In [1]:

```python
#1.Write a program in Jupyter Notebook to declare variables of different data types (integer, float, stri
integer_var = 42
float_var = 3.14
string_var = "Hello, WORLD!"
boolean_var = True
print("Integer Variable:")
print(f"Value: {integer_var}, Type: {type(integer_var)}\n")
print("Float Variable:")
print(f"Value: {float_var}, Type: {type(float_var)}\n")
print("String Variable:")
print(f"Value: '{string_var}', Type: {type(string_var)}\n")
print("Boolean Variable:")
print(f"Value: {boolean_var}, Type: {type(boolean_var)}")
```

```
Integer Variable:
Value: 42, Type: <class 'int'>

Float Variable:
Value: 3.14, Type: <class 'float'>

String Variable:
Value: 'Hello, WORLD!', Type: <class 'str'>

Boolean Variable:
Value: True, Type: <class 'bool'>
```

In [2]:

```
#2.Create a List, tuple and Dictionary with 5 elements in it and how to access few elements based on the
my_list = [10, 20, 30, 40, 50]
print("List:", my_list)
print("First element of List:", my_list[0])
print("Last element of List:", my_list[-1])
print("Slice elements (2nd to 4th):", my_list[1:4])
print("\n")
my_tuple = ("apple", "banana", "cherry", "date", "elderberry")
print("Tuple:", my_tuple)
print("Second element of Tuple:", my_tuple[1])
print("Last element of Tuple:", my_tuple[-1])
print("Slice elements (3rd to last):", my_tuple[2:])
print("\n")
my_dict = {
    "name": "Alice",
    "age": 25,
    "city": "New York",
    "country": "USA",
    "hobby": "reading"
}
print("Dictionary:", my_dict)
print("Value of key 'name':", my_dict["name"])
print("Value of key 'hobby':", my_dict.get("hobby"))
print("Keys of the Dictionary:", list(my_dict.keys()))
print("Values of the Dictionary:", list(my_dict.values()))
```

```
List: [10, 20, 30, 40, 50]
First element of List: 10
Last element of List: 50
Slice elements (2nd to 4th): [20, 30, 40]


Tuple: ('apple', 'banana', 'cherry', 'date', 'elderberry')
Second element of Tuple: banana
Last element of Tuple: elderberry
Slice elements (3rd to last): ('cherry', 'date', 'elderberry')


Dictionary: {'name': 'Alice', 'age': 25, 'city': 'New York', 'country': 'USA', 'hobby': 'reading'}
Value of key 'name': Alice
Value of key 'hobby': reading
Keys of the Dictionary: ['name', 'age', 'city', 'country', 'hobby']
Values of the Dictionary: ['Alice', 25, 'New York', 'USA', 'reading']
```

In [4]:

```python
#3.Write a Python program that takes a student's marks in three subjects as input.
#If the average is greater than or equal to 90, print "Grade: A".
#If the average is between 80 and 89, print "Grade: B".
#If the average is between 70 and 79, print "Grade: C".
#Otherwise, print "Grade: Fail".
try:
    subject1 = float(input("Enter marks for Subject 1: "))
    subject2 = float(input("Enter marks for Subject 2: "))
    subject3 = float(input("Enter marks for Subject 3: "))
    average = (subject1 + subject2 + subject3) / 3
    if average >= 90:
        grade = "A"
    elif 80 <= average < 90:
        grade = "B"
    elif 70 <= average < 80:
        grade = "C"
    else:
        grade = "Fail"
    print(f"Average Marks: {average:.2f}")
    print(f"Grade: {grade}")
except ValueError:
    print("Invalid input. Please enter numeric values for marks.")
```

```
Enter marks for Subject 1: 90
Enter marks for Subject 2: 100
Enter marks for Subject 3: 90
Average Marks: 93.33
Grade: A
```

