Exercise 1 - For loops

Question 1.

In [1]:
```
Words = 'New word'
for Letter in Words:
    print(Letter)
```
```
N
e
w

w
o
r
d
```

Question 2. Question2.py sums all of the even integers between 1 and 10

Question 3. The value of 11 is used to indicate that the final number should be 10. This leads to `range(1, 11, 1)` producing the values 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10.

Question 4. Calling `range(1, 11, 2)` will produce integers between 1 and 10 in steps of 2. That is, it will produce the numbers 1, 3, 5, 7, 9. These are all of the odd numbers between 1 and 10.

Question 5.

In [8]:
```
Words = 'Hello World'

# use len to compute the length of the string
num_char = len(Words)

for c in range(num_char):
    print('The letter', Words[c], 'is at position', c)
```
```
The letter H is at position 0
The letter e is at position 1
The letter l is at position 2
The letter l is at position 3
The letter o is at position 4
The letter   is at position 5
The letter W is at position 6
The letter o is at position 7
The letter r is at position 8
The letter l is at position 9
The letter d is at position 10
```

Question 6. There are three mistakes here:

1. the second argument to `range` should be 11
2. the colon is missing
3. the indent is missing

In [1]:
```
sum = 0

for i in range(1,11):
    sum += 5 * i

print(sum)
```
```
275
```

Question 7

In [7]:
```
n = 10
factorial = 1

for i in range(1, n):
    factorial *= i

print(factorial)
```
```
362880
```

Question 8

In [6]:
```
num_cubes = 0

for n in range(1, 2000):
    if n**3 < 2000:
        num_cubes += 1
    else:
        break

print(num_cubes)
```
```
12
```

Exercise 2 - While loops

Question 1.

In [39]:
```
Word = "Hello World"
TargetLetter = "W"

i = 0
while TargetLetter != Word[i]:
    i += 1
print("Target letter is at position", i)
```
```
Target letter is at position 6
```

We need to increment i to avoid an infinite loop

Question 2. It is best to use a `for` loop for this problem because we want to compare TargetLetter against all of the letters in Word

In [41]:
```
Word = "Hello World"
TargetLetter = "o"

i = 0
for l in Word:
    if l == TargetLetter:
        print("Target letter is at position", i)
    i += 1
```
```
Target letter is at position 4
Target letter is at position 7
```

Question 3.

In [43]:
```
Word.find('W')
```
Out[43]:
```
6
```

Question 4.

In [45]:
```
cumulativeSum=0
counter=0

while cumulativeSum<1000000:
    counter += 1
    cumulativeSum += (counter**2)

print("The answer is", str(counter))
```
```
The answer is 144
```

Question 5.

