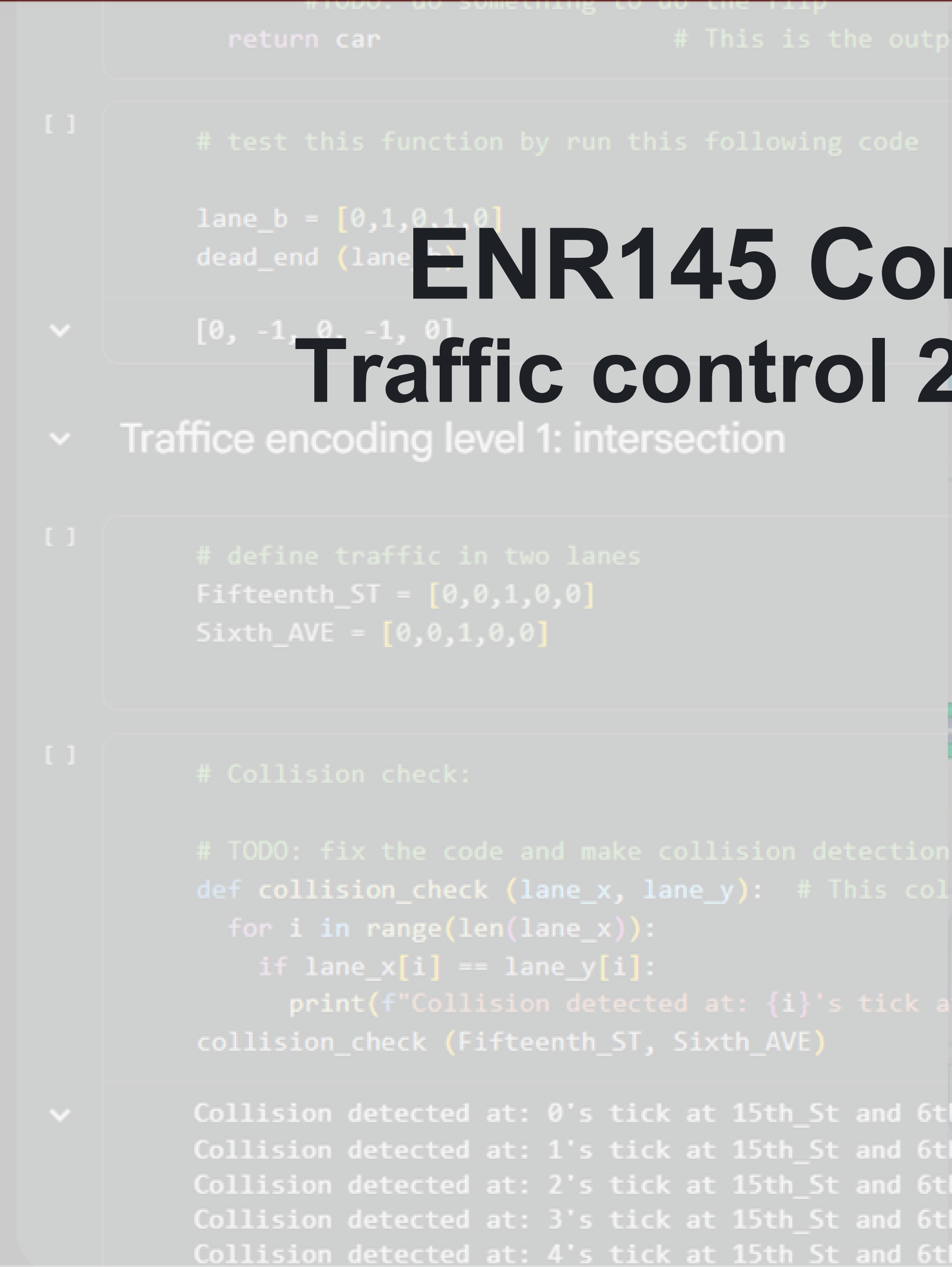
ENR145 Computational Methods: Traffic control 202: Cloverleaf interchange

Xiang Li

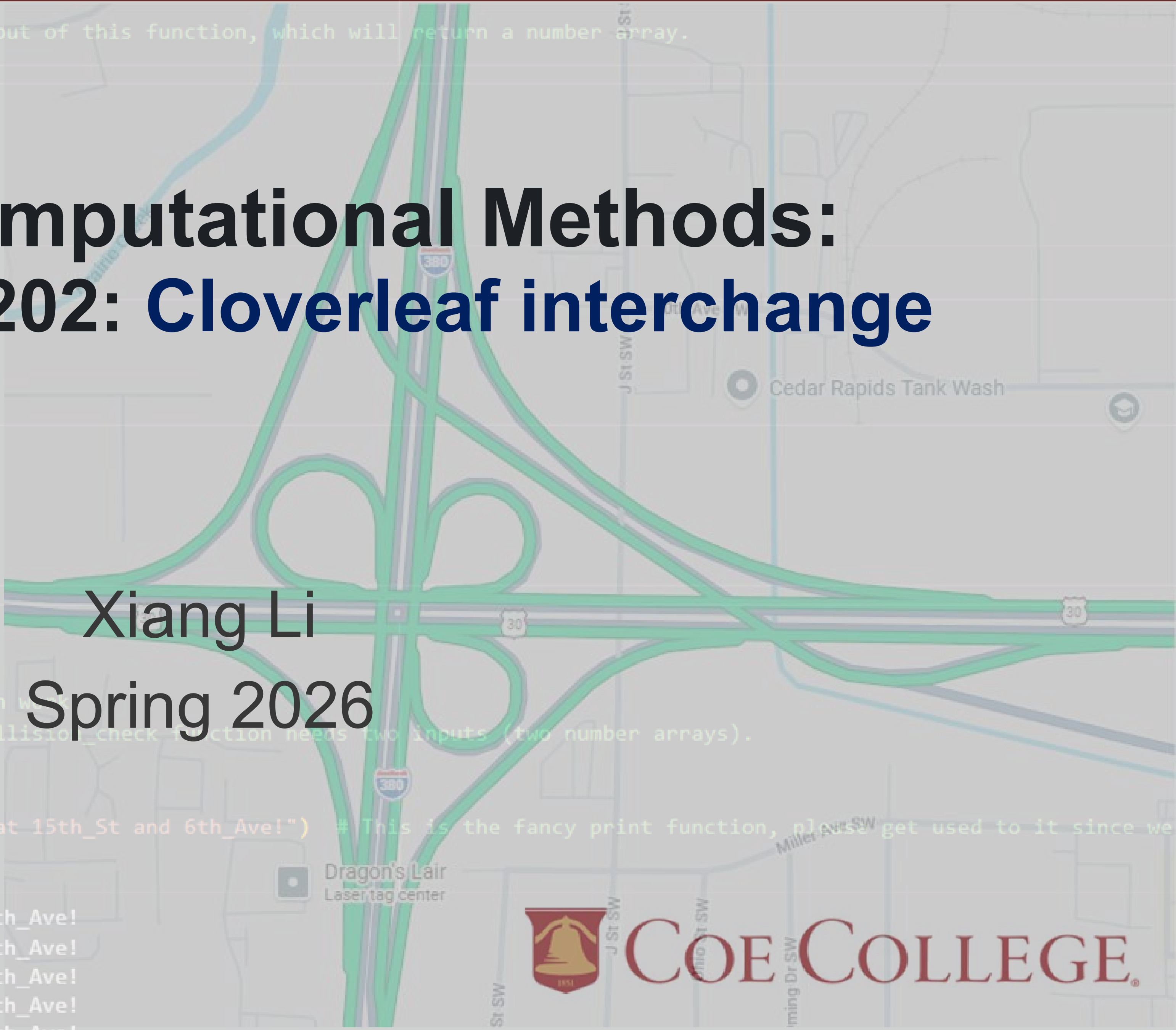
Spring 2026



```
      #TODO: do something to do the flip
    return car                  # This is the output of this function, which will return a number array.

[ ]
    # test this function by run this following code

    lane_b = [0,1,0,1,0]
    dead_end (lane

    [0, -1, 0, -1, 0]
```

