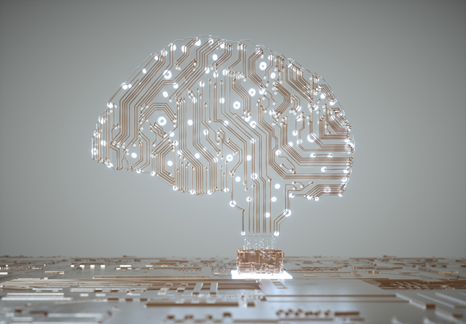
Introduction to Azure OpenAI Service

**Introduction**

Completed100 XP

* 3 minutes



Suppose you want to help your team understand the latest artificial intelligence (AI) innovations in the news. Your colleagues have questions about several headline-making key words, such as *ChatGPT*, *OpenAI*, and *machine learning models*. They would like to evaluate the opportunities these innovations support and understand what is done to keep AI advancements ethical.

You share with your team that today, stable AI models are regularly put into production and used commercially around the world. For example, Microsoft's existing Azure AI services have been handling the needs of businesses for many years to date.

Recently, OpenAI, an AI research company, published a chatbot known as ChatGPT and an image generation application known as DALL-E. Both displayed AI capabilities many have not seen before. These technologies were built with AI models which can take almost any natural language input from a user and return a machine-created human-like response.

You share with your team that Azure OpenAI Service enables users to build enterprise-grade solutions with OpenAI models. With Azure OpenAI, users can summarize text, get code suggestions, generate images for a web site, and much more. This module dives into these capabilities.

**Capabilities of OpenAI AI models**

There are three main categories of capabilities found in OpenAI AI models:

| **Capability** | **Examples** |
| --- | --- |
| *Generating natural language* | Such as: summarizing complex text for different reading levels, suggesting alternative wording for sentences, *and much more* |
| *Generating code* | Such as: translating code from one programming language into another, identifying and troubleshooting bugs in code, *and much more* |
| *Generating images* | Such as: generating images for publications from text descriptions *and much more* |

**What is generative AI**

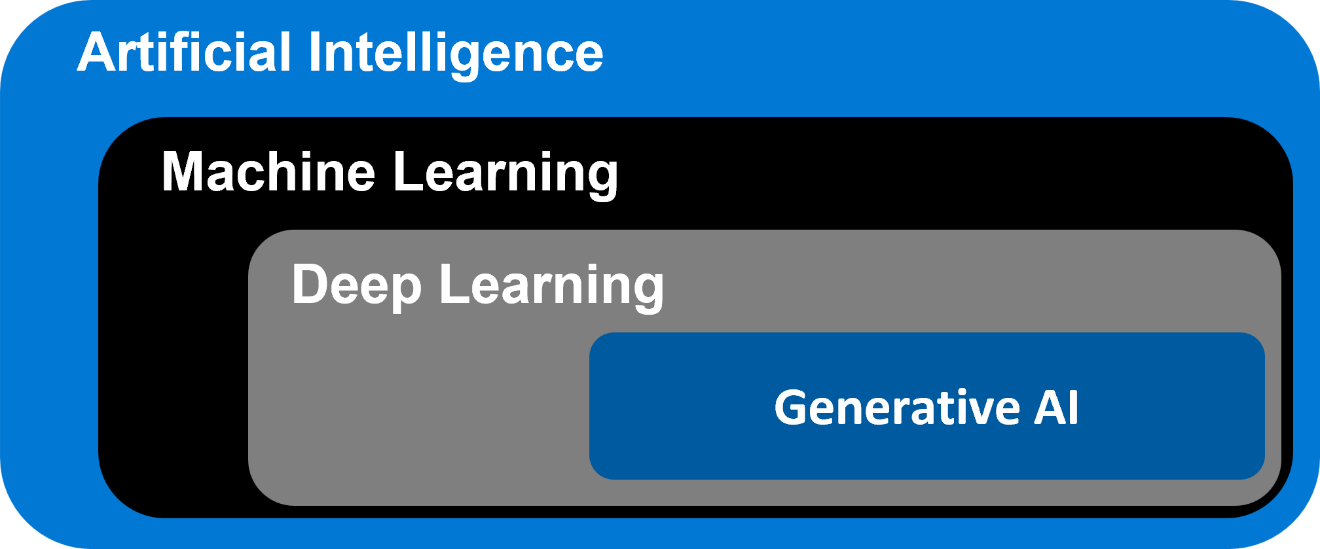
Completed100 XP

* 3 minutes

OpenAI makes its AI models available to developers to build powerful software applications, such as ChatGPT. There are tons of [other examples of OpenAI](https://platform.openai.com/examples) applications on the OpenAI site, ranging from practical, such as generating text from code, to purely entertaining, such as making up scary stories.

Let's identify where OpenAI models fit into the AI landscape.

