



# OPEN AI MODELS

GPT3 | Codex | Content Filter



# FAMILY OF OPENAI MODELS

## GPT3

A set of models that can understand and generate natural language

## CODEX

A set of models that can understand and generate code, including translating natural language to code

## CONTENT FILTER

A fine-tuned model that can detect whether text may be sensitive or unsafe



# GPT-3

understand and generate natural language.

# COMPARING GPT-3 MODELS

## DAVINCI

It is the most capable model family and can perform any task the other models can perform and often with less instruction

Ideal for: **Complex intent, cause and effect, summarization for audience**

## CURIE

Curie is extremely powerful, yet very fast

Curie is quite capable for many nuanced tasks like sentiment classification and summarization. Curie is also quite good at answering questions and performing Q&A and as a general service chatbot

Ideal for: **Language translation, complex classification, text sentiment, summarization**

## BABBAGE

It can perform straightforward tasks like simple classification. It's also quite capable when it comes to Semantic Search ranking how well documents match up with search queries.

Ideal for: **Moderate classification, semantic search classification**

## ADA

Fastest Model

Ideal for: **Parsing text, simple classification, address correction, keywords**

Ada's performance can often be improved by providing more context.



# CODEx

Codex models are descendants of our GPT-3 models



# CODEx

## Descendants of GPT3

Codex models are descendants of our GPT-3 models that can understand and generate code. Their training data contains both natural language and billions of lines of public code from GitHub.

They're most capable in Python and proficient in over a dozen languages including JavaScript, Go, Perl, PHP, Ruby, Swift, TypeScript, SQL, and even Shell.

We can use Codex for a variety of tasks including:

- Turn comments into code
- Complete your next line or function in context
- Bring knowledge to you, such as finding a useful library or API call for an application
- Add comments
- Rewrite code for efficiency

# COMPARING CODEX MODELS

## code-davinci-002

Most capable Codex model

Particularly good at translating natural language to code

In addition to completing code, also supports inserting completions within code.

Max TOKEN REQUEST: 8,000

## code-cushman-001

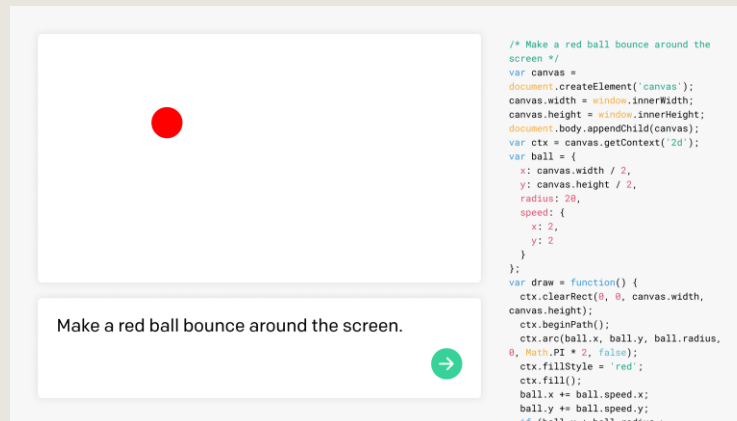
As capable as Davinci Codex, but slightly faster.

This speed advantage may make it preferable for real-time applications.

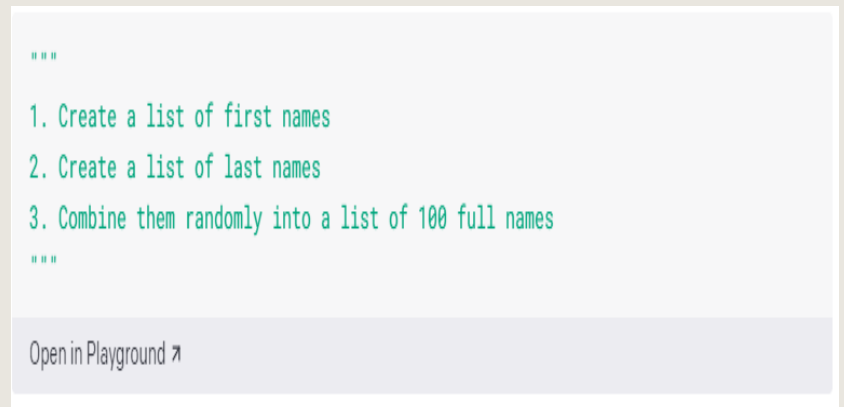
Max TOKEN REQUEST: up to 2400

# CODE COMPLETION ( LIMITED BETA)

see Codex in action, check out [Codex JavaScript Sandbox](#)



Create random names  
(Python)





# CODE COMPLETION ( LIMITED BETA)

## Create a MySQL query (Python)

```
"""  
Table customers, columns = [CustomerId, FirstName, LastName, Company, Address, Ci  
Create a MySQL query for all customers in Texas named Jane  
"""
```

query =

◀  ▶

[Open in Playground ↗](#)

## Explaining code (JavaScript)

```
// Function 1  
var fullNames = [];  
for (var i = 0; i < 50; i++) {  
  fullNames.push(names[Math.floor(Math.random() * names.length)]  
    + " " + lastNames[Math.floor(Math.random() * lastNames.length)]);  
}
```

// What does Function 1 do?

[Open in Playground ↗](#)



# CONTENT FILTERING

Detect generated text that could be sensitive/unsafe coming from the API

0

The text is safe.

1

his text is sensitive. This means that the text could be talking about a sensitive topic, something political, religious, or talking about a protected class such as race or nationality.

2

This text is unsafe. This means that the text contains profane language, prejudiced or hateful language, something that could be NSFW, or text that portrays certain groups/people in a harmful manner

Content Filter

The filter aims to detect generated text that could be sensitive or unsafe coming from the API. It's currently in beta mode and has three ways of classifying text- as safe, sensitive, or unsafe. The filter will make mistakes and we have currently built it to err on the side of caution, thus, resulting in higher false positives.

## LABEL DESCRIPTIONS



# THANK YOU

## References

- Models: <https://beta.openai.com/docs/models>
- Code Completion : <https://beta.openai.com/docs/guides/code/code-completion-limited-beta>
- Text completion: <https://beta.openai.com/docs/guides/completion>