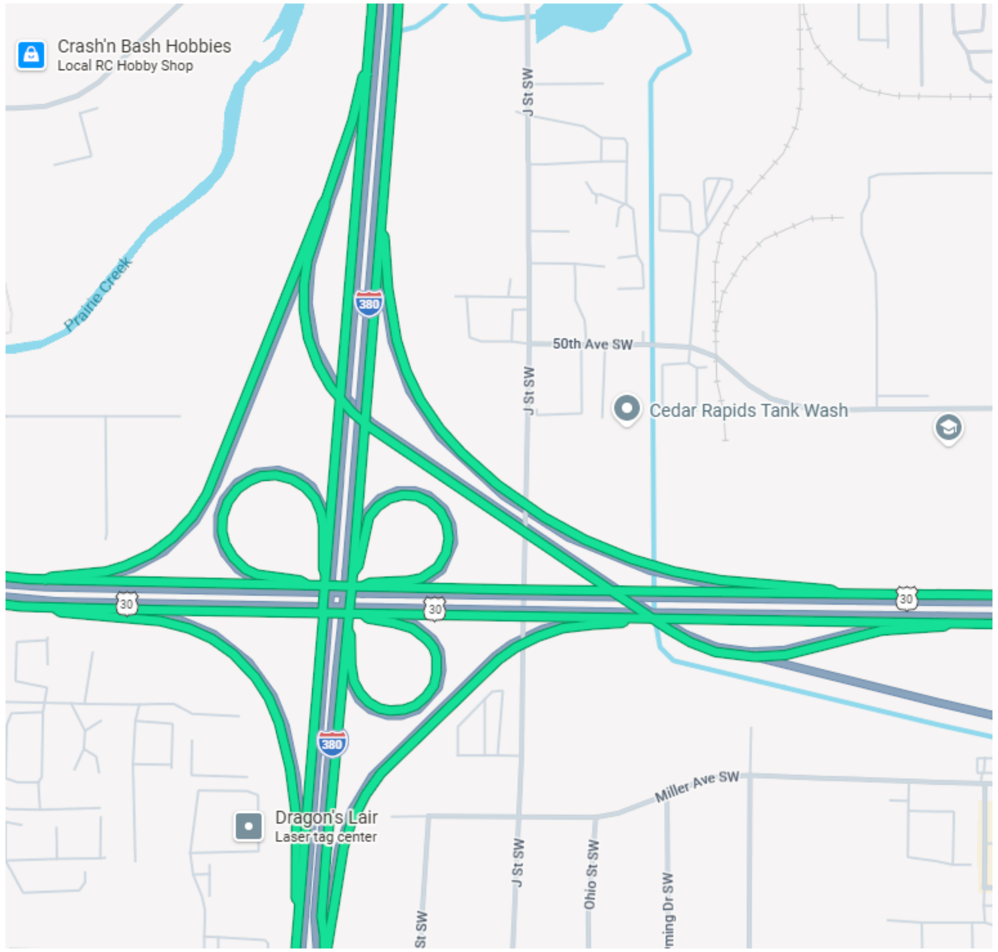
Traffice encoding level 1: intersection

```
[ ]
    # define traffic in two lanes
    Fifteenth_ST = [0,0,1,0,0]
    Sixth_AVE = [0,0,1,0,0]

[ ]
    # Collision check:

    # TODO: fix the code and make collision detection w
    def collision_check (lane_x, lane_y):  # This collision_check function needs two inputs (two number arrays).
      for i in range(len(lane_x)):
        if lane_x[i] == lane_y[i]:
          print(f"Collision detected at: {i}'s tick at 15th_St and 6th_Ave!")  # This is the fancy print function, please get used to it since we
    collision_check (Fifteenth_ST, Sixth_AVE)

    Collision detected at: 0's tick at 15th_St and 6th_Ave!
    Collision detected at: 1's tick at 15th_St and 6th_Ave!
    Collision detected at: 2's tick at 15th_St and 6th_Ave!
    Collision detected at: 3's tick at 15th_St and 6th_Ave!
    Collision detected at: 4's tick at 15th_St and 6th_Ave!
```



