**PROJECT Console Programming with Python - Tic Tac Toe Game**

**Objective** To use Python to demonstrate a game - based Console Application.

***PROJECT DESCRIPTION***

Create and code a Python application for the following scenario:

a TIC TAC TOE game

Follow the steps below to design and create this game application.

Wishing you a successful gameplay!

***Information About This Project***

The initial game screen consists of nine sequentially numbered slots, placed in a grid - like display.

**[ Initial Game Screen ]**

|  |  |  |
| --- | --- | --- |
| **1** | **2** | **3** |
| **4** | **5** | **6** |
| **7** | **8** | **9** |

**[ The Sample Game Play ]**

First Move: The X's Player begins the game and enters this slot number → 1

Second Move: To counteract, the O's Player then enters this slot number → 3

Third Move: The X's Player strikes again and enters this slot number → 9

Fourth Move: To counteract, the O's Player then enters this slot number → 7

Fifth Move: The X's Player now moves and enters this slot number to win → 5

**[ The " X " Player Wins! ]**

|  |  |  |
| --- | --- | --- |
| **X** | **2** | **0** |
| **4** | **X** | **6** |
| **O** | **8** | **X** |

***Steps to Complete This Project***

**STEP 1**  **Open a Web Browser**

Launch a Web Browser on your computer, such as Microsoft Edge, MAC Safari, Google Chrome, Firefox, Mozilla, etc.

**STEP 2**  **Travel to a Web Link**

Navigate to the Web site that is given by the link below.

[Online Python Compiler - online editor (onlinegdb.com)](https://www.onlinegdb.com/online_python_compiler)

**STEP 3**  **Observe the Initial Code Statements**

At the above Web site, view the given test code that is provided.

'''

Welcome to GDB Online.

GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,

C#, OCaml, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.

Code, Compile, Run and Debug online from anywhere in world.

'''

print ('Hello World')

**STEP 4**  **Copy the Program Code**

Highlight and delete all the code statements that are initially provided at the above Web site.

Then, copy all the program code statements that are given in **Figure 1** , which follows.

Be sure to copy just the code statements and to include all the statements.