In [7]:
```python
#4.Write a Python program to calculate the sum of all even numbers between 1 and a given positive integer
def sum_of_even_numbers(n):
    if n < 1:
        return 0
    total_sum = 0
    for i in range(2, n + 1, 2):
        total_sum += i
    return total_sum
try:
    n = int(input("Enter a positive integer: "))
    if n <= 0:
        print("Please enter a positive integer greater than 0.")
    else:
        result = sum_of_even_numbers(n)
        print(f"The sum of all even numbers between 1 and {n} is: {result}")
except ValueError:
    print("Invalid input. Please enter a valid positive integer.")
```

```
Enter a positive integer: 8
The sum of all even numbers between 1 and 8 is: 20
```

In [25]:
```python
#5.Write a Python program to calculate the frequency of each word in a given text. Print the words and th
from collections import Counter
def calculate_word_frequency(text):
    words = text.lower().split()
    word_count = Counter(words)
    print("Word frequencies:")
    for word, count in word_count.items():
        print(f"{word}: {count}")
text = input("Enter a text: ")
calculate_word_frequency(text)
```

```
Enter a text: srinidhi
Word frequencies:
srinidhi: 1
```

In [6]:
```
1 !pip install nltk
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: nltk in e:\programdata\anaconda3\lib\site-packages (3.7)
Requirement already satisfied: joblib in e:\programdata\anaconda3\lib\site-packages (from nltk) (1.1.0)
Requirement already satisfied: regex>=2021.8.3 in e:\programdata\anaconda3\lib\site-packages (from nltk) (20
22.7.9)
Requirement already satisfied: tqdm in e:\programdata\anaconda3\lib\site-packages (from nltk) (4.64.1)
Requirement already satisfied: click in e:\programdata\anaconda3\lib\site-packages (from nltk) (8.0.4)
Requirement already satisfied: colorama in e:\programdata\anaconda3\lib\site-packages (from click->nltk) (0.
4.5)
```