* **Artificial Intelligence** imitates human behavior by relying on machines to learn and execute tasks without explicit directions on what to output.
* **Machine learning** models take in data like weather conditions and fit the data to an algorithm, to make predictions like how much money a store might make in a given day.
* **Deep learning** models use layers of algorithms in the form of artificial neural networks to return results for more complex use cases. Azure AI services are built on deep learning models. You can check out this article to learn more about the [difference between machine learning and deep learning.](https://learn.microsoft.com/en-us/azure/machine-learning/concept-deep-learning-vs-machine-learning)
* **Generative AI** models are a subset of deep learning models that can produce new content based on what is described in the input. The OpenAI models are a collection of generative AI models that can produce language, code, and images.



Next you'll learn how Azure OpenAI gives users the ability to combine Azure's enterprise-grade solutions with many of OpenAI's same generative AI models.

# Describe Azure OpenAI

Completed100 XP

* 5 minutes

Microsoft has partnered with OpenAI to deliver on three main goals:

* To utilize Azure's infrastructure, including security, compliance, and regional availability, to help users build enterprise-grade applications.
* To deploy OpenAI AI model capabilities across Microsoft products, including and beyond Azure AI products.
* To use Azure to power all of OpenAI's workloads.

## Introduction to Azure OpenAI Service

Azure OpenAI Service is a result of the partnership between Microsoft and OpenAI. The service combines Azure's enterprise-grade capabilities with OpenAI's generative AI model capabilities.

Azure OpenAI is available for Azure users and consists of four components:

* Pre-trained generative AI models
* Customization capabilities; the ability to fine-tune AI models with your own data
* Built-in tools to detect and mitigate harmful use cases so users can implement AI responsibly
* Enterprise-grade security with role-based access control (RBAC) and private networks

Using Azure OpenAI allows you to transition between your work with Azure services and OpenAI, while utilizing Azure's private networking, regional availability, and responsible AI content filtering.

### Understand Azure OpenAI workloads

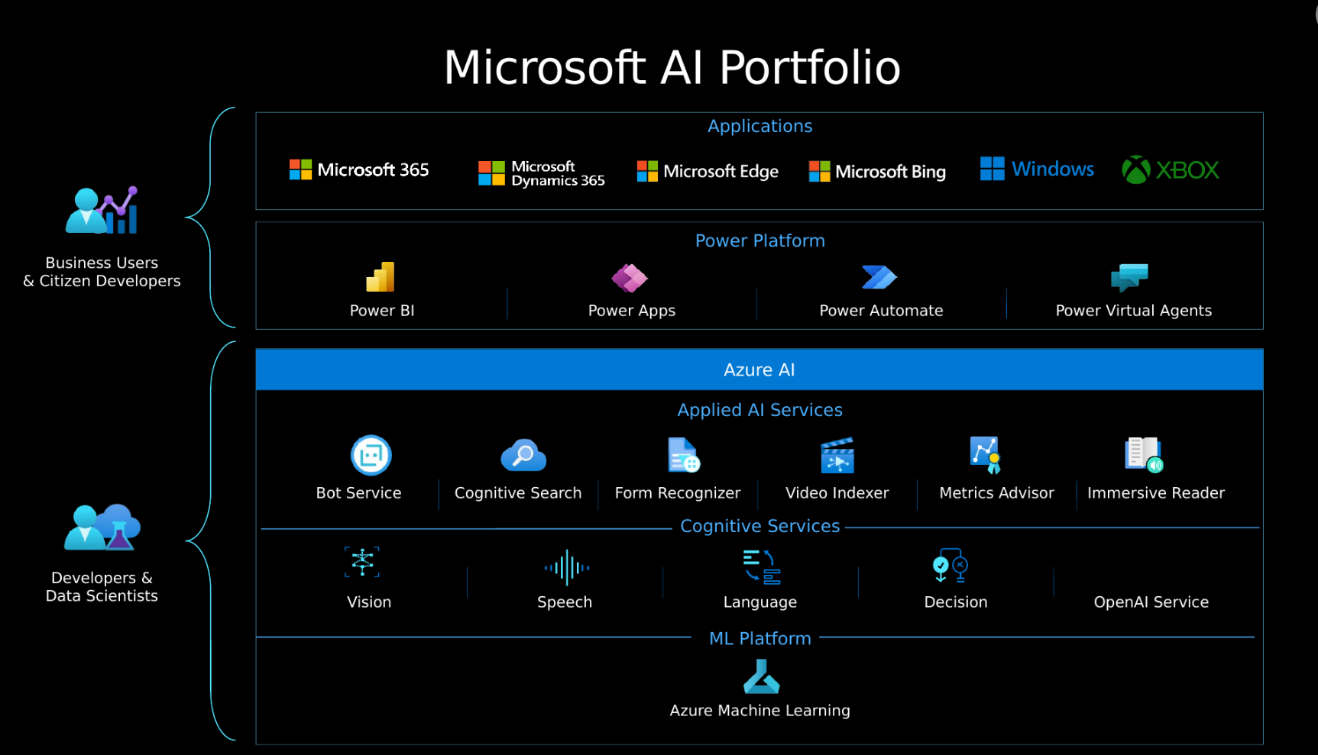
Azure OpenAI supports many common AI workloads and solves for some new ones.

