



---

# SEMINAR I – INTRODUCTION TO PYTHON

---

## CONTENTS

---

What you should know after attending .....	1
Python programs ... ..	2
Basic Data Types (Int, Boolean) .....	2
Problems With Strings .....	3
Problems with Lists .....	4

---

## WHAT YOU SHOULD KNOW AFTER ATTENDING

---

- Python basics – easy to understand, interpreted, dynamically typed, modern, fully-featured language
- Write and run simple programs using console input/output and the list basic structured data type
- Use procedural programming, making sure each function has a single responsibility

## PYTHON PROGRAMS ...

### BASIC DATA TYPES (INT, BOOLEAN)

1. Given 2 ints, a and b, return True if one if them is 10 or if their sum is 10  
(<http://codingbat.com/prob/p124984>)

**Question -** What happens if we enter a non-integer number, or alphanumeric characters?

```
def makes10(a, b):  
    '''  
    Returns True if one of the parameters or their sum is 10  
    a,b - integers  
    '''  
    return (a == 10 or b == 10 or a + b == 10)  
  
a = int(input("Give first int: "))  
b = int(input("Give second int: "))  
print("Result is ", makes10(a, b))
```

2. The parameter weekday is True if it is a weekday, and the parameter vacation is True if we are on vacation. We sleep in if it is not a weekday or we're on vacation. Return True if we sleep in.  
(<http://codingbat.com/prob/p173401>)

```
def sleepIn(weekday, vacation):  
    '''  
    Checks whether we can sleep in  
    weekday - Boolean parameter for weekday  
    vacation - Boolean parameter for vacation  
    Return True if it is not a weekday or we are on vacation, False otherwise  
    '''  
    if not weekday or vacation:  
        return True  
    else:  
        return False  
  
weekday = input("Is it a weekday? ")  
weekday = (weekday == 'yes')  
  
vacation = input("Are we on vacation? ")  
vacation = (vacation == 'yes')  
print("Do we sleep in? ", sleepIn(weekday, vacation))
```

**Question -** How do we validate input in the code above? What should happen if the user enters something else than yes/no?

## PROBLEMS WITH STRINGS

3. Given a non-empty string like "Code" return a string like "CCoCodCode"

stringSplosion('Code') → 'CCoCodCode'  
stringSplosion('abc') → 'aababc'  
stringSplosion('ab') → 'aab'  
(<http://codingbat.com/prob/p118366>)

```
def stringSplosion(string):  
    '''  
    TODO - Write specification  
    '''  
    result = ""  
    for i in range(len(string)):  
        result = result + string[:i + 1]  
    return result  
  
string = input("Enter string to explode ")  
print(stringSplosion(string))
```

4. Given 2 strings, a and b, return the number of the positions where they contain the same length 2 substring. So "xxcaazz" and "xxbaaz" yields 3, since the "xx", "aa", and "az" substrings appear in the same place in both strings.

stringMatch('xxcaazz', 'xxbaaz') → 3  
stringMatch('abc', 'abc') → 2  
stringMatch('abc', 'axc') → 0  
(<http://codingbat.com/prob/p182414>)

```
def stringMatch(a, b):  
    '''  
    TODO - Write specification  
    '''  
    # Figure which string is shorter.  
    shorter = min(len(a), len(b))  
    count = 0  
  
    # Loop i over every substring starting spot.  
    # Use length-1 here, so can use char str[i+1] in the loop  
    for i in range(shorter - 1):  
        a_sub = a[i:i + 2]  
        b_sub = b[i:i + 2]  
        if a_sub == b_sub:  
            count = count + 1  
  
    return count  
  
stringA = input("Give first string:")  
stringB = input("Give second string:")  
print("Number of positions that match: ", stringMatch(stringA, stringB))
```

## PROBLEMS WITH LISTS

5. Return the number of even ints in the given list. Note: the % "mod" operator computes the remainder, e.g.  $5 \% 2$  is 1.

(<http://codingbat.com/prob/p189616>)

```
def readNumberList():
    myList = []
    while True:
        x = int(input("Enter list item (-1 to finish):"))
        if x == -1:
            break
        myList.append(x)
    return myList

def evenCount(myList):
    """
    TODO - Write specification
    """
    result = 0
    for n in myList:
        if n % 2 == 0:
            result += 1
    return result

lst = readNumberList()
print("Number of even elements is ", evenCount(lst))
```

6. Return the sum of the numbers in a list, returning 0 for an empty list. Except the number 13 is very unlucky, so it does not count and numbers that come immediately after a 13 also do not count.

(<http://codingbat.com/prob/p167025>)

```
def sumWithout13(lst):
    """
    TODO - Write specification
    """
    result = 0
    if lst[0] != 13:
        result = lst[0]
    index = 1

    while index < len(lst):
        if lst[index] != 13 and lst[index - 1] != 13:
            result += lst[index]
        index += 1
    return result

lst = readNumberList()
print("Sum without 13s is ", sumWithout13(lst))
```