# How to work with large language models

## How large language models work

[Large language models][Large language models Blog Post] are functions that map text to text. Given an input string of text, a large language model predicts the text that should come next.

The magic of large language models is that by being trained to minimize this prediction error over vast quantities of text, the models end up learning concepts useful for these predictions. For example, they learn:

- how to spell

- how grammar works

- how to paraphrase

- how to answer questions

- how to hold a conversation

- how to write in many languages

- how to code

- etc.

They do this by “reading” a large amount of existing text and learning how words tend to appear in context with other words, and uses what it has learned to predict the next most likely word that might appear in response to a user request, and each subsequent word after that.

GPT-3 and GPT-4 power [many software products][OpenAI Customer Stories], including productivity apps, education apps, games, and more.

## How to control a large language model

Of all the inputs to a large language model, by far the most influential is the text prompt.

Large language models can be prompted to produce output in a few ways:

- \*\*Instruction\*\*: Tell the model what you want

- \*\*Completion\*\*: Induce the model to complete the beginning of what you want

- \*\*Scenario\*\*: Give the model a situation to play out

- \*\*Demonstration\*\*: Show the model what you want, with either:

- A few examples in the prompt

- Many hundreds or thousands of examples in a fine-tuning training dataset

An example of each is shown below.

### Instruction prompts

Write your instruction at the top of the prompt (or at the bottom, or both), and the model will do its best to follow the instruction and then stop. Instructions can be detailed, so don't be afraid to write a paragraph explicitly detailing the output you want, just stay aware of how many [tokens](https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them) the model can process.

Example instruction prompt:

```text

Extract the name of the author from the quotation below.

“Some humans theorize that intelligent species go extinct before they can expand into outer space. If they're correct, then the hush of the night sky is the silence of the graveyard.”

― Ted Chiang, Exhalation

```

Output:

```text

Ted Chiang

```

### Completion prompt example

Completion-style prompts take advantage of how large language models try to write text they think is mostly likely to come next. To steer the model, try beginning a pattern or sentence that will be completed by the output you want to see. Relative to direct instructions, this mode of steering large language models can take more care and experimentation. In addition, the models won't necessarily know where to stop, so you will often need stop sequences or post-processing to cut off text generated beyond the desired output.

Example completion prompt:

```text

“Some humans theorize that intelligent species go extinct before they can expand into outer space. If they're correct, then the hush of the night sky is the silence of the graveyard.”

― Ted Chiang, Exhalation

The author of this quote is

```

Output:

```text

Ted Chiang

```

### Scenario prompt example

Giving the model a scenario to follow or role to play out can be helpful for complex queries or when seeking imaginative responses. When using a hypothetical prompt, you set up a situation, problem, or story, and then ask the model to respond as if it were a character in that scenario or an expert on the topic.

Example scenario prompt:

```text

Your role is to extract the name of the author from any given text

“Some humans theorize that intelligent species go extinct before they can expand into outer space. If they're correct, then the hush of the night sky is the silence of the graveyard.”

― Ted Chiang, Exhalation

```

Output:

```text

Ted Chiang

```

### Demonstration prompt example (few-shot learning)

Similar to completion-style prompts, demonstrations can show the model what you want it to do. This approach is sometimes called few-shot learning, as the model learns from a few examples provided in the prompt.

Example demonstration prompt:

```text

Quote:

“When the reasoning mind is forced to confront the impossible again and again, it has no choice but to adapt.”

― N.K. Jemisin, The Fifth Season

Author: N.K. Jemisin

Quote:

“Some humans theorize that intelligent species go extinct before they can expand into outer space. If they're correct, then the hush of the night sky is the silence of the graveyard.”

― Ted Chiang, Exhalation

Author:

```

Output:

```text

Ted Chiang

```

### Fine-tuned prompt example

With enough training examples, you can [fine-tune][Fine Tuning Docs] a custom model. In this case, instructions become unnecessary, as the model can learn the task from the training data provided. However, it can be helpful to include separator sequences (e.g., `->` or `###` or any string that doesn't commonly appear in your inputs) to tell the model when the prompt has ended and the output should begin. Without separator sequences, there is a risk that the model continues elaborating on the input text rather than starting on the answer you want to see.

Example fine-tuned prompt (for a model that has been custom trained on similar prompt-completion pairs):

```text

“Some humans theorize that intelligent species go extinct before they can expand into outer space. If they're correct, then the hush of the night sky is the silence of the graveyard.”

― Ted Chiang, Exhalation

###

```

Output:

```text

Ted Chiang

```

## Code Capabilities

Large language models aren't only great at text - they can be great at code too. OpenAI's [GPT-4][GPT-4 and GPT-4 Turbo] model is a prime example.

GPT-4 powers [numerous innovative products][OpenAI Customer Stories], including:

- [GitHub Copilot] (autocompletes code in Visual Studio and other IDEs)

- [Replit](https://replit.com/) (can complete, explain, edit and generate code)

- [Cursor](https://cursor.sh/) (build software faster in an editor designed for pair-programming with AI)

GPT-4 is more advanced than previous models like `gpt-3.5-turbo-instruct`. But, to get the best out of GPT-4 for coding tasks, it's still important to give clear and specific instructions. As a result, designing good prompts can take more care.

### More prompt advice

For more prompt examples, visit [OpenAI Examples][OpenAI Examples].

In general, the input prompt is the best lever for improving model outputs. You can try tricks like:

- \*\*Be more specific\*\* E.g., if you want the output to be a comma separated list, ask it to return a comma separated list. If you want it to say "I don't know" when it doesn't know the answer, tell it 'Say "I don't know" if you do not know the answer.' The more specific your instructions, the better the model can respond.

- \*\*Provide Context\*\*: Help the model understand the bigger picture of your request. This could be background information, examples/demonstrations of what you want or explaining the purpose of your task.

- \*\*Ask the model to answer as if it was an expert.\*\* Explicitly asking the model to produce high quality output or output as if it was written by an expert can induce the model to give higher quality answers that it thinks an expert would write. Phrases like "Explain in detail" or "Describe step-by-step" can be effective.

- \*\*Prompt the model to write down the series of steps explaining its reasoning.\*\* If understanding the 'why' behind an answer is important, prompt the model to include its reasoning. This can be done by simply adding a line like "[Let's think step by step](https://arxiv.org/abs/2205.11916)" before each answer.

[Fine Tuning Docs]: https://platform.openai.com/docs/guides/fine-tuning

[OpenAI Customer Stories]: https://openai.com/customer-stories

[Large language models Blog Post]: https://openai.com/research/better-language-models

[GitHub Copilot]: https://github.com/features/copilot/

[GPT-4 and GPT-4 Turbo]: https://platform.openai.com/docs/models/gpt-4-and-gpt-4-turbo

[GPT3 Apps Blog Post]: https://openai.com/blog/gpt-3-apps/

[OpenAI Examples]: https://platform.openai.com/examples