

## Jogo da Forca

Nesta aula vamos criar mais um joguinho interativo para fixar os conhecimentos adquiridos até o momento, o jogo escolhido é o jogo da forca, é um joguinho bastante simples, onde uma palavra aleatória é escolhida e o objetivo do jogador é tentar adivinhar, caractere por caractere, qual a palavra escolhida. A princípio, a única informação disponível para o jogador é a quantidade de caracteres da palavra, e a cada rodada ele pode digitar apenas um caractere, quando o caractere digitado é o correto, ele consegue visualizar a posição do caractere na palavra a ser adivinhada, quando o caractere digitado não está presente na palavra, ele simplesmente perde uma "vida".

Bom, com as devidas introduções fora do caminho, vamos começar a escrever o nosso programa, o primeiro passo será definir algumas variáveis importantes para o funcionamento do jogo:

```
palavras = {"arroz", "pato", "avestruz"} # Pool de palavras aleatórias
letras = "" # Caracteres informados pelo usuário, começa vazio.
max_erros = 3 # Número máximo de tentativas
erros = 0 # Erros cometidos pelo usuário, começa em zero.
palavra_escolhida = palavras.pop() # .pop retorna um valor aleatório do nosso set
```

Apesar de cada variável conter seu próprio comentário explicativo, vamos entrar em um pouco mais de detalhe do porquê de cada uma delas

- palavras: Essa variável contém um set que funciona como o nosso banco de dados de possíveis palavras, escolhemos um set aqui estrategicamente para poder utilizar o método pop, que vamos explicar mais a frente;
- letras: Nesta variável vamos guardar todos os caracteres digitados pelo usuário;
- max\_erros: Aqui vamos definir a quantidade máxima de vezes que o jogador pode errar;
- erros: Vamos salvar a quantidade de vezes que o jogador errou o caractere aqui;
- palavra\_escolhida: Essa é a palavra que fará parte do jogo, vamos selecionar uma palavra aleatória do conjunto de palavras disponíveis utilizando o método .pop() do set palavras, que retorna um valor aleatório contido no set.

Como o jogo só acontece se o usuário ainda tiver vidas disponíveis, ou seja, o valor de erros é menor do que max\_erros, então, vamos definir um loop principal para controlar isso. Aproveitando, vamos utilizar a instrução else (que também é válida com while) para dar uma mensagem de game over quando o jogador tiver perdido todas as chances.

```
palavras = {"arroz", "pato", "avestruz"} # Pool de palavras aleatórias
letras = "" # Caracteres informados pelo usuário, começa vazio.
max_erros = 3 # Número máximo de tentativas
erros = 0 # Erros cometidos pelo usuário, começa em zero.
palavra_escolhida = palavras.pop() # .pop retorna um valor aleatório do nosso set

while erros < max_erros:
    # Lógica do jogo
else:
    print("Que pena, todas as suas vidas acabaram :)")
```

Agora sim, vamos a parte mais desafiadora desta criação, originalmente, no jogo da forca as letras ainda não descobertas pelo jogador aparecem apagadas, no nosso jogo vamos representar essas letras com o caractere underline ("\_"), para isso vamos utilizar uma ferramenta fenomenal da linguagem python chamada de list comprehension (não vou nem tentar traduzir para não confundir a cabeça, mas seria algo como compressão de listas).

Basicamente, as list comprehensions permitem executar expressões relativamente complexas dentro dos elementos da lista e retornar instantaneamente uma nova lista com os resultados, vamos utilizar essa técnica para esconder as letras da palavra aleatória do jogo. Vamos ver o código primeiro, separado do restante do jogo, depois vamos tentar explicar o que está acontecendo da maneira mais simples possível:

```
palavra_underlines = [letra if letra in letras else "_" for letra in palavra_escolhida]
```

Sim, eu sei que em um primeiro momento parece uma coisa de outro mundo, mas nós vamos juntos tentar desvendar o que significa essa expressão que mais parece um palavrão.

