JFSS Data Science Club

Libraries & NumPy

Meeting #4




Table of contents

Libraries

What is that?

- A collection of functions

(yea that’s that simple)

Libraries

Why we use it?

- Import functions created by other people.
- Make codes tidier.

Libraries

How to do it?

```
import libraryname
```

import --> a built-in statement in Python

libraryname --> where you put the name of the library

Libraries

Example:

So there’s a built-in library called “random” in Python...

Libraries

Example:

```
import random
```

Libraries

Example:

Let’s try some function in the random library!

Libraries

Example:

Time to check for some functions we can use in the library...

Libraries

Example:

Call the function name

Call the library name

```
import random

print(random.randint(1, 5))
```

Give you a random number between 1 - 5 (inclusive)

Libraries

Question:

What happen if the library has a really long name?

Libraries

How to do it?

```
import libraryname as ln
```

import --> a built-in statement in Python

libraryname --> where you put the name of the library

as --> another built-in statement in Python

ln --> the “abbreviation” name (you can customize it)

Libraries

Example:

```
import random as rnd
```

Libraries

Example:

Call the function name

```
import random as rnd

print(rnd.randint(1, 5))
```

Call the library name

Give you a random number between 1 - 5 (inclusive)

Libraries

Question:

What happen if we only need a few functions from the library?

Libraries

How to do it?

```
from libraryname import function1, function2
```

from --> another built-in statement in Python

function1, **function2** --> function name that you want to import from the library (you can import more!)

Libraries

Example:

Call the function name

```
from random import randint, choice

print(randint(1, 5))
print(choice([1, 2, 5, 6, 9, 0]))
```

Numpy

- Python library
- Helps us manipulate lists more easily (used a lot in DS)

Quick Refresher on Lists:

```
colors = ["blue", "green", "yellow"]
```

Numpy

Making a Numpy Array

```
import numpy as np

arr_1 = np.array([1, 2, 3, 4])

print(arr_1)
```

importing the library

creating an array

printing the array

Result:

```
[1 2 3 4]
```

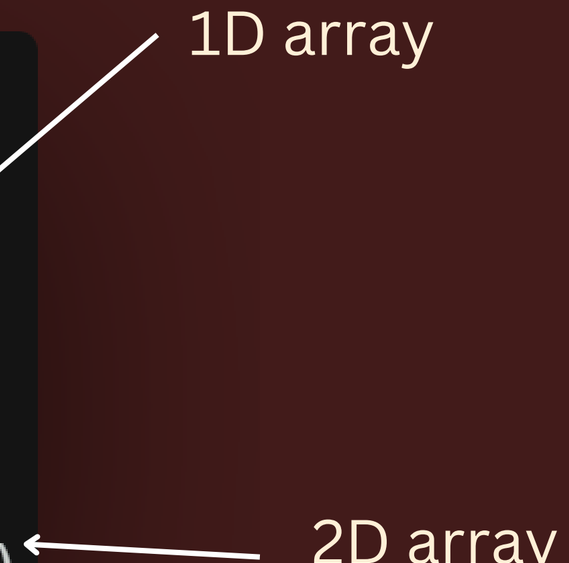
Numpy

Dimensions of Arrays

```
import numpy as np

arr_1 = np.array([1, 2, 3, 4])

arr_2 = np.array([[1,2,3,4], [5,6,7,8]])
```



Numpy

Accessing Elements

```python
import numpy as np

arr_1 = np.array([1, 2, 3, 4])
print(arr_1[1])    # prints 2

arr_2 = np.array([[1,2,3,4], [5,6,7,8]])
print(arr_2[0][3])   # prints 4
```

- Indexing starts at 0
- If 2D array: mention the outer array index then inner array index

Numpy

Returns the elements and dimensions

```
arr_2 = np.array([[1,2,3,4], [5,6,7,8]])
print(arr_2.shape)   # prints (2,4)
```

- 2 represents the number of elements in 1st layer
- 4 represents the number of elements in second layer

Numpy

Joining Arrays

```
num_1 = np.array([1,2,3,4])
num_2 = np.array([5,6,7,8])

all_nums = num_1.concatenate(num_2)
```

- Joins the lists and makes 1 GIANT LIST

Result: [1,2,3,4,5,6,7,8]

Numpy

Looking for Elements

```
nums = np.array([5,12,14,15,12,9,3])

loc = np.where(nums == 12)
print(loc)  # returns [1,4]
```

- np.where()
- Argument: array and the value searched
- Returns all indexes of value in arrays
- Returns empty list of indexes if value not found

QUESTIONS?