Encoding the traffic: level 3

Cloverleaf interchange

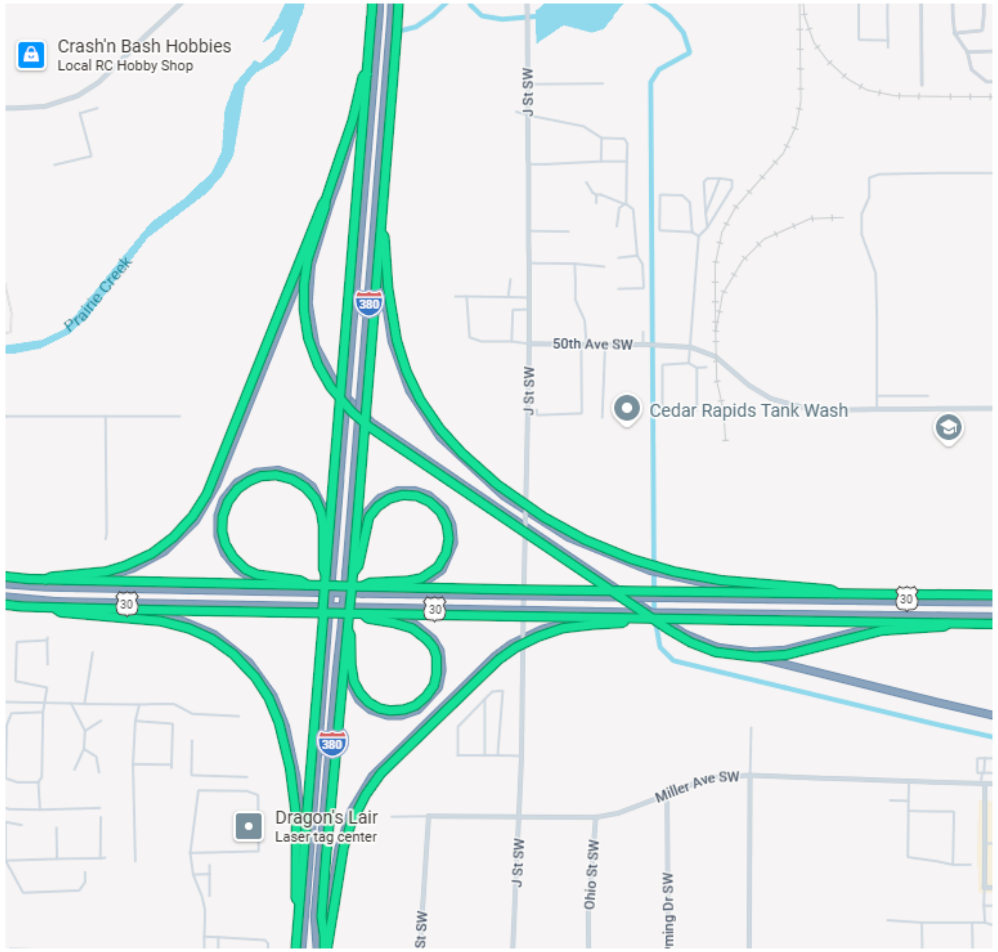


How many lanes?
380N, 380S, 30E, 30W

How many options for output?
3 for each lane, maybe?

*Lane map is found on google map in Cedar Rapids.

The technical challenge:

