Method 🡪 attribute –> class

Diagram, shape, arrow

Description automatically generated

1. Array()
2. from numpy import \*
3. arr = array([1,2,3,4])
4. print(arr.dtype)
5. print(arr)

now change the value or add a decimal value and observe the difference.

* We can mention type int at the end of the array like int and float as well – observe the differences

2) linspace()

from numpy import \*

arr = linspace(0,15,16)

print(arr)

observe the syntax and notice how the output is coming – this is default float as we are breaking the values into pieces so we NEED decimal.

from numpy import \*

arr = linspace(0,20)

print(arr)

Q: what will happen if we don’t mention the third number – it will break the values in to 50 parts which is a default value.

3)arange – again go for syntax and try the below example and observe the output

from numpy import \*

arr = arange(0,20,2)

print(arr)

4) zeros

from numpy import \*

arr = zeros(5) # or arr = zeros(5, int)

print(arr)

5)ones :

from numpy import \*

arr = ones(5, int) # or arr = ones(5)

print(arr)

6)logspace()

<https://numpy.org/doc/stable/reference/generated/numpy.logspace.html>

**operations on array**

1. To add a value to an existing array

from numpy import \*

arr = array([1,2,3,4,5])

arr = arr + 5

print(arr)

Q: now can someone say how to add two arrays? Can we do it ? if yes, how?

This is also called vectorized operation

Mathematical functions

* We can find the sin, cos, log, sqrt, sum, min, max on the final print statement will give the results . let’s try unique sort concatenate **(print(concatenate([arr1][arr2])) - clarification**
* let’s copy an array from another array

from numpy import \*

arr1 = array([1,2,3,4])

arr2 = arr1

print(arr1)

print(arr2)

print(id(arr1))

print(id(arr2))

Graphical user interface, text

Description automatically generated

This is also called aliasing which will point to the same memory address.

Now we can try view function(arr2 = arr1.view()) and print the output and observe the difference , now we will have two different variables with two different addresses and same values of course

Graphical user interface, text, application, chat or text message

Description automatically generated

* Here we have a problem with copying , if we change the value of a number in first array it will impact the second array as well. arr1[1] = 7 add the value to array 1 and observe the output it will change value for second array as well which is a shallow copying.

A picture containing text, businesscard, vector graphics, screenshot

Description automatically generated

* Now we doesn’t want this to happen, which means we need to copy the array and they both should stay independent which means changing the value in arr1 should not change value in arr2, to achieve this we need to use copy() function (arr2 = arr1.copy())
* Add a value to arr1 and observe the output

Graphical user interface, text

Description automatically generated