso de IA em sala de aula. Qual atitude você toma?

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.



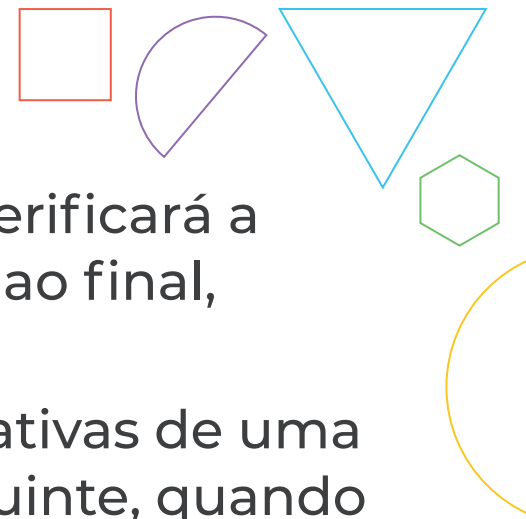
Antes de seguir com o código, precisamos pensar no que queremos que ele faça. Para criarmos uma condição de parada que indique que nossas perguntas acabaram e o jogo terminou, é necessário que o programa execute uma série de instruções para mostrar o resultado ao chegarmos na pergunta final.

Em código, isso significa utilizar um **if** que verificará se nosso contador atual está na última pergunta. Assim, adicionaremos uma linha de código à função **mostraPergunta()**, observe:

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

Ao chegarmos à posição final **length** das nossas perguntas, queremos executar instruções para mostrar o resultado e parar o programa após isso. Desse modo, adicionaremos uma função **mostraResultado()**, que criaremos a seguir, e utilizaremos o comando **return** para parar a função. Seu código ficará da seguinte forma:

```
function mostraPergunta() {  
    if (atual >= perguntas.length) {  
        mostraResultado();  
        return;  
    }  
    perguntaAtual = perguntas[atual];  
    caixaPerguntas.textContent =  
    perguntaAtual.enunciado;  
    mostraAlternativas();  
}
```

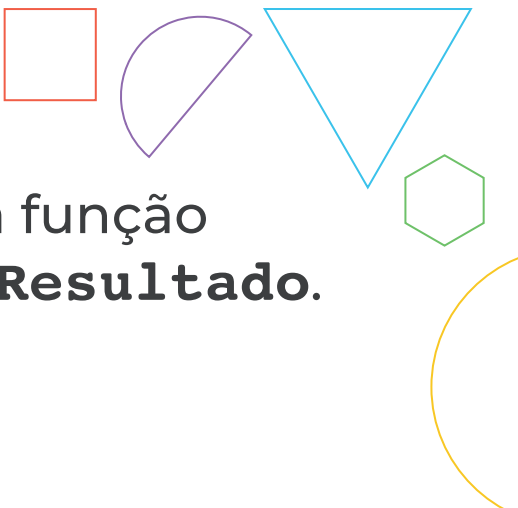


Salve seu código e teste o programa pelo LiveServer. Você verificará a condição de parada está funcionando e o jogo agora chega ao final, porém, as alternativas ainda estão se empilhando.

Para resolver isso, precisamos limpar o conteúdo das alternativas de uma pergunta antes de carregar as alternativas da pergunta seguinte, quando a função **mostraAlternativas** é chamada. Ou seja, precisamos inserir um texto vazio no **textContent** da **caixaAlternativas**.

Podemos fazer isso com o seguinte comando:

```
function mostraPergunta() {  
    if (atual >= perguntas.length) {  
        mostraResultado();  
        return;  
    }  
    perguntaAtual = perguntas[atual];  
    caixaPerguntas.textContent = perguntaAtual.enunciado;  
    caixaAlternativas.textContent = " ";  
    mostraAlternativas();  
}
```



Agora que já estamos limpando as alternativas e temos uma função de parada, precisamos criar as instruções da função **mostraResultado**. Para isso, vamos adicionar essa nova função ao código:

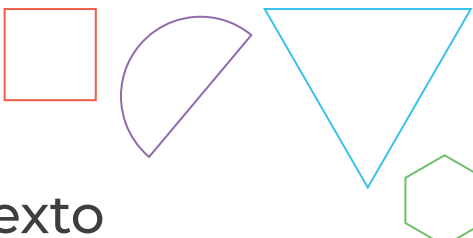
```
function mostraResultado() {  
  
}
```

Agora, essa função precisa seguir três instruções.

A **primeira** é colocar o texto *Em 2049...* em **caixaPerguntas**.