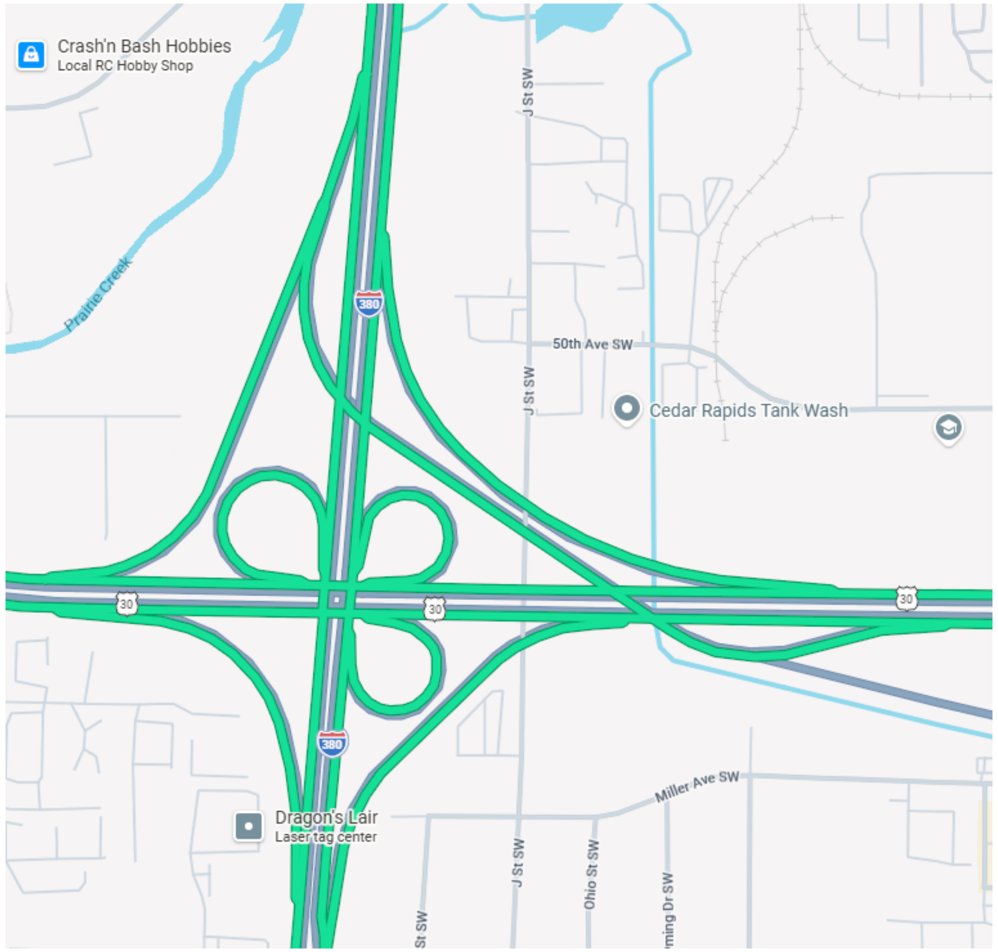


- For collision check between four lanes, how many times we have to check between two lanes?

*Lane map is found on google map in Cedar Rapids.

COE COLLEGE

The technical challenge:

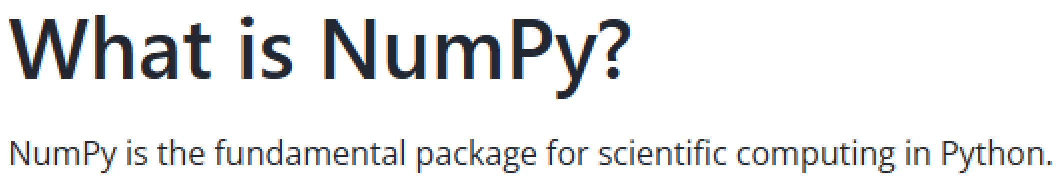


- For collision check between four lanes, how many times we have to check between two lanes?
- We cannot iterate the collision check from lane to lane IRL.
- We have to do it altogether.

*Lane map is found on google map in Cedar Rapids.

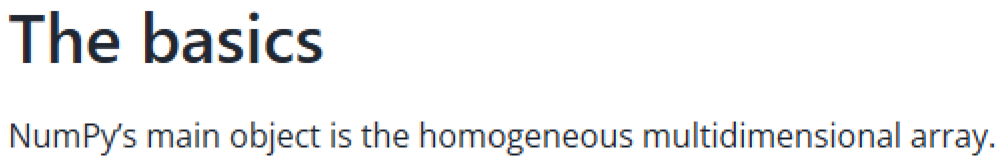
Introducing NumPy

https://numpy.org/devdocs/user/absolute_beginners.html



What is NumPy?

NumPy is the fundamental package for scientific computing in Python.

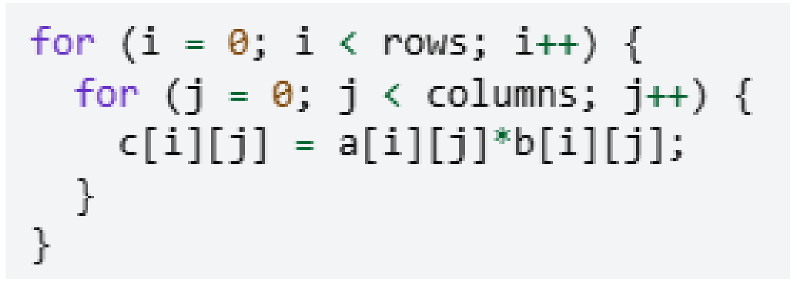


The basics

NumPy's main object is the homogeneous multidimensional array.

