Raul Rodriguez
Python intro to data science

```python

'''Exercise 1
Create a lambda expression that computes the square root of a number (including real num-
bers, e.g., 3.14); the number is given as user input. Do not use the built-in function sqrt(), use the
** operator instead'''
squareroot = lambda args: args**(1/2)
print(squareroot(3.14))
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER

/usr/local/bin/python3 /Users/raulrodriguez/Documents/WorkSpaceVSPython/HW4_1.py
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % /usr/local/bin/python3 /Users/raulrodriguez/Documents/WorkSpaceVSPython/HW4_1.py
1.772004514666935
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython %
```

```python
'''Exercise 2
Given the following list: words = ['Anna', 'hELLo', 'rotor', 'wow', 'CS', 'kayAK', 'program-
ming'], use the filter() function to filter out the non-palindrome words, i.e., your output list
should contain palindrome words only
Note 1: You can use the built-in function upper() to convert a string to uppercase or the
round() function to round a number
Note 2: Your algorithm should be able to cope with both odd as well as even length words,
e.g., 'wow' or 'Anna'''
def palindromeWords(words):
    if(words == words[::-1]) or words.upper()==words.upper()[::-1]:
        return True
    else:
        return False

words = ['Anna', 'hELLo', 'rotor', 'wow', 'CS', 'kayAK', 'programming']
check = filter(palindromeWords,words)
print(list(check))
```

```
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % /usr/local/bin/python3 /Users/raulrodriguez/Documents/WorkSpaceVSPython/HW4_2.py
['Anna', 'rotor', 'wow', 'kayAK']
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython %
```

```python
'''Exercise 3
Given the following list: numbers = [23, 2, 9, 7, 14, 18, 3, 24, 16, 5, 8, 97], use the filter()
function to filter out non-prime numbers, i.e., your output list should contain Prime numbers
only
Note 1: A Prime number is a number that is divisible by 1 and by itself only, e.g., 2, 3, 5, 7, 11
Note 2: Do not manually hardcode any list that contains some of the Prime numbers and use
it for comparison against the numbers list'''
def primeNumbers(n):
    if n%2==0:
        return n==2
    for i in range(3,int(n**.5)+1,2):
        if n % i == 0:
            return False
    return True
numbers = [23, 2, 9, 7, 14, 18, 3, 24, 16, 5, 8, 97]
check = filter(primeNumbers,numbers)
print(list(check))
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER

raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % /usr/local/bin/python3 /Users/raulrodriguez/Documents/WorkSpaceVSPython/HW4_3.py
[23, 2, 7, 3, 5, 97]
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython %
```

```python
'''Exercise 4
Ask user to enter a sentence, e.g., Computer Science is an amazing field of study. Pass
sentence to function str2words() and convert the string/sentenct into words; place individual
words, e.g., 'Computer', 'Science', etc. into a list and return list to main program (your list
should contain 8 elements if the above sentence is entered). In addition, the individual words
should not contain any whitespaces appended to the end of the word such as 'Science '
Note: Do not use any built-in functions to convert a string to words'''
def str2words(sentence):
    li=[]
    words=''
    for i in range(len(sentence)):
        words+=sentence[i]
        if sentence[i]==' ' or sentence[i]=='.':
            words=words[:-1]
            li.append(words)
            words=''
    return li
sentence = input("Enter a sentence: ")
li = str2words(sentence)
print(li)
```

```
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % /usr/local/bin/python3 /Users/raulrodriguez/Documents/WorkSpaceVSPython/HW4_4.py
Enter a sentence: Computer Science is an amazing field of study.
['Computer', 'Science', 'is', 'an', 'amazing', 'field', 'of', 'study']
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython %
```