HANGMAN

What you will need to do

Here you will create a **game of hangman.**

The rules are:

1. There is a word the player needs to guess
2. They will guess a letter each turn
3. If they guess it correctly, it is shown where in the word
4. If they guess wrong they lose a life
5. They need to guess all of the letters before their lives are up

The problem has been **decomposed** for you so you just need to complete each step to make the entire game.

Use www.usingpython.com to help you if you get stuck.

Make sure that you **test each step works before moving on.**

What skills you will cover in this

In this you will be using

- Variables
- If / Else
- While loops
- String Manipulation

First store your words in a list

At the top of your file store a **list of possible words to be guessed.** These will be used by the computer to pick a word that the **player will be trying to figure out.**

Make a **list** containing **ten words** which the user might guess.

```
words_to_guess = ["Apple", "Bear", "Cat", "Dog"]
```

Be careful! Make sure to use words where **a letter is only used once.**

In the list above **apple** is not ok as it has two **p's.**

This reason for this will be explained later.

Get python to pick a random word

As well as guessing **random numbers** python can also **pick an item out of a list at random.**

Use this to get python to **pick out a word at random** for the player to guess. You will need to **store this** so it can be checked as the game goes on.

```
player_word = random.choice(words_to_guess)
```

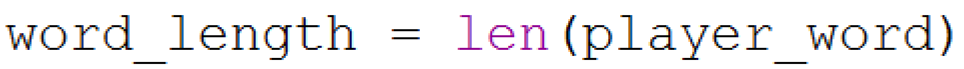
Now make a new word of the same length, but with ???s

Now you'll make the **word you display to the user.** We need to display the word as **question marks** so that they know **how many letters the word is.**

Create a new **string** which is the same **length** as the word but all question marks.

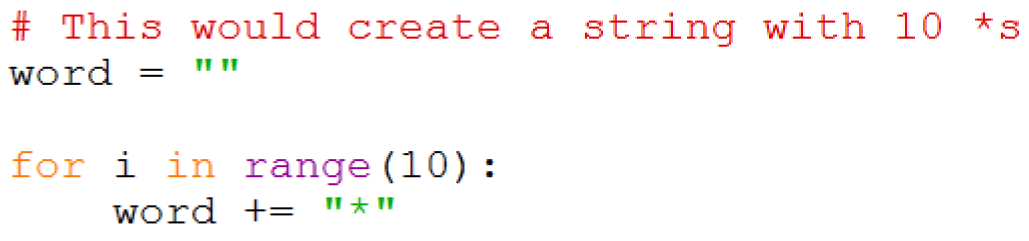
E.g. Hello would become ?????

You can find a string's **length** using



```
word_length = len(player_word)
```

You can make a string of ? **with a certain length** using a **for loop**



```
# This would create a string with 10 *s
word = ""

for i in range(10):
    word += "*"
```

Use these to **help you make the string of the right length.**

Get the user to input a letter

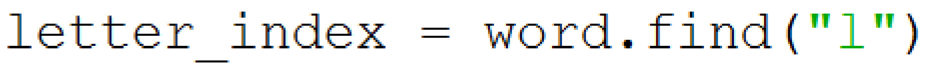
Now have the user **input their guess.** Make sure you assign it to a variable to be used later.

Check if that letter is in the word to be guessed

Now it gets tougher. Using **string manipulation** you'll need to check **if the letter is in the word to be guessed** and **where.**

Remember! Like with lists string **indexes** start at **0** not **1.**

This code will **return the index value where it's first found in the word.**



```
letter_index = word.find("l")
```

The problem is this will **only find the first instance of the letter.**

This is why you should **only be guessing words where each letter is used once.**

If it is, change a ? in the word shown to the user to show they guessed it

Now you know **which letter** they've guessed correctly you need to **change this in the word displayed to the user.**

You have the **index** so change the **word displayed** using **string manipulation.**

You can replace a letter using its index. It gets **quite complex** as you need to turn it into a **list** then back into a **string.**

This code below will **do this for you.** You will need to **change the variable names** to match your own:

```
# Here are the variables you will need
letter_index = 1        # Index you want to change
letter_changed = "a"    # What you want to change it to
word_changed = "?????"  # Word to be changed

# Copy and use this code exactly as it is
# This will turn it into a list and recombine
# it for you
word_list = list(word_changed)
word_list[letter_index] = letter_changed
word_changed = "".join(word_list)
```

Tell them they've won if they guess the entire word

If the **entire word** has been guessed then **tell them they've won.**

What would be the **simplest way** to check if they've won?

As a clue **you can check the word you've displayed to the user.**

Add lives, tell them they've lost if they don't get it in time

Now add **lives** to your game. This can be done by just displaying "You have x lives left!" each turn.

There is **no set number of lives** so it's up to you.

Make sure that they **lose a life** only if they **guess incorrectly.**

You'll need to use a **loop** to let the player **keep guessing until they run out of lives.**

Extras!

- Add timing to the system so it paces the game rather than writing everything at once
- Add a choice to guess the entire word
- Add validation so they can only input one letter
- Currently you can guess the same letter more than once. Make it so they can only guess a letter one time.
- Add it so they can use letters with the same word more than once e.g. Hello
- Add some ASCII art to show the hangman instead of art
- Use file handling to store the words in a text file and import them rather than store them as a list in the program