In [46]:
```
cumulativeSum=0
counter=0

while cumulativeSum<1000000:
    counter += 1
    cumulativeSum += (counter**2)
    print('The cumulative sum after', counter, 'terms is', cumulativeSum)

print("The answer is", str(counter))
```
```
The cumulative sum after 1 terms is 1
The cumulative sum after 2 terms is 5
The cumulative sum after 3 terms is 14
The cumulative sum after 4 terms is 30
The cumulative sum after 5 terms is 55
The cumulative sum after 6 terms is 91
The cumulative sum after 7 terms is 140
The cumulative sum after 8 terms is 204
The cumulative sum after 9 terms is 285
The cumulative sum after 10 terms is 385
The cumulative sum after 11 terms is 506
The cumulative sum after 12 terms is 650
The cumulative sum after 13 terms is 819
The cumulative sum after 14 terms is 1015
The cumulative sum after 15 terms is 1240
The cumulative sum after 16 terms is 1496
The cumulative sum after 17 terms is 1785
The cumulative sum after 18 terms is 2109
The cumulative sum after 19 terms is 2470
The cumulative sum after 20 terms is 2870
The cumulative sum after 21 terms is 3311
The cumulative sum after 22 terms is 3795
The cumulative sum after 23 terms is 4324
The cumulative sum after 24 terms is 4900
The cumulative sum after 25 terms is 5525
The cumulative sum after 26 terms is 6201
The cumulative sum after 27 terms is 6930
The cumulative sum after 28 terms is 7714
The cumulative sum after 29 terms is 8555
The cumulative sum after 30 terms is 9455
The cumulative sum after 31 terms is 10416
The cumulative sum after 32 terms is 11440
The cumulative sum after 33 terms is 12529
The cumulative sum after 34 terms is 13685
The cumulative sum after 35 terms is 14910
The cumulative sum after 36 terms is 16206
The cumulative sum after 37 terms is 17575
The cumulative sum after 38 terms is 19019
The cumulative sum after 39 terms is 20540
The cumulative sum after 40 terms is 22140
The cumulative sum after 41 terms is 23821
The cumulative sum after 42 terms is 25585
The cumulative sum after 43 terms is 27434
The cumulative sum after 44 terms is 29370
The cumulative sum after 45 terms is 31395
The cumulative sum after 46 terms is 33511
The cumulative sum after 47 terms is 35720
The cumulative sum after 48 terms is 38024
The cumulative sum after 49 terms is 40425
The cumulative sum after 50 terms is 42925
The cumulative sum after 51 terms is 45526
The cumulative sum after 52 terms is 48230
The cumulative sum after 53 terms is 51039
The cumulative sum after 54 terms is 53955
The cumulative sum after 55 terms is 56980
The cumulative sum after 56 terms is 60116
The cumulative sum after 57 terms is 63365
The cumulative sum after 58 terms is 66729
The cumulative sum after 59 terms is 70210
The cumulative sum after 60 terms is 73810
The cumulative sum after 61 terms is 77531
The cumulative sum after 62 terms is 81375
The cumulative sum after 63 terms is 85344
The cumulative sum after 64 terms is 89440
The cumulative sum after 65 terms is 93665
The cumulative sum after 66 terms is 98021
The cumulative sum after 67 terms is 102510
The cumulative sum after 68 terms is 107134
The cumulative sum after 69 terms is 111895
The cumulative sum after 70 terms is 116795
The cumulative sum after 71 terms is 121836
The cumulative sum after 72 terms is 127020
The cumulative sum after 73 terms is 132349
The cumulative sum after 74 terms is 137825
The cumulative sum after 75 terms is 143450
The cumulative sum after 76 terms is 149226
The cumulative sum after 77 terms is 155155
The cumulative sum after 78 terms is 161239
The cumulative sum after 79 terms is 167480
The cumulative sum after 80 terms is 173880
The cumulative sum after 81 terms is 180441
The cumulative sum after 82 terms is 187165
The cumulative sum after 83 terms is 194054
The cumulative sum after 84 terms is 201110
The cumulative sum after 85 terms is 208335
The cumulative sum after 86 terms is 215731
The cumulative sum after 87 terms is 223300
The cumulative sum after 88 terms is 231044
The cumulative sum after 89 terms is 238965
The cumulative sum after 90 terms is 247065
The cumulative sum after 91 terms is 255346
The cumulative sum after 92 terms is 263810
The cumulative sum after 93 terms is 272459
The cumulative sum after 94 terms is 281295
The cumulative sum after 95 terms is 290320
The cumulative sum after 96 terms is 299536
The cumulative sum after 97 terms is 308945
The cumulative sum after 98 terms is 318549
The cumulative sum after 99 terms is 328350
The cumulative sum after 100 terms is 338350
The cumulative sum after 101 terms is 348551
The cumulative sum after 102 terms is 358955
The cumulative sum after 103 terms is 369564
The cumulative sum after 104 terms is 380380
The cumulative sum after 105 terms is 391405
The cumulative sum after 106 terms is 402641
The cumulative sum after 107 terms is 414090
The cumulative sum after 108 terms is 425754
The cumulative sum after 109 terms is 437635
The cumulative sum after 110 terms is 449735
The cumulative sum after 111 terms is 462056
The cumulative sum after 112 terms is 474600
The cumulative sum after 113 terms is 487369
The cumulative sum after 114 terms is 500365
The cumulative sum after 115 terms is 513590
The cumulative sum after 116 terms is 527046
The cumulative sum after 117 terms is 540735
The cumulative sum after 118 terms is 554659
The cumulative sum after 119 terms is 568820
The cumulative sum after 120 terms is 583220
The cumulative sum after 121 terms is 597861
The cumulative sum after 122 terms is 612745
The cumulative sum after 123 terms is 627874
The cumulative sum after 124 terms is 643250
The cumulative sum after 125 terms is 658875
The cumulative sum after 126 terms is 674751
The cumulative sum after 127 terms is 690880
The cumulative sum after 128 terms is 707264
The cumulative sum after 129 terms is 723905
The cumulative sum after 130 terms is 740805
The cumulative sum after 131 terms is 757966
The cumulative sum after 132 terms is 775390
The cumulative sum after 133 terms is 793079
The cumulative sum after 134 terms is 811035
The cumulative sum after 135 terms is 829260
The cumulative sum after 136 terms is 847756
The cumulative sum after 137 terms is 866525
The cumulative sum after 138 terms is 885569
The cumulative sum after 139 terms is 904890
The cumulative sum after 140 terms is 924490
The cumulative sum after 141 terms is 944371
The cumulative sum after 142 terms is 964535
The cumulative sum after 143 terms is 984984
The cumulative sum after 144 terms is 1005720
The answer is 144
```

