Game Design

Tic Tac Toe

Mac Cross

Complete the Following

Requirements

1. Decide on the elements for a game.
2. List five online safety rules that you use while using the internet.
3. Create your game
4. Teach an adult or another scout how to play the game.

1. Elements for a game

- **Narrative or Story**: what is your game about?
- **Goals and Mechanics**: how the game works and what it takes to win.
- **Aesthetics**: what your game will look like
- **Medium**: the materials you need to make (e.g., dice, cards, computer)

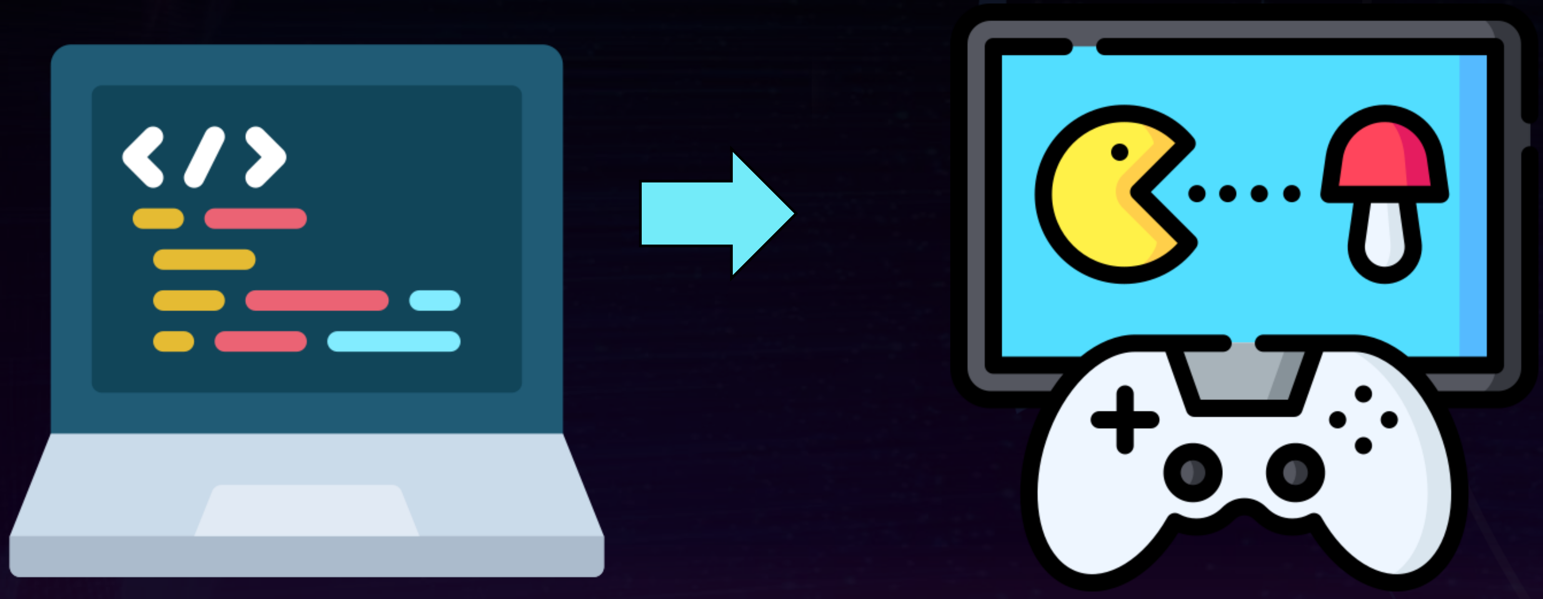
2. Online Safety

5 ways to stay safe online

1. I will tell my trusted adult if anything makes me feel sad, scared, confused, or uncomfortable.
2. I will ask my trusted adult before posting photos or sharing information like my name, address, current location, or phone number.
3. I won’t meet face-to-face with anyone I meet in the digital world.
4. I will respect the online property of others.
5. I will always use good etiquette and not be rude or mean online.

3. Make a Game

How are video games made?



Write Code

Play Game

Computer Languages

How do you write this “code”? What languages are they written in?

Assembly

C

Python, C++, Java

Zelda's Computer Languages

The Legend of Zelda (1986)

Assembly

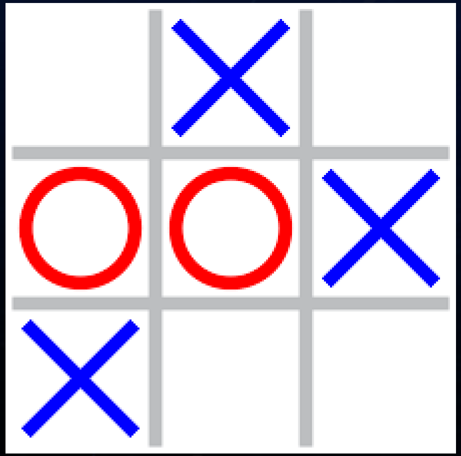
The Legend of Zelda: Ocarina of Time (1998)

C

The Legend of Zelda: Breath of the Wild (2017)

C++
&
Havok Physics Engine

Easy Game



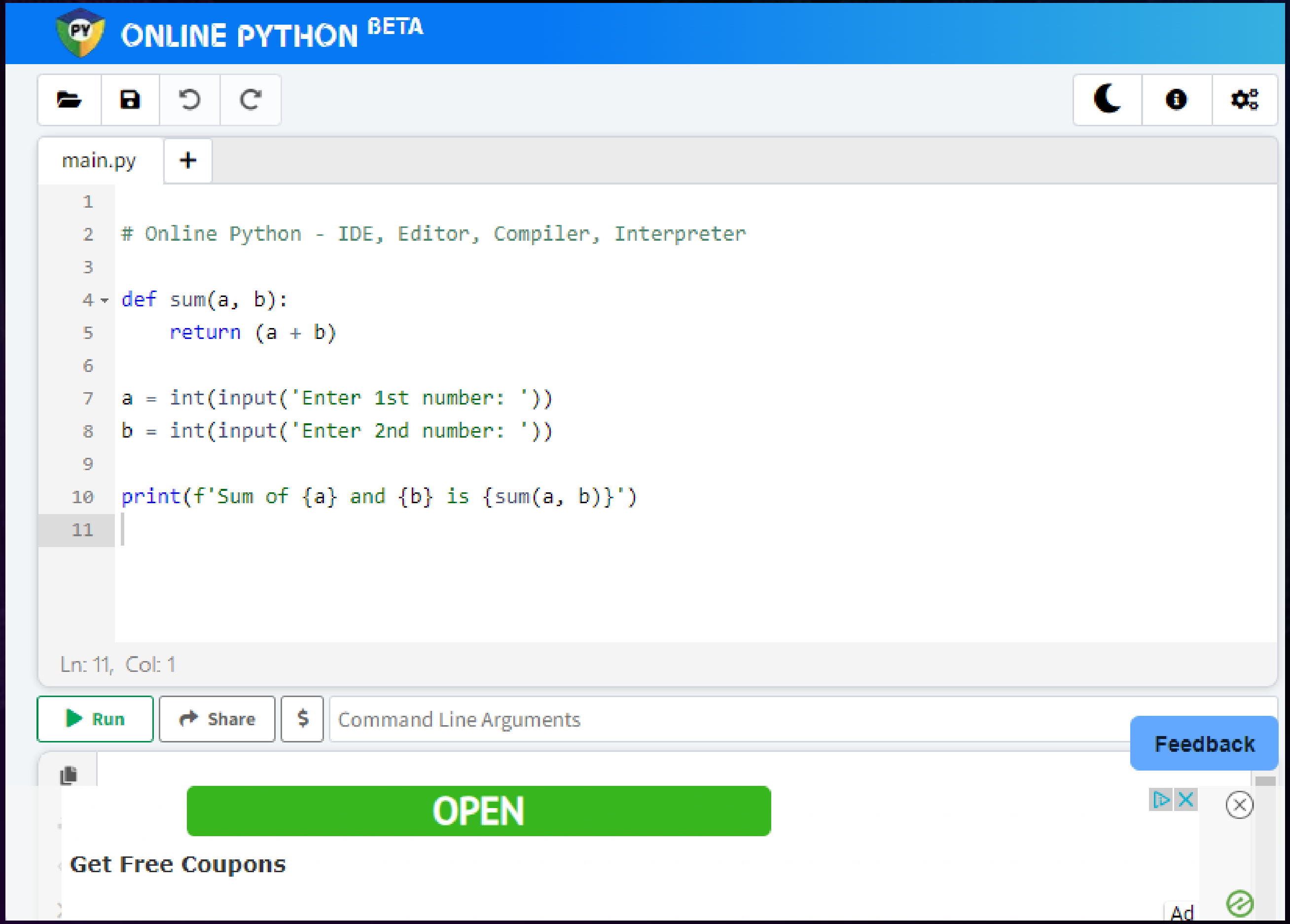
Tic Tac Toe with Python

- Python is an easy language to learn
- Tic Tac Toe is a simple game

Step 1: go to website

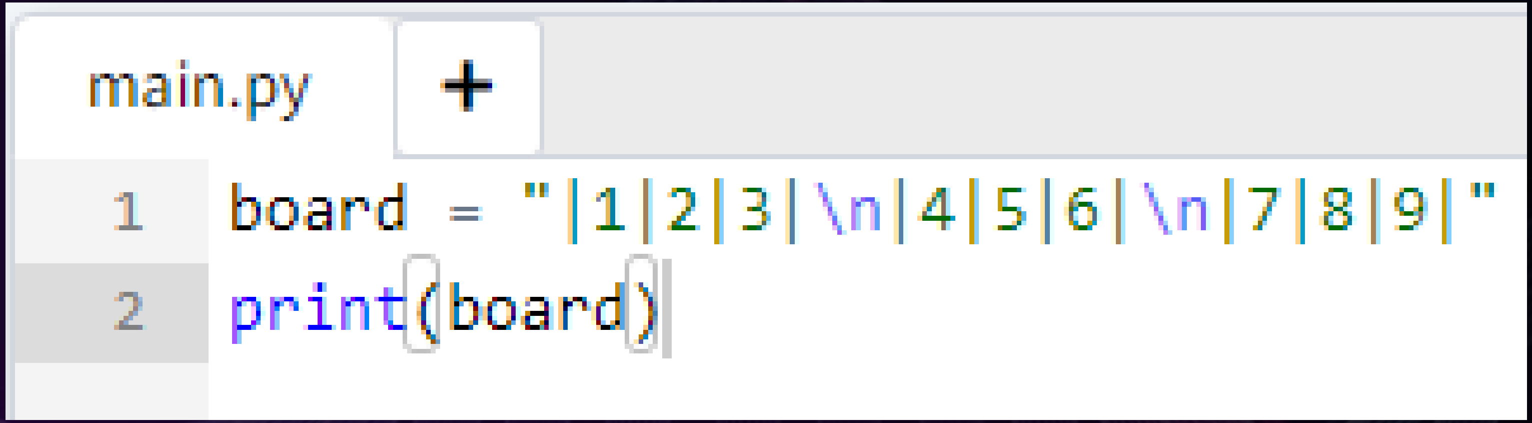
Go to https://www.online-python.com/

- It should look like this:

