HANGMAN GAME IN PYTHON

A Simple Console-Based Word Guessing Game

PROJECT OVERVIEW:

- *A classic Hangman game implemented in Python.*
- *Text-based, using basic programming concepts.*
- *Player guesses one letter at a time to find the hidden word.*
- *Limited to 6 wrong attempts.*

TOOLS & TECHNOLOGIES:

- *Language: Python*
- *Concepts Used:*
  - *random module*
  - *while loops*
  - *if-else statements*
  - *Strings & Lists*
  - *Console I/O*

Programiz Python Online Compiler

main.py

Share Run

```python
import random

# Predefined list of words
word_list = ["apple", "house", "robot", "pizza", "snake"]

# Randomly select a word
secret_word = random.choice(word_list)
guessed_letters = []
incorrect_guesses = 0
max_attempts = 6

# Create a display version of the word with underscores
display_word = ["_"] * len(secret_word)

print("Welcome to Hangman!")
print("Guess the word, one letter at a time.")
print(f"You have {max_attempts} incorrect guesses allowed.\n")

while incorrect_guesses < max_attempts and "_" in display_word:
    print("Word: " + " ".join(display_word))
    guess = input("Enter a letter: ").lower()

    if len(guess) != 1 or not guess.isalpha():
        print("Please enter a single alphabet letter.\n")
```

main.py

Share Run

```python
    if len(guess) != 1 or not guess.isalpha():
        print("Please enter a single alphabet letter.\n")
        continue

    if guess in guessed_letters:
        print("You already guessed that letter. Try another.\n")
        continue

    guessed_letters.append(guess)

    if guess in secret_word:
        print("correct guess!\n")
        for idx, letter in enumerate(secret_word):
            if letter == guess:
                display_word[idx] = guess
    else:
        incorrect_guesses += 1
        print(f"wrong guess! Attempts left: {max_attempts - incorrect_guesses}\n")

# End of game result
if "_" not in display_word:
    print("Congratulations! You guessed the word:", secret_word)
else:
    print("You ran out of attempts. The word was:", secret_word)
```

main.py

Share Run

```python
    if len(guess) != 1 or not guess.isalpha():
        print("Please enter a single alphabet letter.\n")
        continue

    if guess in guessed_letters:
        print("You already guessed that letter. Try another.\n")
        continue

    guessed_letters.append(guess)

    if guess in secret_word:
        print("correct guess!\n")
        for idx, letter in enumerate(secret_word):
            if letter == guess:
                display_word[idx] = guess
    else:
        incorrect_guesses += 1
        print(f"wrong guess! Attempts left: {max_attempts - incorrect_guesses}\n")

# End of game result
if "_" not in display_word:
    print("Congratulations! You guessed the word:", secret_word)
else:
    print("You ran out of attempts. The word was:", secret_word)
```

Output

Clear

```
Word: _ _ _ _ _
Enter a letter: a
wrong guess! Attempts left: 5

Word: _ _ _ _ _
Enter a letter: r
correct guess!

Word: r _ _ _ _
Enter a letter: o
correct guess!

Word: r o _ o _
Enter a letter: b
correct guess!

Word: r o b o _
Enter a letter: t
correct guess!

Congratulations! You guessed the word: robot

=== Code Execution Successful ===
```

GAME FLOW:

1. Choose a random word from a predefined list.
2. Display underscores for each letter in the word.
3. User guesses one letter at a time.
4. Reveal letters if correct.
5. Reduce attempts if incorrect.
6. Game ends when the word is guessed or attempts run out.

CHALLENGES FACED:

- *Validating user input.*
- *Avoiding repeated guesses.*
- *Keeping track of correct vs incorrect letters.*
- *Clear and simple UI in the console.*

CONCLUSION:

- *A fun and educational project.*
- *Reinforced core Python concepts.*
- *Easy to extend (add hints, difficulty levels, etc.)*
- *Great for beginners learning loops and logic.*

THANK YOU