Common AI workloads include machine learning, computer vision, natural language processing, conversational AI, anomaly detection, and knowledge mining.

Other AI workloads Azure OpenAI supports can be categorized into three groups:

* **Generating Natural Language**
  + Text completion: generate and edit text
  + Embeddings: search, classify, and compare text
* **Generating Code**: generate, edit, and explain code
* **Generating Images**: generate and edit images

### Azure OpenAI's relationship to Azure AI services



Azure's AI services are tools for solving AI workloads and can be categorized into Azure's Machine Learning platform, Cognitive Services, and Applied AI Services.

Azure Cognitive Services has five pillars: vision, speech, language, decision, and the Azure OpenAI Service. The services you choose to use depend on what you need to accomplish. In particular, there are several overlapping capabilities between the Cognitive Service's Language service and OpenAI's service, such as translation, sentiment analysis, and keyword extraction.

While there's no strict guidance on when to use a particular service, Azure's existing Language service can be used for widely known use-cases that require minimal tuning, the process of optimizing a model's performance. Azure OpenAI's service may be more beneficial for use-cases that require highly customized generative models, or for exploratory research.

**Note**

Pricing is different for Azure OpenAI and Azure Cognitive Service for Language. [**Learn more here**](https://azure.microsoft.com/pricing/details/cognitive-services/openai-service).

When making business decisions about what type of model to use or to understand why pre-trained machine learning models are so sought after, it's important to understand how time and compute needs factor into machine learning training. In order to get a working machine learning model, it needs to be trained with data. The 'learning' portion of training requires a computer to try all solutions until it identifies a model that best fits the data. A higher performing model, a more complex task, and a larger training dataset results in more time required to run through possible solutions.

# How to use Azure OpenAI

Completed100 XP

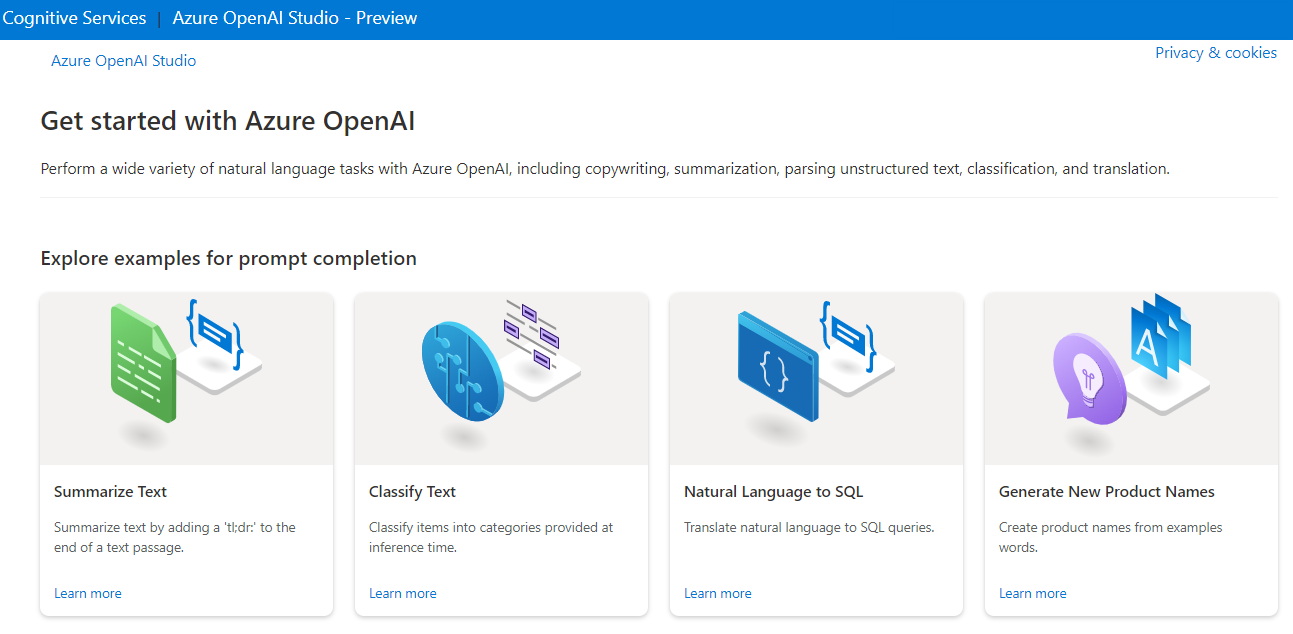
* 3 minutes

Currently you need to [apply](https://aka.ms/oaiapply) for access to Azure OpenAI. Once granted access, you can use the service by creating an Azure OpenAI resource, like you would for other Azure services. Once the resource is created, you can use the service through REST APIs, Python SDK, or the web-based interface in the Azure OpenAI Studio.

**Note**

To learn more about the basics of APIs, check out this infographic on [**how Azure APIs work**](https://github.com/MicrosoftLearning/AI-900-AIFundamentals/raw/main/instructions/infographics/ai900_how_auzre_api_works.pdf).

## Azure OpenAI Studio



In the Azure OpenAI Studio, you can build AI models and deploy them for public consumption in software applications. Azure OpenAI's capabilities are made possible by a few specific generative AI models. Different models are optimized for different tasks; some models excel at simple summarization tasks, some are great at general unstructured responses, and still others are built to generate unique images from text input.



These OpenAI models fall into a few main families:

* Generative Pre-trained Transformer (GPT)
* Codex
* DALL-E

[Embeddings](https://learn.microsoft.com/en-us/azure/cognitive-services/openai/concepts/understand-embeddings) are also available on Azure OpenAI. Embedding models are specifically created to be good at a particular task.

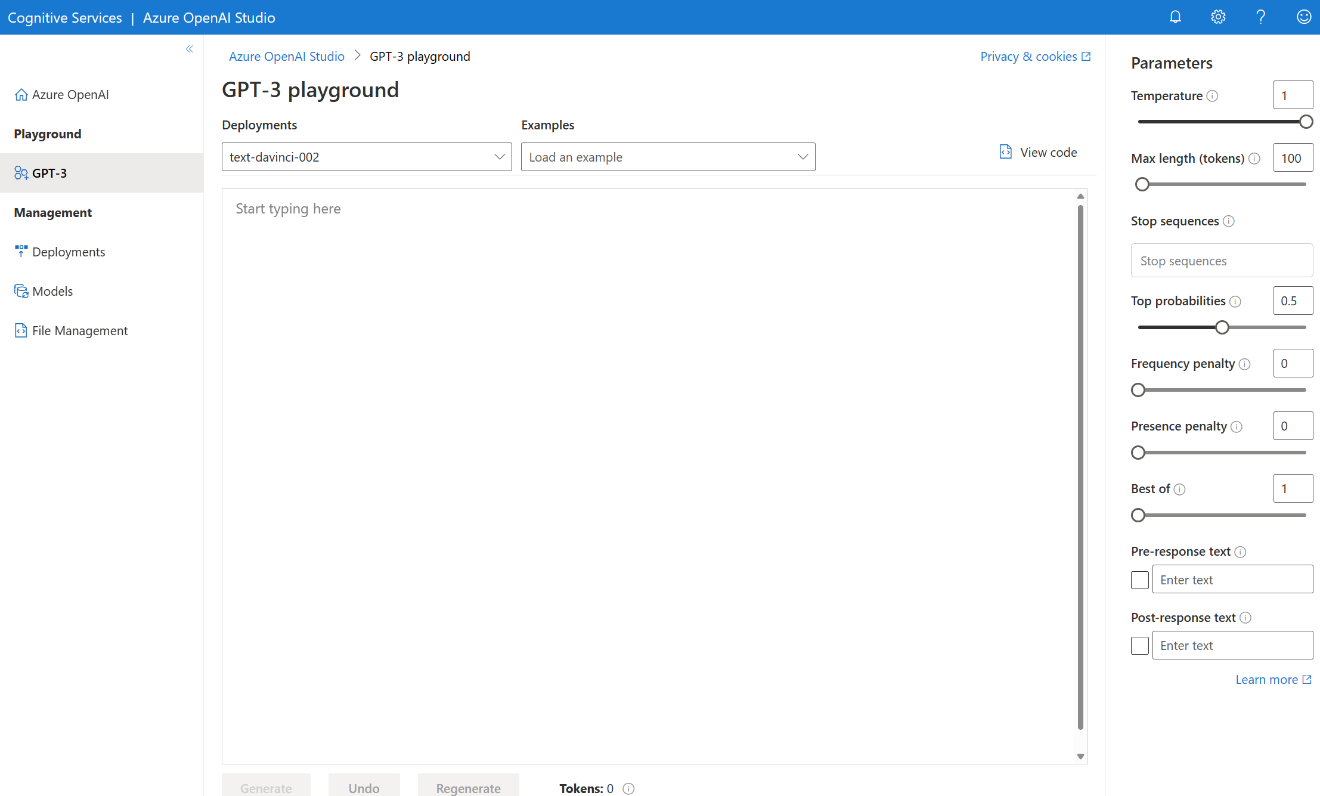
Azure OpenAI's AI models can all be trained and customized with fine-tuning. We won't go into custom models here, but you can learn more on the [fine-tuning your model](https://learn.microsoft.com/en-us/azure/cognitive-services/openai/how-to/fine-tuning?pivots=programming-language-studio%3Fazure-portal%3Dtrue) Azure documentation.

