# Game "Guess the Word"

The game "Guess the word" consists of unraveling a word (or phrase) letter by letter. It is commonly known as the "hangman game". In this tutorial we will only consider one word, for simplicity.

The game starts by visualizing a word, chosen at random. This word will be displayed by a sequence of characters "\*", of length equal to the length of the word.

For example, if the chosen word was "car", the computer would display "\*\*\*".

With each move, the player indicates a letter and, if that letter exists in the word, the word with that letter is displayed in the right position (or positions). If not, the player loses an attempt and the letter is indicated in the list of “wrong” letters.

If the letter “a” was indicated, the sequence "\*a\*" would be displayed.

If the player can guess all the letters that make up the word within the number of defined attempts, s/he wins, otherwise s/he loses.

# Game setup

The game will be started by randomly selecting a word from a list of words.

The game will be implemented by the jogo function that receives two parameters:

* palavras, a list of strings with the words of the game;
* tentativas, the maximum number of attempts.

After randomly selecting the word, which we will call chave, it is necessary to create another word to be deciphered (with the same number of letters but filled with “\*”), in order to be visualized by the player.

Function jogo() ias as follows:

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 + "\*"

Notice that the [for](https://www.w3schools.com/python/python_for_loops.asp) loop traverses with great simplicity the word chave, character by character.

|  |  |
| --- | --- |
| **Head with gears** | It is important to note that strings are immutable, that is, they cannot be changed. In this way, the variable palavra is constructed by creating copies and concatenating (joining) character strings, with the “+” operator.  At each iteration, the memory space that the identifier palavra references is distinct. |

Then, the variables indicating the number of the move and the wrong letters (a string of characters) are initialized, preceding the game cycle.

# inicialização das variáveis

jogada = 1

letras\_erradas = ""

# Iterating through moves – the game loop

The game loop is executed through a [while](https://www.w3schools.com/python/python_while_loops.asp) conditional loop whose condition evaluates whether the player has not yet guessed the word and whether s/he still has more attempts to play.

while chave != palavra and tentativas > 0:

The algorithm for each move is as follows:

1. Print the word to guess and read the player's guess.

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

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

letra = input ("Adivinhe a letra: ")

1. Check if the letter exists in the keyword and in that case unveil the hidden word. A new string is created (npalavra). The index i goes through both the word to be unveiled and the key, in order to verify where the letter exists.

# desvendar a palavra

npalavra = ""

for i in range(len(chave)):

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

npalavra += chave[i]

else:

npalavra += palavra[i]

|  |  |
| --- | --- |
| **Eye** | When comparing letters, method [casefold](https://www.w3schools.com/python/ref_string_casefold.asp) is used on strings to convert both letters to lowercase, avoiding situations where the word to be deciphered has a capital letter and the player inserts a lower case (e.g. the first letter of a name).  Also note that the string concatenation operation can be simplified with the + = operator. |

1. If the player does not guesses the letter (in which case the new word remains the same as the previous one) the number of attempts is reduced and the letter is added to the wrong letters.

if palavra == npalavra:

letras\_erradas += letra

tentativas -= 1

1. Finally, the variables jogada and palavra are updated.

jogada += 1

palavra = npalavra

After the game is over, it is checked whether the player won or lost...

if tentativas == 0:

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

elif chave == palavra:

print ("Parabéns")

|  |  |
| --- | --- |
| **Balloon animal** | Test the game with the following list:  cidades = ["Porto", "Lisboa", "Braga", "Faro", "Bragança"]  jogo(cidades,5).  Experiment later with multiple word lists for different topics. |

# Final challenge

Make a variation of this game in which the player, instead of just having to guess a word, has to guess a sentence. In this case the visualization should show the spaces and the punctuation.

Suggestion: Remember the existence of the method [isalpha](https://www.w3schools.com/python/ref_string_isalpha.asp) of strings to check if a character is a letter.

|  |  |
| --- | --- |
| **Share with person** | **Share** your game! |