

OpenAI Codex

User Manual

Prepared by Ali Asar Khowaja (2012285),
Bilal Pervez (2012292) & Hammad Suleman Khan (2012293)

SZABIST University

14th January, 2023

Overview	3
Working Architecture	3
About OpenAi Codex	4
OpenAi Codex Contributions	5
Reasons to Use OpenAi Codex	6
How to use the OpenAi Codex Platform?	6
Features And Smartness Of OpenAi Codex	7
Easy Code Generation	7
Smart to Understand Everything	7
Smart Code Completion	8
Interview Practice Assistance	8
Working With JavaScript Sandbox	9
Limitations and Gaps in OpenAi Codex	10

Overview

As a language model, OpenAI Codex is a machine learning system that has been trained to generate human-like text based on the input it receives. It does this by analyzing patterns and relationships in large amounts of text data and learning to predict what words and phrases are likely to come next in a given context. Codex is quite good at generating coherent and fluent code. Codex is a natural language processing model, based on OpenAI GPT-3, that can generate working code in over a dozen programming languages. It enables a user to write code just by giving instructions in a natural language like English, French or even Swiss dialect.

Working Architecture

To understand how all this magic happens behind the scenes let's dive into the way OpenAI Codex machine learning model work and its implementation:

First, make sure you have Python installed on your computer. You can check if Python is already installed by opening a terminal window and typing the following command:

```
python --version
```

If Python is not installed, you can download and install it from the Python website (<https://www.python.org/downloads/>).

Next, you will need to install the pip package manager, which is used to install Python packages such as OpenAI Codex. You can check if pip is already installed by typing the following command in a terminal window:

```
pip --version
```

If pip is not installed, you can install it by running the following command:

```
python -m ensurepip --upgrade
```

Once you have pip installed, you can use it to install the OpenAI Codex library. To do this, open a terminal window and type the following command:

```
pip install openai-codex
```

After the installation is complete, you can start using the OpenAI Codex library in your Python scripts. To use the library, you will need to import it at the top of your script:

```
import openai_codex
```

You can now use the functions and features of the OpenAI Codex library to perform machine learning tasks such as training and prediction. For example, you can use the predict function to make predictions with a pre-trained machine learning model:

Load a pre-trained machine learning model

```
model = openai_codex.load_model('model_name')
```

Make a prediction with the model

```
prediction = model.predict(input_data)
```

You can also use the OpenAI Codex library to train your own machine learning models. To do this, you will need to define a machine learning model and some training data. Then, you can use the fit function to train the model on the data:

Define a machine learning model

```
model = openai_codex.create_model('model_type')
```

Load training data

```
data = openai_codex.load_data('data_file.csv')
```

Train the model on the data

```
model.fit(data)
```

About OpenAi Codex

We all have heard that Artificial intelligence (AI) has seen significant growth in recent years. Advances in areas such as machine learning, natural language processing, and computer vision have led to the development of powerful AI systems that can perform tasks such as image and speech recognition, language translation, and decision-making. These technologies have become increasingly accessible, leading to a growing number of businesses and organizations adopting AI to automate processes and make more informed decisions. Additionally, the availability of vast amounts of data and the development of more powerful hardware have played a significant role in driving AI research and development. This growth has led to a wide range of new applications, from virtual personal assistants and self-driving cars to improved healthcare and financial services. AI is now impacting nearly every industry and is expected to continue to revolutionize the way we live and work. Codex's most famous contributions include Github co-pilot and ChatGPT. These contributions changed the way we used to work because of their extraordinary features and smartness.

OpenAi Codex has been used to develop a number of products and tools. Some examples include:

1. Codex Playground: An online platform that allows developers to experiment with Codex and generate code in a variety of programming languages.

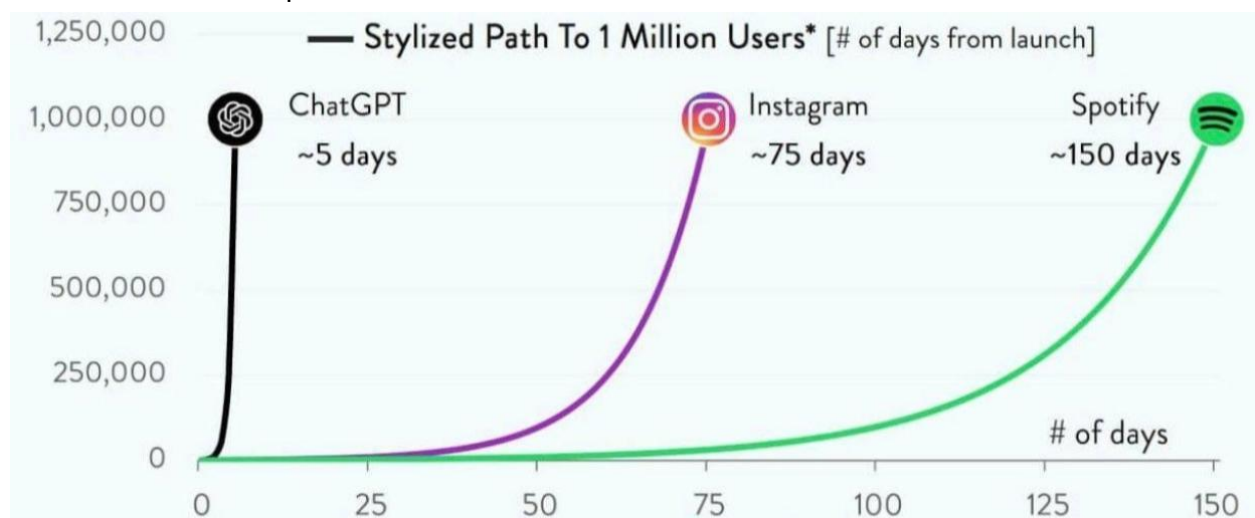
2. Codex Assist: A tool that uses Codex to help developers write code more efficiently by suggesting code snippets and completing lines of code.
3. Codex Code Search: A tool that allows developers to search for code snippets and examples across a large codebase.
4. Codex Code Completion: A tool that uses Codex to suggest code completions as the user types, to make the code writing process more efficient.
5. Codex Code Generation: A tool that uses Codex to generate code automatically based on input such as natural language descriptions or examples.
6. Codex Code Refactoring: A tool that uses Codex to automatically refactor and improve the quality of the existing codebase.

These products are intended to help developers write code more efficiently, and to help teams collaborate more effectively.

OpenAI Codex Contributions

In late 2022, we heard about the new product released by OpenAI Codex which is ChatGPT. It is a conversational language model developed by OpenAI. It is based on the GPT (Generative Pre-training Transformer) architecture, which uses unsupervised learning to pre-train a large neural network on a large corpus of text data, and then fine-tunes the network on a smaller dataset for a specific task. ChatGPT is trained on a large dataset of conversational text and is able to generate human-like responses to a wide range of topics. It is designed to be used in conversational applications such as chatbots, virtual assistants, and customer service bots. It can also be fine-tuned for specific use cases, such as answering questions or providing explanations. Additionally, ChatGPT is capable of understanding context and can continue a conversation based on the previous conversation history.

Here is the graph comparison of the popularity ChatGPT gained in few days compared to other famous social media platforms:



Reasons to Use OpenAi Codex

There are several reasons why one might choose to use OpenAI Codex, which is a language model for coding and programming. Some of these reasons include:

1. **Efficiency:** Codex can help developers write code more quickly and efficiently by suggesting code snippets and completing lines of code.
2. **Collaboration:** Codex can help teams collaborate more effectively by allowing developers to search for code snippets and examples across a large codebase.
3. **Automation:** Codex can be used to generate code automatically based on input such as natural language descriptions or examples, this can speed up the development process.
4. **Quality improvement:** Codex can be used to improve the quality of the existing codebase by automatically refactoring and detecting the potential bugs.
5. **Innovation:** Codex can be used to generate new ideas and solutions by providing developers with new perspectives on coding and programming.
6. **Learning aid:** Codex can be used as a learning aid for developers who are new to a programming language or who want to learn more about a specific topic.

This all can be used as a complementary tool to speed up the development process, it's important to have a good understanding of the underlying technology and best practices.

How to use the OpenAi Codex Platform?

We will be considering ChatGPT as the platform to use because of its ease and popularity. These steps can be used to access the power of OpenAI Codex:


1. Go to the website <https://openai.com/blog/chatgpt/> and **press the try ChatGPT** button.
2. As a **new user signup** with valid fields or directly with gmail or other social media accounts.
3. A mail will be dispatched to the account holder and **verify the account**.
4. Now **login** with your credentials to ChatGPT.
5. **Explore** the website's features.
6. **Write the prompt** on the textfield to get a response and **click the done button**.
7. A **response** will be generated which can be regenerated if required.



Features And Smartness Of OpenAi Codex


Easy Code Generation

A

Write a JAVA code to find even and odd number.



Here's a basic Java program that takes an integer as input and checks whether it's an even or odd number:  

 Copy code

```
import java.util.Scanner;


public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter an integer: ");
        int num = scanner.nextInt();


        if (num % 2 == 0) {
            System.out.println(num + " is an even number.");
        } else {
            System.out.println(num + " is an odd number.");
        }
    }
}
```



This program uses the modulus operator (`%`) to check the remainder of dividing the input number by 2. If the remainder is 0, the number is even. If the remainder is 1, the number is odd.

Smart to Understand Everything

A

Restructure this sentence: playing we ground football 





"We are playing football on the ground."  

Smart Code Completion



A

Complete this code of Python to find Fibonacci series:
def fibonacci(n):
 if n <= 0:
 return []
 elif n == 1:
 return [0]
 elif n == 2:
 return [0, 1]
 else:





Copy code





```
fib_list = [0, 1]
for i in range(2, n):
    next_fib = fib_list[i-1] + fib_list[i-2]
    fib_list.append(next_fib)
return fib_list
```

Language Translation with Accurate Responses



A

Dunya mein sab se bara mulk konsa hai?






Dunya mein sab se bara mulk China hai. China ki area 9.6 million square kilometers hai. Uske baad Russia hai jis ka area 17.1 million square kilometers hai.




Interview Practice Assistance



A

Ask me 3 question related to flutter for interview.





1. What are the core concepts of the Flutter framework and how do they differ from other mobile development frameworks?
2. How does the widget tree in Flutter work and how does it differ from a traditional component-based architecture?
3. Can you explain the role of the Dart programming language in Flutter and how it differs from other languages commonly used for mobile app development?



Working With JavaScript Sandbox

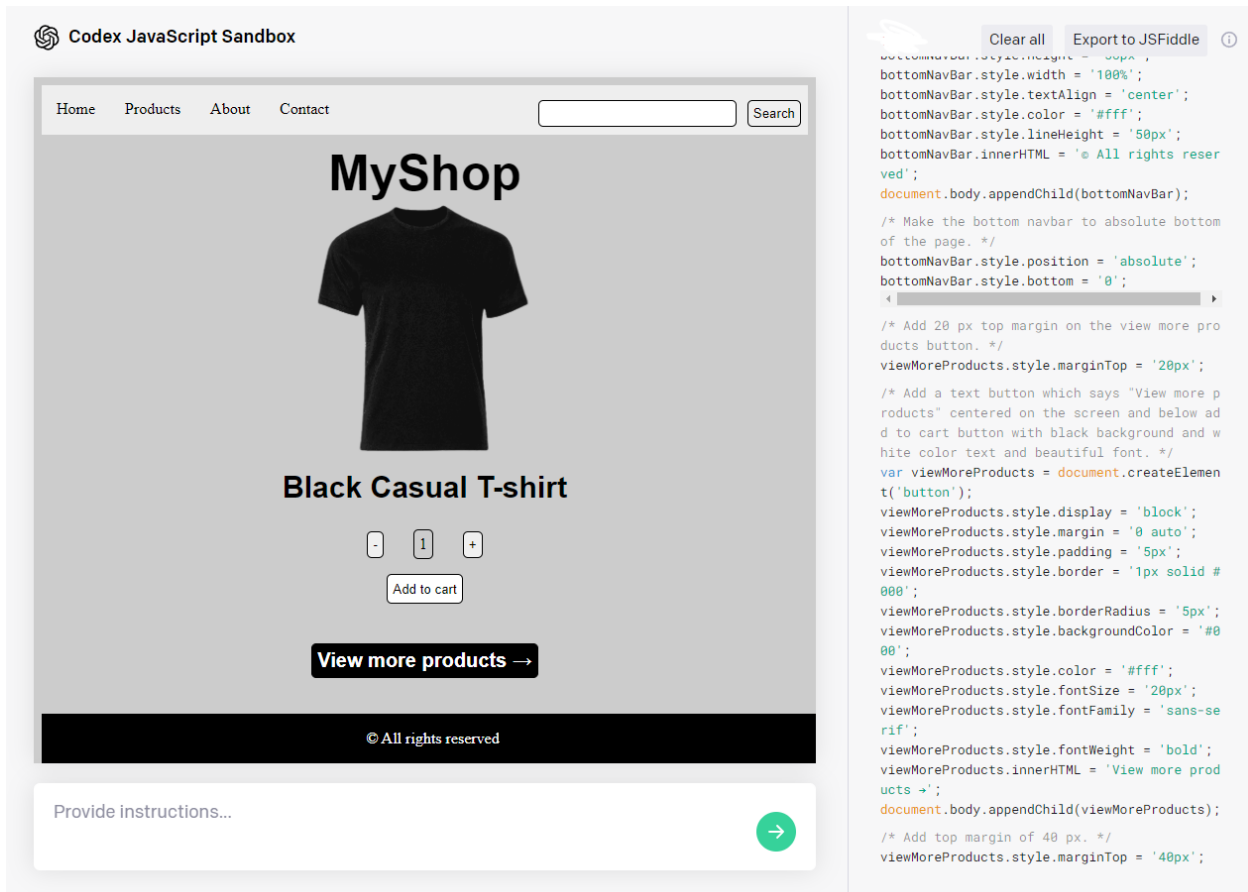
If we want to create something using JavaScript Sandbox we must know the proper details of the inputs we are giving. It must be accurate with proper technicalities and attributes that we require as output.

We created a webpage for a shopping website using the JS sandbox playground of OpenAI Codex. This demonstrates how the playground converts meaningful text into a visual context. By utilizing all of the features and functionalities provided by Codex, we can experiment with this tool.

Here are the steps to create a beautiful webpage for a shopping website:

1. Add a navigation bar on top of the page.
2. Add 4 texts on navbar with black color which says Home, Products, About and Contact.
3. On the right side of navbar add a search text input and button.
4. Align the search Button and input to absolute right hand side of navbar.
5. Add right margin of 80px for text input field.
6. Add a title of MyShop which beautiful font style in a box and centered top on screen.
7. Change top margin to 10px of title.
8. Add this image on center of the screen in a box below the title:
<https://o.remove.bg/downloads/65734bed-9572-42e9-bae4-713329bd8ff6/ts-removebg-preview.png>
9. Reduce size of image by 70%
10. Add a text "Black Casual T-shirt" centered and below the tshirt image in a box.
11. Add + and - button with a text in between it in a box below the tshirt text label.
12. Make top margin 10px.
13. Add a "Add to cart" button on the center of the screen in a box with white background color and black text.
14. Add a text button which says "View more products" centered on the screen and below add to cart button with black background and white color text and beautiful font.
15. Add top margin of 40 px. Add bottom navbar with black background color and white text on it which says "© All copyright reserved" and is centered on navbar.
16. Add bottom navbar with black background color and white text on it which says "© All copyright reserved" and is centered on navbar.
17. Make the bottom navbar to absolute bottom of the page.

After writing all these instructions steps wise, the output website will look something like this:



Limitations and Gaps in OpenAi Codex

As a language model, OpenAI Codex is a machine learning system that has been trained to generate human-like text based on the input it receives. It does this by analyzing patterns and relationships in large amounts of text data and learning to predict what words and phrases are likely to come next in a given context. While Codex is quite good at generating coherent and fluent code, it is still limited by the data it has been trained on and the algorithms that it uses to generate code.

1. One of the main limitations of Codex and other language models is that they are not able to understand or provide information about all topics. They are only able to generate text based on the patterns and relationships that they have learned from their training data, and they may not be able to generate accurate or relevant responses to input that is outside of their training data. This means that if you ask Codex a question about a topic that it has not been trained on or that is not covered in its training data, it is likely to provide an inaccurate or unrelated response.
2. Another important limitation of Codex and other language models is that they do not have access to the internet or other external sources of information. They are only able

to generate text based on the patterns and relationships that they have learned from their training data, and they are not able to update or modify their responses based on new information. This means that if you ask Codex a question about a current event or a topic that has changed since its training data was compiled like the new library version, it may not be able to provide an up-to-date or accurate response.

3. It is also important to keep in mind that Codex and other language models are not real people and do not have the same level of understanding or intelligence as humans. They are not able to think, reason, or learn in the same way that humans do, and they are limited by the capabilities of the machine learning algorithms that they use. They may not be able to understand or respond appropriately to complex or nuanced inputs, and they may not be able to generate responses that are as thoughtful or insightful as those of a human.
4. Since the codex model is trained to generate meaningful text and image responses, it cannot generate diagrams like sequence, flowcharts, etc because of its limitations. Moreover, if we ask a small complex riddle, it may generate a false or inaccurate response in return because of the way the data is trained.
5. Codex may help to find sentiments in context or inputs but it may not find the difference in what a joke is or a sarcasm is due to its non-human engineering which means it cannot sense the feelings of the inputs given by the users.
6. It have limited domain specific knowledge which means Codex has been trained on a wide range of programming languages and code, but it may not have the same level of expertise in a specific domain or language as a human developer
7. It also has limited ability to debug the code, since the methodology humans use are way different and efficient which are not developed in machine learning models otherwise they might be debugging their own models themselves. Moreover it has limited ability to understand the codebase structure. Codex can generate code, but it may not have the ability to understand the codebase structure, dependencies and the relationship between different parts of the codebase.

These limitations can be mitigated by providing Codex with more data and fine-tuning it for specific domains or languages. However, it's not a replacement for human developers for now.