

Chap.-3 R.E.(Regular Expression)

• Metacharacters

- [] - Set of characters
- \ - Signals a special sequence
- (Escape special characters)
- .(dot) - Any character
- ^ - Starts with (but in [...] - inverse)
- \$ - Ends with - it's only check at 0 index
- '*' - Zero or More occurance
- '+' - One or More occurance
- '?' - Zero or One occurance
- {} - Exact occurance
- ! - Either or
- \d - only digit(0-9)
- \D - Except Digit Everything
- \s - whitespace
- \w - AlphaNumeric [a-z, A-Z, 0-9]
- \W - (Not a Word Character): Matches any character that is not a letter, digit, or underscore. This includes punctuation, spaces, and other symbols.
- \b - begining or end of a word
- \B - NOT or end of a word

• re.findall(pattern, string)

```
In [1]: 1 import re
        2 text = 'This is the Python Programming class of 2025'
        3 reg = r't\w*'
        4 ans = re.findall(reg,text)
        5 print(ans) # ['the', 'thon']
```

['the', 'thon']

Type *Markdown* and LaTeX: α^2

```
In [2]: 1 import re
        2 text = 'This is the Python Programming class of 2025'
        3 reg = r'\bt\w*'
        4 ans = re.findall(reg,text)
        5 print(ans)
```

['the']

```
In [4]: 1 import re
        2 text = 'This is the Python Programming class of 2025'
        3 reg = r'\b[t|T]\w*'
        4 ans = re.findall(reg,text)
        5 print(ans)
```

['This', 'the']

```
In [5]: 1 import re
        2 text = 'This is the Python Programming class of 2025'
        3 reg = r'\bt\w*'
        4 ans = re.findall(reg,text,re.I)
        5 print(ans)
```

['This', 'the']

• re.I - Ignore Case(Capital or small).

• re.search(reg,text)

```
In [6]: 1 import re
        2 text = 'This is the Python Programming class of 2025'
        3 reg = r'\bt\w*'
        4 ans = re.search(reg,text,re.I)
        5 print(ans)
```

<re.Match object; span=(0, 4), match='This'>

```
In [7]: 1 import re
2 text = 'This is the Python Programming class of 2025'
3 reg = r'\bt\w*'
4 ans = re.search(reg, text, re.I)
5 print(ans.group())
```

This

```
In [8]: 1 import re
2 text = 'This is the Python Programming class of 2025'
3 reg = r'\bt\w*'
4 ans = re.search(reg,text,re.I)
5 print(ans.span())
```

 $(0, 4)$

```
In [11]: 1 import re
2 text = 'This is the Python Programming class of 2025'
3 reg = r'\bt\w*'
4 ans = re.search(reg,text,re.I)
5 print(ans.start())
```

0

```
In [12]: 1 import re
2 text='This is the python Programming class of 2025'
3 reg=r'\bz\w*'
4 ans=re.search(reg,text,re.I)
5 print(ans)
6 # if not in string then return none and other method dont work on none so it gives error
```

None

```
In [16]: 1 import re
          2 text = 'This is the Pyth-on Prog*ramming class of 2025'
          3 reg = r'\W'
          4 ans = re.split(reg,text)
          5 print(ans)
```

```
['This', 'is', 'the', 'Pyth', 'on', 'Prog', 'ramming', 'class', 'of', '2025']
```

```
In [17]: 1 import re
2 text = 'This is the Pyth-on Prog*ramming class of 2025'
3 reg = r'\W'
4 ans = re.split(reg,text,2) # max 2 split
5 print(ans) # ['This', 'is', 'the Pyth-on Prog*ramming class of 2025']
6
7 # if 2 split then 3 peice of String
```

```
['This', 'is', 'the Pyth-on Prog*ramming class of 2025']
```

```
In [18]: 1 import re
          2 text = 'This is the Pyth-on Prog*ramming class of 2025'
          3 reg = r'\w'
          4 ans = re.split(reg,text)
          5 print(ans)
```

