

[to PDF](#) ▶

# Sequential Chains

## Contents

- SimpleSequentialChain
- Sequential Chain

The next step after calling a language model is make a series of calls to a language model. This is particularly useful when you want to take the output from one call and use it as the input to another.

In this notebook we will walk through some examples for how to do this, using sequential chains. Sequential chains are defined as a series of chains, called in deterministic order. There are two types of sequential chains:

- `SimpleSequentialChain`: The simplest form of sequential chains, where each step has a singular input/output, and the output of one step is the input to the next.
- `SequentialChain`: A more general form of sequential chains, allowing for multiple inputs/outputs.

## SimpleSequentialChain

In this series of chains, each individual chain has a single input and a single output, and the output of one step is used as input to the next.

Let's walk through a toy example of doing this, where the first chain takes in the title of an imaginary play and then generates a synopsis for that title, and the second chain takes in the synopsis of that play and generates an imaginary review for that play.

```
%load_ext dotenv
%dotenv
```

[Skip to main content](#)

cannot find .env file

```
from langchain.llms import OpenAI
from langchain.chains import LLMChain
from langchain.prompts import PromptTemplate
```

```
# This is an LLMChain to write a synopsis given a title of a play.
llm = OpenAI(temperature=.7)
template = """You are a playwright. Given the title of play, it is your job to
write a synopsis for that title.
```

```
Title: {title}
Playwright: This is a synopsis for the above play:"""
prompt_template = PromptTemplate(input_variables=["title"], template=template)
synopsis_chain = LLMChain(llm=llm, prompt=prompt_template)
```

```
# This is an LLMChain to write a review of a play given a synopsis.
llm = OpenAI(temperature=.7)
template = """You are a play critic from the New York Times. Given the synopsis of
play, it is your job to write a review for that play.
```

```
Play Synopsis:
{synopsis}
Review from a New York Times play critic of the above play:"""
prompt_template = PromptTemplate(input_variables=["synopsis"], template=template)
review_chain = LLMChain(llm=llm, prompt=prompt_template)
```

```
# This is the overall chain where we run these two chains in sequence.
from langchain.chains import SimpleSequentialChain
overall_chain = SimpleSequentialChain(chains=[synopsis_chain, review_chain],
verbose=True)
```

```
review = overall_chain.run("Tragedy at sunset on the beach")
```

> Entering new SimpleSequentialChain chain...

Tragedy at Sunset on the Beach is a story of a young couple, Jack and Sarah, who are in love and looking forward to their future together. On the night of their anniversary, they decide to take a walk on the beach at sunset. As they are walking, they come across a mysterious figure, who tells them that their love will be tested in the near future.

[Skip to main content](#)

The figure then tells the couple that the sun will soon set, and with it, a tragedy will strike. If Jack and Sarah can stay together and pass the test, they will be granted everlasting love. However, if they fail, their love will be lost forever.

The play follows the couple as they struggle to stay together and battle the forces that threaten to tear them apart. Despite the tragedy that awaits them, they remain devoted to one another and fight to keep their love alive. In the end, the couple must decide whether to take a chance on their future together or succumb to the tragedy of the sunset.

Tragedy at Sunset on the Beach is an emotionally gripping story of love, hope, and sacrifice. Through the story of Jack and Sarah, the audience is taken on a journey of self-discovery and the power of love to overcome even the greatest of obstacles.

The play's talented cast brings the characters to life, allowing us to feel the depths of their emotion and the intensity of their struggle. With its compelling story and captivating performances, this play is sure to draw in audiences and leave them on the edge of their seats.

The play's setting of the beach at sunset adds a touch of poignancy and romanticism to the story, while the mysterious figure serves to keep the audience enthralled. Overall, Tragedy at Sunset on the Beach is an engaging and thought-provoking play that is sure to leave audiences feeling inspired and hopeful.

> Finished chain.

```
print(review)
```

Tragedy at Sunset on the Beach is an emotionally gripping story of love, hope, and sacrifice. Through the story of Jack and Sarah, the audience is taken on a journey of self-discovery and the power of love to overcome even the greatest of obstacles.

The play's talented cast brings the characters to life, allowing us to feel the depths of their emotion and the intensity of their struggle. With its compelling story and captivating performances, this play is sure to draw in audiences and leave them on the edge of their seats.

