

```
= GraphRAG for Python
:order: 1
:type: lesson
:branch: main
```

The link:<https://neo4j.com/docs/neo4j-graphrag-python/current/>[GraphRAG for Python^] package (`neo4j-graphrag`) allows you to access Neo4j Generative AI functions including:

- Retrievers
- GraphRAG pipelines
- Knowledge graph construction

The purpose is to provide a first party package to developers, where Neo4j can guarantee long term commitment and maintenance as well as being fast to ship new features and high performing patterns and methods.

You will use the `neo4j-graphrag` package to create retrievers and implement simple applications that use GraphRAG to provide context to LLM queries.

You must set up a development environment to run the code examples and exercises.

```
include:.../.../.../.../.../shared/courses/codespace/get-started.adoc[]
```

```
[%collapsible]
```

```
.Develop on your local machine
```

```
====
```

You will need link:<https://python.org>[Python] installed and the ability to install packages using `pip`.

You may want to set up a virtual environment using

link:<https://docs.python.org/3/library/venv.html>[`venv`^] or

link:<https://virtualenv.pypa.io/en/latest/>[`virtualenv`^] to keep your dependencies separate from other projects.

Clone the

link:{repository-link}[[github.com/neo4j-graphacademy/genai-fundamentals](https://github.com/neo4j-graphacademy/genai-fundamentals)]  
repository:

```
[source,bash]
```

```
----
```

```
git clone https://github.com/neo4j-graphacademy/genai-fundamentals
```

```
----
```

Install the required packages using `pip`:

```
[source,bash]
```

```
----
```

```
cd genai-fundamentals
```

```
pip install -r requirements.txt
```

```
----
```

You do not need to create a Neo4j database as you will use the provided sandbox instance.

The sandbox uses Neo4j's GenAI functions, you can find out more about how to configure them in the [link:https://neo4j.com/docs/cypher-manual/current/genai-integrations/\[Neo4j GenAI integration documentation^\]](https://neo4j.com/docs/cypher-manual/current/genai-integrations/[Neo4j GenAI integration documentation^]).

====

== Setup the environment

Create a copy of the ``.env.example`` file and name it ``.env``.  
Fill in the required values.

[source]

.Create a .env file

----

include::{repository-raw}/{branch}/.env.example[]

----

Add your Open AI API key (``OPENAI_API_KEY``), which you can get from [link:https://platform.openai.com\[platform.openai.com^\]](https://platform.openai.com[platform.openai.com^]).

Update the Neo4j sandbox connection details:

NEO4J\_URI:: [copy]#neo4j://{instance-ip}:{instance-boltPort}#

NEO4J\_USERNAME:: [copy]#{instance-username}#

NEO4J\_PASSWORD:: [copy]#{instance-password}#

== Test your setup

You can test your setup by running

``genai-fundamentals/test_environment.py`` - this will attempt to connect to the Neo4j sandbox and the OpenAI API.

You will see an ``OK`` message if you have set up your environment correctly. If any tests fail, check the contents of the ``.env`` file.

== Continue

When you are ready, you can move on to the next task.

read::Success - let's get started![]

[.summary]

== Summary

You have setup your environment and are ready to start this module.

In the next lesson, you will create a vector retriever and use the data to provide context to an LLM.