[illegible]

- **re.sub(reg,'\$',text)** - for replace

```
In [20]: 1 import re
          2 text = 'This is the Pyth-on Prog*ramming class of 2025'
          3 reg = r'\s'
          4 ans = re.sub(reg, '$', text)
          5 print(ans)
```

This is the Pyth-on Prog*ramming class of 2025

```
In [23]: 1 # replace digits with P
2 import re
3 text = 'This is the Pyth-on Prog*ramming class of 2025'
4 reg = r'\d'
5 ans = re.sub(reg, 'P', text)
6 print(ans)
```

This is the Pyth-on Prog*ramming class of PPPP

```
In [24]: 1 import re
2 text = 'This is the Pyth-on Prog*ramming class of 2025'
3 reg = r'\d+'
4 ans = re.sub(reg, 'P', text)
5 print(ans)
```

This is the Pyth-on Prog*ramming class of P

```
In [33]: 1 import re
2 text = ' This is the Pyth-on Prog*ramming class of 2025'
3 reg = r'^\s'
4 ans = re.sub(reg, '$', text)
5 print(ans)
6 # starting with
```

\$This is the Pyth-on Prog*ramming class of 2025

```
In [41]: 1 import re
2 text = ' This is the Pyth-on Prog*ramming class of 2025 '
3 reg = r'^\s|\s$'
4 ans = re.sub(reg, '$', text)
5 print(ans)
```

\$This is the Pyth-on Prog*ramming class of 2025\$

```
In [40]: 1 import re
2 text = ' This is the Pyth-on Prog*ramming class of 2025   '
3 reg = r'^\s|\s$'
4 ans = re.sub(reg, '$', text)
5 print(ans)
```

\$This is the Pyth-on Prog*ramming class of 2025 \$

```
In [39]: 1 import re
2 text = ' This is the Pyth-on Prog*ramming class of 2025   '
3 reg = r'^\s|\s*$'
4 ans = re.sub(reg, '$', text)
5 print(ans)
```

\$This is the Pyth-on Prog*ramming class of 2025\$\$

```
In [42]: 1 import re
2 text = ' This is the Pyth-on Prog*ramming class of 2025   '
3 reg = r'^\s|\s+$'
4 ans = re.sub(reg, '$', text)
5 print(ans)
```

\$This is the Pyth-on Prog*ramming class of 2025\$

```
In [57]: 1 import re
2 text = 'This is the Pyth-on Prog*ramming class of 2025'
3 reg = r'^\bP\w*'
4 ans = re.findall(reg, text)
5 print(ans)
```

[]

```
In [58]: 1 import re
2 text = 'PThis is the Pyth-on Prog*ramming class of 2025'
3 reg = r'^\bP\w*'
4 ans = re.findall(reg, text)
5 print(ans)
```

['PThis']

```
In [59]: 1 import re
2 text = 'PThis is the Pyth-on Prog*ramming class of 2025'
3 reg = r'^P*'
4 ans = re.findall(reg, text)
5 print(ans)
```

['P']

In [72]:

```
1 import re
2 f = open('alice_in_wonderland.txt', encoding='utf-8')
3 text = f.read()
4 print(text)
```

Alice's Adventures in Wonderland

ALICE'S ADVENTURES IN WONDERLAND

Lewis Carroll

THE MILLENNIUM FULCRUM EDITION 3.0

CHAPTER I

Down the Rabbit-Hole

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no

In [71]:

```
1 import re
2 f = open('alice_in_wonderland.txt', encoding='utf-8')
3 text = f.read()
4 # print(text)
5 reg = r'\b[aeiou]\w*'
6 ans = re.findall(reg,text)
7 print(ans)
```

