# OpenAI

* The OpenAI API can be applied to virtually any task that involves understanding or generating natural language or code.
* OpenAi offer a spectrum of [models](https://beta.openai.com/docs/engines) with different levels of power suitable for different tasks, as well as the ability to [fine-tune](https://beta.openai.com/docs/guides/fine-tuning)  custom models.
* These models can be used for everything from content generation to semantic search and classification.

# Completion

* The [completions](https://beta.openai.com/docs/api-reference/completions) endpoint is at the center of API.
* It provides a simple text-in, text-out interface to models.
* Input some text as a **prompt**, and the model will generate a text **completion** that attempts to match whatever context or pattern you gave it.
* The completions endpoint can be used for virtually any task including content or code generation, summarization, expansion, conversation, creative writing, style transfer, and more.

# Prompts

* [Designing prompt](https://beta.openai.com/docs/guides/completion/prompt-design) is essentially how you “program” the model, usually by providing some instructions or a few examples.
* This is different from most other NLP services which are designed for a single task, such as sentiment classification or named entity recognition

# Tokens

* Models understand and process text by breaking it down into tokens.
* Tokens can be words or just chunks of characters.
* 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.
* Many tokens start with a whitespace, for example “ hello” and “ bye”.

# Engines

* BASE SERIES: A set of GPT-3 models that can understand and generate natural language.
* INSTRUCT SERIES: A set of specialized models that are similar to the base series, but better at following your instructions.
* CODEX SERIES: 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.

# BASE SERIES

* Base GPT-3 models can understand and generate natural language.
* Ffour base models called Davinci, curie, babbage, and ada with different levels of power suitable for different tasks are offered.
* Davinci is the most capable model, and Ada is the fastest.

# DAVINCI

* Davinci is the most capable engine
* Davinci can perform any task the other models can perform and often with less instruction.
* For applications requiring a lot of understanding of the content, like summarization for a specific audience and creative content generation, Davinci is going to produce the best results.
* These increased capabilities require more compute resources.
* Davinci costs more per API call .
* Davinci is not as fast as the other engines.
* Davinci also shines is in understanding the intent of text.
* Davinci is quite good at solving many kinds of logic problems and explaining the motives of characters.
* Davinci has been able to solve some of the most challenging AI problems involving cause and effect.

Good at: **Complex intent, cause and effect, summarization for audience.**

# CURIE

* Curie is extremely powerful
* Curie is  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.

Good at: **Language translation, complex classification, text sentiment, summarization**

# BABBAGE

* Babbage can perform straightforward tasks like simple classification
* Babbage is also quite capable when it comes to Semantic Search ranking how well documents match up with search queries

Good at:  **Moderate classification, semantic search classification**

# ADA

* Ada is usually the fastest model
* Ada can perform tasks like parsing text, address correction and certain kinds of classification tasks that don’t require too much nuance.
* Ada’s performance can often be improved by providing more context.

Good at:  P**arsing text, simple classification, address correction, keywords.**

# FINE-TUNED MODELS

Fine tuned models are trained to accomplish specific tasks.

# INSTRUCT SERIES

* Ada is usually the fastest model
* Ada can perform tasks like parsing text, address correction and certain kinds of classification tasks that don’t require too much nuance.
* Ada’s performance can often be improved by providing more context.

Good at:  P**arsing text, simple classification, address correction, keywords**