

GPT-3 and Prompts: A quick primer



The key to building your wildest ideas with GPT3 is having a deep understanding of prompts.

The field of Artificial Intelligence has witnessed incredible advances in recent years. Most recently, you see Generative AI trickling into the mainstream. Text-to-image models like DALL-E, and Midjourney have demonstrated their extraordinary ability to create vibrant, life-like art from just a few words (often referred to as prompts), while the open-source Stable Diffusion has spawned an entirely new paradigm of generated images, ranging from uses in interior design to stock photography to AI generated avatars.

But it's not just drawing pretty pictures that AI has proven to be so skilled at - it's also harnessing the power of language. Text generation models, like OpenAI's GPT-3, are able to conjure up human-like text with a simple prompt.

The above image is an example of input sequence (not highlighted) and GPT-3 output (highlighted in green).

At its core, GPT-3 is an excellent next-word predictor. This might not seem so exciting at first glance, but it does unlock some pretty incredible capabilities.



When neural networks are large enough and trained on complicated enough problems like next-word prediction on a massive dataset from the internet, they take on some pretty surprising, magical properties.

Andrej Karpathy, Founding Member of OpenAI

So what are these “magical properties” of GPT-3 that Andrej talks about?

Well, it turns out, GPT-3 is a *meta-learner*. It has *learnt to learn*. And this remarkable, emergent ability to follow directions is the key to producing complex, tailored behaviour for use in various applications.

The key to obtaining good outputs is capitalizing on GPT-3’s powerful ability to follow directions. Now that we know that, let’s try to build something — a chatbot!

✂ OpenAI provides an excellent playground to help understand the GPT-3 API. This is where we’ll spend most of our time, so go ahead and sign up for a free OpenAI account [here](#), and navigate to the [playground](#).

The prompt that you provide is the bread and butter of your chatbot. This is where you get to instruct GPT-3 about what you want it to do. Remember, the core medium of interfacing with the API is text, so besides the prompt, there’s not much you need to set up in order for it to work.

💡 **NOTE:** In all interactions below, GPT-3 generated text is highlighted in green, whereas everything else is my input, formatted without highlighting

Key Understanding #1: Garbage In, Garbage Out

Probably the single most important rule for good outputs is providing GPT-3 with good context. Do you want it to write a humorous poem? A detailed scientific report? A persuasive argument? Simply instruct it to do so with a clear, unambiguous context.

Take a look at the prompt below.

You are a chatbot designed to help humans with technology. Respond appropriately to the user's texts.

user:

The initial line gives GPT-3 the base context it needs to respond appropriately to chat messages. Without it, GPT-3 might give you unrelated answers and go off on tangents, so let's stick to it for now. Go ahead and paste it into the playground so you can play with it.

Here's a screenshot of an interaction I had:

The text highlighted in green is a direct output from GPT3. Everything else is my input.

Cool right!? By simply asking GPT-3 to behave like a chatbot, I was able to obtain a decent set of answers for to questions. But we can

take it a lot further.

Alright, back to the note.

Let's say I want GPT-3 to include links to Wikipedia for any relevant terms in the generated answer. To accomplish this, all I have to do is describe it in the context.



Checkpoint

You should have created an account, looked around playground and generated a few things by now.

[Show us what you got so far.](#)

I have highlighted the content in red so you can deeply understand what all I wrote as the prompt.

What if I want GPT-3 to roleplay an alien? No problem!



Are you building in public?!

The GPT-3 playground gives you the ability to share your prompt via a share link in the top right corner of the page.

[Show us whatever you've cooked up so far.](#)

Obviously, the Xenon Tech Blasters aren't a real pair of headphones, but the name does sound unique and otherworldly. Our prompt seems to be working as intended! The more specific and detailed your context, the better GPT-3 is able to generate relevant and high-quality output.

Key Understanding 2: Prompting With Examples

If you're following along, the prompt you have right now is what's called a **zero-shot** prompt — where you instruct GPT-3 to generate text based on its existing knowledge and understanding of language, without providing any pattern of answering to help guide the response. Just the context alone makes a massive difference in the quality of answers GPT-3 generates, but changing the prompt from a **zero-shot** prompt to a **few-shot** prompt (one that provides multiple examples) can greatly improve the accuracy and coherence of the generated text.

For the purposes of a chatbot, you can go as [wild](#) as you want with your prompt, as long as it maintains the general structure. Earlier,

I developed askada.me, a GPT-3 powered chatbot that specializes on technology questions. Here's the prompt I used, verbatim:

Notice how I have structured the questions in a manner that resembles a real-life scenario and provided ideal responses to them for providing GPT-3 with some reference.

Key Understanding 3: Using Easily Query-able Return Types

While a prompt featuring the first two key concepts may be sufficient for personal experimentation with the GPT-3 API, you'll soon realize that using these prompts for apps in production may result in responses with extra whitespace or random line breaks. This can be frustrating when trying to extract specific information from the output, which is why it's important for the output to be structured in a format that can be easily parsed and queried on frontend, such as JSON.

How do we do this? Easy, Simply ask GPT-3 to respond to all messages in JSON! Here's a one-shot prompt that accomplishes this.

For demonstration purposes, here's what a response to a GPT-3 API request looks like:

Asking GPT-3 to respond in JSON makes it easier to extract the response on the frontend by calling a JSON parser on

choices[0].text of the response as follows.

Just the tip of the iceberg.

It has never been a better time to create powerful, AI-first applications, and GPT-3 makes it incredibly easy to get started. It's as easy as copying a snippet of code from the playground and plugging it into your codebase.

Think outside the box. Text is the universal interface — a wide range of what you can encode as text and decode from text is content that GPT-3 can be taught to generate. This leads to some surprising and unique applications, like a [GPT-3 chess engine built from encoding chess moves in PGN](#).

I truly believe that GPT-3 is one of the most exciting innovations of the past decade, and I implore you to play with it, experiment with it, and see what amazing things you can create. From personalized emails to a fantasy tongue like Elvish to a [math tutor speaking in the uwu voice](#), the possibilities are endless!

Till next time.

— Aarya

PS. a lil about writing good prompts.

Your understanding of prompts will help you interact better with GPT-3. This very skills is also extremely useful when you are dealing with images or using generative AI Tools such as Stable Diffusion, Dall-E, or Midjourney!

If you want to learn more about Prompt Engineering and how it applies to images, check out:



Prompt Engineering 101

[Read this note to learn more about prompt engineering.](#)

