1.
```
Def print (board):
input
  For I in range(0,9,3):
     Print("|".join(board[i:i+3]))
   If I < 6:
      Print("_"*9)
```

print_board() The function : it takes a list as input and print it to the console in a grid format. the function iterates over the list in group of three and print each group on a separate line.if the list is not divisible by three the function will print the Remaining element on the last line .

For i in range (0,9,3): here for loop is used over the range of number from 0 to 8 ,in increment of 3.for each iteration ,the code print the contents of the board list .

Print ("|". join (board[i:i+3])) : it print the element of the current row of the matrix , separated by the character "|" . the **join()** function is used to combine the element of the list **board [i : i+3]** into a single string ,with the character "|" inserted between each element.

if i <6 : this is used to checks if the current row is less than the sixth row of the matrix . if it is, then the code print a line of **nine hyphens(_)** to separate the current row from the next row .

print ("_"*9): this is used to print "_" nine times

2.
```
def is _board_full (board):
    return ' ' not in board
```

is_board_full(): it takes a board as an argument and return **TRUE** if the board is full and FALSE otherwise

*The board represented as a list of strings , where each string represent a row on the board. The function check if the **space (' ') character** is present in any of the row . if it is, then the board is **not full** and the function return **FALSE** . if the space character is not present in any of the row ,then the board is **FULL** and the function return **TRUE** .

3. **#FUNCTION TO CHECK IF SOMEONE HAS OWN**

Winning_combination : a list of all possible winning combination
Board: A list representing the current state of the tic tac toe board

*The function work by iterating over the **winning_combination** list . for each winning combination, ,the function checks if all the squares in the combination are **occupied by the same player**. If so, the function return **TRUE** . **Otherwise** ,the function **return FALSE**

4. **# MAIN GAME FUNCTION**

Board =[' ' for _ in range(9)]: here a list is initialized called board with 9 spaces ,representing the 9 space on the board .

Current_player = 'X': set the current player to 'X' symbol will make the first move

```
While True :
```

The loop begins by printing the current state of

Print_board (board) the board . Then,it print a message to the
Print("\nPlayer {current_player}'s turn player whose turn it is , telling them to take

their turn. The player then take their turn,and the loop repeat . The loop will repeat until the current player wins the game , or until the game is a draw .

move to which place?(1-9):") if the current player is player1 then it ask the user to enter a no. between 1-9

move =int(input())-1 this line means , it take an input from the user and convert it to an integer . The code then subtract 1 from the input , because the board position are numbered from 0-8 , but the input from the user is numbered from 1-9

if board [move]==' ': it checks if the position that the user entered is empty. If it is, the code then executes the following code block :
board[move]=current_player this line of code places the current player's symbol on the board at the position that the user entered

* this means the code first take an input from the user ,which is the position where the user wants to make their move . The code then check if the position is empty . If it is , then place the current player symbol on the board

If check_winner(board,current_player) here it takes board and current player as an input and return **TRUE** if the current player won and **FALSE** otherwise

Print_board(board) this function print the current state of the game board

Print (f"player{current_player} wins!") it will print the message player{current_player} wins! The current_player placeholder will be replaced with the name of the current player

Break this line of code breaks out of the loop

If is_board_full(board): this line checks if the board is full. If it is, the code inside the if Statement will be executed.

Print_board(board) this line print the board

Print("it's a tie!") it print the message it's a tie

Break this line break the function

* It will print the board and then print "it's a tie". The function then break which means it exits

#SWITCH PLAYER

Current_player = '0' if current_player == 'X' this code first check if the current player is 'X'.

Else 'X' if it is, then the code set the current player to 'O'.

otherwise the code set the current player to 'X'. It keep the track of which player's turn it is