In [2]:

```
import re

def filter_regional_language(text, language_pattern):
    filtered_text = re.sub(language_pattern, '', text)
    return filtered_text

hindi_pattern = re.compile(r'[\u0900-\u097F]+')
input_sentence = input("Enter a sentence: ")
print("Input Sentence:", input_sentence)
filtered_sentence = filter_regional_language(input_sentence, hindi_pattern)
print("Filtered Sentence:", filtered_sentence)
```

```
Enter a sentence: Hello प्रसाद जावळे
Input Sentence: Hello प्रसाद जावळे
Filtered Sentence: Hello
```

In [4]:

```
#Stop word Filtraton

import re

def filter_stop_words(sentence) :
    stop_words = set([
        "i", "me", "my", "myself", "we", "our", "ours","you", "your",
        "yours", "he", "him", "his", "himself", "she", "her", "hers", "it",
        "its", "itself", "they", "them", "their", "theirs", "what",
        "which", "who", "whom", "this", "that", "these", "those", "am",
        "is", "are", "was", "were", "be", "been", "being", "have", "has",
        "had", "having", "do","through", "during", "before", "after", "above",
        "below", "to", "from", "up", "down", "in", "out", "on", "off", "over", "und
        "again", "further", "then", "once", "here", "there", "when", "where",
        "why", "how", "all", "any", "both", "each", "few", "more", "most",
        "other", "some", "such", "no", "nor", "not", "only", "own", "same",
        "so", "than", "too", "very", "s", "t", "can", "will", "just", "don",
        "should", "now"
    ])

    word_pattern = re.compile(r'\b\w+\b')
    filtered_sentence = word_pattern.sub(lambda match: match.group() if match.group
    return filtered_sentence

input_sentence = input("Enter a sentence: ")
filtered_sentence = filter_stop_words(input_sentence)
print("Filtered Sentence:", filtered_sentence)
```

```
Enter a sentence:  Despite their high frequency in language, these words carry lit
tle substantial meaning on their own. Their primary purpose is to facilitate the g
rammatical structure of sentences and paragraphs. However, when conducting text an
alysis, it is common practice to filter out these stop words to focus on the more
informative content words and phrases. This helps improve the quality and relevanc
e of the extracted information for various NLP tasks.
Filtered Sentence:  Despite  high frequency  language,  words carry little substan
tial meaning   .  primary purpose   facilitate the grammatical structure of senten
ces and paragraphs. However,  conducting text analysis,   common practice  filter
stop words  focus  the  informative content words and phrases.  helps improve the
quality and relevance of the extracted information for various NLP tasks.
```

In [3]:

```
# Punctuation FIltration

import re

def filter_punctuation(sentence) :
    filtered_sentence = re.sub(r'[^\w\s]', '', sentence)
    return filtered_sentence

input_sentence = input("Enter a sentence: ")
filtered_sentence = filter_punctuation(input_sentence)
print("Filtered Sentence:", filtered_sentence)
```

```
Enter a sentence: Hello! Welcome world, how are you? Have a good day.
Filtered Sentence: Hello Welcome world how are you Have a good day
```

In [6]:

```
#Email verification

import re

def is_valid_email(email) :
    pattern = r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$'
    return re.match(pattern, email)

input_email = input("Enter an email address: ")

if is_valid_email(input_email) :
   print("Email is valid.")
else :
   print("Email is not valid.")
```

```
Enter an email address: prasad.32
Email is not valid.
```

In [5]:

```
# Phone no validation

import re

def is_valid_phone_number(phone_number) :
    # Regular expression for basic phone number format validation
    pattern = r'^[2-9]\d{2}-\d{3}-\d{4}$'
    return re.match(pattern, phone_number)

input_phone_number = input("Enter a phone number : ")

if is_valid_phone_number(input_phone_number) :
  print("Phone number is valid.")
else :
  print("Phone number is not valid.")
```

```
Enter a phone number : 145879632a
Phone number is not valid.
```

In [7]:

```
# Name validation

import re

def is_valid_name(name) :
    # Regular expression for name validation (letters and spaces only)
    pattern = r'^[a-zA-Z\s]+$'
    return re.match(pattern, name)

input_name = input("Enter a name : ")

if is_valid_name(input_name) :
    print("Name is valid.")
else :
    print("Name is not valid.")
```

```
Enter a name : Prasad
Name is valid.
```

In [7]:

```
input = input("Enter a sentence : ")
```

```
Enter a sentence : hello how are you
```

In [8]:

```
# Tokenization

list = []
for item in input.split(" ") :
  list.append(item)
print(list)
```

```
['hello', 'how', 'are', 'you']
```