To identify what a web page is about using NLTK in Python, you would generally perform the following steps:

1. **Fetch the Web Page Content**: Use libraries like requests or BeautifulSoup to scrape the text content of the web page.
2. **Text Preprocessing**: Clean the text by removing HTML tags, stop words, and punctuation.
3. **Topic Identification**: Use NLP techniques such as word frequency analysis, named entity recognition (NER), or topic modeling to identify the main topics or themes of the page.

```
# Import necessary libraries

import requests

from bs4 import BeautifulSoup

import nltk

from nltk.corpus import stopwords

from nltk.tokenize import word_tokenize

from nltk.probability import FreqDist

from nltk import ne_chunk, pos_tag


# Download NLTK data

nltk.download('punkt')

nltk.download('stopwords')

nltk.download('averaged_perceptron_tagger')

nltk.download('maxent_ne_chunker')

nltk.download('words')


# Step 1: Fetch the Web Page Content

def fetch_webpage_content(url):

    try:

        response = requests.get(url)

        if response.status_code == 200:

            soup = BeautifulSoup(response.content, 'html.parser')

            # Extract text from all paragraph tags
```

```
            page_text = ' '.join([p.text for p in soup.find_all('p')])
            return page_text
        else:
            print(f"Error: Unable to fetch the webpage. Status code {response.status_code}")
            return None
    except Exception as e:
        print(f"An error occurred: {e}")
        return None

# Step 2: Preprocess the Text
def preprocess_text(text):
    stop_words = set(stopwords.words('english'))
    # Tokenize the text
    tokens = word_tokenize(text)
    # Convert to lowercase
    tokens = [word.lower() for word in tokens]
    # Remove punctuation and non-alphabetic characters
    words = [word for word in tokens if word.isalpha()]
    # Remove stopwords
    words = [word for word in words if word not in stop_words]
    return words

# Step 3: Frequency Analysis to Identify Topics
def identify_topics(words, num_topics=10):
    fdist = FreqDist(words)
    common_words = fdist.most_common(num_topics)
    return common_words

# Step 4: Named Entity Recognition (NER)
```

```
def named_entity_recognition(text):
    tokens = word_tokenize(text)
    pos_tags = pos_tag(tokens)
    ner_tree = ne_chunk(pos_tags, binary=False)
    return ner_tree

# Main Program
if __name__ == '__main__':
    # URL of the web page to analyze
    url = 'https://en.wikipedia.org/wiki/Natural_language_processing'

    # Step 1: Fetch Web Page Content
    page_content = fetch_webpage_content(url)
    if page_content:
        print("Web Page Content Fetched Successfully!")

        # Step 2: Preprocess the Text
        processed_text = preprocess_text(page_content)
        print(f"Processed Text Sample: {processed_text[:20]}")

        # Step 3: Identify Topics
        topics = identify_topics(processed_text)
        print("\nMost Common Topics Based on Word Frequency:")
        for word, freq in topics:
            print(f"{word}: {freq}")

        # Step 4: Named Entity Recognition (NER)
        print("\nNamed Entities in the Web Page:")
        ner_result = named_entity_recognition(page_content)
```

```
ner_result.pprint()
```