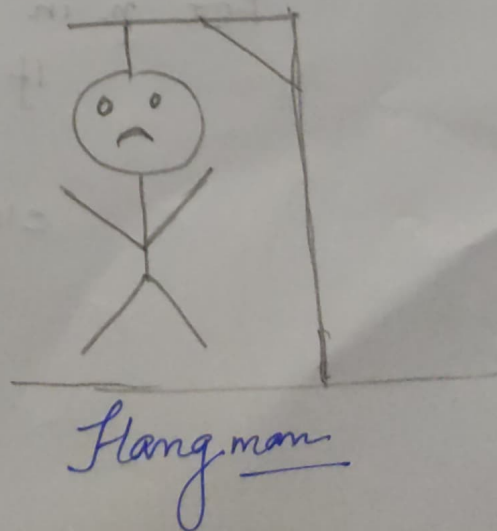
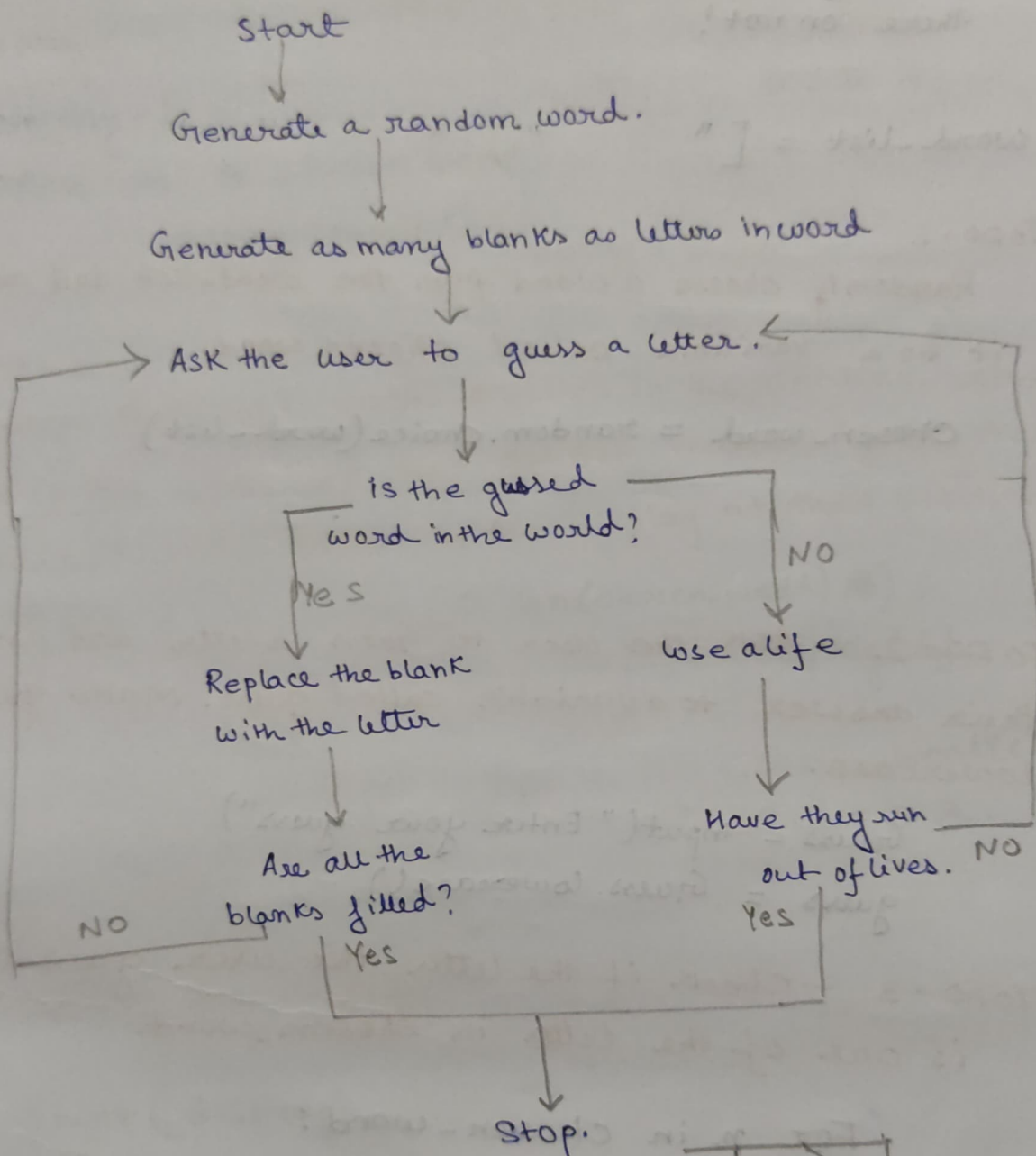