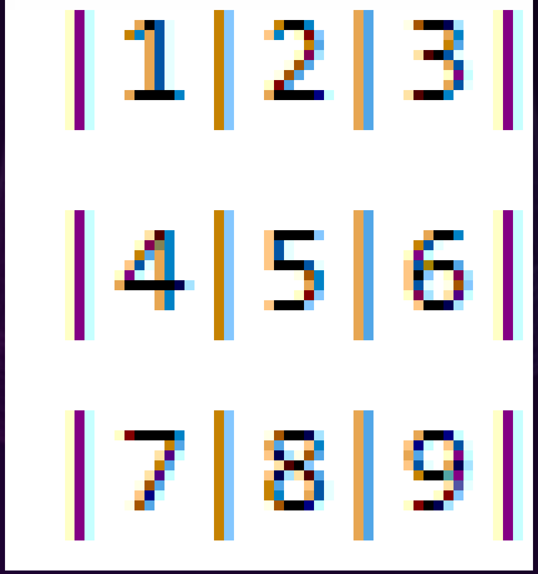


```python

# Online Python - IDE, Editor, Compiler, Interpreter

def sum(a, b):
    return (a + b)

a = int(input('Enter 1st number: '))
b = int(input('Enter 2nd number: '))

print(f'Sum of {a} and {b} is {sum(a, b)}')

```

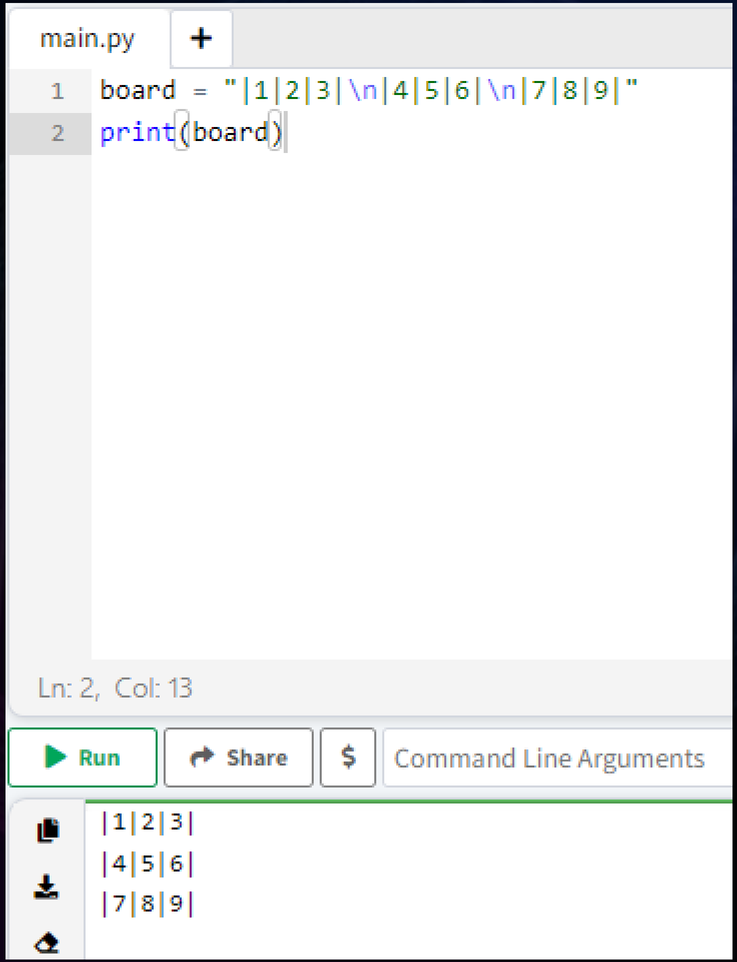
Step 2: Make your first board



```python
board = "|1|2|3|\n|4|5|6|\n|7|8|9|"
print(board)
```



```
|1|2|3|
|4|5|6|
|7|8|9|
```



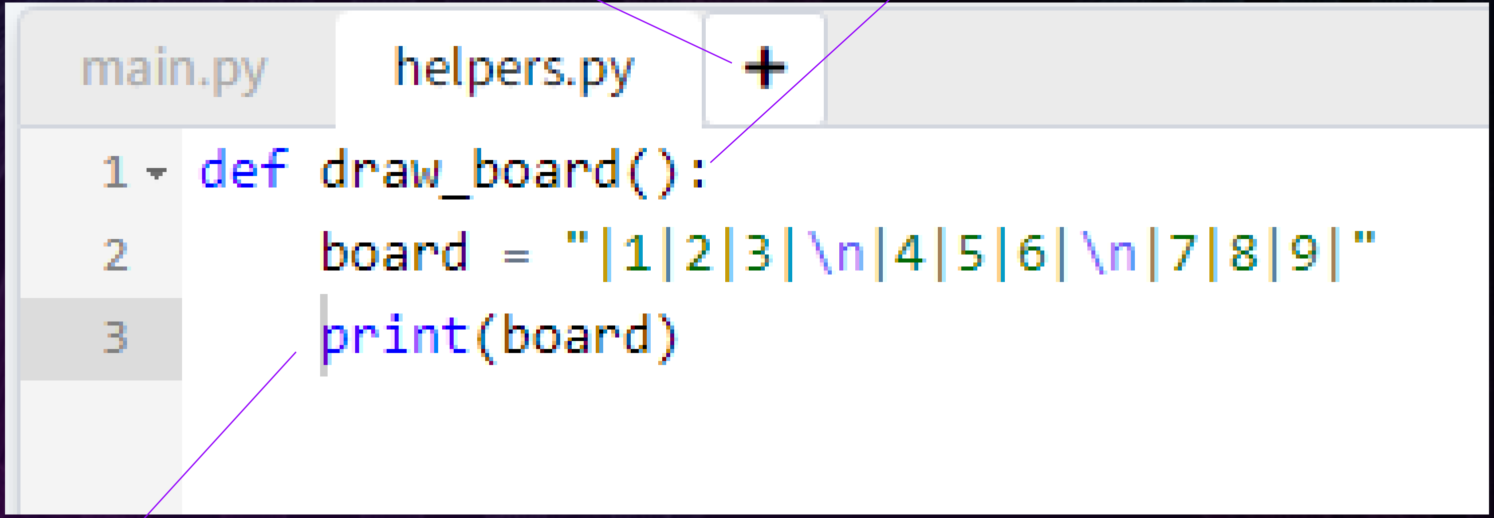
```python
board = "|1|2|3|\n|4|5|6|\n|7|8|9|"
print(board)
```

```
|1|2|3|
|4|5|6|
|7|8|9|
```

Step 3: Make helpers.py and the draw_board() function

Click the + sign to make a new file

Always put a colon : at the end

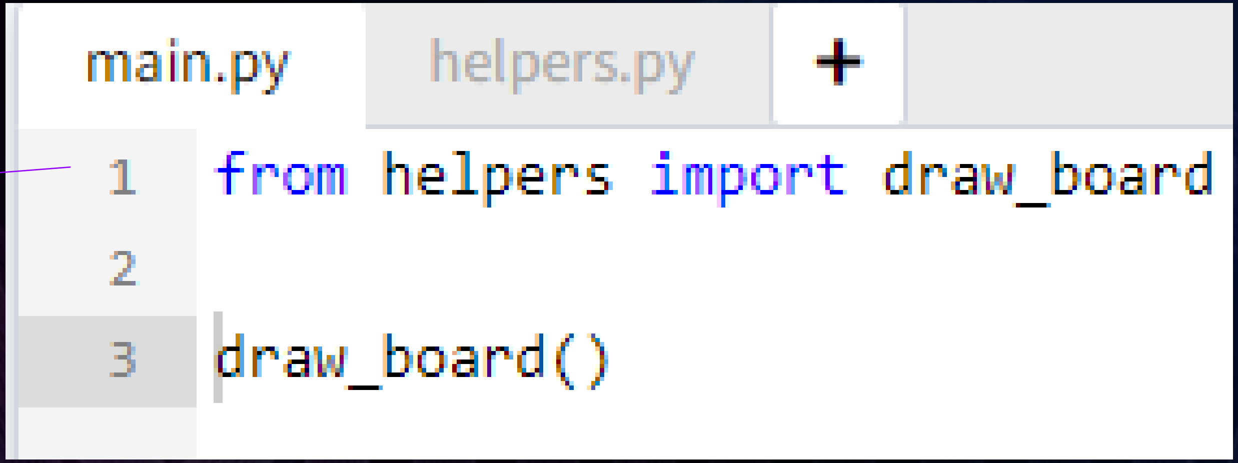


```
def draw_board():
    board = "|1|2|3|\n|4|5|6|\n|7|8|9|"
    print(board)
```