['in', 'of', 'on', 'and', 'of', 'once', 'or', 'into', 'it', 'or', 'in', 'it', 'and', 'is', 'use', 'of', 'a', 'or', 'i
n', 'own', 'as', 'as', 'and', 'of', 'a', 'of', 'up', 'and', 'a', 'eyes', 'in', 'it', 'out', 'of', 'itself', 'it', 'ov
er', 'afterwards', 'it', 'occurred', 'ought', 'at', 'at', 'it', 'all', 'actually', 'and', 'at', 'it', 'and', 'on', 'i
t', 'across', 'a', 'either', 'a', 'or', 'a', 'out', 'of', 'it', 'and', 'across', 'after', 'it', 'and', 'in', 'it',
'a', 'under', 'another', 'after', 'it', 'once', 'in', 'out', 'again', 'on', 'a', 'and', 'a', 'about', 'a', 'or', 'o
f', 'as', 'about', 'and', 'and', 'out', 'it', 'anything', 'at', 'of', 'and', 'and', 'and', 'and', 'upon', 'a', 'one',
'of', 'as', 'it', 'it', 'empty', 'of', 'it', 'into', 'one', 'of', 'as', 'it', 'after', 'a', 'as', 'of', 'all', 'at',
'anything', 'about', 'it', 'even', 'if', 'off', 'of', 'an', 'end', 'aloud', 'of', 'earth', 'of', 'in', 'in', 'and',
'a', 'opportunity', 'off', 'as', 'one', 'it', 'it', 'over', 'about', 'or', 'idea', 'or', 'either', 'again', 'if', 'ea
rth', 'it', 'out', 'among', 'one', 'as', 'it', 'at', 'all', 'ask', 'of', 'is', 'am', 'is', 'or', 'and', 'as', 'as',
'air', 'it', 'an', 'ignorant', 'asking', 'it', 'ask', 'it', 'up', 'else', 'again', 'of', 'at', 'are', 'in', 'air', 'a
fraid', 'a', 'and', 'a', 'eat', 'and', 'on', 'in', 'a', 'of', 'eat', 'eat', 'and', 'eat', 'as', 'answer', 'either',
'it', 'it', 'off', 'and', 'in', 'and', 'earnestly', 'ever', 'eat', 'a', 'upon', 'a', 'of', 'and', 'and', 'over', 'a',
'and', 'up', 'on', 'in', 'a', 'up', 'it', 'all', 'overhead', 'another', 'and', 'in', 'it', 'a', 'away', 'and', 'in',
'it', 'as', 'it', 'a', 'ears', 'and', 'it', 'it', 'in', 'a', 'up', 'a', 'of', 'all', 'all', 'and', 'all', 'one', 'an
d', 'up', 'other', 'every', 'ever', 'out', 'again', 'upon', 'a', 'all', 'of', 'on', 'it', 'except', 'a', 'and', 'it',
'one', 'of', 'of', 'alas', 'either', 'or', 'at', 'any', 'it', 'open', 'any', 'of', 'on', 'upon', 'a', 'and', 'it',
'a', 'about', 'inches', 'in', 'and', 'it']

```
In [77]: 1 import re
2 f = open('alice_in_wonderland.txt', encoding='utf-8')
3 text = f.read()
4 # print(text)
5 reg = r'\b[^aeiou ]\w*'
6 ans = re.findall(reg,text)
7 print(ans)
8
9 # ^ - inverse starts with
```

```
[ 'Alice', "'s", 'Adventures', 'Wonderland', '\n', 'ALICE', "'S", 'ADVENTURES', 'IN', 'WONDERLAND', '\n', 'Lewis', 'Ca
rroll', '\n', 'THE', 'MILLENNIUM', 'FULCRUM', 'EDITION', '3', '.0', '\n', 'CHAPTER', 'I', '\n', 'Down', 'the', 'Rabbi
t', '-Hole', '\n', 'Alice', 'was', 'beginning', 'to', 'get', 'very', 'tired', 'sitting', 'by', 'her', 'sister', '\no
n', 'the', 'bank', ',', 'having', 'nothing', 'to', 'do', ':', 'twice', 'she', 'had', '\npeeped', 'the', 'book', 'he
r', 'sister', 'was', 'reading', ',', 'but', 'had', 'no', '\npictures', 'conversations', ',', 'what', 'the', 'book',
',', 'thought', 'Alice', 'without', 'pictures', 'conversation', '?', 'So', 'she', 'was', 'considering', 'her', 'min
