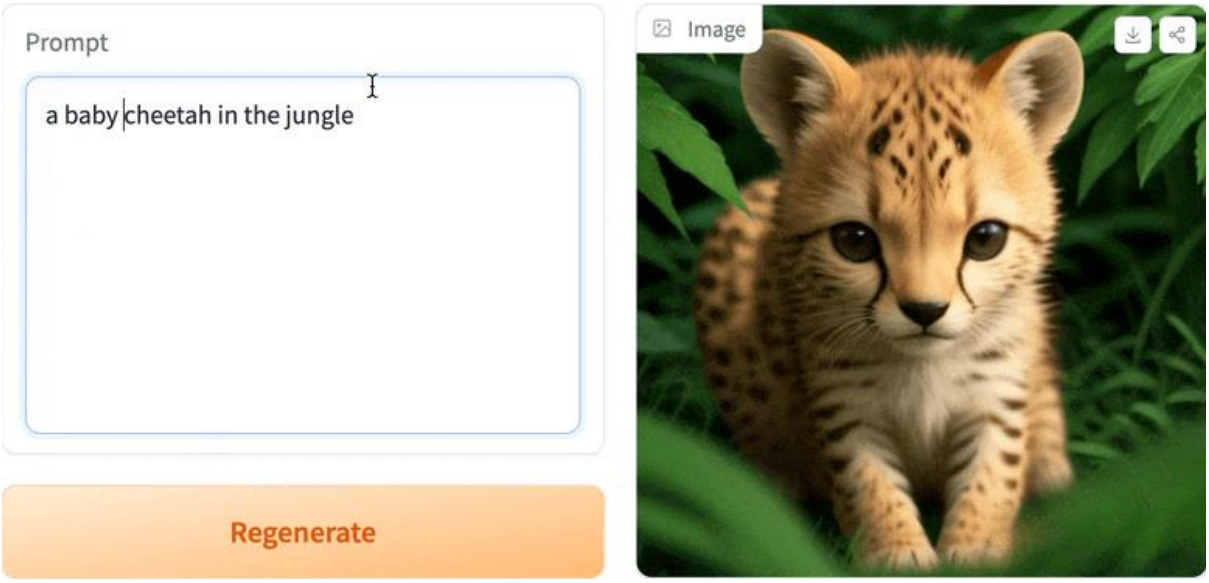
Gradio

Gradio is an open-source python package that allows you to quickly build a demo or web application for your machine learning model, API, or any arbitrary python function. You can then share a link to your demo or web application in just a few seconds using Gradio's built-in sharing features. No JavaScript, CSS, or web hosting experience is needed!



It just takes a few lines of Python to create a beautiful demo like the one above.

Building Your First Demo

You can run Gradio in your code editor, jupyter notebook, google colab, or anywhere else you write Python. Let's write your first Gradio app:

```
import gradio as gr

def greet(name, intensity):
    return "Hello " * intensity + name + "!"

demo = gr.Interface(
    fn=greet,
    inputs=["text", "slider"],
    outputs=["text"],
)

demo.launch()
```

Understanding the Interface Class

You'll notice that to make your first demo, you created an instance of the gr. Interface class. The Interface class is designed to create demos for machine learning models that accept one or more inputs and return one or more outputs.

The Interface class has three core arguments:

- **fn:** the function to wrap a user interface (UI) around.
- **inputs:** the Gradio component(s) to use for the input. The number of components should match the number of arguments in your function.
- **outputs:** the Gradio component(s) to use for the output. The number of components should match the number of return values from your function.

The fn argument is very flexible — you can pass any Python function that you want to wrap with a UI. In the example above, we saw a relatively simple function, but the function could be anything from a music generator to a tax calculator to the prediction function of a pretrained machine learning model.

The input and output arguments take one or more Gradio components. As we'll see, Gradio includes more than 30 built-in components (such as the **gr.Textbox()**, **gr.Image()**, and **gr.HTML()** components) that are designed for machine learning applications.

If your function accepts more than one argument, as is the case above, pass a list of input components to inputs, with each input component corresponding to one of the arguments of the function, in order. The same holds if your function returns more than one value: simply pass in a list of components to outputs. This flexibility makes the Interface class a very powerful way to create demos.