Functions -- Practice Problems

(Included as a repl assignment in GWC PDX Class)

Question 1: Absolute Values!

Write a function so for any number the function returns the absolute value.
After you write the function, use it with 2 different numbers

Hint:

- Don't forget to fill in the '__' spaces too.

Example Input/Output:

```
Input:  -5
Output:  5

Input: 34
Output: 34
```

Starting code in repl assignment:

```
def absolute_value(______):
  """  Write what the function does here:    """


  ## Write the function here

  return _____

## TRY YOUR FUNCTION HERE:
```

Question 2: Largest of two numbers

Create a function to find the largest value (ie: the maximum) of two numbers. Name it max_of_two. Once you write the function, test the following cases to make sure it works!:

Test Cases #1 and #2:

```
test1 = max_of_two(3,1)
print(test1)

test2 = max_of_two(2,4)
print(test2)
```

Once you are able to run the two test cases above, create a function to find the maximum of 3 numbers and name it max_of_three. BUT call (ie: use the function) max_of_two within your new function. Once you write the function, test the following cases to make sure it works!:

Test Cases #3 and #4:

```
test3 = max_of_three(3,1,5)
print(test3)

test4 = max_of_three(10,12,4)
print(test4)
```

Starting code in repl assignment:

```
## Write the function, max_of_two
def max_of_two( _________ , _________):
  """  Write what this function does here  """



### Try Test Cases #1 and #2:



# Write function, max_of_three



### Try Test Cases #3 and #4:
```

Question 3: Get Unique Numbers

Write a Python function that takes a list and returns a new list with unique elements of the first list. Name your function: unique_list

HINT:

(Don't forget to fill in the '__')

1. In your function, initiate a new list.
2. Write a loop to iterate through the entire list
    - If the item in the old list (the list you pass in the function) is NOT in the new list (the one you just initiated)....
    - Append the item to the new list!
3. return the new list!

Try out the following test cases:

```
list1 = [1,2,3,3,3,3,4,5]
test1 = unique_list(list1)
print(test1)


list2 = ['m', 'o', 'l', 'l', 'y']
test2 = unique_list(list2)
print(test2)


list3 = ['p', 'r', 'i', 'y', 'a']
test3 = unique_list(list3)
print(test3)
```

Starting code in repl assignment:

```
def unique_list( _______ ):

  # Initiate your list


  # Write a loop (you choose between a for and a while)


  # Return the result


# Try the test cases here:
```

Question 4: Tic-Tac-Toe

Make a two-player Rock-Paper-Scissors game. (Hint: Ask for player plays (using input), compare them, print out a message of congratulations to the winner, and ask if the players want to start a new game)

Remember the rules:

- Rock beats scissors
- Scissors beats paper
- Paper beats rock
- The same play is a TIE!

Starting code in repl assignment:

```
import sys

#Ask the user for play
player1 = input("What's your name, Player 1?")
player2 = input("And your name, Player 2?")

player1_answer = input("%s, do you want to choose rock, paper or scissors?" % playe
player2_answer = input("%s, do you want to choose rock, paper or scissors?" % playe

def compare(_______, _________):

    #If it's a tie....



    #Else If p1 picks rock...
      #if p2 picks scissors...


      #else

    #else if p1 picks scissors...
      #if p2 picks paper...

      #else..

    #else if p1 picks paper
      # if p2 picks rock....

      #else

    #else
        #say its invalid input!

        #leave this here:
        sys.exit()

 #### CALL THE Function In A print STATEMENT
```