Battleship!

Test Run

Congratulations! Now you should have a game of Battleship! that is fully functional for one guess.

Make sure you play it a couple of times and try different kinds of incorrect guesses. This is a great time to stop and do some serious debugging.

In the next step, we'll move on and look at how to give the user 4 guesses to find the battleship.

Instructions

Thoroughly test your game. Make sure you try a variety of different guesses and look for any errors in the syntax or logic of your program.

Q&A Forum Glossary

script.py

```
    for row in board:
        print " ".join(row)

print_board(board)

def random_row(board):
    return randint(0, len(board) - 1)

def random_col(board):
    return randint(0, len(board[0]) - 1)

ship_row = random_row(board)
ship_col = random_col(board)
guess_row = int(raw_input("Guess Row:"))
guess_col = int(raw_input("Guess Col:"))

print ship_row
print ship_col

# Write your code below!
if guess_row == ship_row and guess_col == ship_col:
    print "Congratulations! You sank my battleship!"
else:
    if guess_row not in range(5) or guess_col not in range(5):
        print "Oops, that's not even in the ocean."
    elif board[guess_row][guess_col]=="X":
        print "You guessed that one already."
    else:
        board[guess_row][guess_col] ='X'
        print_board(board)
        print"You missed my battleship!"
```

```
Guess Col: 3
3
0
O O O O O
O O O O O
O O O O O
O O O X O
O O O O O
You missed my battleship!
None
```

Congratulations, you've finished this section! Next: You Sunk My Battleship! →