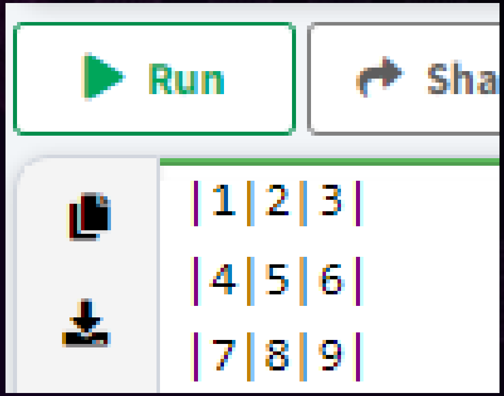
Keep these lines indented

Step 4: Update your main.py

Import the draw_board function from helpers.py

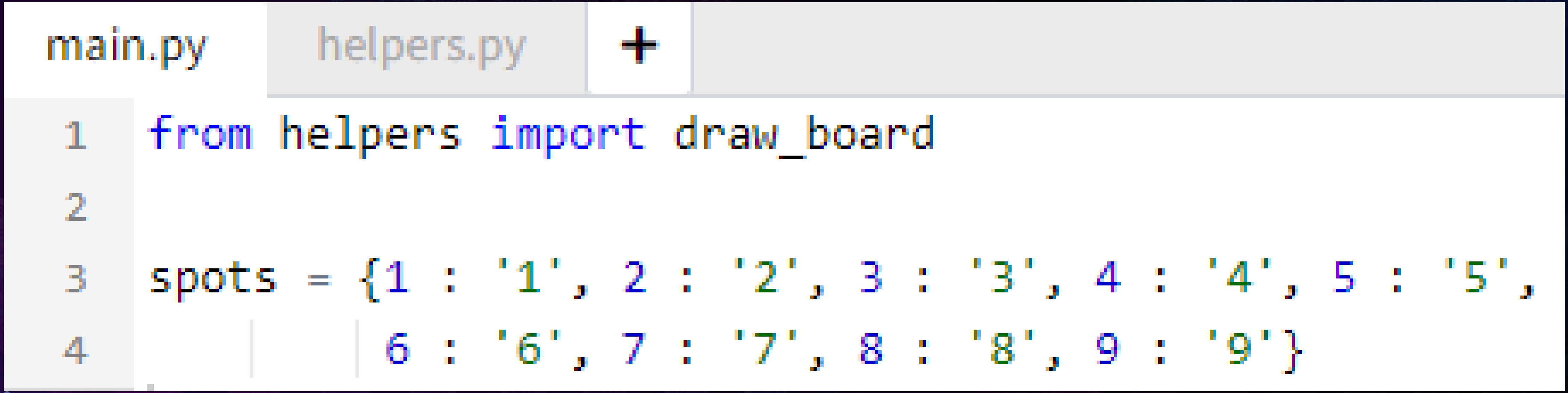


```python
from helpers import draw_board

draw_board()
```



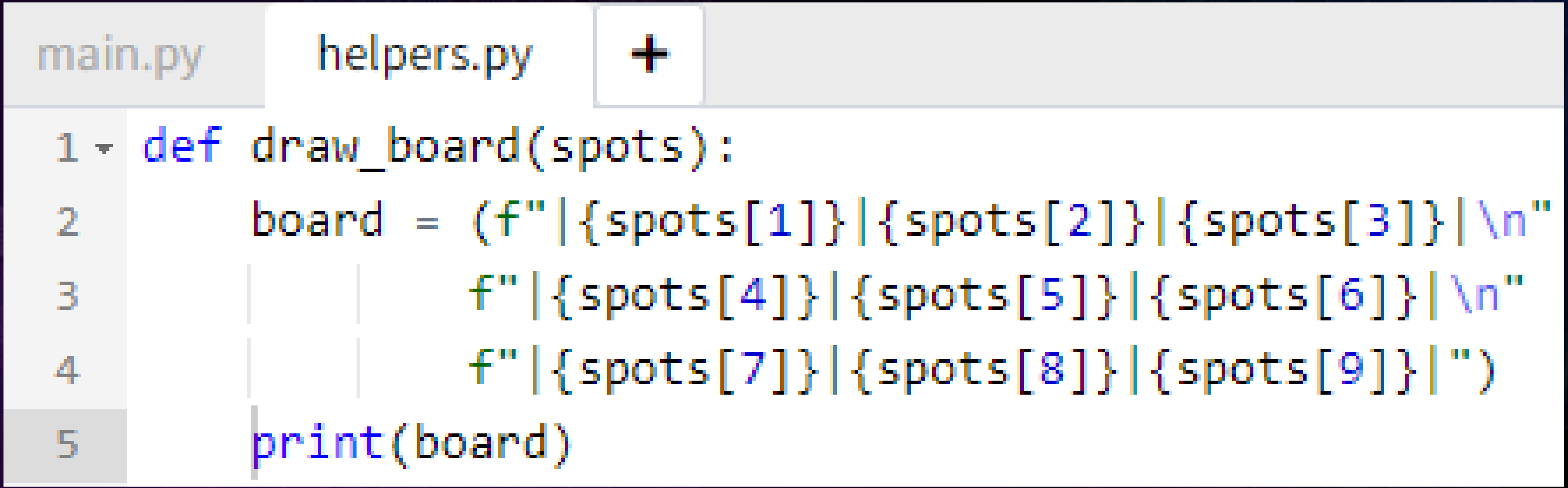
```
|1|2|3|
|4|5|6|
|7|8|9|
```

Step 5: Create Spots Dictionary



```python
from helpers import draw_board

spots = {1 : '1', 2 : '2', 3 : '3', 4 : '4', 5 : '5',
         6 : '6', 7 : '7', 8 : '8', 9 : '9'}
```

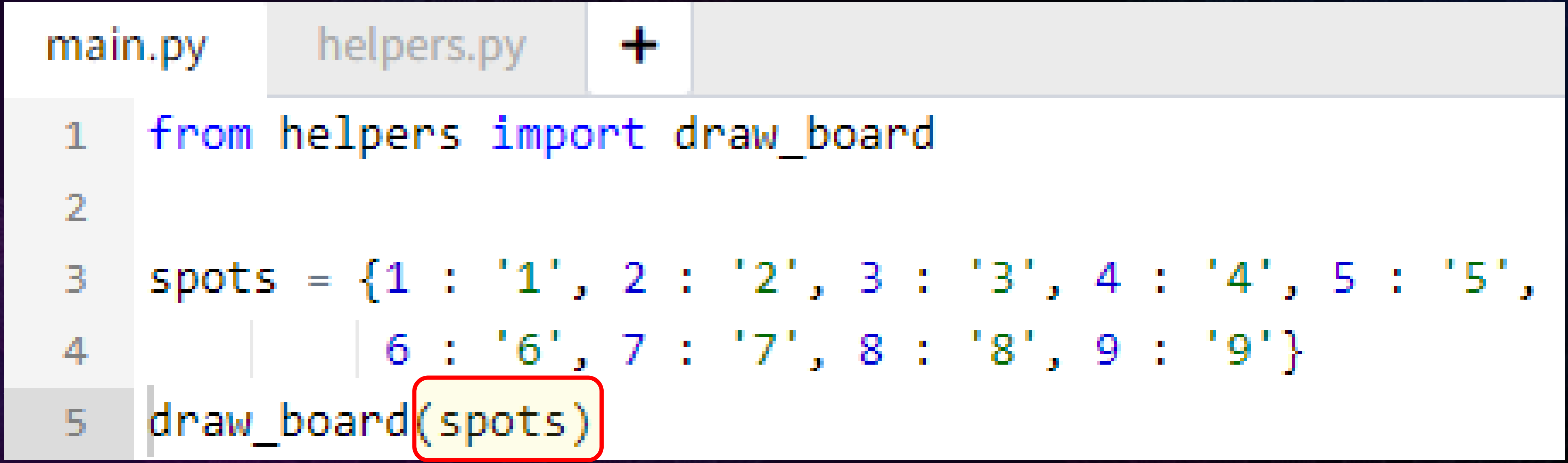
Step 6: Create a F-function in helpers.py



```python
def draw_board(spots):
    board = (f"|{spots[1]}|{spots[2]}|{spots[3]}|\n"
            f"|{spots[4]}|{spots[5]}|{spots[6]}|\n"
            f"|{spots[7]}|{spots[8]}|{spots[9]}|")
    print(board)
```