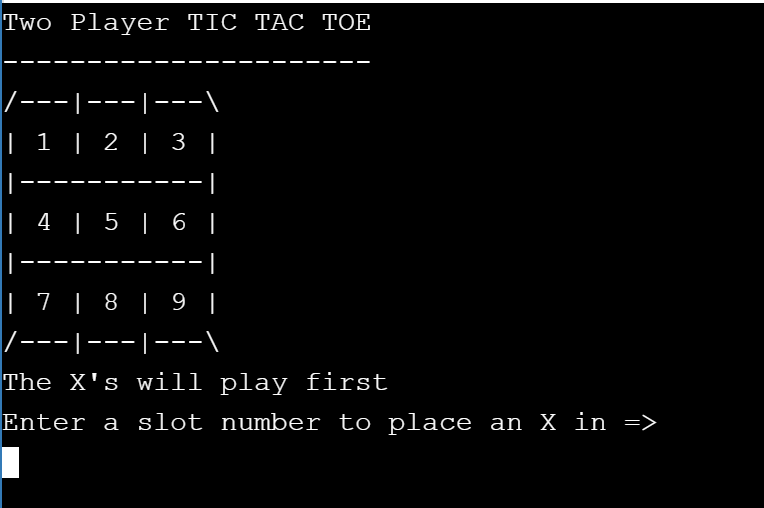
**Figure 1 Program Code: TIC TAC TOE**

|  |
| --- |
| **board = [" " for \_ in range(9)]**  **turn = "X"**  **winner = None**  **def populateEmptyBoard():**  **global board**  **board = [" " for \_ in range(9)]**  **def printBoard():**  **print("----------------------")**  **print("|", board[0], "|", board[1], "|", board[2], "|")**  **print("----------------------")**  **print("|", board[3], "|", board[4], "|", board[5], "|")**  **print("----------------------")**  **print("|", board[6], "|", board[7], "|", board[8], "|")**  **print("----------------------")**  **def checkWinner():**  **global winner**  **# Check rows**  **for i in range(0, 9, 3):**  **if board[i] == board[i+1] == board[i+2] != " ":**  **winner = board[i]**  **return**  **# Check columns**  **for i in range(3):**  **if board[i] == board[i+3] == board[i+6] != " ":**  **winner = board[i]**  **return**  **# Check diagonals**  **if board[0] == board[4] == board[8] != " ":**  **winner = board[0]**  **return**  **if board[2] == board[4] == board[6] != " ":**  **winner = board[2]**  **return**  **# Check for draw**  **if " " not in board:**  **winner = "draw"**  **def playGame():**  **global turn**  **print("Two Player TIC TAC TOE")**  **print("----------------------")**  **printBoard()**  **print("The X's will play first")**  **print("Enter a slot number to place an X in => ")**  **while winner == None:**  **try:**  **numInput = int(input())**  **if not (numInput > 0 and numInput <= 9):**  **print("Invalid Input! Re-enter the slot number:")**  **continue**  **except ValueError:**  **print("Invalid Input! Re-enter the slot number:")**  **continue**  **if board[numInput-1] == " ":**  **board[numInput-1] = turn**  **printBoard()**  **checkWinner()**  **if winner:**  **break**  **if turn == "X":**  **turn = "O"**  **else:**  **turn = "X"**  **else:**  **print("Slot is already taken! Re-enter a slot number:")**  **if winner == "draw":**  **print("The game is a draw! Thanks for playing.")**  **else:**  **print("Congratulations! " + winner + " has won! Thanks for playing.")**  **populateEmptyBoard()**  **playGame()** |

After you have copied the program code into that online compiler, click the   
[ Run ] button there to execute the program.



An output screen such as shown below will then appear.



A sample screen snapshot of the game play is given below.

**STEP 6 Play the Game**

After you compile your program, run the program with multiple executions using each of these three scenarios.

A sample run is shown in **Figure 2** , which follows.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Scenario One**

player X wins the game

**Scenario Two**

player O wins the game

**Scenario Three**

the game ends in a " draw " or tie

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Figure 2 Program Output: TIC TAC TOE**

Two Player TIC TAC TOE

----------------------

/---|---|---\

| 1 | 2 | 3 |

|-----------|

| 4 | 5 | 6 |

|-----------|

| 7 | 8 | 9 |

/---|---|---\

The X's will play first

Enter a slot number to place an X in => 1

O's turn; enter a slot number to place O in: 2

X's turn; enter a slot number to place X in: 5

O's turn; enter a slot number to place O in: 9

X's turn; enter a slot number to place X in: 6

O's turn; enter a slot number to place O in: 7

X's turn; enter a slot number to place X in: 4

/---|---|---\

| X | O | 3 |

|-----------|

| X | X | X |

|-----------|

| O | 8 | O |

/---|---|---\

Congratulations! X's has won! Thanks for playing.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**STEP 7 For Further Exploration**

For further discussion, here are some intriguing questions for pondering that concern this application.

**(1)** Consider thevarious categories of electronic and non - electronic games.

Howwould you classify a TIC TAC TOE game? Explain your answer.

**(2)** What are the allowed player moves for a TIC TAC TOE game?

**(3)** What is the minimum number of moves that a player can make to win at TIC TAC TOE ? What are the maximum number of moves that can be made by an individual player?

**(4)** Let us now think in three - dimensions!Is it possible to construct a 3D version from the usual TIC TAC TOE game? Explain your answer.

**(5)** Consider a game of chess and consider a game of hearts for playing cards.  
 How do the player moves of chess and hearts differ from a TIC TAC TOE game?

Why is a TIC TAC TOE game referred to as a zero - sum game?