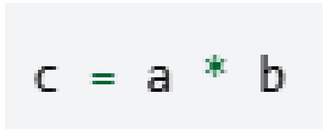
One example of why NumPy is handy:

In vanilla Python:



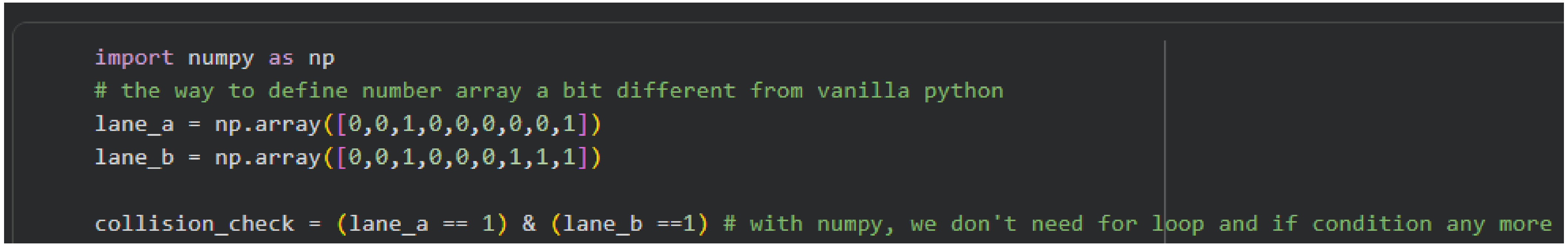
```
for (i = 0; i < rows; i++) {
  for (j = 0; j < columns; j++) {
    c[i][j] = a[i][j]*b[i][j];
  }
}
```

In NumPy:

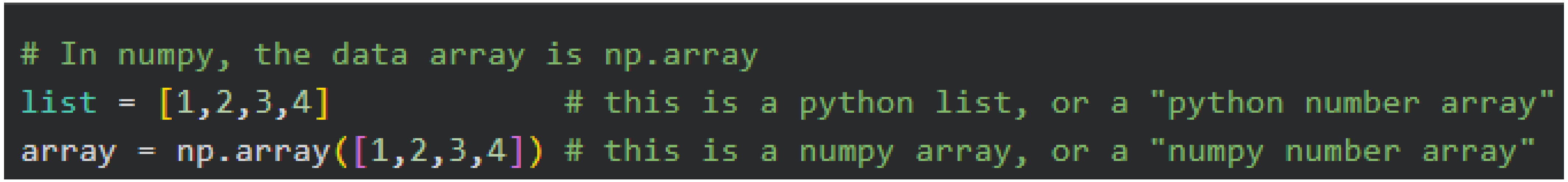


```
c = a * b
```

NumPy’s array is built different

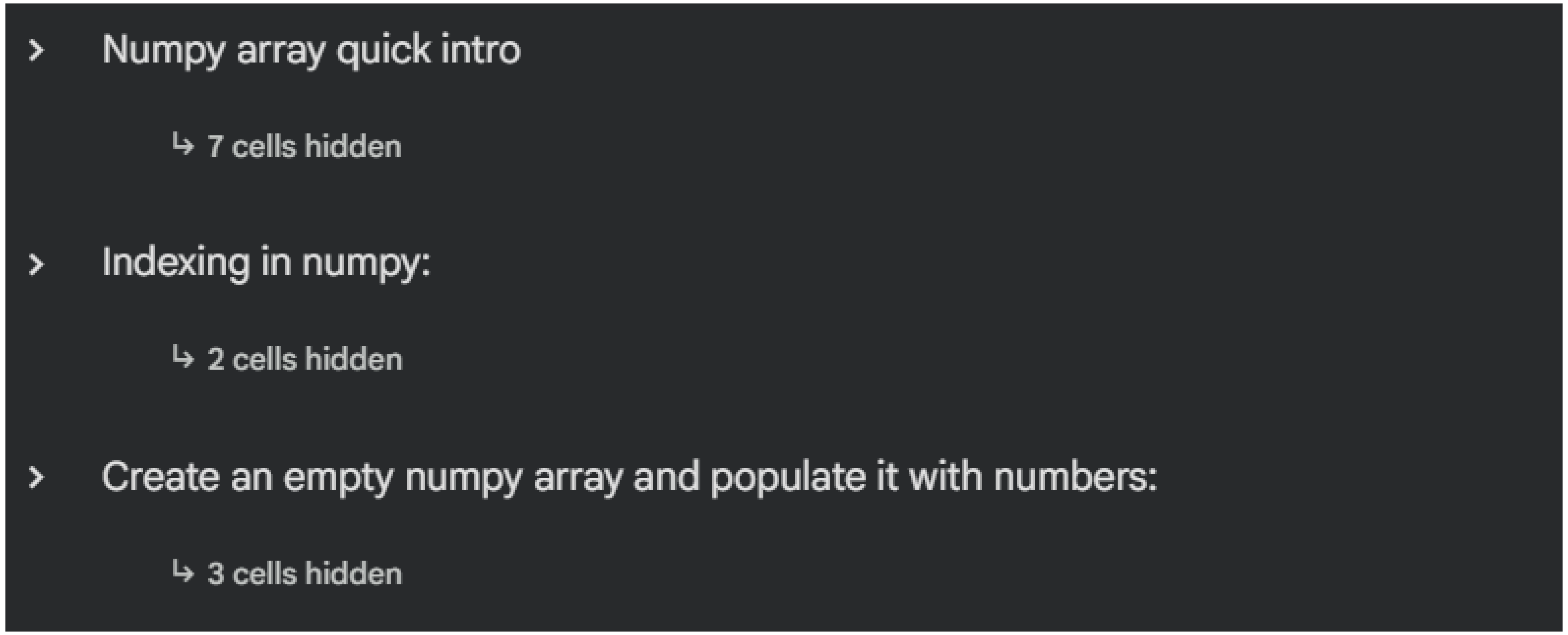


```
import numpy as np
# the way to define number array a bit different from vanilla python
lane_a = np.array([0,0,1,0,0,0,0,0,1])
lane_b = np.array([0,0,1,0,0,0,1,1,1])

collision_check = (lane_a == 1) & (lane_b ==1) # with numpy, we don't need for loop and if condition any more
```



```
# In numpy, the data array is np.array
list = [1,2,3,4]              # this is a python list, or a "python number array"
array = np.array([1,2,3,4]) # this is a numpy array, or a "numpy number array"
```

Some more NumPy basics:



> Numpy array quick intro

↳ 7 cells hidden

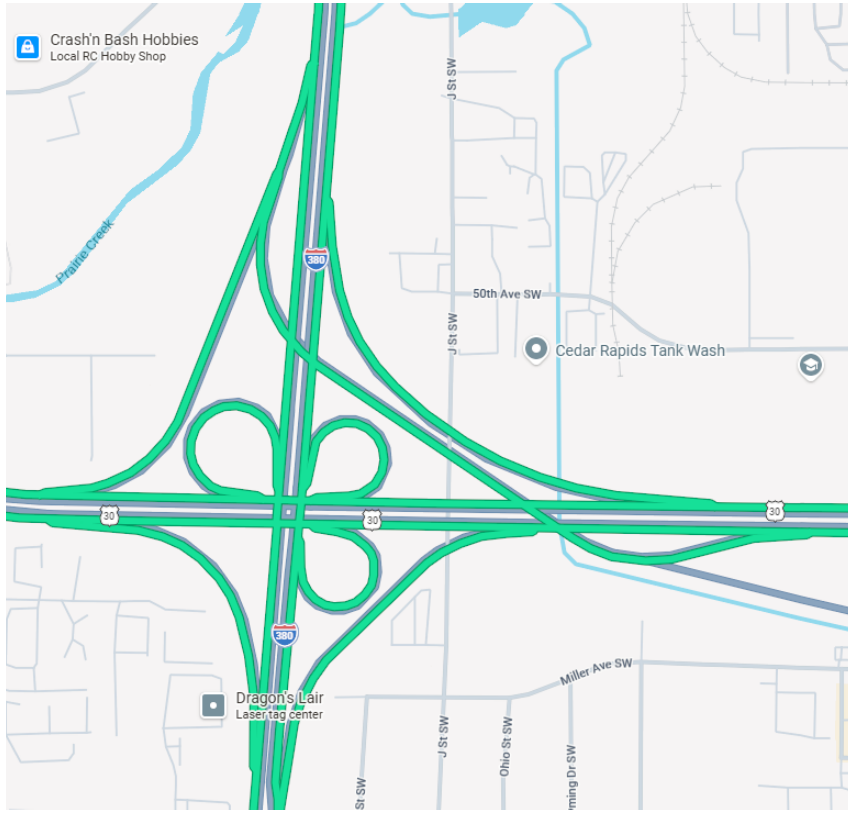
> Indexing in numpy:

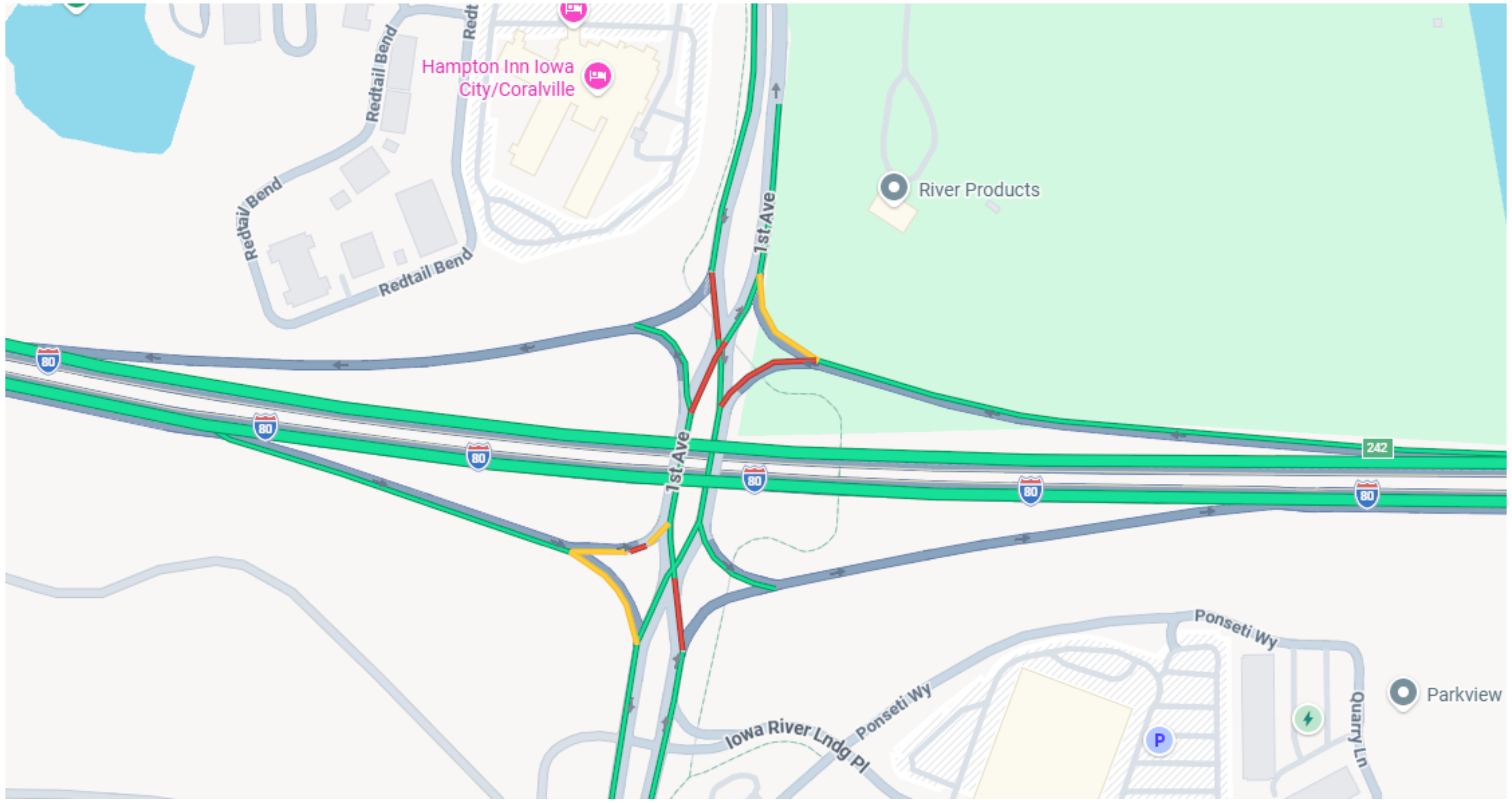
↳ 2 cells hidden

> Create an empty numpy array and populate it with numbers:

↳ 3 cells hidden

Next week: traffic control with NumPy





- **Some nice final project topics, don't you think?**