In [12]:
```
1 !pip install spacy
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting spacy
  Using cached spacy-3.8.3-cp39-cp39-win_amd64.whl (12.3 MB)
Collecting thinc<8.4.0,>=8.3.0
  Using cached thinc-8.3.4-cp39-cp39-win_amd64.whl (1.5 MB)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in c:\users\admin\appdata\roaming\python\python39
\site-packages (from spacy) (3.0.12)
Collecting pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4
  Using cached pydantic-2.10.6-py3-none-any.whl (431 kB)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in c:\users\admin\appdata\roaming\python\python39
\site-packages (from spacy) (1.0.5)
Requirement already satisfied: packaging>=20.0 in e:\programdata\anaconda3\lib\site-packages (from spacy) (2
1.3)
Collecting murmurhash<1.1.0,>=0.28.0
  Using cached murmurhash-1.0.12-cp39-cp39-win_amd64.whl (25 kB)
Collecting wasabi<1.2.0,>=0.9.1
  Using cached wasabi-1.1.3-py3-none-any.whl (27 kB)
Collecting catalogue<2.1.0,>=2.0.6
  Using cached catalogue-2.0.10-py3-none-any.whl (17 kB)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in e:\programdata\anaconda3\lib\site-packages (from spac
y) (4.64.1)
Requirement already satisfied: numpy>=1.19.0 in e:\programdata\anaconda3\lib\site-packages (from spacy) (1.2
1.5)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in e:\programdata\anaconda3\lib\site-packages (from s
pacy) (2.28.1)
Collecting weasel<0.5.0,>=0.1.0
  Using cached weasel-0.4.1-py3-none-any.whl (50 kB)
Collecting srsly<3.0.0,>=2.4.3
  Using cached srsly-2.5.1-cp39-cp39-win_amd64.whl (633 kB)
Collecting typer<1.0.0,>=0.3.0
  Using cached typer-0.15.1-py3-none-any.whl (44 kB)
Collecting langcodes<4.0.0,>=3.2.0
  Using cached langcodes-3.5.0-py3-none-any.whl (182 kB)
Requirement already satisfied: setuptools in e:\programdata\anaconda3\lib\site-packages (from spacy) (63.4.
1)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in c:\users\admin\appdata\roaming\python\python39\site-pa
ckages (from spacy) (2.0.11)
Requirement already satisfied: jinja2 in e:\programdata\anaconda3\lib\site-packages (from spacy) (2.11.3)
Collecting preshed<3.1.0,>=3.0.2
  Using cached preshed-3.0.9-cp39-cp39-win_amd64.whl (122 kB)
Collecting language-data>=1.2
  Using cached language_data-1.3.0-py3-none-any.whl (5.4 MB)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in e:\programdata\anaconda3\lib\site-packages (from
```

```
packaging>=20.0->spacy) (3.0.9)
Requirement already satisfied: typing-extensions>=4.12.2 in c:\users\admin\appdata\roaming\python\python39\s
ite-packages (from pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy) (4.12.2)
Collecting pydantic-core==2.27.2
  Using cached pydantic_core-2.27.2-cp39-cp39-win_amd64.whl (2.0 MB)
Collecting annotated-types>=0.6.0
  Using cached annotated_types-0.7.0-py3-none-any.whl (13 kB)
Requirement already satisfied: charset-normalizer<3,>=2 in e:\programdata\anaconda3\lib\site-packages (from
requests<3.0.0,>=2.13.0->spacy) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in e:\programdata\anaconda3\lib\site-packages (from reques
ts<3.0.0,>=2.13.0->spacy) (2022.9.14)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in e:\programdata\anaconda3\lib\site-packages (from req
uests<3.0.0,>=2.13.0->spacy) (1.26.11)
Requirement already satisfied: idna<4,>=2.5 in e:\programdata\anaconda3\lib\site-packages (from requests<3.
0.0,>=2.13.0->spacy) (3.3)
Collecting confection<1.0.0,>=0.0.1
  Using cached confection-0.1.5-py3-none-any.whl (35 kB)
Collecting blis<1.3.0,>=1.2.0
  Using cached blis-1.2.0-cp39-cp39-win_amd64.whl (6.2 MB)
Requirement already satisfied: colorama in e:\programdata\anaconda3\lib\site-packages (from tqdm<5.0.0,>=4.3
8.0->spacy) (0.4.5)
Requirement already satisfied: shellingham>=1.3.0 in c:\users\admin\appdata\roaming\python\python39\site-pac
kages (from typer<1.0.0,>=0.3.0->spacy) (1.5.4)
Requirement already satisfied: click>=8.0.0 in e:\programdata\anaconda3\lib\site-packages (from typer<1.0.0,
>=0.3.0->spacy) (8.0.4)
Collecting rich>=10.11.0
  Using cached rich-13.9.4-py3-none-any.whl (242 kB)
Collecting colorama
  Using cached colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Collecting cloudpathlib<1.0.0,>=0.7.0
  Using cached cloudpathlib-0.20.0-py3-none-any.whl (52 kB)
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in e:\programdata\anaconda3\lib\site-packages (from
weasel<0.5.0,>=0.1.0->spacy) (5.2.1)
Requirement already satisfied: MarkupSafe>=0.23 in e:\programdata\anaconda3\lib\site-packages (from jinja2->
spacy) (2.0.1)
Collecting marisa-trie>=1.1.0
  Using cached marisa_trie-1.2.1-cp39-cp39-win_amd64.whl (152 kB)
Collecting markdown-it-py>=2.2.0
  Using cached markdown_it_py-3.0.0-py3-none-any.whl (87 kB)
Collecting pygments<3.0.0,>=2.13.0
  Using cached pygments-2.19.1-py3-none-any.whl (1.2 MB)
Collecting mdurl~=0.1
  Using cached mdurl-0.1.2-py3-none-any.whl (10.0 kB)
```

```
Installing collected packages: pygments, pydantic-core, murmurhash, mdurl, marisa-trie, colorama, cloudpathl
ib, catalogue, blis, annotated-types, wasabi, srsly, pydantic, preshed, markdown-it-py, language-data, rich,
langcodes, confection, typer, thinc, weasel, spacy

  WARNING: The script pygmentize.exe is installed in 'C:\Users\ADMIN\AppData\Roaming\Python\Python39\Script
s' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-lo
cation.
  WARNING: The script markdown-it.exe is installed in 'C:\Users\ADMIN\AppData\Roaming\Python\Python39\Script
s' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-lo
cation.
  WARNING: The script typer.exe is installed in 'C:\Users\ADMIN\AppData\Roaming\Python\Python39\Scripts' whi
ch is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-lo
cation.
  WARNING: The script weasel.exe is installed in 'C:\Users\ADMIN\AppData\Roaming\Python\Python39\Scripts' wh
ich is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-lo
cation.
ERROR: Could not install packages due to an OSError: [WinError 32] The process cannot access the file becaus
e it is being used by another process: 'C:\\Users\\ADMIN\\AppData\\Roaming\\Python\\Python39\\site-packages
\\spacy\\lang\\fi\\examples.py'
Check the permissions.
```

