Lecture 7 – Arrays

* Prayer

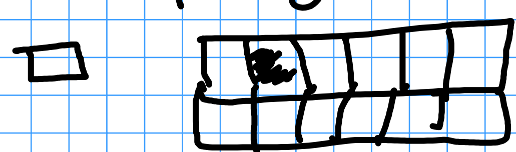
I. Array Basics

* In math we encounter vectors and matrices

$$\underset{\sim}{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$$

$$\underline{\underline{\sigma}} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix}$$

* Num. computing $\Rightarrow$ arrays



* Excel: A1–A5

* Python has 3 types
    * List
    * Tuple
    * Numpy array

* List

my_list = [4, 7, 5]

storing lists

[4, 'a', 3.2]

* Tuple

my_tuple=(4, 7, 5)

functions

return 3, 4

* Numpy Array ← Math

```
import numpy as np
my_array = np.array([4, 7, 5])
```

([4, 7, 5])

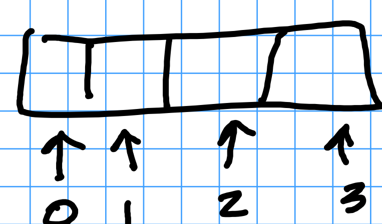
* np.arrays have useful math functions
* Lists don't
* Tuples are "immutable", can't be changed.

*Elements

my_list [2] → 5

[4, 7, 5]

index 0 1 2



my_list [0] = 8   [8, 7, 5]

print(my_list[1]) → 7

print(my_array [2]) → 5

" _tuple [2] → 5

my_tuple [2] = 10   ERROR

print(my_array [0:3]) → 4, 7, 5

slicing

0, 1, 2

0 → N-1

* Functions vs. Arrays

• Call a function : my_func(5)

* Access an array : my_array [2]

index

$\boxed{\text{Activity}}$ Ex1.

II. Loops and Array Math

- Filling arrays
- 2D arrays
- summation formulas
- matrix math / vector math

Filling arrays

np.zeros
np.empty.

```
vel = np.zeros(3)
for i in range(3):      [0,1,2]
    vel[i] = i
```

vel[0] ← 0
vel [1] ← 1
vel [2] ← 2

vel ⇒ [0,1,2]

2D Arrays

- nested loop

```
sig = np.empty((3,3))
for i in range(3):   # Rows
    for j in range(3):   # cols
        sig[i,j] = i*j
```

i	j	$i * j$
0	0	0
0	1	0
0	2	0
1	0	0
1	1	1
1	2	2
2	0	0
2	1	2
2	2	4

$$
\begin{array}{c c}
 & \begin{array}{ccc} 0 & 1 & 2 \end{array} \\
\begin{array}{c} 0 \\ 1 \\ 2 \end{array} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 2 & 4 \end{bmatrix}
\end{array}
$$

- summation formulas

$$\sum_i v_i$$

```
v = np.array([ ... ])
sum_v = 0
for i in range(len(v)):
    sum_v += v[i]
```

$\vec{v}$ $\underline{\underline{\sigma}}$

- vector | matrix

$$\underline{v} \cdot \underline{w} = v_x w_x + v_y w_y + v_z w_z$$
$$= \sum_i v_i w_i$$

```
dot_product = 0
for i in range(len(v)):
    dot_product += v[i]*w[i]
```

* matrix-vector mult:

$$\underline{\underline{A}} \cdot \underline{x} = \sum_j A_{ij} x_j$$

* matrix-matrix

$$\underline{\underline{A}} \cdot \underline{\underline{B}} = \sum_j A_{ij} B_{jk}$$

Activity   Array + Loops
    Array Math.