d', 'well', 'she', 'could', ',', 'for', 'the', 'hot', 'day', 'made', 'her', 'feel', 'very', 'sleepy', 'stupid', ')',
'whether', '\nthe', 'pleasure', 'making', 'daisy', '-chain', 'would', 'be', 'worth', 'the', 'trouble', '\nof', 'getti
ng', 'picking', 'the', 'daisies', ',', 'when', 'suddenly', 'White', '\nRabbit', 'with', 'pink', 'ran', 'close', 'by',
'her', '.', 'There', 'was', 'nothing', 'so', 'VERY', 'remarkable', 'that', ';', 'nor', 'did', 'Alice', '\nthink', 's
o', 'VERY', 'much', 'the', 'way', 'to', 'hear', 'the', 'Rabbit', 'say', 'to', '\nitself', ',', 'Oh', 'dear', '!', 'O
h', 'dear', '!', 'I', 'shall', 'be', 'late', '!', 'when', 'she', 'thought', '\nit', ',', 'to', 'her', 'that', 'she',
'to', 'have', '\nwondered', 'this', ',', 'but', 'the', 'time', 'seemed', 'quite', 'natural', ')', 'but', 'when', 'th
e', 'Rabbit', 'TOOK', 'A', 'WATCH', 'OUT', 'OF', 'ITS', 'WAISTCOAT', '-', 'POCKET', ',', 'looked', ',', 'then', 'hurri
ed', ',', 'Alice', 'started', 'to', '\nher', 'feet', ',', 'for', 'flashed', 'her', 'mind', 'that', 'she', 'had', 'ne
ver', '\nbefore', 'seen', 'rabbit', 'with', 'waistcoat', '-pocket', ',', 'watch', 'to', '\ntake', ',', 'burning', 'wi
th', 'curiosity', ',', 'she', 'ran', 'the', '\nfield', ',', 'fortunately', 'was', 'just', 'time', 'to', 'see', 'pop',
'\ndown', 'large', 'rabbit', '-hole', 'the', 'hedge', '.', 'In', 'moment', 'down', 'went', 'Alice', ',', 'never', '\n
considering', 'how', 'the', 'world', 'she', 'was', 'to', 'get', '.', 'The', 'rabbit', '-hole', 'went', 'straight', 'l
ike', 'tunnel', 'for', 'some', 'way', ',', 'then', 'dipped', 'suddenly', 'down', ',', 'so', 'suddenly', 'that', 'Alic
e', 'had', 'not', '\nmoment', 'to', 'think', 'stopping', 'herself', 'before', 'she', 'found', 'herself', '\nfalling',
'down', 'very', 'deep', 'well', '.', 'Either', 'the', 'well', 'was', 'very', 'deep', ',', 'she', 'fell', 'very', 'slo
wly', ',', 'for', 'she', '\nhad', 'plenty', 'time', 'she', 'went', 'down', 'to', 'look', 'her', 'to', '\nwonder', 'wh
at', 'was', 'going', 'to', 'happen', 'next', '.', 'First', ',', 'she', 'tried', 'to', 'look', '\ndown', 'make', 'wha
t', 'she', 'was', 'coming', 'to', ',', 'but', 'was', 'too', 'dark', 'to', '\nsee', ';', 'then', 'she', 'looked', 'th
e', 'sides', 'the', 'well', ',', '\nnoticed', 'that', 'they', 'were', 'filled', 'with', 'cupboards', 'book', '-shelve
s', ';', 'here', 'there', 'she', 'saw', 'maps', 'pictures', 'hung', 'pegs', '.', 'She', '\ntook', 'down', 'jar', 'fro
m', 'the', 'shelves', 'she', 'passed', ';', 'was', '\nlabelled', 'ORANGE', 'MARMALADE', '"', 'but', 'to', 'her', 'gre
at', 'disappointment', '\nwas', ':', 'she', 'did', 'not', 'like', 'to', 'drop', 'the', 'jar', 'for', 'fear', 'killin
g', '\nsomebody', ',', 'so', 'managed', 'to', 'put', 'the', 'cupboards', 'she', '\nfell', 'past', '.', 'Well', '!',
