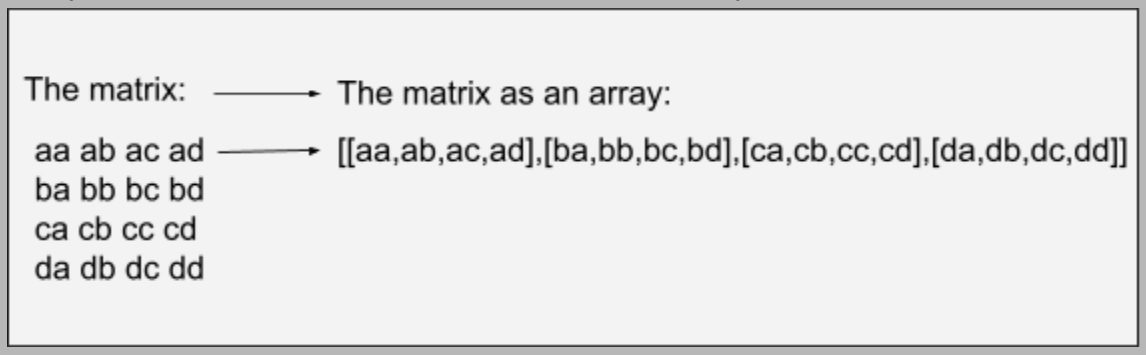
by Falk Geurten

Explanation - ex03_cfd0

Exercise 1

a.
First you need to understand how a matrix looks like in python:

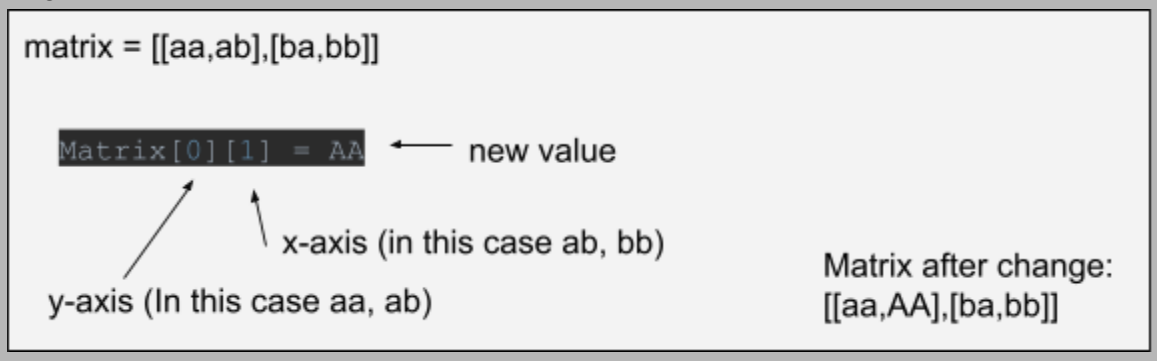


To make it easier you can still picture them like you did before.
We will store it in an numpy array, which is similar to a list and print it:

```
A = np.array([[11, 12, 13], [21, 22, 23]])
print(A)
```

---

b.
To get specific parts at a position use this:



In this exercise it would be this:

```
A[1][2] = 99
print(A)
```

---

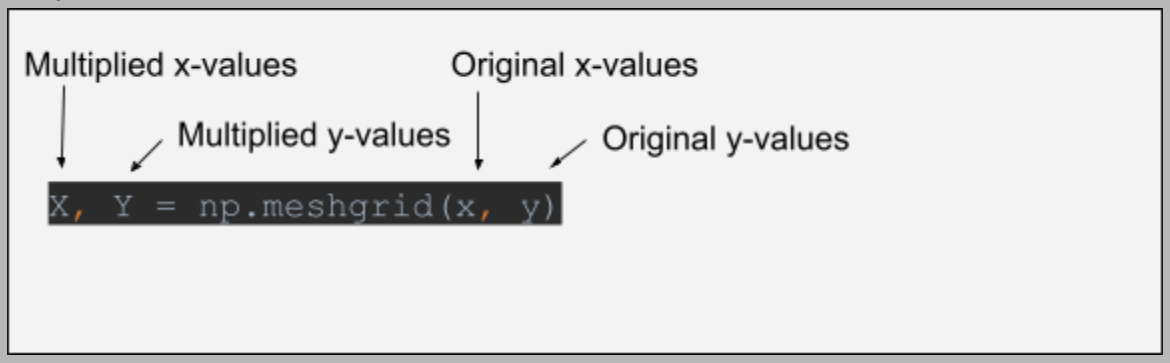
c.
When trying to print a non-existing element it will produce an error, for example:

```
print(A[100][100]          output:    IndexError: index out of bounds
```

by Falk Geurten

Exercise 2

a.
To make a grid out of two arrays we will use the numpy meshgrid function. This will multiply the arrays:



In this exercise we will make our original x and y values with the previously explained linspace function:

```
x = np.linspace(0, 4, 50)
y = np.linspace(-1, 1, 30)
```

And then use the meshgrid function:

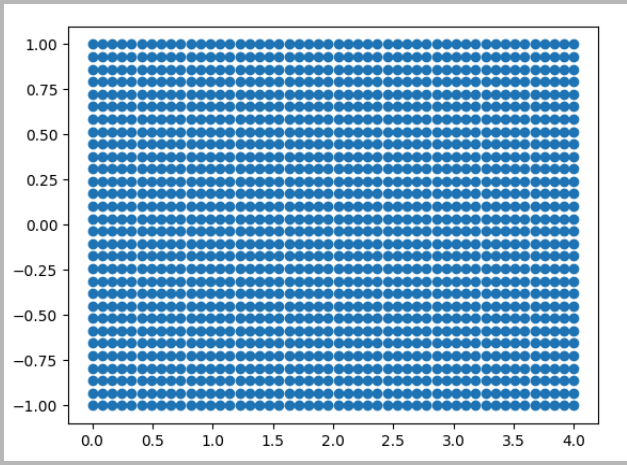
```
X, Y = np.meshgrid(x, y)
```

---

b.
When plotting only dots in matplotlib we will use scatter():

```
plt.scatter(X, Y)
plt.show()
```

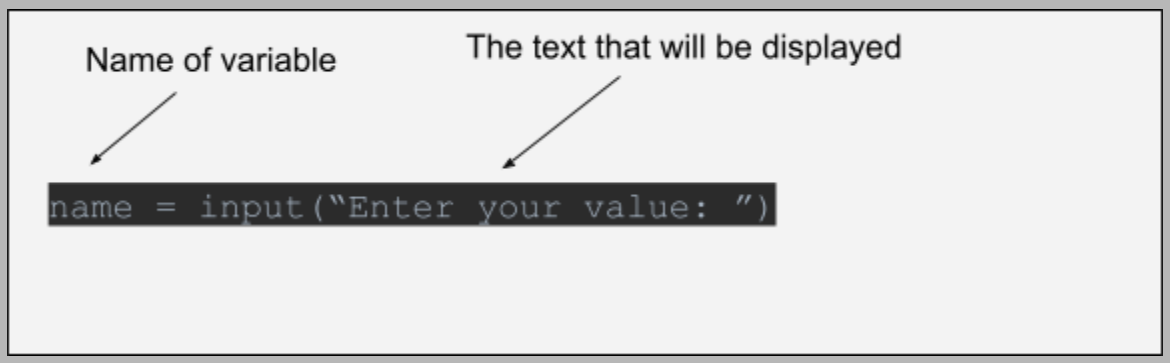
Your graph should look similar to this:



c.
The next step is to change these lines of code from pre chosen numbers to inputs that can be chosen during the run of the programm each time.

```
x = np.linspace(0, 4, 50)
y = np.linspace(-1, 1, 30)
```

The red marked numbers are the ones we will change with “imput()”:



To be sure only integers are used and no errors occur, we will put a int() around the input(), wich turns the input into an integer:

```
x_lower_limit = int(input("Please enter the lower limit of x: "))
x_higher_limit = int(input("Please enter the higher limit of x: "))
x_number_of_cells = int(input("Please enter the number of cells on the x axis: "))
y_lower_limit = int(input("Please enter the lower limit of y: "))
y_higher_limit = int(input("Please enter the higher limit of y: "))
y_number_of_cells = int(input("Please enter the number of cells on the y axis: "))
```

The last thing is the as always, the plotting:

```
x = np.linspace(x_lower_limit, x_higher_limit, x_number_of_cells)
y = np.linspace(y_lower_limit, y_higher_limit, y_number_of_cells)
X, Y = np.meshgrid(x, y)
```

Exercise 3

a.
Now we want to apply a stretch for the y - coordinate on our grid. To do this we will use the provided formula. First we declare the variables, with example values:

```
a = 0.9
b = ½*np.log((1+a)/(1-a))
E = x/100
```

(fyi: we will use E instead of ξ, because we won't use special letters in this script)
The next step is defining the stretched y, for that we will just use the provided formula again:

```
stretched_y = np.tanh(b*(E-1/2))/np.tanh(b/2)
```

The last step is calculating the grid with meshgrid again and plotting it

```
X, Y = np.meshgrid(x, stretched_y)
plt.plot(X, Y, "b.")
plt.show()
```

An example grid, with a stretched y could look like this: