

Session 3: Loop Logic

Discussion Question: Referral Marketing

A start-up's marketing team is trying to predict the growth of a new product launch. Assuming that each person who buys the product will refer a friend (who hasn't seen the product before) in the next month, as well as two months after the sale, but will stop referring after that. Each referred friend will buy the product upon hearing about it and follow the same pattern of referral.

a) Assuming that there is one person who buys the product in month zero. How many people would have bought the product in month 12, assuming the only buyers are those referred?

b) What is the first month in which the monthly sales exceeds 100?

Loop Examples: Counting Permutations

For loops are used for iterating through a given collection of items. The following for loop is used to calculate $n! = 1 \times 2 \times \dots \times n$, which is the number of ways to arrange n distinct items in order.

```
[1]: n=5
    factorial=1
    for i in range(1,n+1):
        factorial=factorial*i
    print(f'{n}! is equal to {factorial}.')
```

5! is equal to 120.

It is often helpful to use auxiliary print statements to illustrate what's going on in each iteration of a loop, as follows:

```
[2]: n=5
    factorial=1
    print('i\tfactorial')
    print(f'-\t{factorial}')
    for i in range(1,n+1):
        factorial=factorial*i
        print(f'{i}\t{factorial}')
    print(f'\n{n}! is equal to {factorial}.')
```

i	factorial
-	1
1	1
2	2
3	6
4	24
5	120

5! is equal to 120.

While loops are used to iterate as long as a given condition is met. The following while loop finds the smallest n such that $n!$ is at least 1000.

```
[3]: factorial=1
    i=1
    while factorial<1000:
        i=i+1
        factorial=factorial*i
    n=i
    print (f'{n}!>=1000.')
```

7!>=1000.

Adding auxiliary print statements reveals what is going on in each iteration.

```
[4]: factorial=1
    i=1
    print('i\tfactorial\tfactorial<1000?')
    print(f'{i}\t{factorial}\t\t{factorial<1000}')
    while factorial<1000:
        i=i+1
        factorial=factorial*i
        print(f'{i}\t{factorial}\t\t{factorial<1000}')
    n=i
    print (f'{n}!>=1000.')
```

i	factorial	factorial<1000?
1	1	True
2	2	True
3	6	True
4	24	True
5	120	True
6	720	True
7	5040	False

7!>=1000.

For loops and while loops are to a certain extent interchangeable, although one is usually more convenient than the other depending on the circumstance. The following examples replicate the above using the other kind of loop.

```
[5]: n=5
    factorial=1
    i=1
    while i<=n:
        factorial=factorial*i
        i+=1
    print(f'{n}! is equal to {factorial}.')
```

5! is equal to 120.

```
[6]: factorial=1
    for i in range(1,9999999):
        factorial=factorial*i
        if factorial>=1000:
            break
    n=i
    print (f'{n}!>=1000.')
```

7!>=1000.

Case 4. Referral Marketing

Solve the discussion question using for or while loops. In each case, denote the numerical inputs (12, 100, and 100) by a variable so that your code would work with any other input. In particular, find the monthly sale in month 36, and number of months before monthly sale exceeds 1 million. **Hint:** Use a list to store the sales for each month.

```
[7]: # Solution to part a)
```

The number of items sold in month 36 is 24157817.

```
[11]: # Solution to part b)
```

In month 30, the sales is 1346269, which exceeds 1000000.

Case 5. Hospital Capacity Planning

A hospital is applying for funding to expand its number of beds. Given a list of estimated number of arrivals each day, and assuming each patient stays 3 days, how many beds does the hospital need so as to be able to not turn away anyone?

```
demand=[5,8,3,10,7,4,9,5,8]
```

```
[16]:
```

The number of beds needed to accommodate demand is 22.