In [11]:

```
1 !python -m spacy download en_core_web_sm
```

```
e:\ProgramData\Anaconda3\python.exe: No module named spacy
```

In [2]:
```
#6.Write a Python program to using NLTK and Spacy
#Convert text to lowercase.
#Remove stopwords using NLTK
import nltk
import spacy
from nltk.corpus import stopwords
nltk.download('stopwords')
nlp = spacy.load("en_core_web_sm")
def process_text(text):
    doc = nlp(text.lower())
    stop_words = set(stopwords.words('english'))
    filtered_tokens = [token.text for token in doc if token.text not in stop_words]
    return " ".join(filtered_tokens)
if __name__ == "__main__":
    input_text = "This is a simple example to demonstrate text processing with NLTK and SpaCy."
    processed_text = process_text(input_text)
    print("Original Text:", input_text)
    print("Processed Text:", processed_text)
```

```
[nltk_data] Error loading stopwords: <urlopen error [Errno 11001]
[nltk_data]     getaddrinfo failed>

Original Text: This is a simple example to demonstrate text processing with NLTK and SpaCy.
Processed Text: simple example demonstrate text processing nltk spacy .
```

In [13]:
```
!pip install gensim
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: gensim in e:\programdata\anaconda3\lib\site-packages (4.1.2)
Requirement already satisfied: numpy>=1.17.0 in e:\programdata\anaconda3\lib\site-packages (from gensim) (1.
21.5)
Requirement already satisfied: smart-open>=1.8.1 in e:\programdata\anaconda3\lib\site-packages (from gensim)
(5.2.1)
Requirement already satisfied: scipy>=0.18.1 in e:\programdata\anaconda3\lib\site-packages (from gensim) (1.
9.1)
```

In [14]:
```
import nltk
nltk.download('punkt')
nltk.download('wordnet')
nltk.download('stopwords')
```

```
  WARNING: The script pygmentize.exe is installed in 'C:\Users\ADMIN\AppData\Roaming\Python\Python39\Scrip
ts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-
location.
  WARNING: The script markdown-it.exe is installed in 'C:\Users\ADMIN\AppData\Roaming\Python\Python39\Scri
pts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-
location.
  WARNING: The script typer.exe is installed in 'C:\Users\ADMIN\AppData\Roaming\Python\Python39\Scripts' w
hich is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-
location.
  WARNING: The script weasel.exe is installed in 'C:\Users\ADMIN\AppData\Roaming\Python\Python39\Scripts'
which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-
location.
  WARNING: The script spacy.exe is installed in 'C:\Users\ADMIN\AppData\Roaming\Python\Python39\Scripts' w
hich is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-
```