Verificando o protótipo, você perceberá que o resultado sempre começa com *Em 2049...* e é seguido pelo texto das afirmações selecionadas, portanto, ele deve aparecer no resultado. Observe:

```
function mostraResultado() {  
    caixaPerguntas.textContent = "Em 2049...";  
}
```



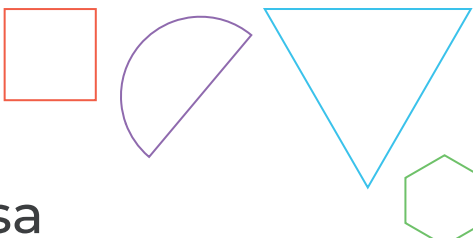
A **segunda** coisa que essa função precisa fazer é mostrar o texto das alternativas escolhidas. Você deve se lembrar que a função **respostaSelecionada**, que criamos nas aulas anteriores, estava salvando todo esse conteúdo na variável **historiaFinal**.

```
function respostaSelecionada(opcaoSelecionada) {  
    const afirmacoes = opcaoSelecionada.afirmacao;  
    historiaFinal = afirmacoes;  
    atual++;  
    mostraPergunta();  
}
```

Desse modo, podemos salvar a **historiaFinal** no **textContent** da div **textoResultado**. Observe:

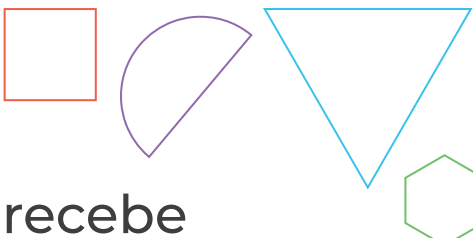
```
function mostraResultado() {  
    caixaPerguntas.textContent = "Em 2049...";  
    textoResultado.textContent = historiaFinal;  
}
```





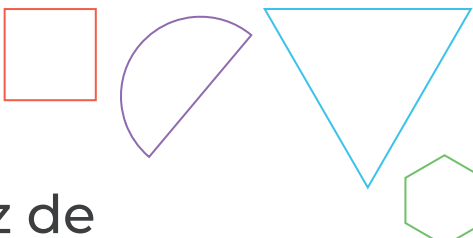
Por fim, o **terceiro** e último comando que essa função precisa executar é limpar as caixas de alternativas, afinal, não queremos os botões aparecendo junto com o resultado final. Para isso, conforme feito anteriormente, basta inserir um texto vazio no **textContent** da **caixaAlternativas**. Observe:

```
function mostraResultado() {  
    caixaPerguntas.textContent = "Em 2049...";  
    textoResultado.textContent = historiaFinal;  
    caixaAlternativas.textContent = "";  
}
```



Contudo, agora temos um novo problema: **historiaFinal** recebe sempre a afirmação da última opção selecionada. E, quando uma nova seleção ocorre, esse valor é atualizado para o último escolhido. Dessa forma, no final, teremos uma variável **historiaFinal** apenas com o valor da última alternativa escolhida.

```
function respostaSelecionada(opcaoSelecionada) {  
    const afirmacoes = opcaoSelecionada.afirmacao;  
    historiaFinal = afirmacoes;  
    atual++;  
    mostraPergunta();  
}
```



Dessa forma, precisamos concatenar o texto. Ou seja, em vez de substituir, juntaremos mais um pedaço ao texto atual. Para isso, basta mudarmos o `=` por `+=`. Porém, não queremos que o texto fique colado um no outro, então, além do conteúdo de **afirmacoes**, adicionaremos também um espaço em branco para separá-los. Para isso, basta somar um texto em branco, entre aspas `" "`.

```
function respostaSelecionada(opcaoSelecionada) {  
  const afirmacoes = opcaoSelecionada.afirmacao;  
  historiaFinal += afirmacoes + " ";  
  atual++;  
  mostraPergunta();  
}
```



Se testarmos nosso código agora, poderemos ver a página final:

# Você decide o futuro da IA

Em 2049...

afirmação afirmação afirmação afirmação afirmação

Contudo, ainda não incluímos os textos das afirmações. Faremos isso na próxima aula.

## ► Desafio

Nesta aula, criamos uma função de parada e resolvemos o problema das alternativas empilhadas. Além disso, criamos também a função para mostrar o resultado da nossa história.

Seu desafio é pensar no que você faria diferente na hora de mostrar o resultado final. Mostraria o nome do usuário? Mudaria a cor da página de acordo com as respostas? Colocaria imagens? Faça um Protótipo das suas ideias usando uma ferramenta digital ou analógica e tente implementar ao menos uma dessas ideias.

Para isso, analise o código que fizemos e busque utilizar os conhecimentos adquiridos até aqui. Copie a pasta com os arquivos HTML, CSS e JS para mexer à vontade nos arquivos sem bagunçar o código atual, e teste o seu trabalho abrindo o arquivo HTML para verificar seus resultados.



CLIQUE AQUI PARA AVALIAR ESTE MATERIAL