To download Anaconda and install it

https://www.anaconda.com/download

Dictionary:

1. It is mutable.
2. It stores key, value pair. Keys are unique. Any immutable object can be the key
3. It is ordered collection.
4. The value can be retrieved very fast, if you know the key. But if you want to find the key based on value, then you need to navigate through the dictionary.
5. Dictionary is represented by using {}
6. To create a empty dictionary you can use, d={} , to create empty set s=set()

To add a key

C={}

c['DAC']=240 # a new key -value pair will be added

c['DAC']=300 #if key exists it will overwrite the value

D1.update(d2)	Add all keys of d2 in d1, if any of the key is common, then d1's keys value will be overwritten by d2's keys value D1={'a':10,'b':20} d2={'b':300,'c':30} d1.update(d2) will change d1 ={'a':10,'b':300,'c':30}
D1.fromkeys(lst,default value)	create a new dictionary by adding every value from list as key value of all keys will be None, if the default value is not given
d.pop(key,[default value)	It will delete the pair of key-value if the given key exists and returns the value. But if key does not exist: And default value is given, then it will return default value, otherwise it will throw andexception
d.popitem()	It will delete the last key-value pair, it will return a tuple
d.copy()	It will create a shallow copy of the dictionary
d.clear()	It will delete all key-value pairs from the the dictionary
d.get(key,[default value])	It will return the value of the given key, if key exists, otherwise, if key does not exists, and if default value is not given, then it will return None, if deafault value is given, it will return default value
d.setdefault(key,[default value])	It will return the value of the given key, if key exists, otherwise, if key does not exist, and it will add a new key-value pair in the dictionary and returns default value.
d.Keys()	It will return all keys
d.values()	It will return all values
d.items()	It will return a iterable, each value will be a tuple of (key,value)

Regular expression

\d	One single digit

\D	One non digit character
\b	Boundary character
\B	Nonboundry characters
\w	One word character[0-9A-Za-z_]
\W	Non word charcaters
\s	One space character
\S	One nonspace character

	or	^or	Or$	\bor\b	\bor	
There is tailor	Y	N	Y	N	n	
There is a origami class	Y	N	N	N	Y	
This or that	Y	N	N	Y	Y	
Origami is good	y	Y	N	N	Y	
This is normal	y	n	n	n	n	

S=This is string

^\w+\s\w+\s\w+$---matches all the lines which has exact 3 words separated by space

^\w+\s\w+\s\w+-→ matches all the line which has atleast 3 words

\w+\s\w+\s\w+$-→ matches with lines ending with 3 words, before that anything can be there

S="something is there somewhere"

s.*e

S="something is there somewhere"

s.*?e

import re

re.search(pattern ,string, ,flags)	It will give the MatchObject with information about first occurrence of the string that matches with the pattern, the pattern can match anywhere in the string
re.match(pattern ,string, flags)	It will give the MatchObject with information about first occurrence of the string that matches with the pattern, the pattern can match only at the beginning of the string, even if ^ is not there still the pattern will be matched at the beginning

Re.findall(pattern ,string, ,flags)	It will return a list of string, where the pattern matches
re.finditer(pattern,string, flags)	It will return a list of matchObjects, where the pattern matches
Re.sub(pattern,string,newstring,flags)	It will replace all occurrences of the given pattern with the new string
Re.compile(pattern,flags)	Returns the regular expression object