Refined Notes on Static vs Dynamic Prompts (Flowchart Explanation)

1. Model Invocation

- At the top, we have the **Model (LLM)**.
- The model is always **invoked** with a **prompt**.
- Prompts can be structured in **two main ways**:
    - **Single Message** → for single-turn queries.
    - **List of Messages** → for multi-turn conversations (chat format).

---

2. Single Message (Single-turn Standalone Queries)

- Used when the user asks **one question only**, without conversation history.
- The prompt is a single block of text.

a) Static Message

- Hardcoded, fixed prompt.
- Doesn't adapt to input.
- Example:

```python
prompt = "Write a summary of climate change."
```

Output will always be about climate change, no matter what the user wants.

b) Dynamic Message (PromptTemplate)

- Flexible prompt with **variables**.
- Can take user input and adapt accordingly.
- Example:

```python
from langchain.prompts import PromptTemplate

template = "Write a summary of {topic} in {style}."
prompt = PromptTemplate(
    input_variables=["topic", "style"],
    template=template
)

final_prompt = prompt.format(topic="Artificial Intelligence", style="simple terms")
```

✅ Output changes dynamically based on topic and style.

---

3. List of Messages (Multi-turn Conversation)

- Used in **chat-based interactions** (like ChatGPT).
- The model sees a **sequence of messages**:
    - **System Message** → defines the assistant's role.
    - **Human Message** → user's query.
    - **AI Message** → model's past responses.

a) Static Messages

- Predefined, fixed role messages.
- Example:

```python
messages = [
    {"role": "system", "content": "You are a helpful tutor."},
    {"role": "user", "content": "Explain Newton's 3rd law."}
]
```

Here, the role is fixed: "helpful tutor".

b) Dynamic Messages (ChatPromptTemplate)

- Messages adapt dynamically based on user context, variables, or history.
- Example:

```python
from langchain.prompts.chat import ChatPromptTemplate

template = ChatPromptTemplate.from_messages([
    ("system", "You are a {role} specialized in {domain}."),
    ("human", "{question}")
])

final_prompt = template.format(role="tutor", domain="physics", question="Explain Newton's 3rd law.")
```

✅ Here, the system role and domain are dynamically filled.

Key Takeaways

- **Static Messages:** Fixed, simple, but not flexible.
- **Dynamic Messages (Prompt Templates):** Adaptable, context-aware, and preferred for real-world apps.
- **Single Message:** Best for standalone queries.
- **List of Messages:** Best for multi-turn conversations (chat history).

👉 In short:

- **Static = One-size-fits-all (bad for scaling).**
- **Dynamic = Flexible, user-aware (best practice).**
- **PromptTemplate = Single query dynamic prompting.**
- **ChatPromptTemplate = Multi-turn dynamic prompting.**