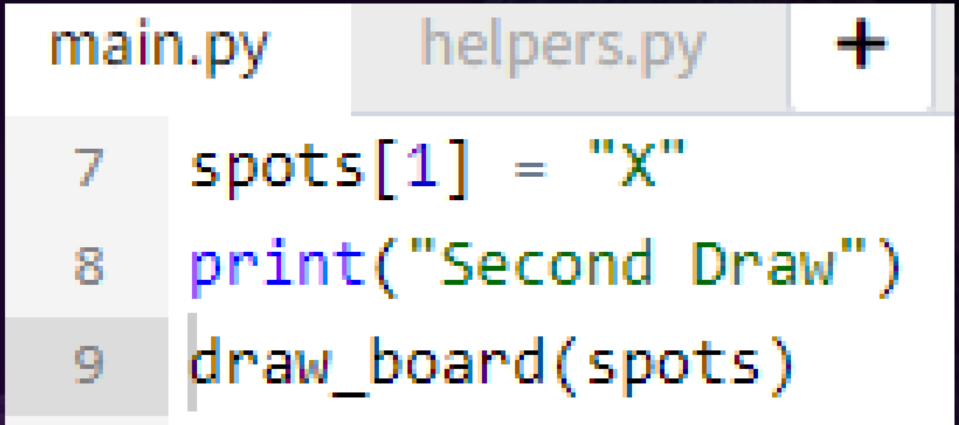
Dictionary{key}

Step 7: Create Spots Dictionary

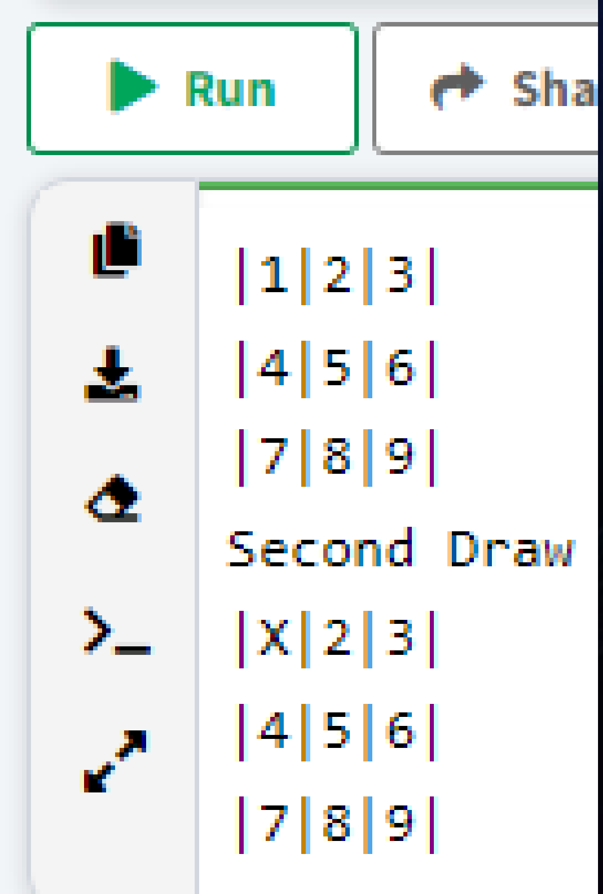


```
from helpers import draw_board

spots = {1 : '1', 2 : '2', 3 : '3', 4 : '4', 5 : '5',
         6 : '6', 7 : '7', 8 : '8', 9 : '9'}
draw_board(spots)
```

Step 8: draw your first X

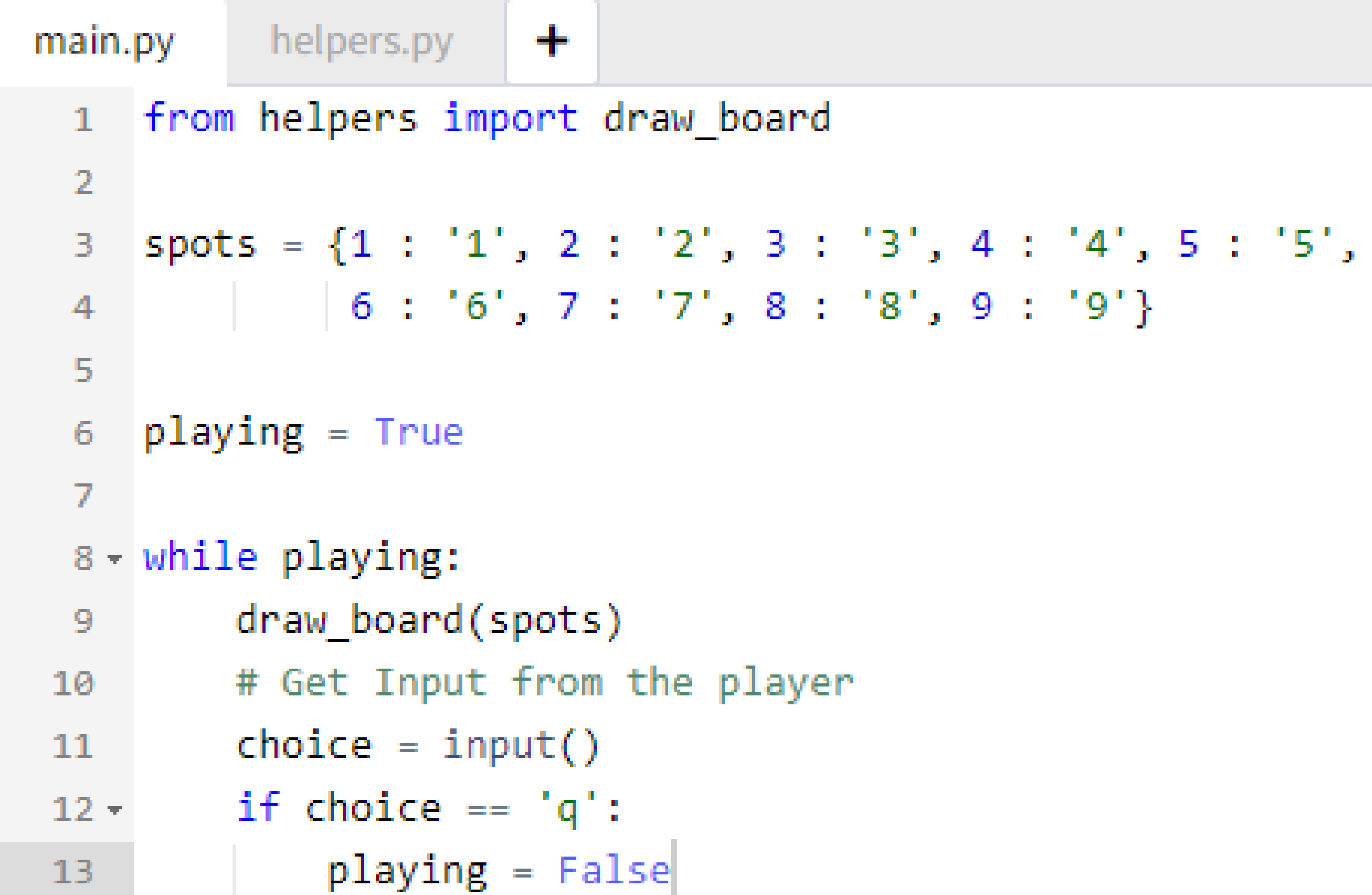


```python
spots[1] = "X"
print("Second Draw")
draw_board(spots)
```



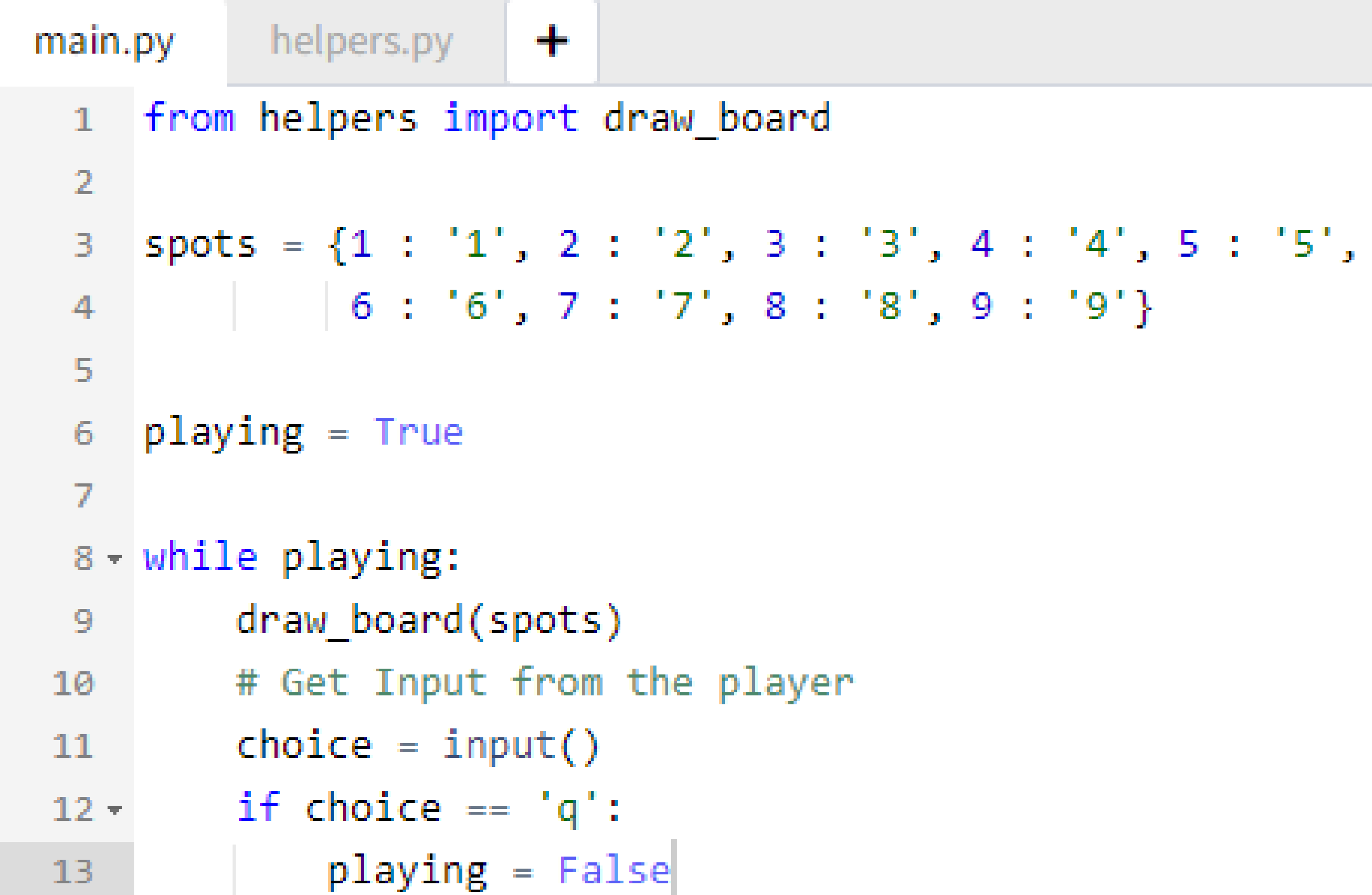
```
|1|2|3|
|4|5|6|
|7|8|9|
Second Draw
|X|2|3|
|4|5|6|
|7|8|9|
```

Step 9: while loop and quit



```python
from helpers import draw_board

spots = {1 : '1', 2 : '2', 3 : '3', 4 : '4', 5 : '5',
         6 : '6', 7 : '7', 8 : '8', 9 : '9'}

playing = True

while playing:
    draw_board(spots)
    # Get Input from the player
    choice = input()
    if choice == 'q':
        playing = False
```

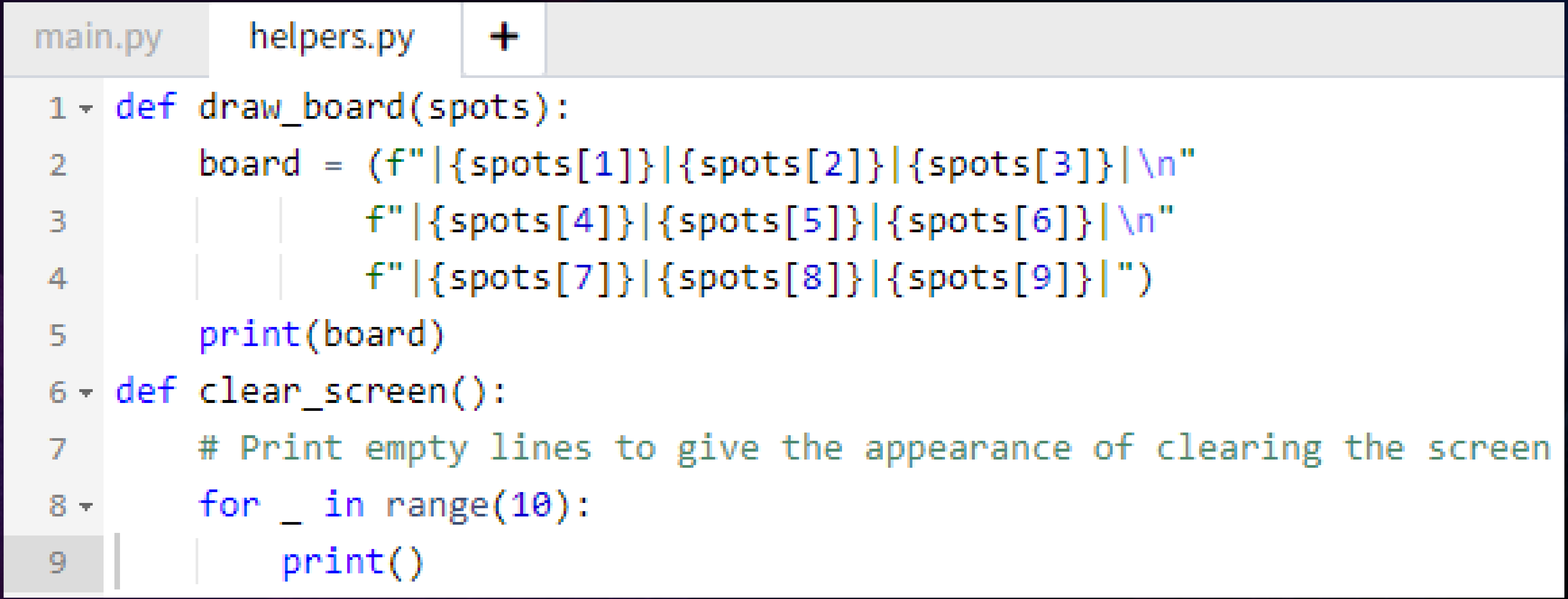
Step 10: while loop and quit



```python
from helpers import draw_board

spots = {1 : '1', 2 : '2', 3 : '3', 4 : '4', 5 : '5',
         6 : '6', 7 : '7', 8 : '8', 9 : '9'}

playing = True

while playing:
    draw_board(spots)
    # Get Input from the player
    choice = input()
    if choice == 'q':
        playing = False
```

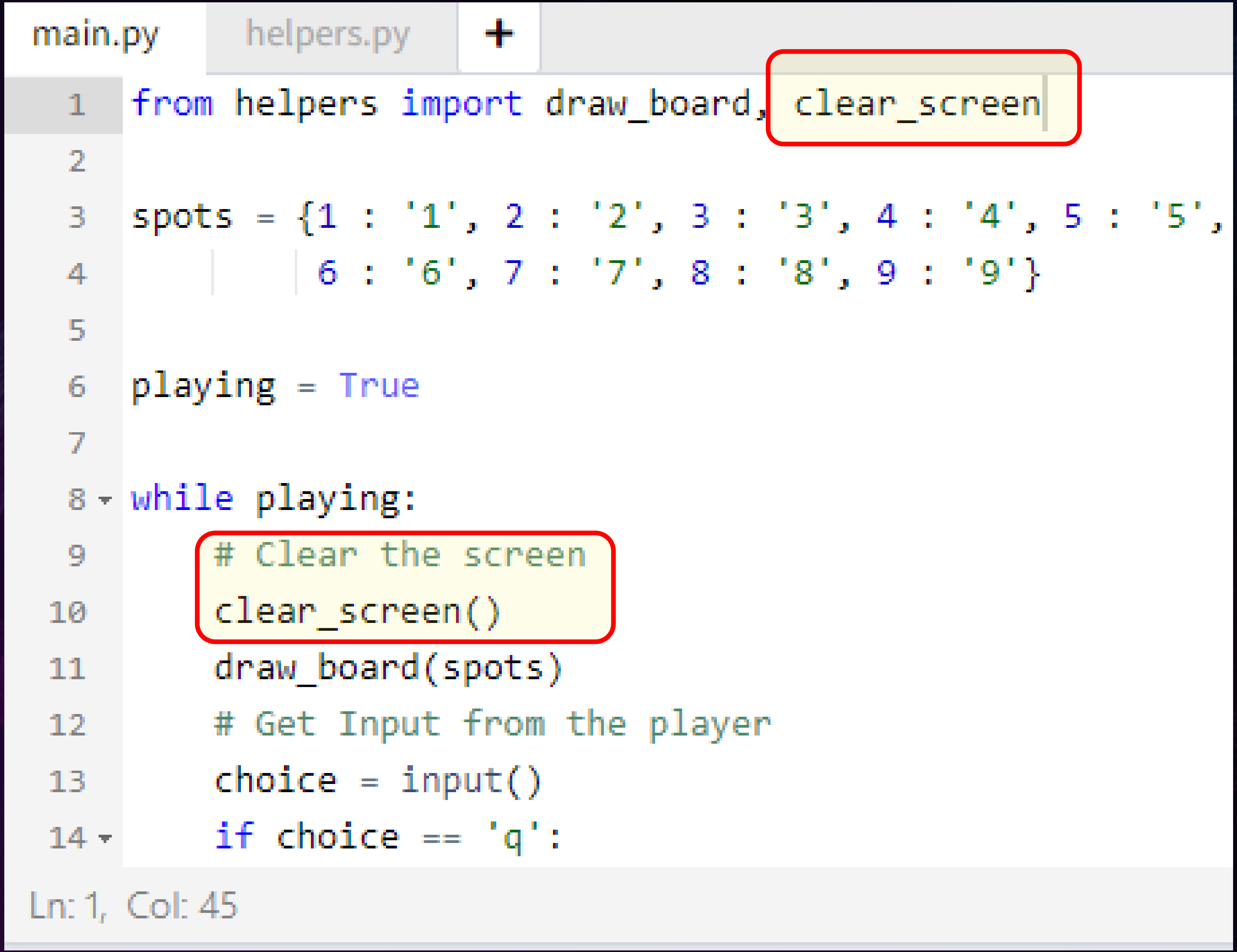
Run and type in letters until you type in q. It should quit.

Step 11: function clear_scree()



```python
def draw_board(spots):
    board = (f"|{spots[1]}|{spots[2]}|{spots[3]}|\n"
             f"|{spots[4]}|{spots[5]}|{spots[6]}|\n"
             f"|{spots[7]}|{spots[8]}|{spots[9]}|")
    print(board)
def clear_screen():
    # Print empty lines to give the appearance of clearing the screen
    for _ in range(10):
        print()
```

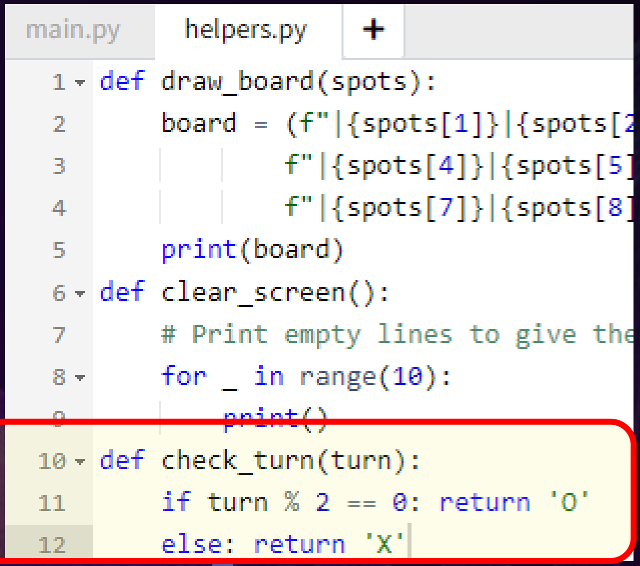
Step 12: add clear_screen() to main.py



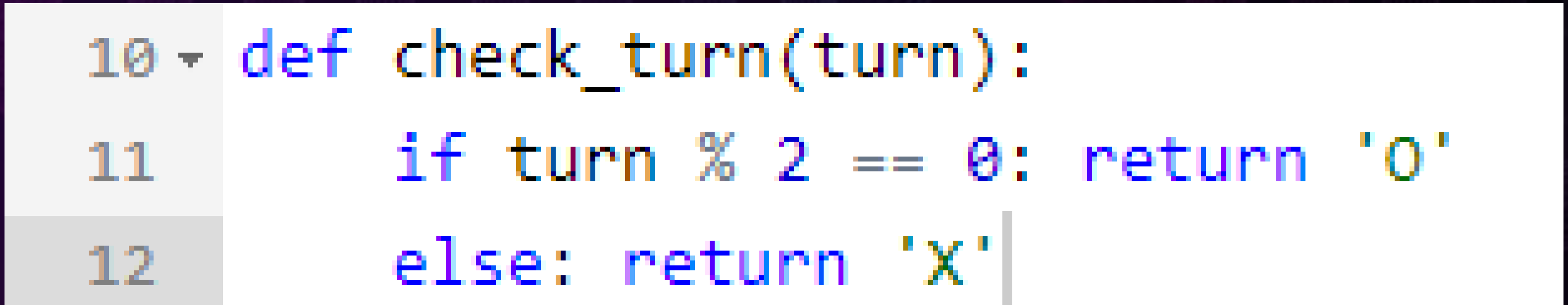
```
from helpers import draw_board, clear_screen

spots = {1 : '1', 2 : '2', 3 : '3', 4 : '4', 5 : '5',
         6 : '6', 7 : '7', 8 : '8', 9 : '9'}

playing = True

while playing:
    # Clear the screen
    clear_screen()
    draw_board(spots)
    # Get Input from the player
    choice = input()
    if choice == 'q':
```

Run and type in letters until you type in q. It should quit.

Step 13: create check_turn() function

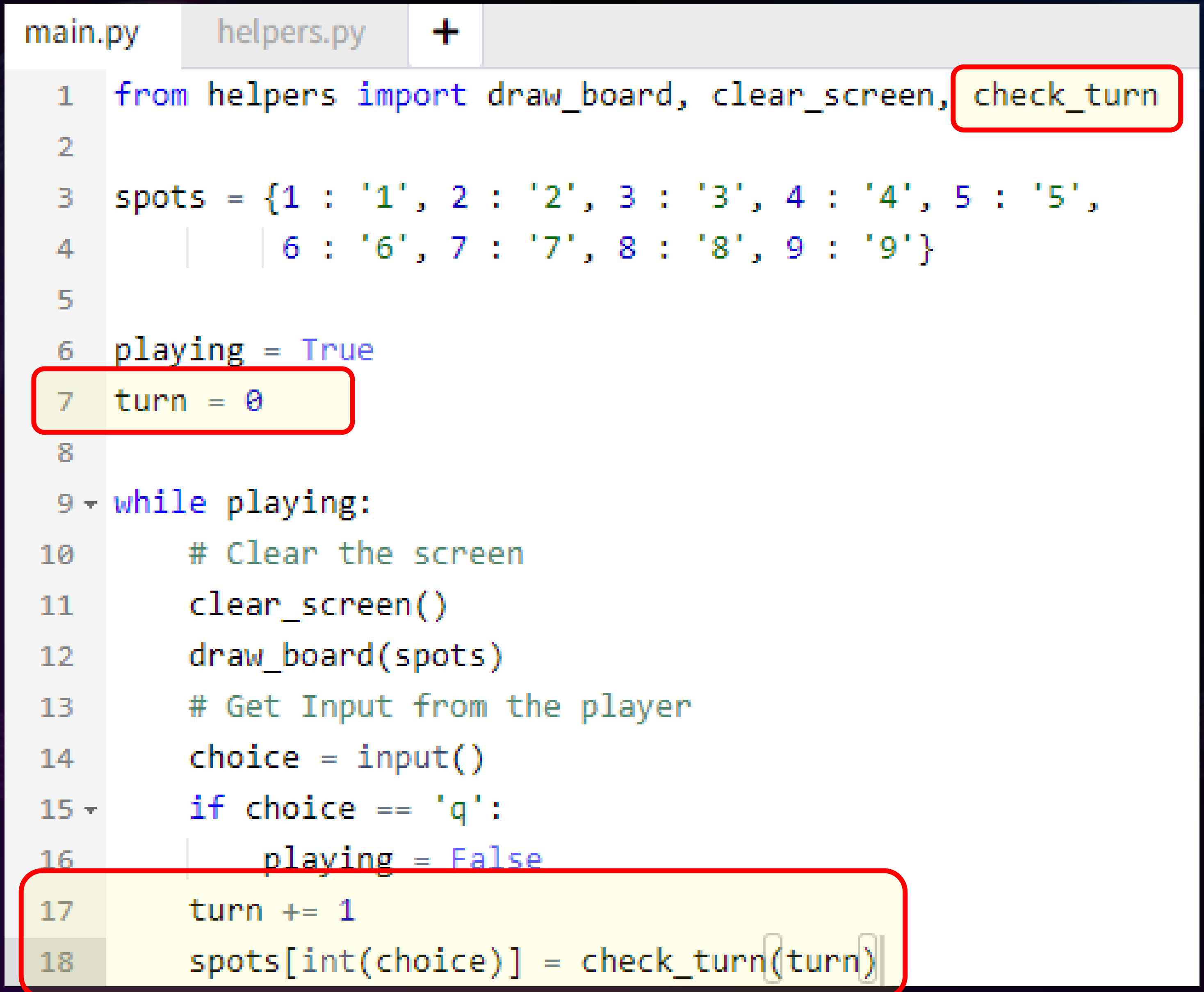


```python
def draw_board(spots):
    board = (f"|{spots[1]}|{spots[2
            f"|{spots[4]}|{spots[5]
            f"|{spots[7]}|{spots[8]
    print(board)
def clear_screen():
    # Print empty lines to give the
    for _ in range(10):
        print()
def check_turn(turn):
    if turn % 2 == 0: return 'O'
    else: return 'X'
```



```python
def check_turn(turn):
    if turn % 2 == 0: return 'O'
    else: return 'X'
```

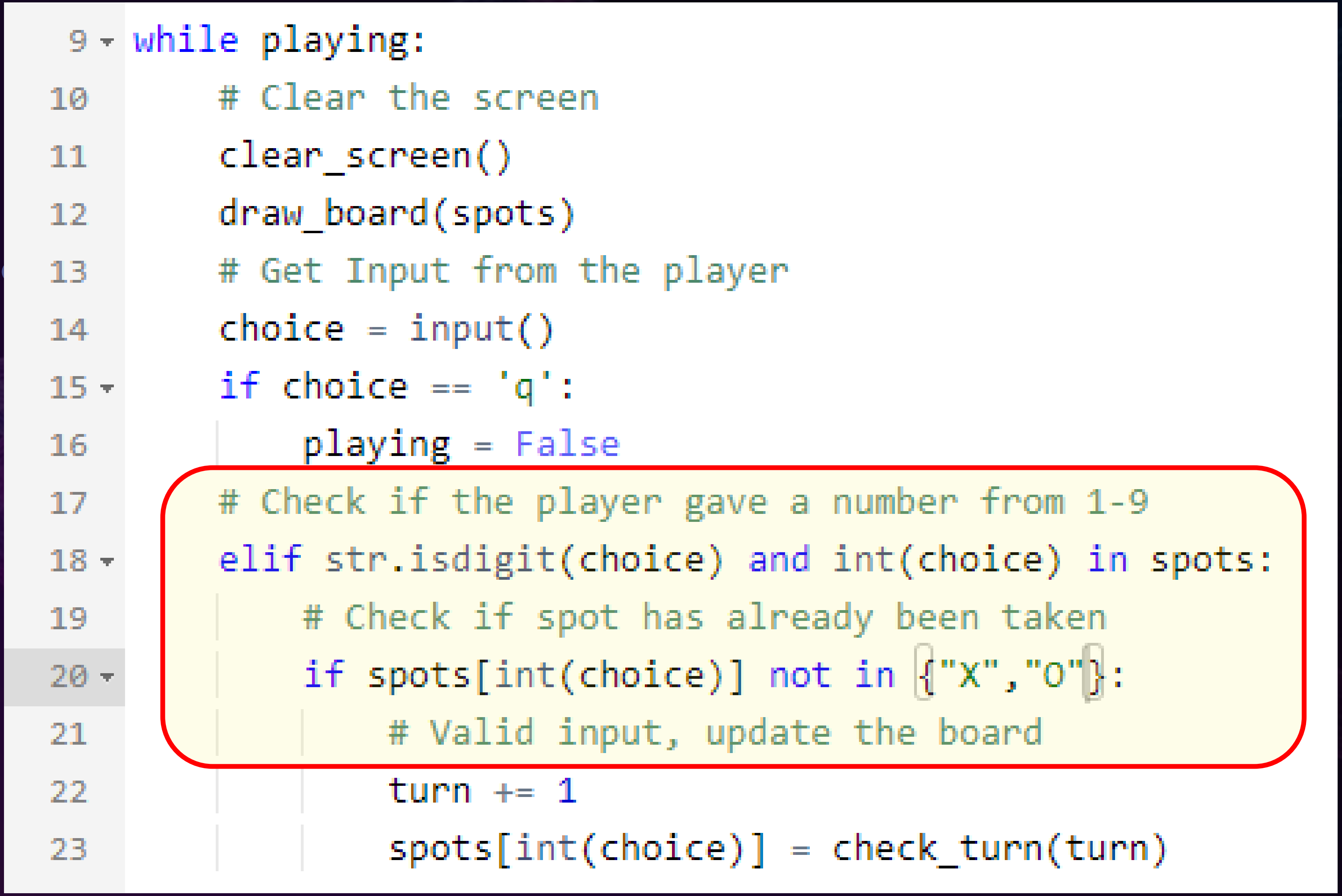
Step 14: add check_turn() to main.py



```python
from helpers import draw_board, clear_screen, check_turn

spots = {1 : '1', 2 : '2', 3 : '3', 4 : '4', 5 : '5',
         6 : '6', 7 : '7', 8 : '8', 9 : '9'}

playing = True
turn = 0

while playing:
    # Clear the screen
    clear_screen()
    draw_board(spots)
    # Get Input from the player
    choice = input()
    if choice == 'q':
        playing = False
    turn += 1
    spots[int(choice)] = check_turn(turn)
```

