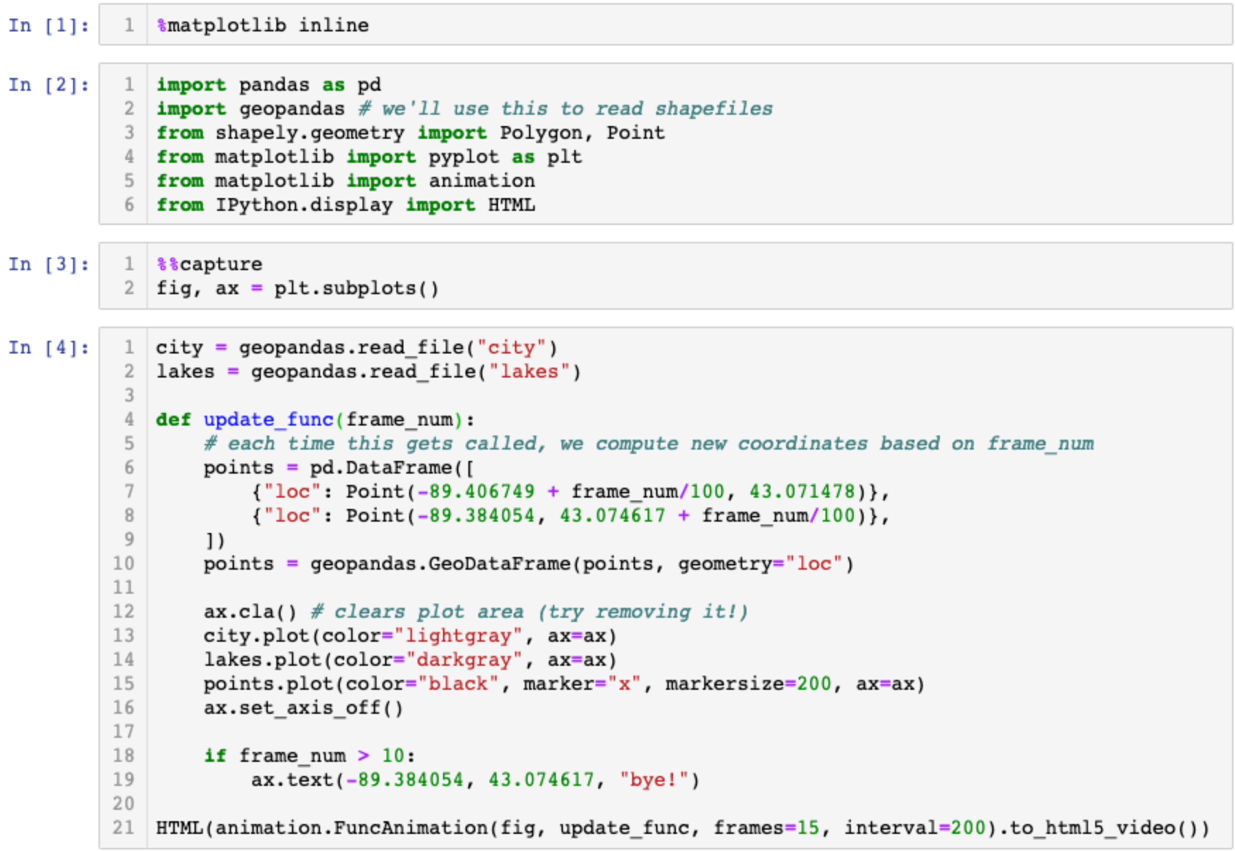
Animation

We'll sometimes want animations to visualize data (e.g., imagine an animated map of Madison showing traffic incidents over time).

Many Python libraries are based on ffmpeg; install it here: https://ffmpeg.org/. Make sure the binary program is in the PATH (i.e., you can execute `ffmpeg` in the terminal). ffmpeg can take many images, use them as frames, and stitch them together into a video clip.



In [1]:
```
%matplotlib inline
```

In [2]:
```
import pandas as pd
import geopandas # we'll use this to read shapefiles
from shapely.geometry import Polygon, Point
from matplotlib import pyplot as plt
from matplotlib import animation
from IPython.display import HTML
```

In [3]:
```
%%capture
fig, ax = plt.subplots()
```

In [4]:
```
city = geopandas.read_file("city")
lakes = geopandas.read_file("lakes")

def update_func(frame_num):
    # each time this gets called, we compute new coordinates based on frame_num
    points = pd.DataFrame([
        {"loc": Point(-89.406749 + frame_num/100, 43.071478)},
        {"loc": Point(-89.384054, 43.074617 + frame_num/100)},
    ])
    points = geopandas.GeoDataFrame(points, geometry="loc")

    ax.cla() # clears plot area (try removing it!)
    city.plot(color="lightgray", ax=ax)
    lakes.plot(color="darkgray", ax=ax)
    points.plot(color="black", marker="x", markersize=200, ax=ax)
    ax.set_axis_off()

    if frame_num > 10:
        ax.text(-89.384054, 43.074617, "bye!")

HTML(animation.FuncAnimation(fig, update_func, frames=15, interval=200).to_html5_video())
```

- **%%capture** suppresses output for that cell
- **FuncAnimation** in the **animation** submodule of **matplotlib** calls the give update function (**update_func**) 15 times (**frames** parameter), once ever 200 milliseconds (**interval** parameter).
- **.to_html5_video()** converts the video to HTML code that you could copy/paste into a site
- **HTML(...)** embeds that returned HTML directly into the notebook