Week 3 Loops and Data Structure

Part 1 For loops

Exercise 1

1

In [1]:

```
Words = "Hello World"
for Letter in Words:
    print(Letter)
```

```
H
e
l
l
o

W
o
r
l
d
```

2

The program prints the sum of the even integers up to 10

3

If only one argument is in range(N), then the requested range will start from 0 up to N-1

In [2]:

```
for i in range(11):
    print(i)
```

4

the first argument is the start of the sequence the secondn is the stop and the last one is the step. You could also re-write the Sum of integers program using a step of 10 starting from 0 instead of using a modulo

In [3]:

```
for i in range(1,11,2):
    print(i)
```

```
1
3
5
7
9
```

5, 6

In [29]:

```
Words = "Hello World"
for i in range(0,len(Words)):
    print("Char "+Words[i]+" at position",i)
```

```
Char H at position 0
Char e at position 1
Char l at position 2
Char l at position 3
Char o at position 4
Char   at position 5
Char W at position 6
Char o at position 7
Char r at position 8
Char l at position 9
Char d at position 10
```

In [5]:

```
for pos,letter in enumerate(Words):
    print("Char "+letter+" at position",pos)
```

```
Char H at position 0
Char e at position 1
Char l at position 2
Char l at position 3
Char o at position 4
Char   at position 5
Char W at position 6
Char o at position 7
Char r at position 8
Char l at position 9
Char d at position 10
```

Exercise 2 While loops

1,2,4

you need to increment i otherwise you will still compare the first character of the string

In [6]:

```
Words = "Hello World"
TargetLetter = 'd'
i = 0
while Words[i] != TargetLetter and i+1 < len(Words):
    i += 1
print("Target letter is at position", i)
```

```
Target letter is at position 10
```

In [31]:

```
Words = "Hello World"
TargetLetter = 'o'
for i in range(len(Words)):
    if Words[i] == TargetLetter:
        print("Found "+ TargetLetter + " at postition", i)

Words.find("o")
Words.rfind("o")
```

```
Found o at postition 4
Found o at postition 7
```

Out[31]:

```
7
```

4,5

Won't do the camel case

In [5]:

```
cumulativeSum=0
counter=0

while cumulativeSum<1000000:
    counter+=+1
    cumulativeSum=cumulativeSum+(counter**2)
    print(cumulativeSum)
print("The answer is", str(counter))
```

1
5
14
30
55
91
140
204
285
385
506
650
819
1015
1240
1496
1785
2109
2470
2870
3311
3795
4324
4900
5525
6201
6930
7714
8555
9455
10416
11440
12529
13685
14910
16206
17575
19019
20540
22140
23821
25585
27434
29370
31395
33511
35720
38024
40425
42925
45526
48230
51039
53955
56980
60116
63365
66729
70210
73810
77531

81375
85344
89440
93665
98021
102510
107134
111895
116795
121836
127020
132349
137825
143450
149226
155155
161239
167480
173880
180441
187165
194054
201110
208335
215731
223300
231044
238965
247065
255346
263810
272459
281295
290320
299536
308945
318549
328350
338350
348551
358955
369564
380380
391405
402641
414090
425754
437635
449735
462056
474600
487369
500365
513590
527046
540735
554659
568820
583220
597861
612745

627874
643250
658875
674751
690880
707264
723905
740805
757966
775390
793079
811035
829260
847756
866525
885569
904890
924490
944371
964535
984984
1005720
The answer is 144

In [9]:

```
cumulativeSum=0
counter=0

while cumulativeSum<1000000:
    counter+=+1
    cumulativeSum=cumulativeSum+(counter**2)
    print(counter, cumulativeSum)
print("The answer is", str(counter))
```

1 1
2 5
3 14
4 30
5 55
6 91
7 140
8 204
9 285
10 385
11 506
12 650
13 819
14 1015
15 1240
16 1496
17 1785
18 2109
19 2470
20 2870
21 3311
22 3795
23 4324
24 4900
25 5525
26 6201
27 6930
28 7714
29 8555
30 9455
31 10416
32 11440
33 12529
34 13685
35 14910
36 16206
37 17575
38 19019
39 20540
40 22140
41 23821
42 25585
43 27434
44 29370
45 31395
46 33511
47 35720
48 38024
49 40425
50 42925
51 45526
52 48230
53 51039
54 53955
55 56980
56 60116
57 63365
58 66729
59 70210
60 73810
61 77531

62 81375
63 85344
64 89440
65 93665
66 98021
67 102510
68 107134
69 111895
70 116795
71 121836
72 127020
73 132349
74 137825
75 143450
76 149226
77 155155
78 161239
79 167480
80 173880
81 180441
82 187165
83 194054
84 201110
85 208335
86 215731
87 223300
88 231044
89 238965
90 247065
91 255346
92 263810
93 272459
94 281295
95 290320
96 299536
97 308945
98 318549
99 328350
100 338350
101 348551
102 358955
103 369564
104 380380
105 391405
106 402641
107 414090
108 425754
109 437635
110 449735
111 462056
112 474600
113 487369
114 500365
115 513590
116 527046
117 540735
118 554659
119 568820
120 583220
121 597861
122 612745

```
123 627874
124 643250
125 658875
126 674751
127 690880
128 707264
129 723905
130 740805
131 757966
132 775390
133 793079
134 811035
135 829260
136 847756
137 866525
138 885569
139 904890
140 924490
141 944371
142 964535
143 984984
144 1005720
The answer is 144
```