- Run again and you should be able to change the board with X's and O's

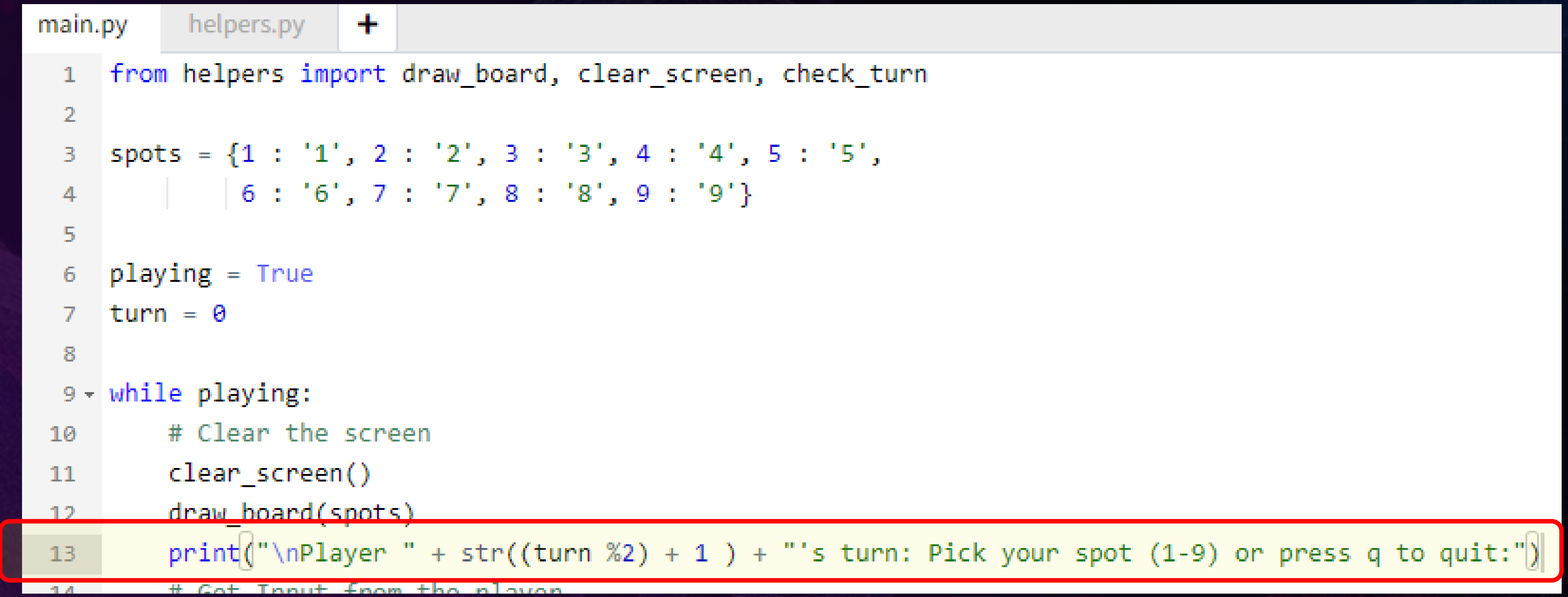
Step 15: only take 1-9 and not overwrite X or O



```
while playing:
    # Clear the screen
    clear_screen()
    draw_board(spots)
    # Get Input from the player
    choice = input()
    if choice == 'q':
        playing = False
    # Check if the player gave a number from 1-9
    elif str.isdigit(choice) and int(choice) in spots:
        # Check if spot has already been taken
        if spots[int(choice)] not in {"X","O"}:
            # Valid input, update the board
            turn += 1
            spots[int(choice)] = check_turn(turn)
```

- Run again and you should be able to change the board with X's and O's

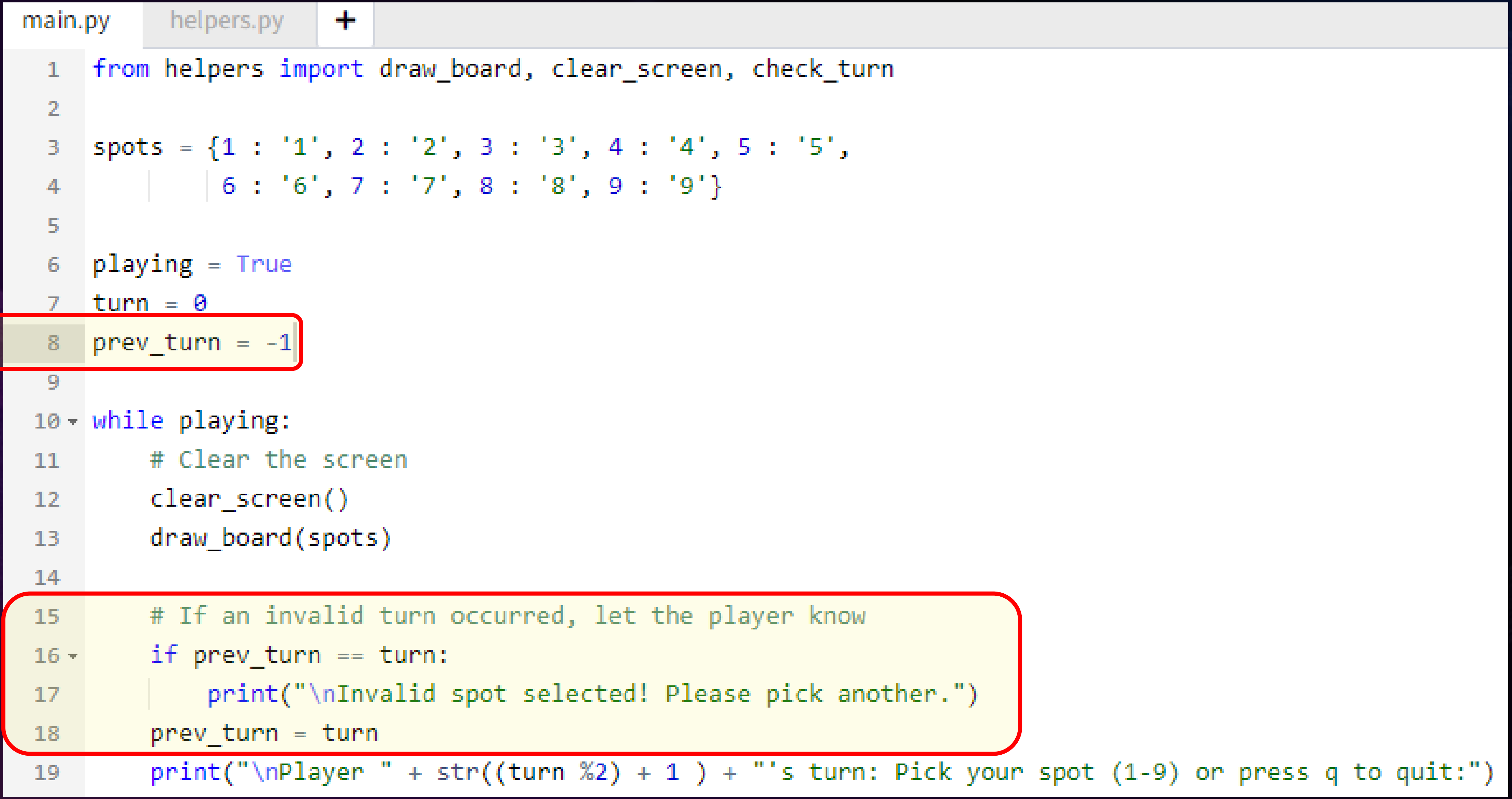
Step 16: tell who‘s turn it is



```
from helpers import draw_board, clear_screen, check_turn

spots = {1 : '1', 2 : '2', 3 : '3', 4 : '4', 5 : '5',
         6 : '6', 7 : '7', 8 : '8', 9 : '9'}

playing = True
turn = 0

while playing:
    # Clear the screen
    clear_screen()
    draw_board(spots)
    print("\nPlayer " + str((turn %2) + 1 ) + "'s turn: Pick your spot (1-9) or press q to quit:")
    # Get Input from the player
```

