Learning objectives

Learners will be able to...

- **Creating AI Playground**
- **Learning about File manipulation**
- **Generating key words to interact with program**

info

Make Sure You Know

You are familiar with Python.

Limitations

This is a gentle introduction. So there is a little bit of Python programming. The information might not be the most up to date as OpenAI releases new features.

Chat GPT-3

Generally we can :

1. Use GPT-3 to generate natural language text for a given topic. This can be used to quickly generate blog posts, essays, and other forms of copy.
2. Utilize GPT-3 to generate product descriptions, titles and other marketing materials.
3. Use GPT-3 to automate customer support tasks such as FAQs and customer service emails.
4. Leverage GPT-3 to generate technical documentation and software manuals.
5. Automate the process of summarizing long articles with GPT-3.
6. Generate personalized emails and messages with GPT-3.
7. Create interactive simulations, tutorials, and interactive demonstrations with GPT-3.
8. Use GPT-3 to generate lyrics for songs and poems.
9. Create engaging and interesting conversations with GPT-3.
10. Automate the process of creating resumes, cover letters, and other job application documents.

Sample Playground

We have used the playground to classify. For example try the following. GPT-3 can classify items into categories. Use the file on the left to write your prompts. Try the prompt below:

```
classify the following : cat, dog , car , plane
```

We are going to create our own playground to try and run the following 5 tasks.

1. Use GPT-3 to generate natural language text for a given topic. This can be used to quickly generate blog posts, essays, and other forms of copy.
2. Utilize GPT-3 to generate product descriptions, titles and other marketing materials.
3. Use GPT-3 to automate customer support tasks such as FAQs and customer service emails.
4. Leverage GPT-3 to generate technical documentation and software manuals.
5. Automate the process of summarizing long articles with GPT-3.

We will get to all of that but first we want to create a text file that is able to do what the playground on our left can do. Again on the idea of creating a more user-friendly program.

Let's ask the box to:

```
generate a blog post about a pizza shop
```

Our goal in this lesson is to recreate the box. The user should be able to write whatever they want and that should be passed on as the prompt. After that the generated answer will also be placed in your file.

Response

On the top left we have our Python file. On the bottom left we have the empty text file where the user will be able to simply write their prompts and generate and get their responses from.

For starters, we are going to use the same structure we have been using before to generate our responses. First, let's get our libraries.

```
import os
import openai
import secret
openai.api_key=secret.api_key
```

We need to read our file and just put the contents of it as our prompt.

```
#open the file
file = open("playground.txt", "r")
#read each line
filelines = file.readlines()
#put all the lines in a single variable
all_lines = ""
for line in filelines:
    all_lines += line
#close the file
file.close()
```

It is customary to close the file when done with it. To test our code, we will try a couple of things.
On the top left in our Python file. Try running the following code.

```
print("hello world")
```

Now instead of printing `hello world` we are going to print the `all_lines`, to try and see the content of the playground.txt file.

```
print(all_lines)
```

Since our playground has no content in it yet we are going to get a blank response. We should have the Codio IDE message that our code successfully ran.

Now let's try adding the following text on our `.txt` file(bottom left), and try running our file again since now our playground will not be empty.

```
write a tagline for an ice cream shop.
```

It should print out `write a tagline for an ice cream shop.`. Now that we know we have access to the contents of our file lets assign it as the prompt instead of printing it.

```
prompts = all_lines
```

We typically employ the following code to prompt users and capture the AI response:

```
response = openai.Completion.create(model="text-davinci-002",
                                    prompt=prompts,
                                    max_tokens=256,
                                    top_p=0.1)
txt_response=response['choices'][0]['text'].strip()
print(txt_response)
```

Now that we can get it to generate a response on the content of our text file we are going to have it write down the response in that same text file. For that we need to first open our file then simply add the response in to the text file instead of printing it.

```
#open our file, and append to it
f = open("playground.txt", "a")
#write the response we want to append
f.write((txt_response))
#close our file
f.close()
```

From that we realize it generates our response on the same line. Delete the text in the playground file. Have it go back to simply:

```
Write a tagline for an ice cream shop
```

Before we write our response we are going to make sure it adds to a new line so we will change our code to the following:

```
#write the response we want to append
f.write(("\n"))
f.write((txt_response))
```

Feel free to try it with different prompts. For example:

```
Give me 5 movie recommendations.
```

Let's just say its a pain to keep erasing. Let's add some code that will clear our text file for us. We will use the key word `//clear//` in our playground to know to clear it instead of generating anything. In other words, if in a our file we see the `//clear//` keyword our program will instead of generating anything will give us a blank page. Copy and paste the following in our code above the code we have already written. For the `else` case, we are going to make the code we have already written the contents for it.

```
# if the code word is present we clear the contents
if '//clear//' in open('playground.txt').read():
  open("playground.txt", "w").close()
else:
```

Opening a file in "write" mode clears it. Again we have to close it after we are done with it. Inside the else statement we can indent the rest of the code we have previously written

Using Our Box

Let's use our new playground to generate a couple more prompts. Feel free to modify at your own leisure. But practice running the following prompts

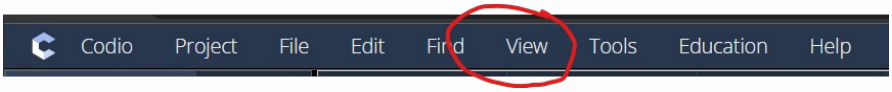
1. Generate blog posts, essays, and other forms of copy.

```
write a short essay about willy wonka
```

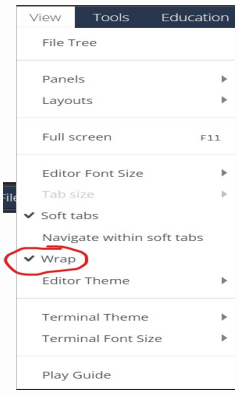
2. Utilize GPT-3 to generate product descriptions, titles and other marketing materials.

```
Write a description for Codio, the best learning website.
```

If the text generated is too long and you have to scroll to read everything. You can click on the View tab.



We can click on the wrap in our drop-down menu.



3. Use GPT-3 to automate customer support tasks such as FAQs and customer service emails.

Clear the previous text in the playground. Feel free to manually do it, or type `//clear//` in your playground. Before asking it to:

```
generate  a customer service email to apologize for our website
being down. Let customers know that the issue was fixed.
```

4. Leverage GPT-3 to generate technical documentation and software manuals.

```
generate a short technical document on how to use gpt-2.
```

5. Automate the process of summarizing long articles with GPT-3.

```
summarise the content of this webpage
(https://arxiv.org/abs/2005.14165).
```

Coding Exercise

For our coding exercise we are going to create an extra feature similar to `//clear`.

You can implement a feature where certain keywords trigger GPT-3 to generate specific types of content. For example,`//joke//` could make the program generate a random joke, `//quote//` could trigger a famous quote generation, and `//news//` could lead to a summary of the day's top news.

For our assignment modify the code for your program so that if the user types in `//joke//` a joke will be outputted. Have your code so that `//clear//` has a priority than joke. Meaning if both key words are present it will just clear the page.

Be sure to click the Try it to run your box.