In [3]:

```python
#7.Use Genism to preprocess data from a sample text file, follow basic procedures like tokenization, stem
import nltk
from gensim.utils import simple_preprocess
from gensim.parsing.preprocessing import STOPWORDS
from nltk.stem import PorterStemmer, WordNetLemmatizer
nltk.download('wordnet')
nltk.download('omw-1.4')
def preprocess_text(text):
    tokens = simple_preprocess(text, deacc=True)
    tokens = [word for word in tokens if word not in STOPWORDS]
    stemmer = PorterStemmer()
    stemmed_tokens = [stemmer.stem(word) for word in tokens]
    lemmatizer = WordNetLemmatizer()
    lemmatized_tokens = [lemmatizer.lemmatize(word) for word in stemmed_tokens]
    return lemmatized_tokens
text_data = "The quick brown fox jumps over the lazy dog. This is a test document for text preprocessing.
preprocessed_data = preprocess_text(text_data)
print("Preprocessed Text Tokens:")
print(preprocessed_data)
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\sindh\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to
[nltk_data]     C:\Users\sindh\AppData\Roaming\nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!

Preprocessed Text Tokens:
['quick', 'brown', 'fox', 'jump', 'lazi', 'dog', 'test', 'document', 'text', 'preprocess']
```

In [15]:
```python
#8.Tokenizes a sample paragraph into words and sentences.
import nltk
nltk.download('punkt')
def tokenize_text(paragraph):
    """
    Tokenizes the input paragraph into sentences and words.

    Args:
        paragraph (str): The input text to be tokenized.

    Returns:
        tuple: A tuple containing a list of sentences and a list of words.
    """
    sentences = nltk.sent_tokenize(paragraph)
    words = nltk.word_tokenize(paragraph)
    return sentences, words
if __name__ == "__main__":
    sample_paragraph = (
        "Natural Language Processing is an exciting field of Artificial Intelligence. "
        "It involves enabling machines to understand and process human languages."
    )
    sentences, words = tokenize_text(sample_paragraph)
    print("Original Paragraph:")
    print(sample_paragraph)
    print("\nTokenized Sentences:")
    print(sentences)
    print("\nTokenized Words:")
    print(words)
```

```
Original Paragraph:
Natural Language Processing is an exciting field of Artificial Intelligence. It involves enabling machines t
o understand and process human languages.

Tokenized Sentences:
['Natural Language Processing is an exciting field of Artificial Intelligence.', 'It involves enabling machi
nes to understand and process human languages.']

Tokenized Words:
['Natural', 'Language', 'Processing', 'is', 'an', 'exciting', 'field', 'of', 'Artificial', 'Intelligence',
'.', 'It', 'involves', 'enabling', 'machines', 'to', 'understand', 'and', 'process', 'human', 'languages',
'.']
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\ADMIN\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

In [16]:

```python
#9.Write a Python function to clean a given text by removing special characters and converting it to lowe
import re
def clean_text(text):
    """
    Cleans the input text by:
    1. Removing special characters.
    2. Converting the text to lowercase.

    Parameters:
        text (str): The input text to clean.

    Returns:
        str: The cleaned text.
    """
    cleaned_text = re.sub(r'[^a-zA-Z0-9\s]', '', text)
    cleaned_text = cleaned_text.lower()
    return cleaned_text
input_text = 'Hello, World! Welcome to NLP 101.'
cleaned_text = clean_text(input_text)
print("Original Text:", input_text)
print("Cleaned Text:", cleaned_text)
```

```
Original Text: Hello, World! Welcome to NLP 101.
Cleaned Text: hello world welcome to nlp 101
```

In [17]:

```python
#10.Write a Python function using regular expressions to extract all email addresses from a given string.
import re
def extract_emails(text):
    """
    Extract all email addresses from the given string using regular expressions.

    Parameters:
        text (str): The input string containing email addresses.

