In [ ]:
```
##Sumedh Bhagat github-->  https://github.com/jnvw/chat_template_formater.git
```

In [1]:
```
import pyparsing as pp

def format_assistant_tag(content):
    return "{{#assistant}}{}\n{{/assistant}}".format(content)

def format_chat_template(input_text):
    # Define grammar for user and assistant segments
    user_segment = pp.nestedExpr('{{#user}}', '{{/user}}')
    assistant_segment = pp.nestedExpr('{{#assistant}}', '{{/assistant}}')

    # Parse the input text into user and assistant segments
    parsed_text = user_segment | assistant_segment
    parsed_segments = parsed_text.searchString(input_text)

    output_segments = []

    for segment in parsed_segments:
        if segment[0] == '#assistant':
            assistant_content = segment[1:-1]
            output_segments.append(format_assistant_tag(''.join(assistant_content))
        else:
            user_content = segment[1:-1]
            output_segments.append("{{#user}}{}\n{{/user}}".format(''.join(user_co

    if not any(format_assistant_tag("{{gen 'write' }}") in seg for seg in output_se
        output_segments.append(format_assistant_tag("{{gen 'write' }}"))

    return ' '.join(output_segments)

# Test Cases
test_case_1_input = "how are things going, tell me about Delhi\n{{gen 'write' }}"
test_case_2_input = "Tweak this proverb to apply to model instructions instead. Whe
test_case_3_input = "{{#user}}Hello\n{{/user}} {{#assistant}}{{gen 'respond'}}{{/as
test_case_4_input = "{{#user}}Greetings\n{{/user}}"

test_cases = [
    (test_case_1_input, "how are things going, tell me about Delhi\n{{gen 'write' }
    (test_case_2_input, "Tweak this proverb to apply to model instructions instead.
    (test_case_3_input, "{{#user}}Hello\n{{/user}} {{#assistant}}{{gen 'respond'}}{
    (test_case_4_input, "{{#user}}Greetings\n{{/user}}")
]

for idx, (input_text, expected_output) in enumerate(test_cases):
    formatted_output = format_chat_template(input_text)
    print(f"Test Case {idx+1} - Input:")
    print(input_text)
    print("\nExpected Output:")
    print(expected_output)
    print("\nFormatted Output:")
    print(formatted_output)
    print("\n" + "="*40 + "\n")
```

```
Test Case 1 - Input:
how are things going, tell me about Delhi
{{gen 'write' }}

Expected Output:
how are things going, tell me about Delhi
{{gen 'write' }}

Formatted Output:
{#assistant}{{gen 'write' }}
{/assistant}

========================================

Test Case 2 - Input:
Tweak this proverb to apply to model instructions instead. Where there is no guida
nce{{gen 'rewrite'}}

Expected Output:
Tweak this proverb to apply to model instructions instead. Where there is no guida
nce{{gen 'rewrite'}}

Formatted Output:
{#assistant}{{gen 'write' }}
{/assistant}

========================================

Test Case 3 - Input:
{{#user}}Hello
{{/user}} {{#assistant}}{{gen 'respond'}}{{/assistant}}

Expected Output:
{{#user}}Hello
{{/user}} {{#assistant}}{{gen 'respond'}}{{/assistant}}

Formatted Output:
{#user}
{/user} {#user}
{/user} {#assistant}{{gen 'write' }}
{/assistant}

========================================

Test Case 4 - Input:
{{#user}}Greetings
{{/user}}

Expected Output:
{{#user}}Greetings
{{/user}}

Formatted Output:
{#user}
{/user} {#assistant}{{gen 'write' }}
{/assistant}

========================================
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: