```
= Introduction to Neo4j & GenAI
:order: 1
:type: lesson

This course will get you started learning about using Neo4j with
Generative AI.

You will learn about:

* Large Language Models (LLMs)
* Knowledge Graphs
* How to use Neo4j for Retrieval Augmented Generation (RAG)
* Integrating Neo4j and LLMs using Python and Langchain

Throughout this course, you will explore how to leverage the capabilities
of Neo4j and Generative AI to build intelligent, context-aware systems.

In this lesson, you will learn about Generative AI and Knowledge Graphs.

[.video]
== Generative AI and Knowledge Graphs

video::S1aXauligDY[youtube,width=560,height=315]

[.transcript]
== Generative AI and Knowledge Graphs

Knowledge graphs are a specific implementation of a Graph Database, where
information is captured and integrated from many different sources,
representing the inherent knowledge of a particular domain.

They provide a structured way to represent entities, their attributes, and
their relationships, allowing for a comprehensive and interconnected
understanding of the information within that domain.

Knowledge graphs break down sources of information and integrate them,
allowing you to see the relationships between the data.

image::images/generic-knowledge-graph.svg[a diagram of an abstract
knowledge graph showing how sources contain chunks of data about topics
which can be related to other topics]

You can tailor knowledge graphs for semantic search, data retrieval, and
reasoning.

You may not be familiar with the term knowledge graph, but you have
probably used one. Search engines typically use knowledge graphs to
provide information about people, places, and things.

The following knowledge graph could represent Neo4j:

image::images/neo4j-google-knowledge-graph.svg[An example of a knowledge
graph of Neo4j showing the relationships between people, places, and
```

things]

This integration from diverse sources gives knowledge graphs a more holistic view and facilitates complex queries, analytics, and insights.

[TIP]
.Knowledge Graphs and Ontologies
For more on Knowledge Graphs, Ontologies, we recommend watching the link:https://www.youtube.com/watch?v=NQqWBnyQlS4&list=PL9Hl4pk2FsvX-5QPvwChB-ni_mFF97rCE[Going Meta â€“ A Series on Graphs, Semantics and Knowledge series on YouTube^].

Knowledge graphs can readily adapt and evolve as they grow, taking on new information and structure changes.

Neo4j is well-suited for representing and querying complex, interconnected data in Knowledge Graphs.
Unlike traditional relational databases, which use tables and rows, Neo4j uses a graph-based model with nodes and relationships.

=== Generative AI & Large Language Models

Generative AI is a class of algorithms and models that can generate new content, such as images, text, or even music.
New content is generated based on user prompting, existing patterns, and examples from existing data.

Large Language Models, referred to as LLMs, learn the underlying structure and distribution of the data and can then generate new samples that resemble the original data.

LLMs are trained on vast amounts of text data to understand and generate human-like text.
LLMs can answer questions, create content, and assist with various linguistic tasks by leveraging patterns learned from the data.

=== Instructing an LLM

The response generated by an LLM is a probabilistic continuation of the instructions it receives.
The LLM provides the most likely response based on the patterns it has learned from its training data.

In simple terms, if presented with the prompt _"Continue this sequence - A B C"_, an LLM could respond _"D E F"_.

To get an LLM to perform a task, you provide a **prompt**, a piece of text that should specify your requirements and provide clear instructions on how to respond.

image::images/llm-prompt-interaction.svg[A user asks an LLM the question 'What is an LLM? Give the response using simple language avoiding jargon.', the LLM responds with a simple definition of an LLM.]

Precision in the task description, potentially combined with examples or context, ensures that the model understands the intent and produces relevant and accurate outputs.

An example prompt may be a simple question.

```
    What is the capital of Japan?
```

Or, it could be more descriptive. For example:

```
    You are a friendly travel agent
    helping a customer to choose
    a holiday destination. Your readers
    may have English as a second
    language, so use simple terms
    and avoid colloquialisms.
    Avoid Jargon at all costs.

    Tell me about the capital of Japan.
```

The LLM will interpret these instructions and return a response based on the patterns it has learned from its training data.

=== Potential Problems

While LLMs provide a lot of potential, you should also be cautious.

At their core, LLMs are highly complex predictive text machines.
LLM's don't know or understand the information they output; they simply predict the next word in a sequence.

The words are based on the patterns and relationships from other text in the training data.
The sources for this training data are often the internet, books, and other publicly available text.
This data could be of questionable quality and maybe be incorrect.
Training happens at a point in time, it may not reflect the current state of the world and would not include any private information.

LLMs are fine-tuned to be as helpful as possible, even if that means occasionally generating misleading or baseless content, a phenomenon known as **hallucination**.

For example, when asked to _"Describe the moon."_ an LLM may respond with _"The moon is made of cheese."_. While this is a common saying, it is not true.

image::images/confused-llm.svg[A diagram of a confused LLM with a question mark thinking about the moon and cheese.]

While LLMs can represent the essence of words and phrases, they don't possess a genuine understanding or ethical judgment of the content.

Large Language Models (LLMs) are often considered "black boxes" due to the difficulty deciphering their decision-making processes.
The LLM would also be unable to provide the sources for its output or explain its reasoning.

image::images/llm-blackbox.svg[An LLM as a black box, responding to the question 'How did you determine that answer?' with 'I don't know.']

These factors can lead to outputs that might be biased, devoid of context, or lack logical coherence.

=== Fixing Hallucinations

Providing additional *contextual* data helps to _ground_ the LLM's responses and make them more accurate.

A knowledge graph is a mechanism for providing additional data to an LLM.
Data within the knowledge graph can guide the LLM to provide more relevant, accurate, and reliable responses.

While the LLM uses its language skills to interpret and respond to the contextual data, it will not disregard the original training data.

You can think of the original training data as the base knowledge and linguistic capabilities, while the contextual information guides in specific situations.

The combination of both approaches enables the LLM to generate more meaningful responses.

== Check Your Understanding

include::questions/1-hallucination.adoc[leveloffset=+1]
include::questions/2-fixing-hallucination.adoc[leveloffset=+1]

[.summary]
== Lesson Summary

In this lesson, you learned about LLMs, their benefits, and challenges.

In the next lesson, you will learn about hallucination and the strategies for avoiding it.