Day 7

HANGMAN PROJECT.

→ Flow chart Programming.



Let's break so down the project into small challenges.

Challenge-1.

Picking a random words and check if the letter is there or not!

```
word_list = ["", "", "", "", ""]
```

TODO-1.

Randomly choose a word from the word-list and assign it to a variable called chosen word.

```
chosen_word = random.choice(word_list)
```

TODO-2 - ASK the user to Guess a letter and assign their answer to a variable called guess. Make guess lowercase.

```
Guess = input("Enter your guess")  
guess = Guess.lowercase()
```

TODO-3 - Check if the letter the user guessed (guess) is one of the letter in chosen_word.

```
for n in chosen_word:
```

```
    if n == guess:
```

```
        print("Right")
```

```
    else:  
        print("Wrong")
```

Challenge - 2

TO DO - 1.

Create an empty list called display.

For each letter in the chosen word add a "_" to display.

```
display = [ ]
```

For n in chosen word:

```
display.append("_")
```

Apple
0 1 2 3

TODO - 2

Loop ~~at~~ through each position in the chosen-word:
if the letter at that position matches 'guess' then
reveal that letter in the display at that position.

```
for i in range(0, len(chosen-word)):
    if chosen-word[i] == display[i]:
        for n in guess:
            if guess n == chosen-word[i]:
                display[i] = n.
```

TO DO - 3.

Display the display.

```
print(display).
```


Challenge 3.

TODO-1.

Use a While loop to let the user guess again. The loop should only stop once the user has guessed all the letters in the chosen word and 'display' has no more blank (" _ "). Then ~~so~~ you can tell the user they've won.

end-of-game = False.

While ~~end-of-game~~: not end-of-game:

} Previous code block.

if " _ " not in display:

end-of-game = True

Now lets punish!!

Challenge-4.

TODO-1.

Create a Variable called 'lives' to keep track of the number of lives left.

lives = 6

TODO-2

If guess is not a letter in chosen_word,

Then reduce lives by 1.

If lives goes down to 0 then the game should stop and it should print "you loose".

```
if guess not in chosen_word:
```

```
    lives -= 1.
```

```
    if lives == 0:
```

```
        end_of_game = True
```

```
        print("You loose")
```

To DO #3.

Print the ASCII ART from 'Stages' the corresponds to the current number of 'lives' the user has remaining.

```
print(stages[lives])
```

Challenge 5.

TODO-01.

Update the word list to use the 'word-list' from hangman-words.py.

```
from hangman import word-list
```

```
word-list = random.choice(word-list)
```

TODO-02

Import the stages from hangman art.py ~~and~~.

```
from hangman art import stages,
```

TODO #3.

Import the logo from hangman and print it at the start of the game.

```
from hangman-art import stages, logo
```

```
print(logo)
```

TODO #4.

If the user has entered a letter they've already guessed, print the letter and let them know.

```
print("You guessed right")
```

TODO #5

If the user has entered a letter they've already guessed, print the letter and let them know

```
print("Your guess is wrong")
```

Now, let's clear the old guesses from the console.

```
from replit import clear.
```

```
clear()
```

← Place it after the print statement for asking the guess!!

→ Day 07 Completed! ←