Question 6.

In [47]:
```
Word = "Hello World"
TargetLetter = "W"
Enum = list(enumerate(Word))
for e in Enum:
    if e[1] == TargetLetter:
        print("Target letter is at position", e[0])
        break
```
```
Target letter is at position 6
```

Exercise 3 - More loops

Question 1

In [9]:
```
S = 0

for i in range(1,11):
    for j in range(0,6):
        S += j * j * (i + j)

print(S)
```
```
5275
```

Question 2. The code to compute π_N for a fixed value of N is:

In [31]:
```
# define the exact value of pi
pi = 3.141592653589793

# create a variable for the approximate value of pi
pi_N = 0

# number of terms in the series
N = 100

# use a for loop to compute the series
for n in range(N):
    pi_N += 8 / (4 * n + 1) / (4 * n + 3)

print(pi_N)
error = abs(pi - pi_N)
print('The error with N =', N, 'is', error)
```
```
3.1365926848388144
The error with N = 100 is 0.004999968750978745
```

We find that:

N	error
100	5.0e-3
1000	5.0e-4
10000	5.0e-5

In [28]:
```
# define the exact value of pi
pi = 3.141592653589793

# create a variable for the approximate value of pi
pi_N = 0

# the number of terms in the sum
n = 0
while abs(pi - pi_N) > 1e-6:
    pi_N += 8 / (4 * n + 1) / (4 * n + 3)
    n += 1

print(pi_N)
print(n, 'terms are needed for an error that is less than 1e-6')
```
```
3.1415916535899084
500000 terms are needed for an error that is less than 1e-6
```

Question 3: The code to compute the number of Fibonacci numbers is:

In [34]:
```
# first Fibonacci number
a = 0

# second Fibonacci number
b = 1

# number of Fibonacci numbers
num = 2

while a + b < 100:

    # compute next Fibonacci number and update counter
    c = a + b
    num += 1

    # update the values of a and b
    a = b
    b = c

print('There are', num, 'Fibonacci numbers below 100')
```
```
There are 12 Fibonacci numbers below 100
```

There are 12 numbers below 100, 17 numbers below 1,000, and 21 numbers below 10,000.

Question 4.

In [102]:
```
import math
import random

# create random integers A and B that lie in the range 1 to 6 (inclusive)
A = random.randint(1, 6)
B = random.randint(1, 6)

# create a variable to count the number of rolls
roll = 1
while A != B:

    A = random.randint(1, 6)
    B = random.randint(1, 6)

    roll += 1

print(roll, 'rolls were needed for the dice to have the same number')
```
```
11 rolls were needed for the dice to have the same number
```

Question 5. See number-guessing-game.py

Exercise 4 - Lists

Question 1.

In [48]:
```
a = [1, 2]
b = [3, 4]
```

Question 2.

In [49]:
```
a[0] = 5
```

Question 3.

In [50]:
```
a.sort()
b.sort()

print(a)
print(b)
```
```
[2, 5]
[3, 4]
```

Question 4.

In [52]:
```
nested = [a, b]
print(nested)
```
```
[[2, 5], [3, 4]]
```

Question 5.

In [55]:
```
# the first loop; l is an element of nested so will be a list
for l in nested:
    # the second list; e is an element of the list l
    for e in l:
        print(e)
```
```
2
5
3
4
```

Question 6. See the list_words.py file

Question 7.

In [57]:
```
l = [e for e in range(101) if e % 2 != 0]
print(l)
```
```
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 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]
```

In [59]:
```
l = [e for e in range(101) if e % 3 == 0]
print(l)
```
```
[0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, 63, 66, 69, 72, 75,
78, 81, 84, 87, 90, 93, 96, 99]
```

In [62]:
```
l = [e for e in range(101) if all(e % x != 0 for x in range(2, e)) and e > 1]
print(l)
```
```
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]
```

Exercise 5 - Tuples

Question 2.

In [63]:
```
FondueIngredients = ('gruyere', 'vacherin')
```

Question 3.

In [64]:
```
print(FondueIngredients)
```
```
('gruyere', 'vacherin')
```

Question 4. It is not possible to change "gruyere" to "cheddar" becauase tuples are immutable

Question 5. There is no function to remove the last element of a tuple. Instead, one can convert a tuple to a list and then convert it back.

Exercise 6 - Sets

Question 2.

In [65]:
```
s1 = {1, 2, 5, 5, 8}
s2 = {1, 2, 4, 9, 2}
print(s1)
print(s2)
```
```
{8, 1, 2, 5}
{1, 2, 4, 9}
```

There are no duplicate entries

Question 3. No, elements of sets cannot be accessed with indices

Question 4.

In [66]:
```
4 in s1
```
Out[66]:
```
False
```

In [67]:
```
4 in s2
```
Out[67]:
```
True
```

Question 5. & is intersection, | is union, - is difference, and ^ is symmetric difference

In [68]:
```
s1 & s2
```
Out[68]:
```
{1, 2}
```

In [69]:
```
s1 | s2
```
Out[69]:
```
{1, 2, 4, 5, 8, 9}
```

In [70]:
```
s1 - s2
```
Out[70]:
```
{5, 8}
```

In [71]:
```
s1 ^ s2
```
Out[71]:
```
{4, 5, 8, 9}
```

Question 6. Elements of a set can be removed using the `remove` or `discard` method. Adding an element to a set can be performed using the `add` method

In [76]:
```
s1.remove(1)
s1.add(6)
print(s1)
```
```
{2, 5, 6, 8}
```

Exercise 7 - Dictionaries

Question 2.

In [77]:
```
ages = {'Jill':21, 'Sally':20, 'Bob':20, 'Harry':21}
```

Question 3. The `keys` method can be used to create a list with all of the keys in a dictionary

In [86]:
```
for l in ages.keys():
    print(l)
```
```
Jill
Sally
Bob
Harry
```

Question 4. A key-value pair can be added as follows:

In [88]:
```
ages['Rachel'] = 19
print(ages)
```
```
{'Jill': 21, 'Sally': 20, 'Bob': 20, 'Harry': 21, 'Rachel': 19}
```

Question 5. The `pop` method can be used to remove a key-value pair

In [89]:
```
ages.pop('Bob')
print(ages)
```
```
{'Jill': 21, 'Sally': 20, 'Harry': 21, 'Rachel': 19}
```

Question 6. It is not possible to create duplicate entries in a dictionary

In [93]:
```
ages['Jill'] = 22
print(ages)
```
```
{'Jill': 22, 'Sally': 20, 'Harry': 21, 'Rachel': 19}
```

Question 7. The `in` keyword can be used to check whether a key exists in a dictionary

In [94]:
```
print('Harry' in ages)
```
```
True
```

Exercise 8 - FizzBuzz Game

See the file FizzBuzz.py