**Important**

Generative AI models always have a probability of reflecting true values. Higher performing models, such as models that have been fine-tuned for specific tasks, do a better job of returning responses that reflect true values. It is important to review the output of generative AI models. Additionally, the models provided by Azure OpenAI are trained on data prior to October 2019 and do not browse the internet, so they may not have knowledge of current events.

#### Playground

In the Azure OpenAI Studio, you can experiment with OpenAI models in a text box user interface called the GPT-3 Playground. You can type in prompts, and see responses without having to code.



# Understand OpenAI's natural language capabilities

Completed100 XP

* 5 minutes

Azure OpenAI's natural language models are able to take in natural language and generate responses.

Natural language learning models are trained on words or chunks of characters known as tokens. For example, the word "hamburger" gets broken up into the tokens ham, bur, and ger, while a short and common word like "pear" is a single token. These tokens are mapped into vectors for a machine learning model to use for training. When a trained natural language model takes in a user's input, it also breaks down the input into tokens.

## Understanding GPT models for natural language generation

Generative pre-trained transformer (GPT) models are excellent at both understanding and creating natural language. If you've seen recent news around AI answering questions or writing a paragraph based on a prompt, it likely could have been generated by a GPT model. GPT models often have the version appended to the end, such as GPT-2 or GPT-3. Azure OpenAI offers access to GPT-3, and will provide access to GPT-3.5 soon.

#### What does GPT-3 look like?

A key aspect of OpenAI's generative AI is that it takes an input, or **prompt**, to return a natural language, visual, or code response. GPT tries to infer, or guess, the context of the user's question based on the prompt.

GPT models are great at completing several natural language tasks, some of which include:

| **Task** | **Prompt** |
| --- | --- |
| Summarizing text | "Summarize this text into a short blurb" |
| Classifying text | "What genre of book is this?" |
| Generating names or phrases | "Write a tagline for my flower company" |
| Translation | "Translate 'How are you' to French" |
| Answering questions | "What does Azure OpenAI do?" |
| Suggesting content | "Give me the five best weddings songs" |

For example, given a prompt where the user types in text asking for a cooking recipe:

Copy

Write a recipe based on these ingredients and instructions:

Fruit Pie

Ingredients:

Strawberries

Blueberries

Flour

Eggs

Milk

Here's an example of the OpenAI API generated response:

Copy

Instructions:

