Strategy 1: Use NumPy's ufuncs

There are many ufuncs available:

- Arithmetic Operators: `+ - * / // % **`
- Bitwise Operators: `& | ~ ^ >> <<`
- Comparison Oper's: `< > <= >= == !=`
- Trig Family: `np.sin, np.cos, np.tan ...`
- Exponential Family: `np.exp, np.log, np.log10 ...`
- Special Functions: `scipy.special.*`

. . . and many, many more.

upx
Move up in life

Chat

assignement
From Varun Gor to All Participants (11:36:54 AM):
aaya is mutable
From Rahul Singh to All Participants (11:36:54 AM):
immutable
From satish to All Participants (11:36:56 AM):
no
From Sudheer to All Participants (11:36:56 AM):
yes
From srinivas bandi to All Participants (11:36:56 AM):
no
From Shobhit to All Participants (11:36:56 AM):
yes
From Aravind to All Participants (11:36:56 AM):
nope
From VSV Narasimha Murthy to All Participants (11:36:56 AM):
no
From sb to All Participants (11:36:58 AM):
yes
From vijay to All Participants (11:36:58 AM):
can be changed
From pragnyan to All Participants (11:36:59 AM):
yes
From sanjeev to All Participants (11:37:02 AM):
mutable
From Krishna to All Participants (11:37:02 AM):
no
From Aravind to All Participants (11:37:03 AM):
but in array you can add
From upender to All Participants (11:37:03 AM):
mutable'
From Sandhya S to All Participants (11:37:04 AM):
can be changed
From srinivas bandi to All Participants (11:37:04 AM):
yes
From Aakash to All Participants (11:37:05 AM):
no
From PK to All Participants (11:37:10 AM):
yes
From Ravish to All Participants (11:37:10 AM):
immutable
From sanjeev to All Participants (11:37:12 AM):
arrays are mutable
From Rahul Singh to All Participants (11:37:17 AM):
but tuple is immutable
From Ravish to All Participants (11:37:18 AM):
okie
From Mohsin Khan to All Participants (11:37:20 AM):
what if there is 0 for division..will numpy catchhold the exceptions implicitly?
From vijay to All Participants (11:37:26 AM):
ndarray mutable
From Mallisaravanan D to All Participants (11:37:39 AM):
is there numpy tuple?
From Vinod to All Participants (11:38:07 AM):
how is it so ?

Strategy 2 : Use NumPy's Aggregations

Aggregations are functions which summarize the values in an array (min, max, sum

Lots of aggregations available . . .

```
np.min() np.max() np.sum() np.prod()
np.mean() np.std() np.var() np.any()
np.all() np.median() np.percentile()
np.argmin() np.argmax() . . .

np.nanmin() np.nanmax() np.nansum()...
```

upx
Move up in life

Chat

From Fayaz to All Participants (11:58:31 AM):
print is in the sum function it seems
From Vipin Gaur to All Participants (11:58:32 AM):
it has added 3, 3.22, 3.444 --
From Talha Siddiqui to All Participants (11:58:53 AM):
sum should be 45 ?
From pragnyan to All Participants (11:58:58 AM):
its difficult for a nonprogrammer to remember teh functions usage .if its np.func() or var.func() any suggestions please
From Aravind to All Participants (11:59:13 AM):
so mostly we will performinag mostly numpy and pandas in ML ?
From VSV Narasimha Murthy to All Participants (11:59:14 AM):
@Sai if the numpy array has all floating poiints all the operations work similary?
From Fayaz to All Participants (11:59:17 AM):
practise :-)
From Varun Gor to All Participants (11:59:24 AM):
numpy support to filter data from array , like filter values wuth grather than 5 ?
From Vinod to All Participants (11:59:38 AM):
these need some practice
From Fayaz to All Participants (11:59:54 AM):
pls share scipy documents if any.
From sanjeev to All Participants (12:00:07 PM):
standart deviation is also there
From sanjeev to All Participants (12:00:09 PM):
wow
From Fayaz to All Participants (12:00:13 PM):
as you said its not in this course scope.
From Ravish to All Participants (12:00:22 PM):
nanmin, nanmax??
From Mohsin Khan to All Participants (12:00:27 PM):
what about dtype('<U11'), you were going to explain ?
From Mohsin Khan to All Participants (12:00:30 PM):
np.all?
From Rupesh to All Participants (12:00:30 PM):
can you explain
From Talha Siddiqui to All Participants (12:00:31 PM):
np.any ?
From Rupesh to All Participants (12:00:31 PM):
c=[val+5 for val in a]
From srinivas bandi to All Participants (12:00:37 PM):
not a number
From Umesh Sharma to All Participants (12:00:37 PM):
np.prod() ..?
From Umesh Sharma to All Participants (12:00:41 PM):
np.all()?
From Spandana Musku to All Participants (12:00:44 PM):
any online book to refer np aggregations?
From Jitendra to All Participants (12:01:13 PM):
Sai, seems your laptop battery seems exhausting
From Mallisaravanan D to All Participants (12:01:30 PM):
can you leave previous slide open
From Mallisaravanan D to All Participants (12:01:46 PM):
thanks sir
From Ajay to All Participants (12:01:58 PM):
No Voice sai?