Primeiro, o que essa expressão gigantesca significa: Retorne **letra** se a **letra** está dentro da lista **letras**, senão retorne "\_" para cada **letra** em **palavra\_escolhida**. Acredito que com essa tradução de python para português as coisas já tenham ficado um pouco mais esclarecidas, mas mesmo assim vamos quebrar essa expressão em pequenas partes para que você identifique o que foi utilizado:

1. Declara a variável **letra** que será retornada se;
2. **letra** está presente na lista **letras** (uso de if para verificar na lista);
3. Caso contrário, retorne underline "\_" (uso de else);
4. Repita 1 a 3 para cada **letra** dentro da str **palavra\_escolhida** (uso de for para repetição);
5. No final temos uma nova lista com underlines e letras;

Agora vamos exibir para o jogador essa nova lista com os underlines e as letras por ele já descobertas, vamos aproveitar para exibir também a quantidade de chances que ele tem para acertar.

```
palavras = {"arroz", "pato", "avestruz"} # Pool de palavras aleatórias
letras = "" # Caracteres informados pelo usuário, começa vazio.
max_erros = 3 # Número máximo de tentativas
erros = 0 # Erros cometidos pelo usuário, começa em zero.
palavra_escolhida = palavras.pop() # .pop retorna um valor aleatório do nosso set

while erros < max_erros:
    palavra_underlines = [letra if letra in letras else "_" for letra in palavra_escolhida]
    print("".join(palavra_underlines)) # Exibe a palavra com os underlines
    print("Vidas: ", max_erros - erros) # Exibe a quantidade de chances disponíveis
else:
    print("Que pena, todas as suas vidas acabaram :(")
```

Pronto, vamos solicitar ao jogador que digite um caractere, vamos verificar se ele digitou apenas um caractere e se este caractere está presente na palavra\_escolhida, caso positivo, vamos adicionar o caractere a lista de letras já digitadas, caso contrário, vamos adicionar um marcador na quantidade de letras erradas, e exibir uma pequena mensagem de erro.

```
palavras = {"arroz", "pato", "avestruz"} # Pool de palavras aleatórias
letras = "" # Caracteres informados pelo usuário, começa vazio.
max_erros = 3 # Número máximo de tentativas
erros = 0 # Erros cometidos pelo usuário, começa em zero.
palavra_escolhida = palavras.pop() # .pop retorna um valor aleatório do nosso set

while erros < max_erros:
    palavra_underlines = [letra if letra in letras else "_" for letra in palavra_escolhida]
    print("".join(palavra_underlines)) # Exibe a palavra com os underlines
    print("Vidas: ", max_erros - erros) # Exibe a quantidade de chances disponíveis

    entrada = input("Digite uma letra: ") # Jogador escolhe uma letra
    if len(entrada) == 1: # Vamos verificar se o jogador digitou apenas uma letra
        if entrada in palavra_escolhida: # Aqui verificamos se a letra está presente na palavra
```

```

        letras += entrada
    else:
        print("A letra escolhida não está presente na palavra. Você perdeu uma vida.")
        erros += 1
    else:
        print("Digite apenas uma letra de cada vez.")

else:
    print("Que pena, todas as suas vidas acabaram :)")

```

Ufa, finalmente estamos acabando o nosso joguinho, ainda falta criarmos algumas condições para validar se o jogador ganhou o jogo, vamos fazer isso utilizando mais um **if not**:

```

palavras = {"arroz", "pato", "avestruz"} # Pool de palavras aleatórias
letras = "" # Caracteres informados pelo usuário, começa vazio.
max_erros = 3 # Número máximo de tentativas
erros = 0 # Erros cometidos pelo usuário, começa em zero.
palavra_escolhida = palavras.pop() # .pop retorna um valor aleatório do nosso set

while erros < max_erros:
    palavra_underlines = [letra if letra in letras else "_" for letra in palavra_escolhida]

    if not "_" in palavra_underlines:
        print("Parabéns você venceu! A palavra é: ", "".join(palavra_underlines))
        break

    print("".join(palavra_underlines)) # Exibe a palavra com os underlines
    print("Vidas: ", max_erros - erros) # Exibe a quantidade de chances disponíveis

    entrada = input("Digite uma letra: ") # Jogador escolhe uma letra
    if len(entrada) == 1: # Vamos verificar se o jogador digitou apenas uma letra
        if entrada in palavra_escolhida: # Aqui verificamos se a letra está presente na palavra
            letras += entrada
        else:
            print("A letra escolhida não está presente na palavra. Você perdeu uma vida.")
            erros += 1
    else:
        print("Digite apenas uma letra de cada vez.")

else:
    print("Que pena, todas as suas vidas acabaram :)")

```

Finalmente concluímos o nosso jogo da forca! Construir pequenos projetos é muito importante no começo da jornada de aprendizado de uma nova linguagem de programação, e ajuda a consolidar os conhecimentos. Enfim, esperamos você na próxima aula 😊.