Run again and you should see it change between Player 1 and Player 2

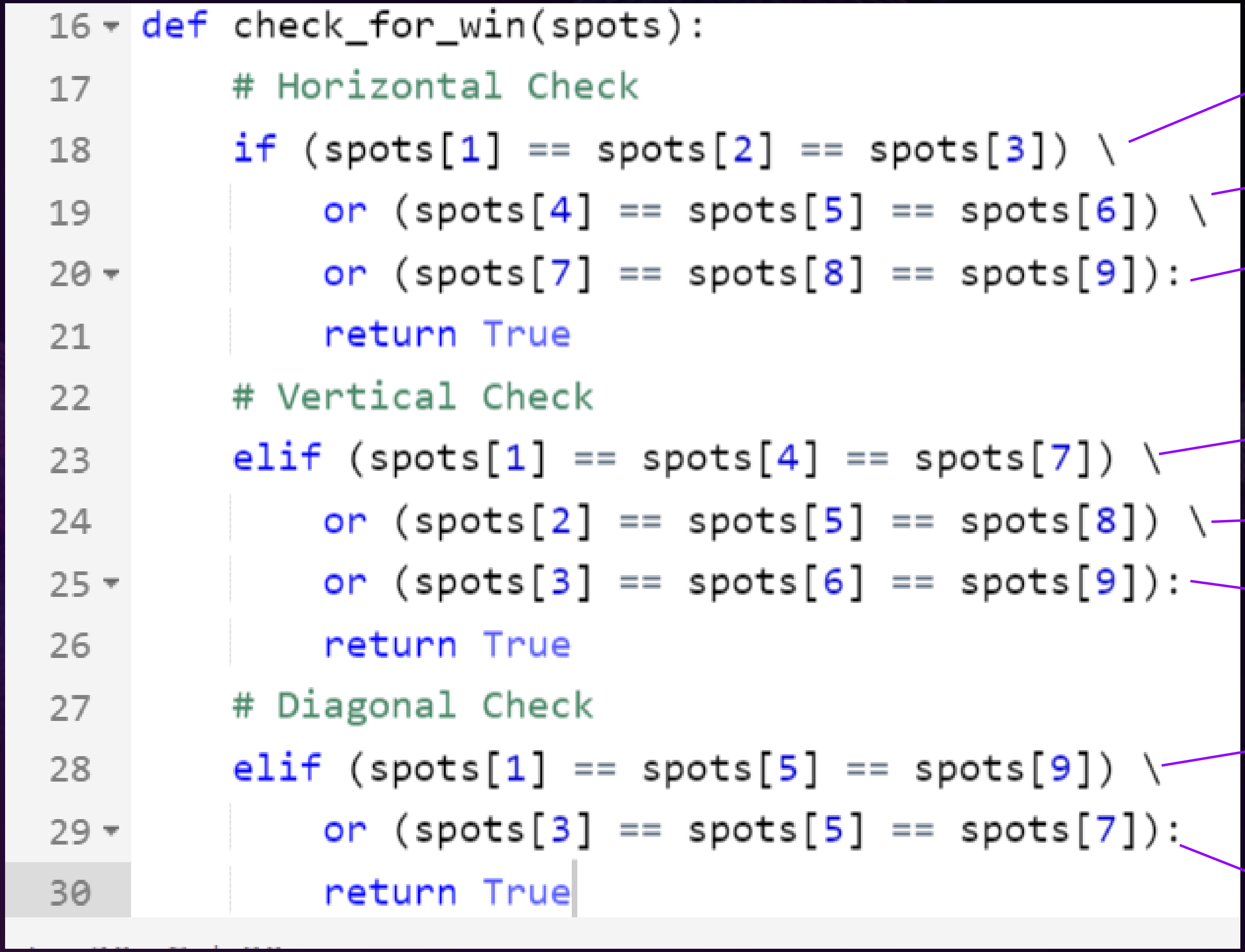
Step 17: tell user if input is invalid



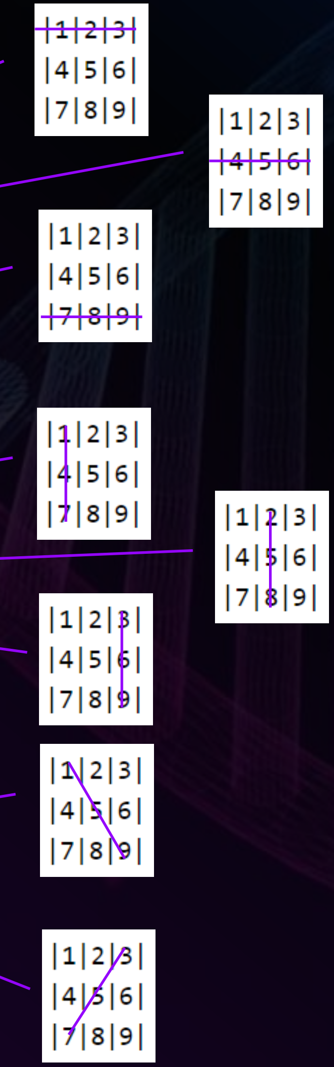
```python
from helpers import draw_board, clear_screen, check_turn

spots = {1 : '1', 2 : '2', 3 : '3', 4 : '4', 5 : '5',
         6 : '6', 7 : '7', 8 : '8', 9 : '9'}

playing = True
turn = 0
prev_turn = -1

while playing:
    # Clear the screen
    clear_screen()
    draw_board(spots)

    # If an invalid turn occurred, let the player know
    if prev_turn == turn:
        print("\nInvalid spot selected! Please pick another.")
    prev_turn = turn
    print("\nPlayer " + str((turn %2) + 1 ) + "'s turn: Pick your spot (1-9) or press q to quit:")
```

Run. Now it will tell you if you had an invalid input to the user.

Step 18: make check_for_win() function in helpers.py



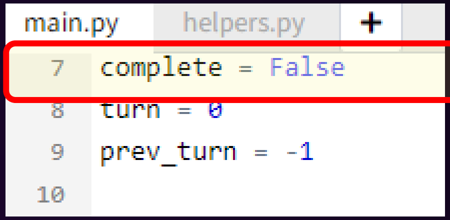
```python
def check_for_win(spots):
    # Horizontal Check
    if (spots[1] == spots[2] == spots[3]) \
        or (spots[4] == spots[5] == spots[6]) \
        or (spots[7] == spots[8] == spots[9]):
        return True
    # Vertical Check
    elif (spots[1] == spots[4] == spots[7]) \
        or (spots[2] == spots[5] == spots[8]) \
        or (spots[3] == spots[6] == spots[9]):
        return True
    # Diagonal Check
    elif (spots[1] == spots[5] == spots[9]) \
        or (spots[3] == spots[5] == spots[7]):
        return True
```



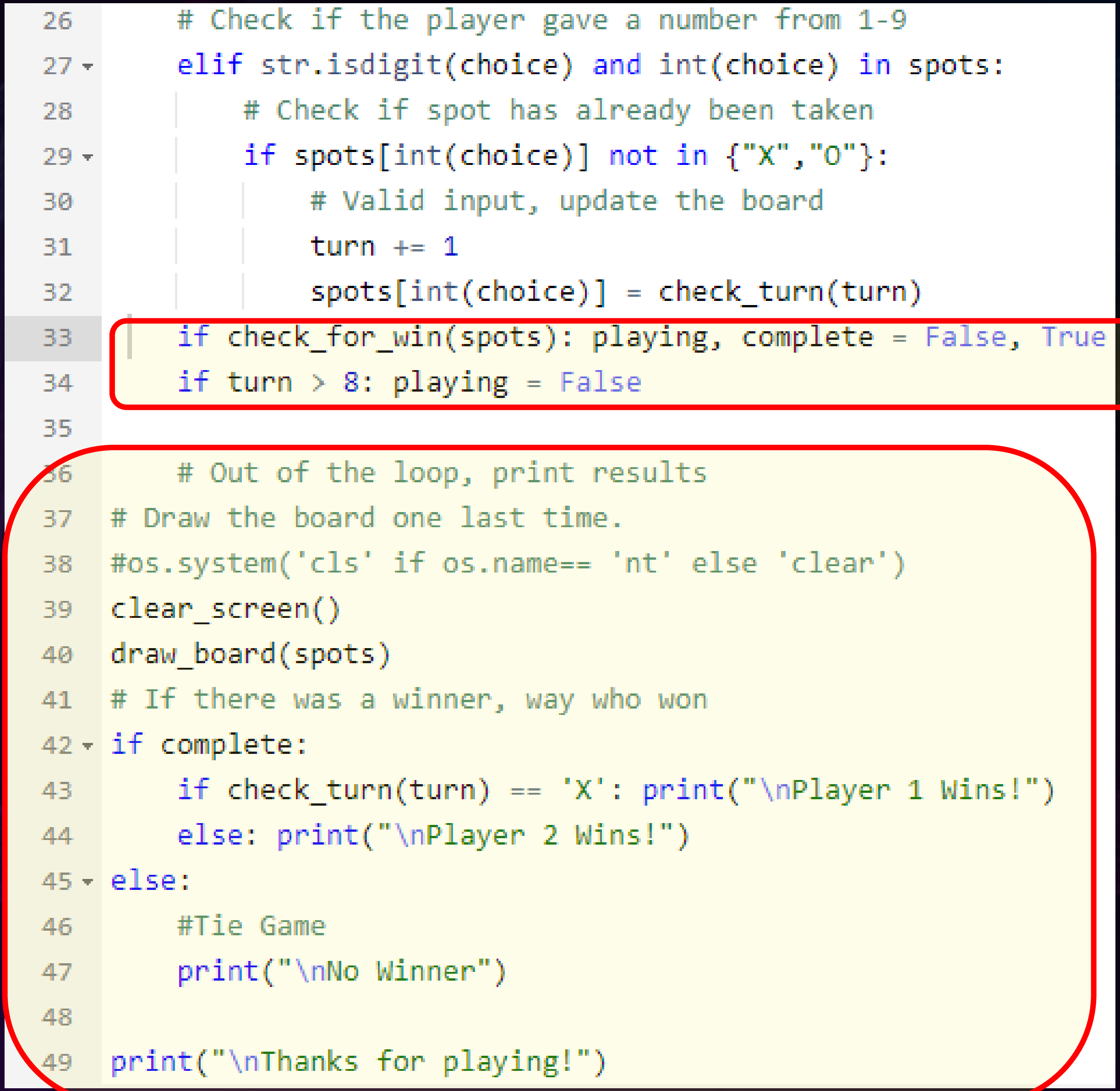
8 Ways to Win

- 3 Horizontal
- 3 Vertical
- 2 Diagonal

Step 19: add check_for_win to main.py



```python
complete = False
turn = 0
prev_turn = -1

```



```python
    # Check if the player gave a number from 1-9
    elif str.isdigit(choice) and int(choice) in spots:
        # Check if spot has already been taken
        if spots[int(choice)] not in {"X","O"}:
            # Valid input, update the board
            turn += 1
            spots[int(choice)] = check_turn(turn)
    if check_for_win(spots): playing, complete = False, True
    if turn > 8: playing = False

    # Out of the loop, print results
# Draw the board one last time.
#os.system('cls' if os.name== 'nt' else 'clear')
clear_screen()
draw_board(spots)
# If there was a winner, way who won
if complete:
    if check_turn(turn) == 'X': print("\nPlayer 1 Wins!")
    else: print("\nPlayer 2 Wins!")
else:
    #Tie Game
    print("\nNo Winner")

print("\nThanks for playing!")
```

Run again and it should end on win, but will not finish drawing correctly.

YOU DID IT!!

4. Teach someone to play your game!

How are video games made?

Tic Tac Toe Game

Teach family

The End