1. Preheat your oven to 350 degrees Fahrenheit.

2. In a medium bowl, mix together 1 cup of flour and 1 egg until it forms a dough.

3. Grease a 9-inch pie dish with butter or non-stick cooking spray.

4. Roll out the dough and place it in the pie dish.

5. In a separate bowl, mix together 1 cup of strawberries, 1 cup of blueberries, and 1 cup of milk.

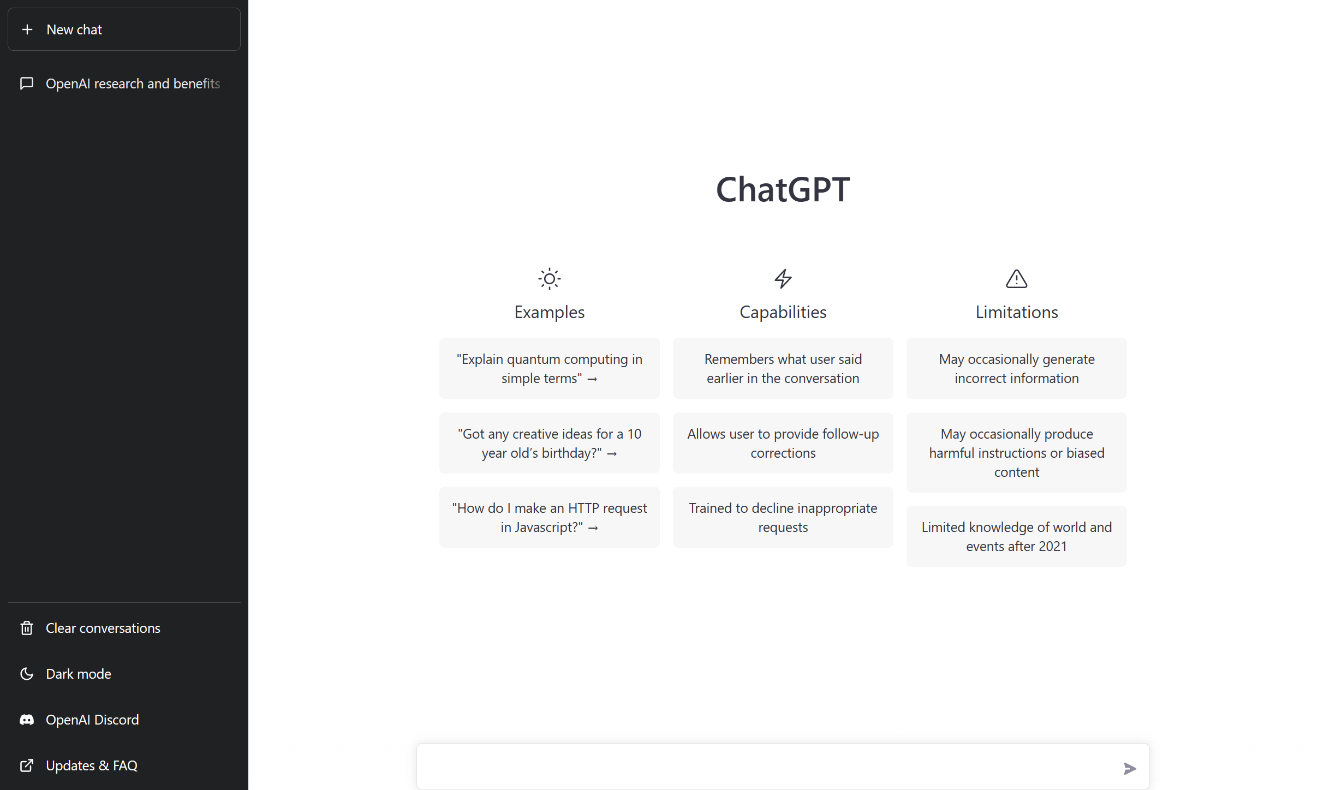
6. Pour the fruit mixture into the pie dish.

7. Bake

Delicious - maybe! It's important to understand that the generated responses are best guesses from a machine. In this case, the generated text may be useful for cooking something that tastes good in real life, or not.

## How models are applied to new use cases

ChatGPT is chatbot built on the GPT-3.5 generative AI model. You may have tried out ChatGPT's predictive capabilities in a chat portal similar to this screenshot, where you can type prompts and receive automated responses. The portal consists of the front-end user interface (UI) users see, and a back-end that includes the GPT-3.5 model. The combination of the front and back end can be described as a chatbot. The model provided on the back end is what is available as a building block with the OpenAI API. When you see ChatGPT-like capabilities in other applications, developers have taken the building blocks, customized them to a use case, and built them into the back end of new front-end user interfaces.



**Understand OpenAI code generation capabilities**

Completed100 XP

* 6 minutes

Code generation AI models are able to take natural language or code snippets and translate them into code. The OpenAI code generation model family, Codex, is proficient in over a dozen languages, such as C#, JavaScript, Perl, PHP, and is most capable in Python.

**Codex**

Codex models are based off of GPT-3 and are optimized to understand and write code. These models have been trained on both natural language and billions of lines of code from public repositories. Codex is able to generate code from natural language instructions such as code comments, and can suggest ways to complete code functions.

For example, given the prompt "Write a for loop counting from 1 to 10 in Python," the following answer is provided:

PythonCopy

for i in range(1,11):

print(i)

Code generation models can help developers code faster, understand new coding languages, and focus on solving bigger problems in their application. Developers can break down their goal into simpler tasks and use Codex to help build those out tasks using known patterns.

**Examples of code generation**

Part of the training data for GPT-3 included programming languages, so it's no surprise that GPT models can answer programming questions if asked. What's unique about the Codex model family is that it's more capable across more languages than GPT models.

Codex goes beyond just writing code from natural language prompts. Given the following code, it can generate unit tests:

PythonCopy

# Python 3

def mult\_numbers(a, b):

return a \* b

# Unit test

def

Codex builds out unit tests for our function:

PythonCopy

# Python 3

def mult\_numbers(a, b):

return a \* b

# Unit test

def test\_mult\_numbers():

assert mult\_numbers(3, 4) == 12

assert mult\_numbers(0, 10) == 0

assert mult\_numbers(4, 0) == 0

# Unit test

def test\_mult\_numbers\_negative():

assert mult\_numbers(-1, 10) == -10

assert mult\_numbers(10, -1) == -10

Codex can also summarize functions that are already written, explain SQL queries or tables, and convert a function from one programming language into another.

When interacting with Codex models, you can specify libraries or language specific tags to make it clear to Codex what we want. For example, we can provide this prompt formatted as an HTML comment: <!-- build a page titled "Let's Learn about AI" -->, and get this as a result:

HTMLCopy

<html>

<head>

<title>Let's Learn about AI</title>

<link rel="stylesheet" type="text/css" href="css/styles.css">

</head>

<body>

<div class="wrapper">

<div class="header">

<h1>Let's Learn about AI</h1>

</div>

<div class="nav">

<ul>

<li><a href="index.php">Home</a></li>

<li><a href="about.php">About</a></li>

<li><a href="resources.php">Resources</a></li>

<li><a href="contact.php">Contact</a></li>

</ul>

</div>

<div class="content">

<h2>Contact</h2>

<p>

<!-- build a form to contact the site owner -->

<form action="contact.php" method="post">

<label for="name">Name:</label>

<input type="text" name="name" id="name" required>

<label for="email">Email:</label>

<input type="email" name="email" id="email" required>

<label for="subject">Subject:</label>

<input type="text" name="subject" id="subject" required>

<label for="message">Message:</label>

<textarea name="message" id="message" required></textarea>

<input type="submit" value="Send">

</form>

</p>

</div>

<div class="footer">

<p>Let's Learn about AI</p>

</div>

</div>

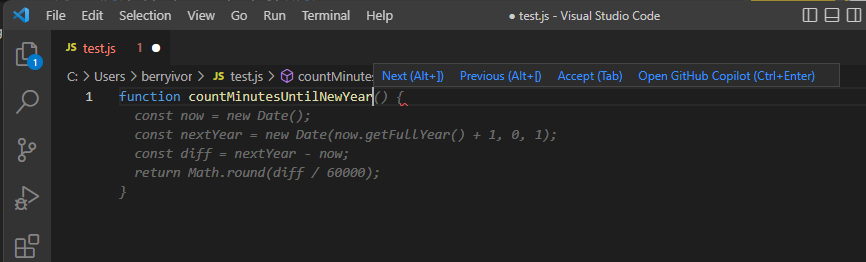
</body>

</html>

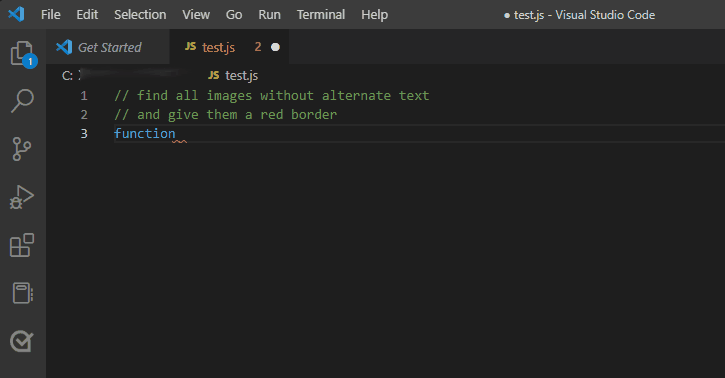
**GitHub Copilot**

OpenAI partnered with GitHub to create GitHub Copilot, which they call an AI pair programmer. GitHub Copilot integrates the power of OpenAI Codex into a plugin for developer environments like Visual Studio Code.

Once the plugin is installed and enabled, you can start writing your code, and GitHub Copilot starts automatically suggesting the remainder of the function based on code comments or the function name. For example, we have only a function name in the file, and the gray text is automatically suggested to complete it.



GitHub Copilot offers multiple suggestions for code completion, which you can tab through using keyboard shortcuts. When given informative code comments, it can even suggest a function name along with the complete function code.



**Understand OpenAI's image generation capabilities**

Completed100 XP

* 5 minutes

Image generation models can take a prompt, a base image, or both, and create something new. These generative AI models can create both realistic and artistic images, change the layout or style of an image, and create variations on a provided image.

**DALL-E**

In addition to natural language capabilities, these generative AI models can edit and create images. The model that works with images is called DALL-E. Much like GPT models, subsequent versions of DALL-E are appended onto the name, such as DALL-E 2. Image capabilities generally fall into the three categories of image creation, editing an image, and creating variations of an image.

**Image generation**

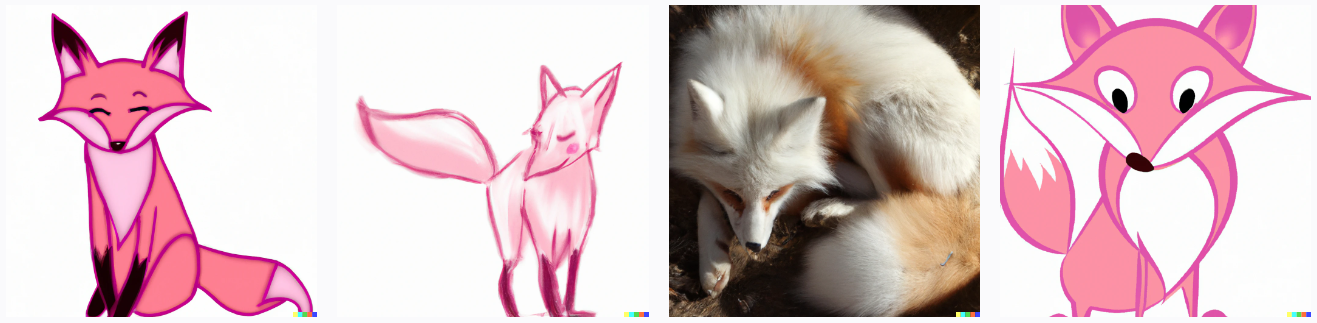
Original images can be generated by providing a text prompt of what you would like the image to be of. The more detailed the prompt, the more likely the model will provide a desired result.

With DALL-E, you can even request an image in a particular style, such as "a dog in the style of Vincent van Gogh". Styles can be used for edits and variations as well.

For example, given the prompt "an elephant standing with a burger on top, style digital art", the model generates digital art images depicting exactly what is asked for.



When asked for something more generic like "a pink fox", the images generated are more varied and simpler while still fulfilling what is asked for.



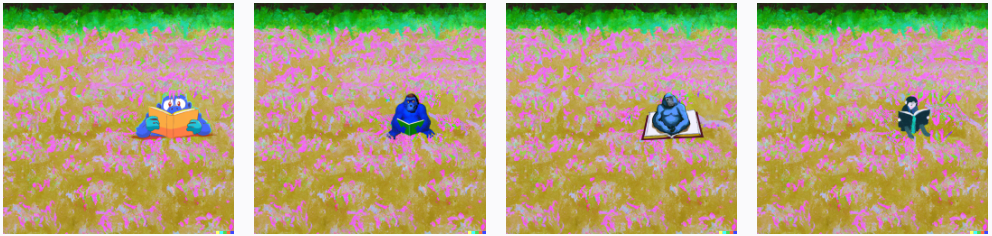
However when we make the prompt more specific, such as "a pink fox running through a field, in the style of Monet", the model creates much more similar detailed images.



**Editing an image**

When provided an image, DALL-E can edit the image as requested by changing its style, adding or removing items, or generating new content to add. Edits are made by uploading the original image and specifying a transparent mask that indicates what area of the image to edit. Along with the image and mask, a prompt indicating what is to be edited instructs the model to then generate the appropriate content to fill the area.

When given one of the above images of a pink fox, a mask covering the fox, and the prompt of "blue gorilla reading a book in a field", the model creates edits of the image based on the provided input.



**Image variations**

Image variations can be created by providing an image and specifying how many variations of the image you would like. The general content of the image will stay the same, but aspects will be adjusted such as where subjects are located or looking, background scene, and colors may change.

For example, if I upload one of the images of the elephant wearing a burger as a hat, I get variations of the same subject.



**Note**

Access to DALL-E is currently granted on an invite basis only.

**Describe Azure OpenAI's access and responsible AI policies**

Completed100 XP

* 3 minutes

It's important to consider the ethical implications of working with AI systems. Azure OpenAI provides powerful natural language models capable of completing various tasks and operating in several different use cases, each with their own considerations for safe and fair use. Teams or individuals tasked with developing and deploying AI systems should work to identify, measure, and mitigate harm.

Usage of Azure OpenAI should follow the six Microsoft [AI principles](https://learn.microsoft.com/en-us/azure/machine-learning/concept-responsible-ai):

* **Fairness**: Al systems shouldn't make decisions that discriminate against or support bias of a group or individual.
* **Reliability and Safety**: Al systems should respond safely to new situations and potential manipulation.
* **Privacy and Security**: Al systems should be secure and respect data privacy.
* **Inclusiveness**: Al systems should empower everyone and engage people.
* **Accountability**: People must be accountable for how Al systems operate.
* **Transparency**: AI systems should have explanations so users can understand how they're built and used.

Responsible AI principles guide [Microsoft's Transparency Notes on Azure OpenAI](https://learn.microsoft.com/en-us/legal/cognitive-services/openai/transparency-note), as well as explanations of other products. Transparency Notes are intended to help you understand how Microsoft's AI technology works, the choices system owners can make that influence system performance and behavior, and the importance of thinking about the whole system, including the technology, the people, and the environment.

If you haven't completed the [Get started with AI on Azure](https://learn.microsoft.com/en-us/training/modules/get-started-ai-fundamentals) module, it's worth reviewing its unit on [responsible AI](https://learn.microsoft.com/en-us/training/modules/get-started-ai-fundamentals/8-understand-responsible-ai).

**Limited access to Azure OpenAI**

As part of Microsoft's commitment to using AI responsibly, access to Azure OpenAI is currently limited. Customers that wish to use Azure OpenAI must submit a registration form for both initial experimentation access, and again for approval for use in production.

Additional registration is required for customers who want to modify content filters and abuse monitoring.

To apply for access and learn more about the limited access policy, see the [Azure OpenAI limited access](https://learn.microsoft.com/en-us/legal/cognitive-services/openai/limited-access) documentation.