

11

Mão na massa: encerrando o jogo

1. No jogo da forca, implemente a lógica para que o jogo dê apenas 6 chances para o jogador tentar acertar a palavra. Primeiro, crie uma variável chamada `erro` fora do loop e initialize-a com zero.

```
erros = 0
```

2. Em seguida, coloque uma condição dentro do loop para verificar se o jogador acertou a letra. Ela deve englobar a inicialização da variável `index` e o loop `for`. Crie um `else` para incrementar a variável `erros`.

```
if(chute in palavra_secreta):
    index = 0
    for letra in palavra_secreta:
        if(chute == letra):
            letras_acertadas[index] = letra
            index = index + 1
else:
    erros = erros + 1
```

3. No final do loop, atualize a variável `enforcou` para que ela se torne verdadeira quando a quantidade de erros for 6. Teste o jogo colocando 6 letras erradas e verifique que o jogo termina!

```
enforcou = erros == 6
```

4. Agora implemente a lógica para que o jogo termine quando o jogador acertar a palavra. Crie mais uma linha no final do loop para atualizar a variável `acertou` com a verificação se o caracter `underscore` não existe em `letras_acertadas`. Teste a aplicação e coloque as letras da palavra secreta. Confirme que o jogo termina.

```
acertou = "_" not in letras_acertadas
```

5. Apesar do jogo encerrar, o jogador ainda não sabe se acertou ou foi enforcado. Após o loop e antes da impressão da mensagem `Fim do jogo`, coloque um `if` para imprimir `Você ganhou!!` caso ele tenha vencido a forca e `Você perdeu!` caso contrário. Teste novamente sua aplicação.

```
if(acertou):
    print("Você ganhou!!")
else:
    print("Você perdeu!!")
```

6. Inicialize a variável `letras_acertadas` dinamicamente a partir de qualquer palavra secreta. Utilize o recurso *List Comprehension* para criar um laço dentro da inicialização da lista.

```
letras_acertadas = ["_" for letra in palavra_secreta]
```

