Battleship!

Bad Aim

Great job! Now we can handle both correct and incorrect guesses from the user. But now let's think a little bit more about the "miss" condition.

01. They can enter a guess that's off the board.

02. They can guess a spot they've already guessed.

03. They can just miss the ship.

We'll add these tests inside our `else` condition. Let's build the first case now!

```
if x not in range(8) or \
   y not in range(3):
       print "Outside the range"
```

The example above checks if either `x` or `y` are outside those ranges. The `\` character just continues the `if` statement onto the next line.

Q&A Forum Glossary

script.py

```
1  from random import randint
2
3  board = []
4
5  for x in range(0, 5):
6      board.append(["O"] * 5)
7
8  def print_board(board):
9      for row in board:
10         print " ".join(row)
11
12 print_board(board)
13
14 def random_row(board):
15     return randint(0, len(board) - 1)
16
17 def random_col(board):
18     return randint(0, len(board[0]) - 1)
19
20 ship_row = random_row(board)
21 ship_col = random_col(board)
22 guess_row = int(raw_input("Guess Row:"))
23 guess_col = int(raw_input("Guess Col:"))
24
25 print ship_row
26 print ship_col
27
28 # Write your code below!
29 if guess_row == ship_row and guess_col == ship_col:
30     print "Congratulations! You sank my battleship!"
31 else:
```

```
Guess Col: 3
0
3
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
```

Way to go! Start Next Lesson

12. Bad Aim