6

In [7]:

```
Words = "Hello World"
TargetLetter = 'o'
i = 0
for i in range(len(Words)):
   if(Words[i] == TargetLetter):
       break
print("Target letter is at position", i)
```

```
Target letter is at position 4
```

Exercise 3 More loops

1

In [33]:

```
import math  as mt
import random
```

In [47]:

```
A = random.randint(1,6)
B = random.randint(1,6) # if you use the function random.randrange() you should go to N
+1
# that function works similarly to range(), if you use randin() you can specify the ran
ge inclusive of the upper
# limit
counter = 0
while  A != B:
    print("Throwing the dices again", A, B)
    A = random.randint(1,6)
    B = random.randint(1,6)
    counter += 1
print("Finally A is equal to B", A, B,"tries", counter)
```

```
Throwing the dices again 4 2
Finally A is equal to B 6 6 tries 1
```

2

Refer to provided code

Part 2 Data structures

Exercise 4 List

1,2,3,4,5

In [49]:

```
a = [1,2]
b = [3,4]
a[0] = 5
print(a)
a.sort()
print(a)
# nested list
c = [a,b]
print(c)
for sublist in c:
    for element in sublist:
        print(element)
```

```
[5, 2]
[2, 5]
[[2, 5], [3, 4]]
2
5
3
4
```

6

refer to the list_words.py code

In [8]:

```
a =[x for x in range(101) if x%2 != 0]
print(a)
b =[x for x in range(101) if x%3 == 0]
print(b)
c = [x for x in range(101) if all(x % y != 0 for y in range(2, x)) and x > 1]
print(c)
```

```
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 3
9, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75,
77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99]
[0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 5
7, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99]
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 7
1, 73, 79, 83, 89, 97]
```

Exercise 5 Tuples

2,3,4,5

In [3]:

```
FondueIngredients = ("gruyere","vacherin")
print(FondueIngredients)
FondueIngredients[0] = "Cheddar"
```

```
('gruyere', 'vacherin')

---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
<ipython-input-3-e616b5661bb7> in <module>
      1 FondueIngredients = ("gruyere","vacherin")
      2 print(FondueIngredients)
----> 3 FondueIngredients[0] = "Cheddar"

TypeError: 'tuple' object does not support item assignment
```

tuples are immutable, to change it convert it to a list and back

In [5]:

```
FondueIngredientsList = list(FondueIngredients)
FondueIngredientsList[0] = "Cheddar"
FondueIngredients = tuple(FondueIngredientsList)
print(FondueIngredients)
```

```
('Cheddar', 'vacherin')
```

```
---------------------------------------------------------------------------
-
AttributeError                            Traceback (most recent call las
t)
<ipython-input-5-df4c56efaa72> in <module>
      4 print(FondueIngredients)
      5 # to remove the last element use the pop method
----> 6 FondueIngredients.remove("Cheddar")

AttributeError: 'tuple' object has no attribute 'remove'
```

Exercise 6 - Sets

2,3,4,5,6

Duplicates are removed
sets can't be accessed using indices
visit https://www.w3schools.com/python/python_sets.asp (https://www.w3schools.com/python/python_sets.asp) for more

In [11]:

```
s1 = {1,2,5,5,8}
s2 = {1,2,4,9,2}
print(s1)
print(s2)
```

```
{8, 1, 2, 5}
{1, 2, 4, 9}
```

In [12]:

```
s1[2]
```

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
<ipython-input-12-7087e83d753a> in <module>
----> 1 s1[2]

TypeError: 'set' object is not subscriptable
```

In [51]:

```
print(4 in s1 and 4 in s2)
```

```
False
```

In [14]:

```
print(s2 & s1)
print(s2 | s1)
print(s1 - s2)
print(s2 - s1) ## notice that that the - operator is not symmetric
print(s2 ^ s1) ## the combinatrion of the 2 above
```

```
{1, 2}
{1, 2, 4, 5, 8, 9}
{8, 5}
{9, 4}
{4, 5, 8, 9}
```

In [15]:

```
s1 = {1,2,5,5,8}
s1.remove(1)
print(s1)
s1.add(6)
print(s1)
```

```
{8, 2, 5}
{2, 5, 6, 8}
```

Exercise 7 Dictionaries

1,2,3,4,5

In [16]:

```
mydict = {"Jill":21, "Sally":20, "Bob":20, "Harry":21}
```

In [19]:

```
mydict.keys()
mydict.values()
```

Out[19]:

```
dict_values([21, 20, 20, 21])
```

In [ ]:

```
mydict["Rachel"] = 19
```

In [54]:

```
mydict = {"Jill":21, "Sally":20, "Bob":20, "Harry":21}
print(mydict.pop("Bob"))
print(mydict)
```

```
20
{'Jill': 21, 'Sally': 20, 'Harry': 21}
```

In [56]:

```
mydict["Jill"] = 24
mydict
```

Out[56]:

```
{'Jill': 24, 'Sally': 20, 'Harry': 21}
```

In [28]:

```
print("Harry" in mydict)
```

```
True
```

Exercise 8 - FizzBuzz Game

refer to code FizzBuzz.py

In [ ]:

```

```