'thought', 'Alice', 'to', 'herself', ',', 'such', 'fall', 'this', ',', 'I', '\nshall', 'think', 'nothing', 'tumblin
g', 'down', 'stairs', '!', 'How', 'brave', 'they', '"ll", '\nall', 'think', 'me', 'home', '!', 'Why', ',', 'I', 'woul
dn', '"t", 'say', ',', 'I', 'fell', 'the', 'top', 'the', 'house', '!', 'Which', 'was', 'very', 'likely', '\ntrue',
',', 'Down', ',', 'down', ',', 'down', '.', 'Would', 'the', 'fall', 'NEVER', 'come', 'to', '!', 'I', '\nwonder', 'ho
w', 'many', 'miles', 'I', '"ve", 'fallen', 'by', 'this', 'time', '?', 'she', 'said', '.', 'I', 'must', 'be', 'gettin
g', 'somewhere', 'near', 'the', 'centre', 'the', '.', 'Let', '\nme', 'see', ':', 'that', 'would', 'be', 'four', 'thou
sand', 'miles', 'down', ',', 'I', 'think', '-', 'for', ',', 'you', 'see', ',', 'Alice', 'had', 'learnt', 'several',
'things', 'this', 'sort', 'her', '\nlessons', 'the', 'schoolroom', ',', 'though', 'this', 'was', 'not', 'VERY', 'goo
d', '\nopportunity', 'for', 'showing', 'her', 'knowledge', ',', 'there', 'was', 'no', 'to', '\nlisten', 'to', 'her',
',', 'still', 'was', 'good', 'practice', 'to', 'say', ')', 'yes', ',', 'that', '"s", 'the', 'right', 'distance', '-',
'but', 'then', 'I', 'wonder', 'what', 'Latitude', '\nor', 'Longitude', 'I', '"ve", 'got', 'to', '?', 'Alice', 'had',
'no', 'what', 'Latitude', 'was', ',', 'Longitude', ',', 'but', 'thought', 'they', 'were', 'nice', 'grand', 'words',
'to', '\nsay', '.', 'Presently', 'she', 'began', '.', 'I', 'wonder', 'I', 'shall', 'fall', 'right', '\nTHROUGH', 'th
e', '!', 'How', 'funny', '"ll", 'seem', 'to', 'come', 'the', '\npeople', 'that', 'walk', 'with', 'their', 'heads', 'd
ownward', '!', 'The', 'Antipathies', ',', 'I', '\nthink', '-', 'she', 'was', 'rather', 'glad', 'there', 'WAS', 'no',
'listening', ',', 'this', '\ntime', ',', 'didn', '"t", 'sound', 'the', 'right', 'word', ')', 'but', 'I', 'shall', '\n
have', 'to', 'them', 'what', 'the', 'name', 'the', 'country', ',', 'you', 'know', '.', 'Please', ',', 'Ma', '"am",
',', 'this', 'New', 'Zealand', 'Australia', '?', 'she', 'tried', '\nto', 'curtsey', 'she', 'spoke', '-', 'fancy', 'CU
RTSEYING', 'you', '"re", 'falling', '\nthrough', 'the', '!', 'Do', 'you', 'think', 'you', 'could', 'manage', '?', 'An
d', 'what', '\nan', 'little', 'girl', 'she', '"ll", 'think', 'me', 'for', '!', 'No', ',', '"ll", '\nnever', 'do', 't
o', ':', 'perhaps', 'I', 'shall', 'see', 'written', 'somewhere', '.', 'Down', ',', 'down', ',', 'down', '.', 'There',
'was', 'nothing', 'to', 'do', ',', 'so', 'Alice', 'soon', '\nbegan', 'talking', '.', 'Dinah', '"ll", 'miss', 'me', 'v
ery', 'much', 'to', '-night', ',', 'I', '\nshould', 'think', '!', 'Dinah', 'was', 'the', 'cat', '.', 'I', 'hope', 'th
ey', '"ll", 'remember', '\nher', 'saucer', 'milk', 'tea', '-time', '.', 'Dinah', 'my', 'dear', '!', 'I', 'wish', 'yo
u', 'were', '\ndown', 'here', 'with', 'me', '!', 'There', 'no', 'mice', 'the', ',', 'I', '"m", ',', 'but', '\nyou',
