Precalc AI Lab

Lab 0 - Working with an LLM to learn Math

Purpose of Lab

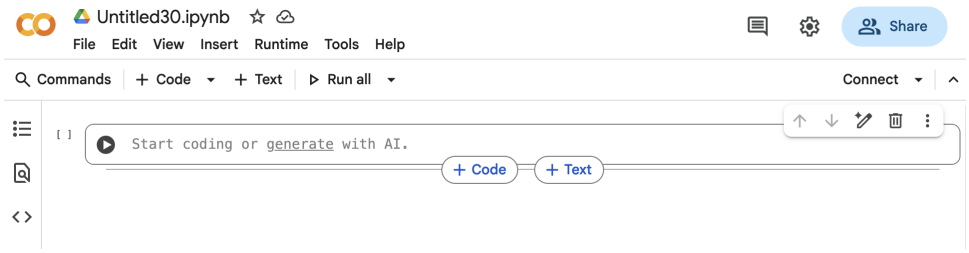
- Use an LLM to generate Python code to do something math-related.
- Run Python code on the web using Colab.
- Ask an LLM questions and explanations about math and Python code.

Context for Lab

In these labs you will use AI to learn and understand the math, not just to get answers. This is the first lab, in which you learn the basics for working through the rest.

Part 1: Running Python code on the web with Colab

1. Log in to your Google account, and navigate to `https://colab.research.google.com`.
2. Click on the button to create a *New notebook*. You will be directed to a website which allows you to execute Python code directly in the browser. It will resemble the following screenshot:



3. Notice the filename at the top-left of the website (`Untitled30.ipynb` in the screenshot). Click on the name to rename it so that you can find it easily later.
4. Paste the following Python code in the *code cell* (the rectangle with the *Play* button).

```
num1 = 10
num2 = 20
sum_result = num1 + num2
print("The sum is:")
print(sum_result)
```

5. Run the Python code in the cell by pressing the *Play* button. You should see the following output right below the cell:

```
The sum is:
30
```

6. Edit the code to change `num1 = 10` to `num1 = 45`, then re-run the code (press *Play* again) to get a new result.
7. You can create as many cells as you like. Each cell has its own *Play* button. Make a new cell and run the code below to compute the difference of two numbers

```
num1 = 10
num2 = 20
difference = num1 - num2
print("The difference is:")
print(difference)
```

Part 2: Asking an LLM to write Python code and running it

You can ask any LLM to write Python code. It will usually give you the code in an easy-to-copy box so that you can paste it into a Colab cell.

In these labs, instructions for using an LLM appear in shaded boxes. Try out the following one!

Use an LLM

1. Ask your favorite LLM to write Python code that multiplies two numbers and shows the result.
2. Copy the code into a new Colab cell and run it.
3. To make changes, edit the code yourself or ask the LLM for an updated version.

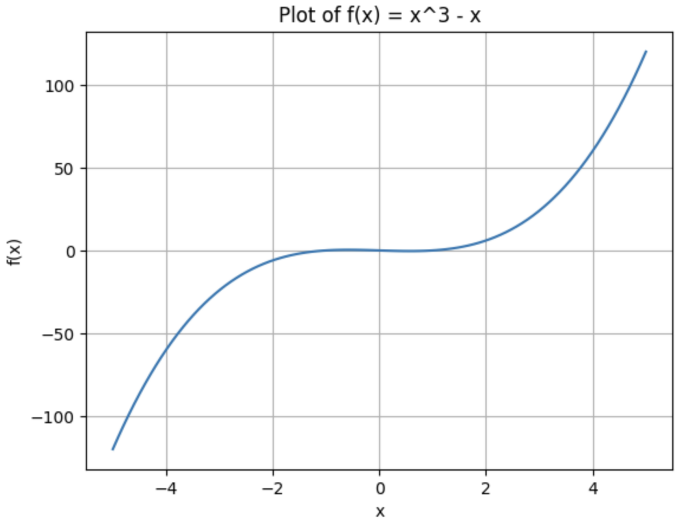
Note: Some Python code may not run in Colab. If you get errors, ask the LLM to *write code that works in Colab*. This usually fixes the problem.

Try the following tasks using an LLM.

Use an LLM

1. Ask the LLM to write Python code to:
   a) receive two numbers a and b and print a/b in decimal form with 10 decimal places.
   b) plot the function $f(x) = x^3 - x$ for x from -5 to 5. **Note:** to type x^3 for an LLM use `x^3` (the LLM, in turn, will use the correct way to type this in Python for your code, which is `x**3`). The graph that you get should look similar to the one displayed below.
2. Run each code in Colab to check that it works. If it does not, ask the LLM to fix it and make sure it is Colab-compatible.

The following figure shows the plot you should get for $f(x) = x^3 - x$. Your plot might not look exactly the same, as the LLM may select a different format or scale in the y-direction.



Important note: You do not need to understand all the Python code that the LLM generates, and becoming a Python expert is *not* the goal. Instead, the point is to learn to ask an LLM to write Python code for you and to ask it questions about the code if you want, as the next part of the lab explains.

Part 3: Asking questions

LLMs can not only write Python code, they can also explain it. If you are curious or unsure about how something works, just ask for an explanation or for the code to be well-documented with comments. If you don't follow the explanation, just say so and ask for a simpler one. You can decide how much detail you want, or even ask the LLM to rewrite the entire code in a way that is easier to understand for someone learning Python.

The same applies to the math involved. Ask questions as you go to make sure you are following along—this is the key to learning! You can even ask for entirely different explanations, and specify that the explanations should be understandable by a Precalculus student.

Try this with the following questions.

Use an LLM

1. Ask the LLM to write Python code that uses a loop to add at least five numbers of your choice. The code should print out the intermediate results (partial sums) and the final total.
   a) Run the code in Colab and make sure it works.
   b) Look at the code that the LLM made. There may be parts of the code you don't understand. Ask the LLM to explain any parts of the code you are curious about. Keep asking until what you are curious about is clear.
2. Ask the LLM to write Python code to plot the line through the points $(1, 10)$ and $(5, 1)$ and display its equation.
   a) Run the code in Colab and check that the line looks correct.
   b) Ask the LLM to explain the math it used and show where it appears in the code. Tell it you are learning Precalculus, so it can adjust the explanation. Keep asking until you feel you follow the math completely.
   c) Ask how to change the line's style or color and what style or color options are available. Then try changing it yourself, or ask for updated code. **Note:** If your edits cause errors, paste your code into the LLM and ask what you did wrong!
   d) Ask the LLM to explain any parts of the code you are curious about. Keep asking until you feel you follow everything you want to know.

Part 4: Pushing back

One last thing: Do not assume the LLM does not work if you are not getting what you want or expect. Remember you can always push back and do so until you get what you expect. It does not always work, but many times it does.

Try this out.

Use an LLM

1. Ask the LLM to give you an interesting problem about lines to practice.
2. If you find the problem boring or uninteresting, say so! Be specific about what you would find more engaging and see how it responds.
3. If the new problem feels too artificial, say so! Continue until you get a problem about lines that really interests you.