20 August

- finalise groups by thursday night

Dictionary Data Structure
- key/index not in sequence order but user defined
- key/index : value pair of items
e.g.
```
a = {}
a[1] = "something"
a[4] = "potato"
```

Set Data Structure
- all unique values inside the data structure
e.g.
```
b = set(a)
```

Tuple Data Structure
- non mutable/changeable lists after definition
e.g.
```
c = (1,2,3)
```

Loops
- For loop
e.g.
```
for x in a:
    print(x)
# range(a,b) := [a,b)
for x in range(0,10):
    for y in range(0,10):
        print(str(x) + " x " + str(y) + " = " + str(x*y))
```

break and continue
- used for skipping an iteration or breaking out of the loop
e.g.
```
for x in range(0,100):
    if x%10 == 0:
        continue
    elif x == 78:
        break
    else:
        print(x)
```

Loops
- While loop
e.g.
```
x = 50
while(x>=20):
    print(x)
    x -= 1
```

Function
- returns an output for some input
e.g.
```
def avg(x, y, z):
    return (x+y+z)/3
def funct1(x, y, z):
    a = x+y+z
    return a
```

Fibonacci series: 1, 1, 2, 3, 5, 8 ...

Fibonacci using loop
# Loops - using processor
e.g.
```
x = 1
y = 1
print(y)
while True:
    z = x + y
    x = y
    y = z
    print(y)
```

Fibonacci using function
# Recursion - using RAM
e.g.
```
x = 1
y = 1
def fib1(x, y):
    print(x + y)
    fib1(y, x + y)
def fib(x, y):
    print(x)
    print(y)
    fib1(x, y)
```

pip - package installer for python
-numpy, scipy etc... python libraries

Python Libraries
- Numpy, Scipy library for calculations and stuff
- Matplotlib library for visualization
- Scikit-learn library for various ML algo implementation
- lasagne and keras library for DL algo implementation
- nltk library for natural language processing

Natural Language Processing (NLP):
early objective: part of speech tagging
- label every word with a part of speech
- supervised classification task
    * different from normal supervised classification (as context)
    e.g. 1. I bat for this team.
         2. This is a bat.
         Note: 1. bat - verb
               2. bat - noun
- segmentation
    * breaking sentence into segments/parts
- tokenization
    * converting segments into tokens/numbers
- Named Entity Detection
    * \start and \stop tokens added before words to find specific phrases/named entities in text
    * State of the Art Models also give ~70-80% accuracy
- stanford coreNLP library
    * for NLP algo implementation - not SOTA but pretty good girl
- most NLP datasets based on news related texts, journalistic text
- direct and not have lot of references, therefore need better datasets for coreference resolution
    * e.g. confusing text with lot of references asked while quizzing