'might', 'catch', 'bat', ',', 'that', '"s", 'very', 'like', 'mouse', ',', 'you', 'know', '.', 'But', 'do', 'cats', 'b
ats', ',', 'I', 'wonder', '?', 'And', 'here', 'Alice', 'began', 'to', 'get', '\nrather', 'sleepy', ',', 'went', 'sayi
ng', 'to', 'herself', ',', 'dreamy', 'sort', '\nway', ',', 'Do', 'cats', 'bats', '?', 'Do', 'cats', 'bats', '?', 'som
etimes', ',', 'Do', '\nbats', 'cats', '?', 'for', ',', 'you', 'see', ',', 'she', 'couldn', '"t", '\nquestion', ',',
'didn', '"t", 'much', 'matter', 'which', 'way', 'she', 'put', '.', 'She', 'felt', '\nthat', 'she', 'was', 'dozing',
',', 'had', 'just', 'begun', 'to', 'dream', 'that', 'she', '\nwas', 'walking', 'hand', 'hand', 'with', 'Dinah', ',',
'saying', 'to', 'her', 'very', '\nearnestly', ',', 'Now', ',', 'Dinah', ',', 'tell', 'me', 'the', 'truth', ':', 'di
d', 'you', '\nbat', '?', 'when', 'suddenly', ',', 'thump', '!', 'thump', '!', 'down', 'she', 'came', 'heap', '\nstick
s', 'dry', 'leaves', ',', 'the', 'fall', 'was', '.', 'Alice', 'was', 'not', 'bit', 'hurt', ',', 'she', 'jumped', 't
o', 'her', 'feet', '\nmoment', ':', 'she', 'looked', ',', 'but', 'was', 'dark', ';', 'before', 'her', '\nwas', 'lon
g', 'passage', ',', 'the', 'White', 'Rabbit', 'was', 'still', '\nsight', ',', 'hurrying', 'down', '.', 'There', 'wa
s', 'not', 'moment', 'to', 'be', 'lost', ':', 'went', 'Alice', 'like', 'the', 'wind', ',', 'was', 'just', 'time', 't
o', 'hear', '\nsay', ',', 'turned', 'corner', ',', 'Oh', 'my', 'whiskers', ',', 'how', 'late', '\nit', '"s", 'gettin
g', '!', 'She', 'was', 'close', 'behind', 'when', 'she', 'turned', 'the', '\ncorner', ',', 'but', 'the', 'Rabbit', 'w
as', 'no', 'longer', 'to', 'be', 'seen', ':', 'she', 'found', '\nherself', 'long', ',', 'low', 'hall', ',', 'which',
'was', 'lit', 'by', 'row', 'lamps', '\nhanging', 'from', 'the', 'roof', '.', 'There', 'were', 'doors', 'round', 'th
e', 'hall', ',', 'but', 'they', 'were', 'locked', ';', 'when', 'Alice', 'had', 'been', 'the', 'way', 'down', 'side',
'the', '\nother', ',', 'trying', 'door', ',', 'she', 'walked', 'sadly', 'down', 'the', 'middle', ',', 'wondering', 'h
ow', 'she', 'was', 'to', 'get', '.', 'Suddenly', 'she', 'came', 'little', 'three', '-legged', 'table', ',', 'made',
'\nsolid', 'glass', ';', 'there', 'was', 'nothing', 'tiny', 'golden', 'key', ',', 'Alice', '"s", 'first', 'thought',
'was', 'that', 'might', 'belong', 'to', 'the', '\ndoors', 'the', 'hall', ';', 'but', ',', '!', 'the', 'locks', 'wer
e', 'too', 'large', ',', '\nthe', 'key', 'was', 'too', 'small', ',', 'but', 'rate', 'would', 'not', '\nthem', '.', 'H
owever', ',', 'the', 'second', 'time', 'round', ',', 'she', 'came', 'low', '\ncurtain', 'she', 'had', 'not', 'notice
d', 'before', ',', 'behind', 'was', 'little', '\ndoor', 'fifteen', 'high', ':', 'she', 'tried', 'the', 'little', 'gol
den', 'key', '\nin', 'the', 'lock', ',', 'to', 'her', 'great', 'delight', 'fitted', '!', 'THE', 'END']
```

In [78]:

```
1 import re
2 f = open('alice_in_wonderland.txt', encoding='utf-8')
3 text = f.read()
4 # print(text)
5 reg = r'\b[aeiou]\w{2}'
6 ans = re.findall(reg,text)
7 print(ans)
```

['and', 'onc', 'int', 'and', 'use', 'own', 'and', 'and', 'eye', 'out', 'its', 'ove', 'aft', 'occ', 'oug', 'all', 'act', 'and', 'and', 'acr', 'eit', 'out', 'and', 'acr', 'aft', 'and', 'und', 'ano', 'aft', 'onc', 'out', 'aga', 'and', 'abo', 'abo', 'and', 'and', 'out', 'any', 'and', 'and', 'and', 'and', 'up', 'one', 'emp', 'int', 'one', 'aft', 'all', 'any', 'abo', 'eve', 'off', 'end', 'alo', 'ear', 'and', 'opp', 'off', 'one', 'ove', 'abo', 'ide', 'eit', 'aga', 'ear', 'out', 'amo', 'one', 'all', 'ask', 'and', 'air', 'ign', 'ask', 'ask', 'els', 'aga', 'are', 'air', 'afr', 'and', 'eat', 'and', 'eat', 'eat', 'and', 'eat', 'ans', 'eit', 'off', 'and', 'and', 'ear', 'eve', 'eat', 'up', 'and', 'and', 'ove', 'and', 'all', 'ove', 'ano', 'and', 'awa', 'and', 'ear', 'and', 'all', 'all', 'and', 'all', 'one', 'and', 'oth', 'eve', 'eve', 'out', 'aga', 'up', 'all', 'exc', 'and', 'one', 'ala', 'eit', 'any', 'ope', 'any', 'up', 'and', 'abo', 'inc', 'and']

In [79]:

```
1 import re
2 f = open('alice_in_wonderland.txt', encoding='utf-8')
3 text = f.read()
4 # print(text)
5 reg = r'\b[aeiou]\w{2}\b'
6 ans = re.findall(reg,text)
7 print(ans)
```

['and', 'and', 'use', 'own', 'and', 'and', 'out', 'all', 'and', 'and', 'out', 'and', 'and', 'out', 'and', 'and', 'and', 'out', 'and', 'and', 'and', 'one', 'one', 'all', 'off', 'end', 'and', 'off', 'one', 'out', 'one', 'all', 'ask', 'and', 'air', 'ask', 'are', 'air', 'and', 'eat', 'and', 'eat', 'eat', 'and', 'eat', 'off', 'and', 'and', 'eat', 'and', 'and', 'and', 'all', 'and', 'and', 'and', 'all', 'all', 'and', 'all', 'one', 'and', 'out', 'all', 'and', 'one', 'any', 'any', 'and', 'and']

In []:

1