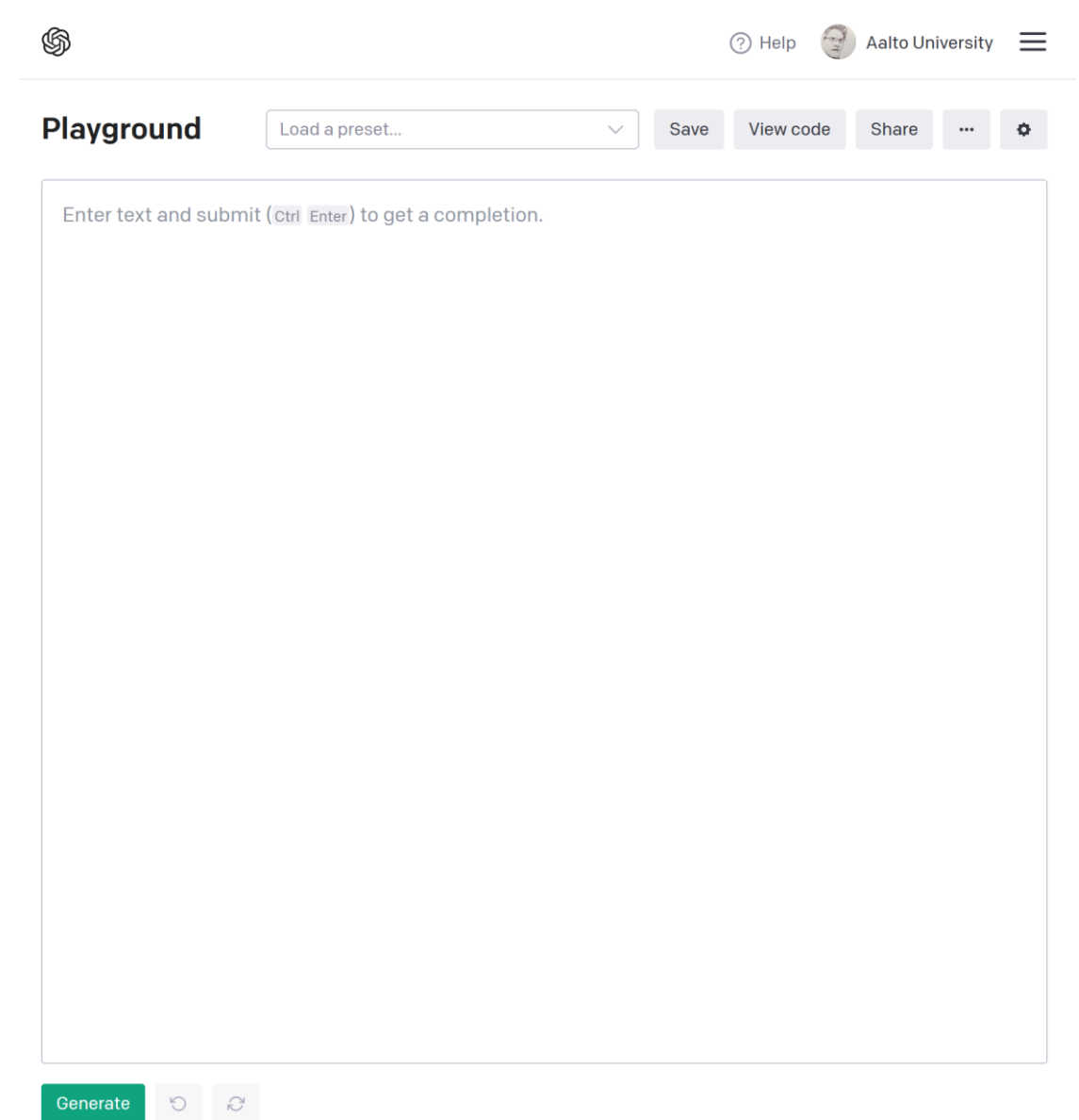


Exercise (entry-level): Back to the Playground

- Use insights from our in-lecture exercise to **generate a script** for e.g. a play, game cutscene, a movie scene, a short story, etc. Tips:
 - **Augment the dialogue starting lines** with additional information on intentions, moods, etc: "Sam (wants to escape the conversation):<generated text>"
 - **Write the beginning of the scene yourself**, giving an example of the tone and describing the setting and actors. Or use the beginning of your favourite book / play script, etc.
 - Select the **davinci engine** from the drop-down box on the right hand side, as it produces better fictional/creative output.
 - Try increasing the **response length parameter** to increase the output length
- **Completion:** Share outcome (prompt clearly separated) on Discord.



The screenshot shows the OpenAI GPT-3 Playground interface. At the top, there is a navigation bar with the OpenAI logo, a 'Help' link, a user profile icon, and the text 'Aalto University'. Below this, the main header reads 'Playground'. To the right of the header is a dropdown menu labeled 'Load a preset...' with a downward arrow. Further right are buttons for 'Save', 'View code', 'Share', a three-dot menu, and a settings gear icon. The central area is a large text input box with the placeholder text 'Enter text and submit (Ctrl Enter) to get a completion.'. At the bottom of the interface, there is a green 'Generate' button, followed by two circular icons: a refresh/clear icon and a redo icon.

OpenAI Gpt-3 Playground:
<https://beta.openai.com/playground>

Exercise (advanced): GPT-3 via OpenAI API

- Implement a Javascript / Python script that uses OpenAI's API to query GPT-3 programmatically. 100k tokens are free!
- **Example:** text game where GPT-3 provides both, player choices and consequences (cf. AI Dungeon: free choices).
- Potentially useful: **Natural Language Toolkit** for Python: <https://www.nltk.org/>
- OpenAI API Docs: <https://beta.openai.com/docs/introduction>
- OpenAI API Quickstart tutorial: <https://beta.openai.com/docs/quickstart>

