```
from string import ascii_lowercase
from words import get_random_word


def get_num_attempts():
    """Get user-inputted number of incorrect attempts for the game."""
    while True:
        num_attempts = input(
            'How many incorrect attempts do you want? [1-25] ')
        try:
            num_attempts = int(num_attempts)
            if 1 <= num_attempts <= 25:
                return num_attempts
            else:
                print('{0} is not between 1 and 25'.format(num_attempts))
        except ValueError:
            print('{0} is not an integer between 1 and 25'.format(
                num_attempts))


def get_min_word_length():
    """Get user-inputted minimum word length for the game."""
    while True:
        min_word_length = input(
            'What minimum word length do you want? [4-16] ')
        try:
            min_word_length = int(min_word_length)
            if 4 <= min_word_length <= 16:
                return min_word_length
            else:
                print('{0} is not between 4 and 16'.format(min_word_length))
        except ValueError:
            print('{0} is not an integer between 4 and 16'.format(
                min_word_length))


def get_display_word(word, idxs):
    """Get the word suitable for display."""
    if len(word) != len(idxs):
        raise ValueError('Word length and indices length are not the same')
    displayed_word = ''.join(
        [letter if idxs[i] else '*' for i, letter in enumerate(word)])
    return displayed_word.strip()


def get_next_letter(remaining_letters):
    """Get the user-inputted next letter."""
    if len(remaining_letters) == 0:
        raise ValueError('There are no remaining letters')
    while True:
        next_letter = input('Choose the next letter: ').lower()
        if len(next_letter) != 1:
            print('{0} is not a single character'.format(next_letter))
        elif next_letter not in ascii_lowercase:
            print('{0} is not a letter'.format(next_letter))
        elif next_letter not in remaining_letters:
            print('{0} has been guessed before'.format(next_letter))
        else:
            remaining_letters.remove(next_letter)
            return next_letter


def play_hangman():
    """Play a game of hangman.
```

```
    At the end of the game, returns if the player wants to retry.
    """
    # Let player specify difficulty
    print('Starting a game of Hangman...')
    attempts_remaining = get_num_attempts()
    min_word_length = get_min_word_length()

    # Randomly select a word
    print('Selecting a word...')
    word = get_random_word(min_word_length)
    print()

    # Initialize game state variables
    idxs = [letter not in ascii_lowercase for letter in word]
    remaining_letters = set(ascii_lowercase)
    wrong_letters = []
    word_solved = False

    # Main game loop
    while attempts_remaining > 0 and not word_solved:
        # Print current game state
        print('Word: {0}'.format(get_display_word(word, idxs)))
        print('Attempts Remaining: {0}'.format(attempts_remaining))
        print('Previous Guesses: {0}'.format(' '.join(wrong_letters)))

        # Get player's next letter guess
        next_letter = get_next_letter(remaining_letters)

        # Check if letter guess is in word
        if next_letter in word:
            # Guessed correctly
            print('{0} is in the word!'.format(next_letter))

            # Reveal matching letters
            for i in range(len(word)):
                if word[i] == next_letter:
                    idxs[i] = True
        else:
            # Guessed incorrectly
            print('{0} is NOT in the word!'.format(next_letter))

            # Decrement num of attempts left and append guess to wrong guesses
            attempts_remaining -= 1
            wrong_letters.append(next_letter)

        # Check if word is completely solved
        if False not in idxs:
            word_solved = True
        print()

    # The game is over: reveal the word
    print('The word is {0}'.format(word))

    # Notify player of victory or defeat
    if word_solved:
        print('Congratulations! You won!')
    else:
        print('Try again next time!')

    # Ask player if he/she wants to try again
    try_again = input('Would you like to try again? [y/Y] ')
    return try_again.lower() == 'y'


if __name__ == '__main__':
    while play_hangman():
        print()
```