[1]:
```
import numpy as np
```

```
# Solving linear system of equations using Python
```

Let's first introduce a simple example, to illustrate how we can use np.linalg.solve to solve system of linear equations.

Given the equations equilibrium

$$\Sigma F_x = 0 = F_1 + 2F_2 - 1$$
$$\Sigma F_y = 0 = 3F_1 + 5F_2 - 2$$

we can rewrite it in the form of $Ax = b$ by defining a coefficient matrix, A, and resultant matrix b, then solve for the unknowns $x = [F_1, F_2]$ by:

[2]:
```
A = np.array([[1, 2], [3, 5]])
b = np.array([1, 2])
x = np.linalg.solve(A, b)
x
```

[2]:
```
array([-1.,  1.])
```

You can use a similar approach to solve the balloon example!

Compute the cable tension

Copy below the variables provided in the PrairieLearn question.

[3]:
```
import numpy as np

F = 45 # Lift force in N
m = 2 # Mass of balloon in kg
h = 7 # Height of balloon in m
```

Define weight of the balloon as the variable `W`. You can use gravity as $9.81 m/s^2$

[4]:
```
#grade (enter your code in this cell - DO NOT DELETE THIS LINE)
W=m*9.81
```

We now want to define the unit vectors that provide the directions for force vectors F_{AB}, F_{AC}, and F_{AD}.

Before we get to that, you first need to define the coordinates of the points A, B, C, D. We store these points as `a`, `b`, `c` and `d` respectively. We already give you the coordinates of `a` and `b`.

[5]:
```
#grade (enter your code in this cell - DO NOT DELETE THIS LINE)
a = np.array([0,0,h])
b = np.array([-1.5,-2,0])
c = np.array([2,-3,0])
d = np.array([0,2.5,0])
```

Determine the position vectors `ab` (from A to B), `ac` (from A to C), `ad` (from A to D).

[6]:
```
#grade (enter your code in this cell - DO NOT DELETE THIS LINE)
ab = b - a
ac = c - a
ad = d - a
```

Define the unit vectors. You can compute the norm of a vector using np.linalg.norm. Recal that the unit vector in the direction from A to C is defined as:

$$u_{AC} = r_{AC}/|r_{AC}|$$

Determine the position vectors u_{AB}, u_{AC} and u_{AD} and store them respectively as `u_ab`, `u_ac` and `u_ad`.

[7]:
```
#grade (enter your code in this cell - DO NOT DELETE THIS LINE)
u_ab = ab/np.linalg.norm(ab)
u_ac = ac/np.linalg.norm(ac)
u_ad = ad/np.linalg.norm(ad)
```

You can now write the equations of equilibrium in the form of $\mathbf{Ax} = \mathbf{y}$, where $\mathbf{x}$ is the array containing the unknown force magnitudes, $\mathbf{x} = [F_{AB}, F_{AC}, F_{AD}]$. We will use the following steps to solve for the unknown forces:

1) Define the coefficient matrix $\mathbf{A}$, and store it as a 2d numpy array `A`

2) Define the resultant vector $\mathbf{y}$, and store it as as a 1d numpy array `y`

3) Compute the unknown vector $\mathbf{x}$ by solving the linear system of equations. Store your result as the 1d numpy array `x`.

[8]:
```
#grade (enter your code in this cell - DO NOT DELETE THIS LINE)
A = np.array([[u_ab[0],u_ac[0],u_ad[0]],[u_ab[1],u_ac[1],u_ad[1]],[u_ab[2],u_ac[2],u_ad[2]]])
y = np.array([0,0,W-F])
x = np.linalg.solve(A, y)
```

You can now define the tension in cable AD, i.e. F_{AD}. Store this as the variable `Fad`.

[9]:

```
#grade (enter your code in this cell - DO NOT DELETE THIS LINE)
Fad = x[2]
```

Compute the tension in cable AB

To complete the example below, you will use the steps below:

1. Identify the known and unknown parameters.
2. Draw the proper free-body-diagrams that best relate known/unknown parameters to each other.
3. Write the corresponding equations of equilibrium.
4. Solve the system of equations for the desired unknowns.

Copy below the variables provided in the PrairieLearn question.

[1]:
```
import numpy as np

m = 2 # Mass of cable EB in kg
```

Define weight of the connector as the variable `W`. You can use gravity as $9.81 m/s^2$.

[2]:
```
#grade (enter your code in this cell - DO NOT DELETE THIS LINE)
W=m*9.81
```

In the cell below, write a code snippet to compute the tension in cable AB.

In summary, you will need to perform the following computations:

1) Define all the unit vectors that provide the directions for force vectors that act on connectors B and E. We give you the example for the unit vector u_{BA} and save it as `BA`.

2) Rewrite your equations of equilibrium in the form of $\mathbf{A}\mathbf{x} = \mathbf{y}$, where $\mathbf{x}$ is the array containing the unknown tension in each cable. Define the coefficient matrix $\mathbf{A}$ and the resultant vector $\mathbf{y}$.

3) Solve the system of equations for the unknown $\mathbf{x}$.

4) Determin the tension in cable AB. Save this result as the variable `T_AB`.

Include all your code in the same cell. We will only check the final value for `T_AB`, so you can choose the variable names for all intermediate steps.

[3]:
```
#grade (enter your code in this cell - DO NOT DELETE THIS LINE)
a = np.array([3.4,1,0])
b = np.array([2,1,0])
```

```python
BA = (a-b)/np.linalg.norm(a-b)
#grade (enter your code in this cell - DO NOT DELETE THIS LINE)
a = np.array([3.4,1,0])
b = np.array([2,1,0])
BA = (a-b)/np.linalg.norm(a-b)
C = np.array([2.2,0,1])
D = np.array([2.2,0,-1])
E = np.array([1,1.2,0])
F = np.array([0,1.4,1.2])
G = np.array([0,1.5,-1.1])
BC = (C-b)/np.linalg.norm(C-b)
BD = (D-b)/np.linalg.norm(D-b)
BE = (E-b)/np.linalg.norm(E-b)
EB = -BE
EF = (F-E)/np.linalg.norm(F-E)
EG = (G-E)/np.linalg.norm(G-E)
A = np.array([[BA[0], BC[0], BD[0], BE[0], 0, 0], \
              [BA[1], BC[1], BD[1], BE[1], 0, 0], \
              [BA[2], BC[2], BD[2], BE[2], 0, 0], \
              [0, 0, 0, EB[0], EF[0], EG[0]], \
              [0, 0, 0, EB[1], EF[1], EG[1]], \
              [0, 0, 0, EB[2], EF[2], EG[2]]])
y = np.array([0, W, 0, 0, W, 0])
x = np.linalg.solve(A,y)
T_AB = x[0]
T_AB
```

[3]: 364.9319999999997