The screenshot shows the OpenAI API documentation website. The browser address bar displays <https://beta.openai.com/docs/libraries>. The page has a navigation sidebar on the left with sections: GET STARTED (Introduction, Quickstart tutorial, Libraries, Python bindings, Node.js library, Community libraries), Engines, Going live, Usage guidelines, and Safety best practices. Under GUIDES, there are links for Completion, Classification, Search, Question answering, Fine-tuning, and Embeddings. Under API REFERENCE, there are links for Introduction, Authentication, Making requests, Engines, and Completions. The main content area is titled 'Libraries' and features a sub-section 'Python bindings'. It states: 'We provide Python bindings, which you can install as follows:'. Below this is a code block showing the command: `$ pip install openai`. Further down, it says: 'Once installed, you can use the bindings and your secret key to run the fol'. Below this is another code block showing a Python script snippet:

```
1 import os
2 import openai
3
4 # Load your API key from an environment variable or secret
5 openai.api_key = os.getenv("OPENAI_API_KEY")
6
7 response = openai.Completion.create(engine="text-davinci-001", prompt="Say hello to my friends and they will say hello to me!")
```

. Below the code block, it says: 'The bindings also will install a command-line utility you can use as follow'. At the bottom, there is a code block showing the command: `$ openai api completions.create -e text-davinci-001 -p "Say hello to my friends and they will say hello to me!"`. The page also has a 'Node.js library' section at the bottom.

Overview Documentation Examples

Libraries

Python bindings

We provide Python bindings, which you can install as follows:

```
$ pip install openai
```

Once installed, you can use the bindings and your secret key to run the fol

```
1 import os
2 import openai
3
4 # Load your API key from an environment variable or secret
5 openai.api_key = os.getenv("OPENAI_API_KEY")
6
7 response = openai.Completion.create(engine="text-davinci-001", prompt="Say hello to my friends and they will say hello to me!")
```

The bindings also will install a command-line utility you can use as follow

```
$ openai api completions.create -e text-davinci-001 -p "Say hello to my friends and they will say hello to me!"
```

Node.js library

We also have a Node.js library, which you can install by running the followi
your Node.js project directory: