

[🏠](#) [■ Modules](#) [■ Memory](#) [■ How-to](#) [■ Conversation buffer memory](#)

Conversation buffer memory

This notebook shows how to use `ConversationBufferMemory`. This memory allows for storing of messages and then extracts the messages in a variable.

We can first extract it as a string.

```
from langchain.memory import ConversationBufferMemory
```

```
memory = ConversationBufferMemory()  
memory.save_context({"input": "hi"}, {"output": "whats up"})
```

```
memory.load_memory_variables({})
```

```
{'history': 'Human: hi\nAI: whats up'}
```

We can also get the history as a list of messages (this is useful if you are using this with a chat model).

```
memory = ConversationBufferMemory(return_messages=True)  
memory.save_context({"input": "hi"}, {"output": "whats up"})
```

```
memory.load_memory_variables({})
```

```
{'history': [HumanMessage(content='hi', additional_kwargs={}),  
             AIMessage(content='whats up', additional_kwargs={})]}
```

Using in a chain

Finally, let's take a look at using this in a chain (setting `verbose=True` so we can see the prompt).

```
from langchain.llms import OpenAI  
from langchain.chains import ConversationChain  
  
llm = OpenAI(temperature=0)  
conversation = ConversationChain(  
    llm=llm,  
    verbose=True,  
    memory=ConversationBufferMemory()  
)
```

```
conversation.predict(input="Hi there!")
```

```
> Entering new ConversationChain chain...
```

Prompt after formatting:

The following is a friendly conversation between a human and an AI. The AI is talkative and provides lots of specific details from its context. If the AI does not know the answer to a question, it truthfully says it does not know.

Current conversation:

Human: Hi there!

AI:

> Finished chain.

" Hi there! It's nice to meet you. How can I help you today?"

```
conversation.predict(input="I'm doing well! Just having a conversation with an AI.")
```

> Entering new ConversationChain chain...

Prompt after formatting:

The following is a friendly conversation between a human and an AI. The AI is talkative and provides lots of specific details from its context. If the AI does not know the answer to a question, it truthfully says it does not know.

Current conversation:

Human: Hi there!

AI: Hi there! It's nice to meet you. How can I help you today?

Human: I'm doing well! Just having a conversation with an AI.

AI:

> Finished chain.

" That's great! It's always nice to have a conversation with someone new. What would you like to talk about?"

```
conversation.predict(input="Tell me about yourself.")
```

> Entering new ConversationChain chain...

Prompt after formatting:

The following is a friendly conversation between a human and an AI. The AI is talkative and provides lots of specific details from its context. If the AI does not know the answer to a question, it truthfully says it does not know.

Current conversation:

Human: Hi there!

AI: Hi there! It's nice to meet you. How can I help you today?

Human: I'm doing well! Just having a conversation with an AI.

AI: That's great! It's always nice to have a conversation with someone new. What would you like to talk about?

Human: Tell me about yourself.

AI:

```
> Finished chain.
```

```
" Sure! I'm an AI created to help people with their everyday tasks. I'm programmed to understand natural language and provide helpful information. I'm also constantly learning and updating my knowledge base so I can provide more accurate and helpful answers."
```

And that's it for the getting started! There are plenty of different types of memory, check out our examples to see them all