```
!pip install pyPDF2
```

```
Collecting pyPDF2
  Downloading pypdf2-3.0.1-py3-none-any.whl.metadata (6.8 kB)
Downloading pypdf2-3.0.1-py3-none-any.whl (232 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 232.6/232.6 kB 3.3 MB/s eta 0:00:00
Installing collected packages: pyPDF2
Successfully installed pyPDF2-3.0.1
```

```
import PyPDF2
from PyPDF2 import PdfFileReader
```

```


pdf = open("file1pdf.pdf","rb")

pdf_reader = PyPDF2.PdfReader(pdf)
print("Number of pages:", len(pdf_reader.pages))
page = pdf_reader.pages[1]
print(page.extract_text())
pdf.close()

```

```
Number of pages: 35


 Development  Plan for Greater Mumbai 2014-2034
Acknowledgements
The Consultant  wishes to thank the following  individuals  from the Municipal  Corporation  of
Greater Mumbai for their invaluable  support, insights and contributions  towards ‘Working  Paper 1
– Preparation  of Base Map’ for the preparation  of the Development  Plan for Greater Mumbai
2014-34.
 Mr. Subodh Kumar, IAS, Municipal  Commissioner;
 Mr. Rajeev Kuknoor, Chief Engineer Development  Plan;
 Mr. Sudhir Ghate, Deputy Chief Engineer Development  Plan;
 Mr. A.G. Marathe, Deputy Chief Engineer Development  Plan;
 Mr. R. Balachandran,  Executive  Engineer and Town Planning Officer, Development  Plan.
 Our gratitude  to the following  experts for their invaluable  insights and support:
```

```
⍰
Mr. V.K Phatak, Former Chief Town Planner (MMRDA);
⍰ Mr. A.N Kale, Former Chief Engineer, (DP);
⍰ Mr. A. S Jain Former Dy. Chief Engineer, (DP).
 We wish to especially  thank MCGM officers, Mr. Jagdish Talreja, Mr. Dinesh Naik, Mr. Hiren
Daftardar,  Ms. Anita Naik for their continual  support since the
 beginning  of the project and their
help towards familiarization  and data collection.  They have been instrumental  in helping to
contact various MCGM departments  as well as in helping to establish contact with personnel  from
other government  departments  and organizations.  Many thanks for the MCGM team, for
deploying  personnel,  particularly  Mr. Prasad Gharat, on extensive  field visits that have helped in
understanding  actual ground conditions.

We apologize  if we have inadvertently  omitted anyone to whom acknowledgement  is due. We hope
and anticipate  the work's usefulness  for the intended purpose.
```

```
import PyPDF2,urllib,nltk
from io import BytesIO
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
```

```
wFile = urllib.request.urlopen('http://www.udri.org/pdf/02%20working%20paper%201.pdf')
pdfreader = PyPDF2.PdfReader(BytesIO(wFile.read()))
```

```
pageObj = pdfreader.pages[2]
page2=pageObj.extract_text()
punctuations=['(',')',':',':',':','[',']',',','....']
tokens=word_tokenize(page2)
stop_words=stopwords.words('english')
keywords=[w for w in tokens if not w in punctuations]
keywords=[w for w in tokens if not w in punctuations]
```

```
import nltk
nltk.download('punkt_tab')
```

```
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
True
```

```
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
keywords
```

```
 ............. ,
 '5',
 'Development',
 'Plan',
 'for',
 'Greater',
 'Mumbai',
 '...........................................................................................',
 '5',
 'ELU',

'...............................................................................................................................'
 '.................',
 '5',
 'Existing',
 'Land',
 'use',

'.............................................................................................................................',
 '5',
 'FSI',

'...............................................................................................................................'
 '...................',
 '5',
 'Floor',
 'Space',
 'Index',
 '............................................................................................................................',
 '5',
 'GIS',

'...............................................................................................................................'
 '..................',
 '5']
```

```
name_list=list()
check=['Mr.','Mrs.','Ms.']
for idx,token in enumerate(tokens):
  if token.startswith(tuple(check)) and idx<(len(token)-1):
    name = token+tokens[idx+1]+' '+tokens[idx+2]
    name_list.append(name)
print(name_list)
```

```
[]
```

```
!pip install python-docx
```

```
Collecting python-docx
  Downloading python_docx-1.1.2-py3-none-any.whl.metadata (2.0 kB)
Requirement already satisfied: lxml>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from python-docx) (5.3.0)
Requirement already satisfied: typing-extensions>=4.9.0 in /usr/local/lib/python3.11/dist-packages (from python-docx) (4.12.2)
Downloading python_docx-1.1.2-py3-none-any.whl (244 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 244.3/244.3 kB 3.3 MB/s eta 0:00:00
Installing collected packages: python-docx
Successfully installed python-docx-1.1.2
```

```
import docx
```

```
doc=open('Task-1-Answers.docx','rb')
document=docx.Document(doc)
```

```
docu=""
for para in document.paragraphs:
  docu+=para.text
print(docu)
```

```
for i in range(len(document.paragraphs)):
  print("The content of the paragraph"+str(i+1)+"is:"+document.paragraphs[i].text+"\n")
```

```
The content of the paragraph1is:

The content of the paragraph2is:
```

```
The content of the paragraph3is:

The content of the paragraph4is:

The content of the paragraph5is:

The content of the paragraph6is:

The content of the paragraph7is:

The content of the paragraph8is:

The content of the paragraph9is:

The content of the paragraph10is:
```

```
!pip install bs4
```

```
Collecting bs4
  Downloading bs4-0.0.2-py2.py3-none-any.whl.metadata (411 bytes)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.11/dist-packages (from bs4) (4.13.3)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.11/dist-packages (from beautifulsoup4->bs4) (2.6)
Requirement already satisfied: typing-extensions>=4.0.0 in /usr/local/lib/python3.11/dist-packages (from beautifulsoup4->bs4) (4.12
Downloading bs4-0.0.2-py2.py3-none-any.whl (1.2 kB)
Installing collected packages: bs4
Successfully installed bs4-0.0.2
```

```
import urllib.request as urllib2
from bs4 import BeautifulSoup
```

```
response=urllib2.urlopen("https://www.bing.com/search?q=nlp+wikipedia+deutsch&form=ANNTH1&refig=8c0cf9e4443346ad8910270abd5da215&pc=ASTS
html_doc=response.read()
```

```
soup=BeautifulSoup(html_doc,'html.parser')
strhtml=soup.prettify()
print(strhtml[:500])
```

```
<!DOCTYPE html>
<html dir="ltr" lang="en" xml:lang="en" xmlns="http://www.w3.org/1999/xhtml" xmlns:web="http://schemas.live.com/Web/">
 <script nonce="SrqDV6g0ZZj1Gp6au8b0vOPAC1igE83I6ArqsQmcEqM=" type="text/javascript">
  //<![CDATA[
si_ST=new Date
//]]>
 </script>
 <head>
  <!--pc-->
  <title>
   nlp wikipedia deutsch - Search
  </title>
  <meta content="text/html; charset=utf-8" http-equiv="content-type"/>
  <meta content="origin-when-cross-origin" name="referrer"/>
  <meta content="A5is4nw
```

```
Start coding or generate with AI.
```