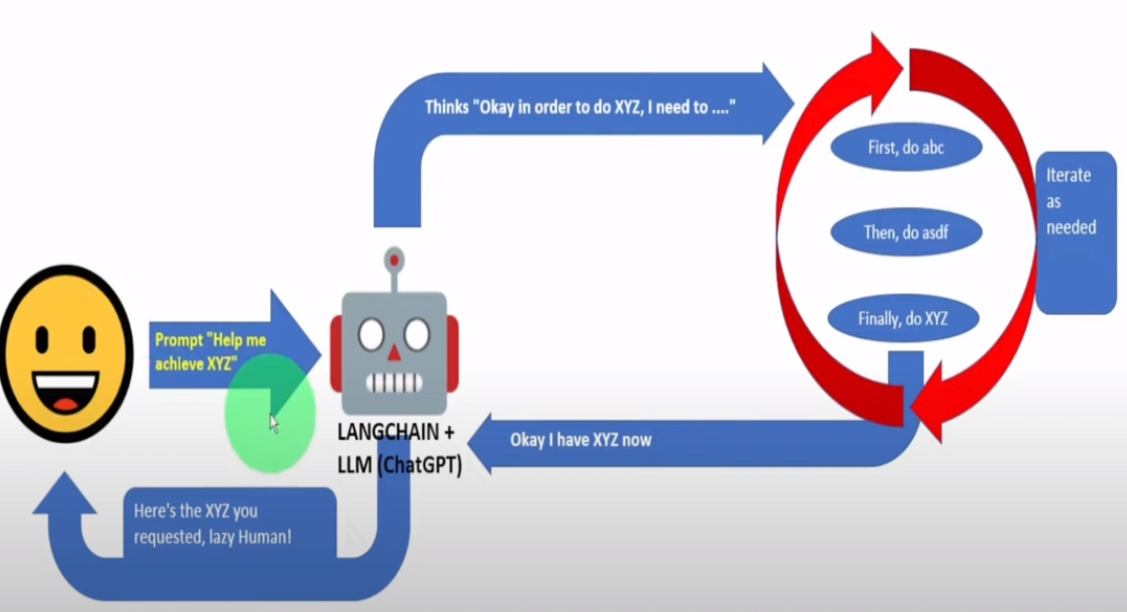
# Agents

The core idea of agents is to use an LLM to choose a sequence of actions to take. In chains, a sequence of actions is hardcoded (in code). In agents, a language model is used as a reasoning engine to determine which actions to take and in which order.



Agents use LLM to determine which action to take and in what order without hardcoding anything in code. An action can either be using a tool and observing its output, or returning a response to the user.

There are multiple types of agents in LangChain but most common is following

**Zero-shot React :** This agent uses the ReAct framework to determine which tool to use based solely on the tool's description. Any number of tools can be provided. This agent requires that a description is provided for each tool.

Note: This is the most general purpose action agent.

Every agent need different things to understand and break down given operation in plain language

into smaller pieces and perform them using given tools

1. Tools to perform operations like hit and get data from linkedin api, get data from google drive etc...

2. LLM for language capabilities

3. Agent type, we will use 'zero-shot React' which is most common

LLM model suffer from one limitation: they are limited to the knowledge on which they have been trained and the additional knowledge provided as context; as a result, if a useful piece of information is missing the provided knowledge, the model cannot “go around” and try to find it in other sources.

This is the reason why we need to introduce the concept of Agents. Agents can be seen as applications powered by LLMs and integrated with a set of tools like search engines, databases, websites, and so on. Within an agent, the LLM is the reasoning engine that, based on the user input, is able to plan and execute a set of actions that are needed to fulfill the request.

\*\*\*\*\*\*\*\*\*

# Python basics

DataClass

Introduced in python 3.7 It is a decorator @dataclass used to define classes which are intended to store data. When you mainly use class to store/represent data objects or store states rather than having multiple operations and methods then dataclass is used.

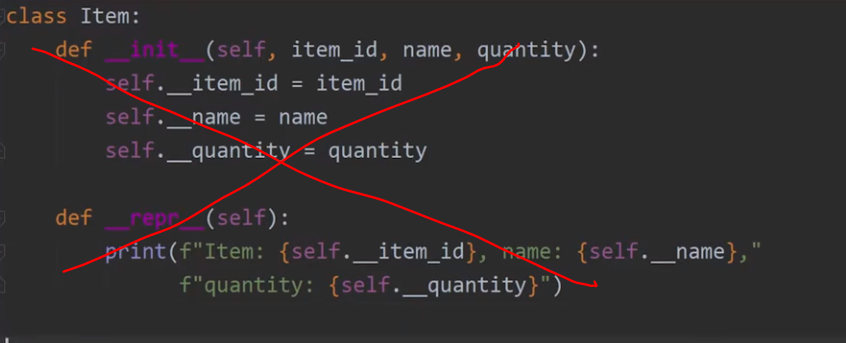
It automatically defines following methods for us when used it as a decorator on class

\_\_init\_\_(): Initialize the objects and assigns provided values to class attributes

\_\_repr\_\_(): provides string representation of a class object

\_\_eq\_\_(): Implements equality comparison between two objects

If we have normal class then nanaging and updating properfites in different methods become tedious over time.



Data class makes them easy by managing lots of things in background just by using @dataclass as decorator to class. It also provides lots more functionality. It also helps with inheriting classes.

