# O jogo "Adivinhar a palavra"

O jogo “Adivinhar a palavra” consiste em desvendar uma palavra (ou uma frase) letra a letra. É normalmente conhecido por "jogo da forca". Neste tutorial iremos apenas considerar uma palavra, para maior simplicidade.

O jogo começa por visualizar uma palavra, escolhida aleatoriamente. Esta palavra será visualizada por uma sequência de caracteres “\*”, de comprimento igual ao comprimento da palavra.

Por exemplo, se a palavra escolhida fosse "carro", o computador visualizaria "\*\*\*\*\*".

A cada jogada, o jogador indica uma letra e, no caso dessa letra existir na palavra, é visualizada a palavra com essa letra na posição (ou posições) certa(s). Caso não exista, o jogador perde uma tentativa e a letra é indicada na lista de letras “erradas”.

Se fosse indicada a letra “a”, seria visualizada a sequência "\*a\*\*\*".

Se o jogador conseguir adivinhar todas as letras que formam a palavra dentro do número de tentativas definidas ganha, caso contrário, perde.

# Inicialização do jogo

A inicialização do jogo será realizada através da seleção aleatória de uma palavra dentro de uma lista de palavras.

O jogo será implementado pela função jogo() que recebe dois parâmetros:

* palavras, uma lista de cadeias de caracteres (*strings*) com as palavras do jogo;
* tentativas, o número máximo de tentativas.

Depois de selecionar aleatoriamente a palavra, que iremos denominar chave, é necessário criar uma outra palavra a ser decifrada (com o mesmo número de letras mas preenchida com “\*”), para poder ser visualizada pelo jogador.

O código da função jogo() é o seguinte:

import random

def jogo(palavras, tentativas):

# seleção da chave da lista de palavras

chave = random.choice(palavras)

# palavra escondida

palavra = ""

for c in chave:

palavra = palavra + "\*"

Repare que o ciclo [for](https://www.w3schools.com/python/python_for_loops.asp) percorre com grande simplicidade a palavra chave, letra a letra.

|  |  |
| --- | --- |
| **Head with gears** | É importante referir que as cadeias de caracteres são imutáveis, ou seja, não podem ser alteradas. Desta forma, a variável palavra é construída criando cópias e **concatenando** (juntando) cadeias de caracteres, com o operador “+”.  A cada iteração, o espaço de memória que o identificador palavra referencia é distinto. |

Seguidamente, inicializam-se as variáveis que indicam o número da jogada e as letras erradas (uma cadeia de caracteres), antecedendo o ciclo de jogo.

# inicialização das variáveis

jogada = 1

letras\_erradas = ""

# Iterando através das jogadas – o ciclo de jogo

O ciclo de jogo é executado através de um ciclo condicional [while](https://www.w3schools.com/python/python_while_loops.asp) cuja condição avalia se o jogador ainda não adivinhou a palavra e se ainda tem tentativas para jogar.

while chave != palavra and tentativas > 0:

O algoritmo de cada jogada é o seguinte:

1. Imprimir a palavra a adivinhar e ler a letra do jogador.

print ("Jogada", jogada, ":", palavra)

print ("letras erradas: ", letras\_erradas)

letra = input ("Adivinhe a letra: ")

1. Verificar se a letra existe na palavra chave e nesse caso desvendar a palavra escondida. É criada uma nova cadeia de caracteres (npalavra). O índice i percorre tanto a palavra a desvendar, como a chave, de forma a verificar onde a letra existe.

# desvendar a palavra

npalavra = ""

for i in range(len(chave)):

if chave[i]. casefold() == letra. casefold():

npalavra += chave[i]

else:

npalavra += palavra[i]

|  |  |
| --- | --- |
| **Eye** | Ao comparar as letras utiliza-se o método [casefold](https://www.w3schools.com/python/ref_string_casefold.asp) das cadeias de caracteres para converter ambas as letras para minúsculas, evitando as situações em que a palavra a decifrar tem uma maiúscula e o jogador insere uma minúscula (ex. A primeira letra de um nome).  Repare também que a operação de concatenação de cadeias de caracteres pode ser simplificada com o operador += . |

1. No caso de o jogador não acertar na letra (caso em que a nova palavra se mantém igual à anterior) diminui o número de tentativas e acrescenta essa letra às letras erradas.

if palavra == npalavra:

letras\_erradas += letra

tentativas -= 1

1. Finalmente atualizam-se as variáveis jogada e palavra.

jogada += 1

palavra = npalavra

Após o fim do jogo é verificado se o jogador ganhou ou perdeu...

if tentativas == 0:

print ("Perdeu. Já não tem mais tentativas... Era a palavra: ", chave)

elif chave == palavra:

print ("Parabéns")

|  |  |
| --- | --- |
| **Balloon animal** | Teste o jogo com a seguida lista:  cidades = ["Porto", "Lisboa", "Braga", "Faro", "Bragança"]  jogo(cidades,5).  Experimente depois com várias listas de palavras de diferentes temas. |

# Desafio final

Faça uma variação deste jogo em que o jogador, em vez de apenas ter que adivinhar uma palavra, adivinhe uma frase. Neste caso a visualização deve mostrar os espaços e a pontuação.

Sugestão: Relembra-se a existência do método [isalpha](https://www.w3schools.com/python/ref_string_isalpha.asp) das cadeias de caracteres para verificar se um caracter é uma letra.

|  |  |
| --- | --- |
| **Share with person** | Partilhe o **seu** jogo! |