Coding Challenge #1

From the real world
11/07/16

CHALLENGE:

Define a function that takes two arguments, an array and an integer, and checks if any two numbers in the array sum to the integer. The function should output True/False based on results.

TIPS:

- Try it by hand first! Use pen and paper to figure it out instead of jumping to code.
- Make sure to test/debug as you are writing code.
- Approach:
  - pseudo-code ⇒ first-draft code (get it working!) ⇒ refactor code

TEST CASES

test_1 = ([3, 7, 88, 24, 9, 0], 9)

test_2 = ([4, 93, 2000, 36, 55], 41)

test_3 = ([4, 44, 4444, 1616, 8, 16], 48)

test_4 =([2, -3, 72, -6, 1], 1)

SOLUTION:

(Well, this is just one solution. There are many. Try to make a better solution than mine...and be able to explain why it's better.)

```
1  def checker(an_array, a_num):
2
3      index_1 = 0
4
5      while index_1 < len(an_array) - 1:
6          index_2 = index_1 + 1
7
8          while index_2 < len(an_array):
9              print an_array[index_1], an_array[index_2]
10             if an_array[index_1] + an_array[index_2] == a_num:
11                 return True
12             index_2 += 1
13
14         index_1 +=1
15
16
17     return False
```

TAKE-AWAYS:

- Talk it out first! Do by hand first!
- Why is a while loop better than a for loop for this problem? Hint: Think about ‘Big O Notation’ (check out this link for more info: https://rob-bell.net/2009/06/a-beginners-guide-to-big-o-notation/)
- Why is there no need for the ‘else’ portion in the if-block of code in my solution? When would we need the ‘else’ part?
- Make sure to test as you are writing your code to make sure it is working properly at every step. Simple ‘prints’ in the right places can make debugging a breeze!
- Think of test cases! Always try to break your code! Be your own worst enemy...or critic.