The play's setting of the beach at sunset adds a touch of poignancy and romanticism to the story, while the mysterious figure serves to keep the audience enthralled. Overall, Tragedy at Sunset on the Beach is an engaging and thought-provoking play that is sure to leave audiences feeling inspired and hopeful.

[Skip to main content](#)

# Sequential Chain

Of course, not all sequential chains will be as simple as passing a single string as an argument and getting a single string as output for all steps in the chain. In this next example, we will experiment with more complex chains that involve multiple inputs, and where there also multiple final outputs.

Of particular importance is how we name the input/output variable names. In the above example we didn't have to think about that because we were just passing the output of one chain directly as input to the next, but here we do have worry about that because we have multiple inputs.

```
# This is an LLMChain to write a synopsis given a title of a play and the era it
is set in.
llm = OpenAI(temperature=.7)
template = """You are a playwright. Given the title of play and the era it is set
in, it is your job to write a synopsis for that title.

Title: {title}
Era: {era}
Playwright: This is a synopsis for the above play:"""
prompt_template = PromptTemplate(input_variables=["title", 'era'],
template=template)
synopsis_chain = LLMChain(llm=llm, prompt=prompt_template, output_key="synopsis")
```

```
# This is an LLMChain to write a review of a play given a synopsis.
llm = OpenAI(temperature=.7)
template = """You are a play critic from the New York Times. Given the synopsis of
play, it is your job to write a review for that play.

Play Synopsis:
{synopsis}
Review from a New York Times play critic of the above play:"""
prompt_template = PromptTemplate(input_variables=["synopsis"], template=template)
review_chain = LLMChain(llm=llm, prompt=prompt_template, output_key="review")
```

```
# This is the overall chain where we run these two chains in sequence.
from langchain.chains import SequentialChain
overall_chain = SequentialChain(
    chains=[synopsis_chain, review_chain],
    input_variables=["era", "title"],
```

[Skip to main content](#)

```
output_variables=["synopsis", "review"],
verbose=True)
```

```
review = overall_chain({"title":"Tragedy at sunset on the beach", "era":
"Victorian England"})
```

```
> Entering new SequentialChain chain...
```

```
> Finished chain.
```

## Memory in Sequential Chains

Sometimes you may want to pass along some context to use in each step of the chain or in a later part of the chain, but maintaining and chaining together the input/output variables can quickly get messy. Using `SimpleMemory` is a convenient way to do manage this and clean up your chains.

For example, using the previous playwright SequentialChain, lets say you wanted to include some context about date, time and location of the play, and using the generated synopsis and review, create some social media post text. You could add these new context variables as `input_variables`, or we can add a `SimpleMemory` to the chain to manage this context:

```
from langchain.chains import SequentialChain
from langchain.memory import SimpleMemory

llm = OpenAI(temperature=.7)
template = """You are a social media manager for a theater company. Given the
title of play, the era it is set in, the date,time and location, the synopsis of
the play, and the review of the play, it is your job to write a social media post
for that play.

Here is some context about the time and location of the play:
Date and Time: {time}
Location: {location}

Play Synopsis:
{synopsis}
Review from a New York Times play critic of the above play:
{review}

Social Media Post:
"""
```

[Skip to main content](#)

```

"location"], template=template)
social_chain = LLMChain(llm=llm, prompt=prompt_template,
output_key="social_post_text")

overall_chain = SequentialChain(
    memory=SimpleMemory(memories={"time": "December 25th, 8pm PST", "location":
"Theater in the Park"}),
    chains=[synopsis_chain, review_chain, social_chain],
    input_variables=["era", "title"],
    # Here we return multiple variables
    output_variables=["social_post_text"],
    verbose=True)

overall_chain({"title": "Tragedy at sunset on the beach", "era": "Victorian
England"})

```

> Entering new SequentialChain chain...

> Finished chain.

```

{'title': 'Tragedy at sunset on the beach',
 'era': 'Victorian England',
 'time': 'December 25th, 8pm PST',
 'location': 'Theater in the Park',
 'social_post_text': "\nSpend your Christmas night with us at Theater in the Park
and experience the heartbreaking story of love and loss that is 'A Walk on the
Beach'. Set in Victorian England, this romantic tragedy follows the story of
Frances and Edward, a young couple whose love is tragically cut short. Don't miss
this emotional and thought-provoking production that is sure to leave you in
tears. #AWalkOnTheBeach #LoveAndLoss #TheaterInThePark #VictorianEngland"}

```