    Returns:
        list: A list of email addresses found in the input string.
    """
    email_pattern = r'[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}'
    return re.findall(email_pattern, text)

input_text = 'Contact us at support@example.com and sales@example.org.'
extracted_emails = extract_emails(input_text)
print(extracted_emails)
```

```
['support@example.com', 'sales@example.org']
```

In [18]:

```python
#11.Write a Python script to fetch and print the title of a webpage using the 'requests' and 'BeautifulSo
import re
import requests
from bs4 import BeautifulSoup
from collections import Counter
import nltk
from nltk.tokenize import word_tokenize, sent_tokenize

# Ensure necessary NLTK resources are available
nltk.download('punkt')

def word_frequency(text):
    # Convert text to lowercase and split into words
    words = text.lower().split()

    # Count frequency of each word
    word_counts = Counter(words)

    # Print words and their frequencies
    for word, count in word_counts.items():
        print(f"{word}: {count}")

def extract_emails(text):
    # Regular expression pattern for email addresses
    email_pattern = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'
    return re.findall(email_pattern, text)

def tokenize_text(text):
    words = word_tokenize(text)
    sentences = sent_tokenize(text)
    return words, sentences

def fetch_webpage_title(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.text, 'html.parser')
    return soup.title.string if soup.title else "No title found"

# Example usage
if __name__ == "__main__":
    text = input("Enter a text: ")
    word_frequency(text)
    emails = extract_emails(text)
    print("Extracted emails:", emails)
```

```
    words, sentences = tokenize_text(text)
    print("Tokenized words:", words)
    print("Tokenized sentences:", sentences)

    url = "https://example.com"
    print("Webpage title:", fetch_webpage_title(url))
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\ADMIN\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!

Enter a text: keerthy
keerthy: 1
Extracted emails: []
Tokenized words: ['keerthy']
Tokenized sentences: ['keerthy']
Webpage title: Example Domain
```

In [19]:
```
1 !pip install wordcloud
```

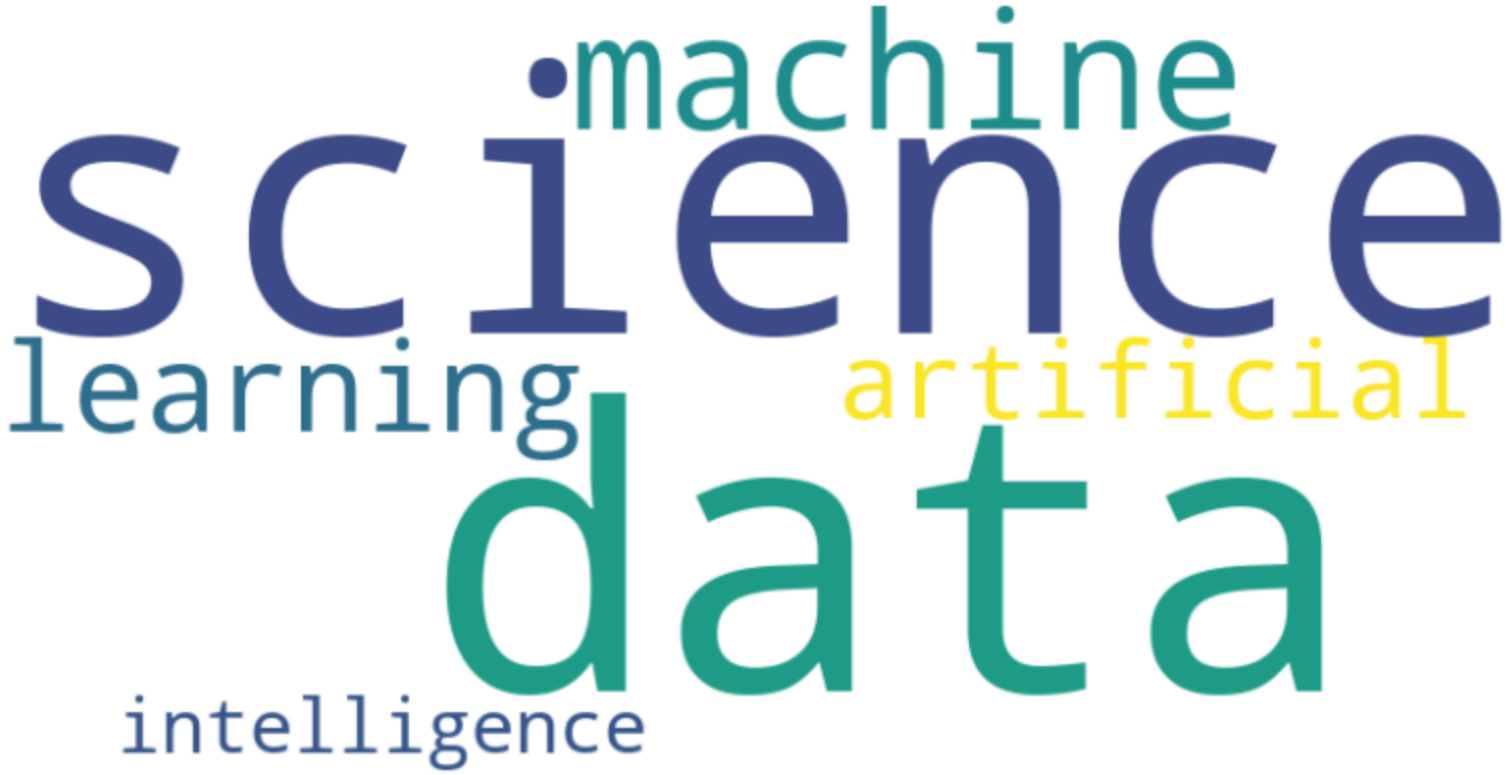
```
Defaulting to user installation because normal site-packages is not writeable

  WARNING: The script wordcloud_cli.exe is installed in 'C:\Users\ADMIN\AppData\Roaming\Python\Python39\Scri
pts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-lo
cation.


Collecting wordcloud
  Downloading wordcloud-1.9.4-cp39-cp39-win_amd64.whl (300 kB)
     ---------------------------------------- 300.4/300.4 kB 3.1 MB/s eta 0:00:00
Requirement already satisfied: pillow in e:\programdata\anaconda3\lib\site-packages (from wordcloud) (9.2.0)
Requirement already satisfied: matplotlib in e:\programdata\anaconda3\lib\site-packages (from wordcloud) (3.
5.2)
Requirement already satisfied: numpy>=1.6.1 in e:\programdata\anaconda3\lib\site-packages (from wordcloud)
(1.21.5)
Requirement already satisfied: packaging>=20.0 in e:\programdata\anaconda3\lib\site-packages (from matplotli
b->wordcloud) (21.3)
Requirement already satisfied: pyparsing>=2.2.1 in e:\programdata\anaconda3\lib\site-packages (from matplotl
ib->wordcloud) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in e:\programdata\anaconda3\lib\site-packages (from matp
lotlib->wordcloud) (2.8.2)
Requirement already satisfied: cycler>=0.10 in e:\programdata\anaconda3\lib\site-packages (from matplotlib->
wordcloud) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in e:\programdata\anaconda3\lib\site-packages (from matplot
lib->wordcloud) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in e:\programdata\anaconda3\lib\site-packages (from matplot
lib->wordcloud) (1.4.2)
Requirement already satisfied: six>=1.5 in e:\programdata\anaconda3\lib\site-packages (from python-dateutil>
=2.7->matplotlib->wordcloud) (1.16.0)
Installing collected packages: wordcloud
Successfully installed wordcloud-1.9.4
```

In [20]:

```
#12.Write a Python script to generate a WordCloud from the text: 'data science machine learning artificia
from wordcloud import WordCloud
import matplotlib.pyplot as plt
text = 'data science machine learning artificial intelligence'
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(text)
output_image = 'wordcloud.png'
wordcloud.to_file(output_image)
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
print(f"WordCloud saved as {output_image}")
```




```
WordCloud saved as wordcloud.png
```

In [ ]:
```
1
```

In [ ]:
```
1
```

In [ ]:
```
1
```