Strategy 3 : Use NumPy's Broadcasting capability

Broadcasting allows ufuncs to operate on arrays of different sizes and dimensions

upx
Move up in life

Chat

no
From Aakash to All Participants (12:14:37 PM):
no
From pragnyan to All Participants (12:14:37 PM):
no
From Sijo to All Participants (12:14:37 PM):
no
From Aakash to All Participants (12:14:46 PM):
hahaa
From Mohsin Khan to All Participants (12:14:48 PM):
+1
From Manoj to All Participants (12:14:50 PM):
Space-Time
From sanjeev to All Participants (12:14:55 PM):
space-time
From Varun Gor to All Participants (12:15:44 PM):
no
From Deekshit to All Participants (12:15:45 PM):
dancing
From Amit to All Participants (12:15:45 PM):
no
From Talha Siddiqui to All Participants (12:15:45 PM):
no
From sundar to All Participants (12:15:45 PM):
nopr
From upender to All Participants (12:15:45 PM):
no
From pragnyan to All Participants (12:15:46 PM):
yes
From VSV Narasimha Murthy to All Participants (12:15:46 PM):
no
From PK to All Participants (12:15:47 PM):
no
From pragnyan to All Participants (12:15:47 PM):
2d
From Munavvar to All Participants (12:15:48 PM):
no
From Motilal Jadhav to All Participants (12:15:50 PM):
no
From satish to All Participants (12:15:51 PM):
not sure
From Narendra to All Participants (12:15:55 PM):
lying
From srinivas bandi to All Participants (12:15:57 PM):
he is stretching
From VSV Narasimha Murthy to All Participants (12:16:01 PM):
dancing
From Mohsin Khan to All Participants (12:16:04 PM):
yes
From Sandhya S to All Participants (12:16:04 PM):
standing
From srinath to All Participants (12:16:10 PM):
yes
From Anish to All Participants (12:16:21 PM):
doesn't look too happy standing on 1 leg

Broadcasting Rules

- Broadcasting in NumPy follows a strict set of rules to determine the interaction between the two arrays:

➢ Rule 1: If the two arrays differ in their number of dimensions, the shape of the one with fewer dimensions is *padded* with ones on its leading (left) side.

➢ Rule 2: If the shape of the two arrays does not match in any dimension, the array with shape equal to 1 in that dimension is stretched to match the other shape.

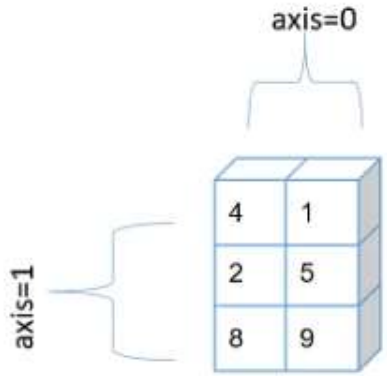
➢ Rule 3: If in any dimension the sizes disagree and neither is equal to 1, an error is raised.

Chat

3
From Sunil Pednekar to All Participants (12:24:26 PM):
3
From Shobhit to All Participants (12:24:26 PM):
3
From Motilal Jadhav to All Participants (12:24:26 PM):
3
From srinivas bandi to All Participants (12:24:26 PM):
3
From Mehul to All Participants (12:24:26 PM):
3
From Sandhya S to All Participants (12:24:26 PM):
3
From Munavvar to All Participants (12:24:26 PM):
3
From sundar to All Participants (12:24:26 PM):
3
From VIKASH to All Participants (12:24:26 PM):
3
From upender to All Participants (12:24:26 PM):
3
From Deepthi Kunapareddy to All Participants (12:24:27 PM):
3
From naresh to All Participants (12:24:27 PM):
3
From jashuva mudusu to All Participants (12:24:27 PM):
3
From Umesh Sharma to All Participants (12:24:28 PM):
43
From Rashmi to All Participants (12:24:30 PM):
we csnot remove
From Umesh Sharma to All Participants (12:24:30 PM):
3
From srinivas bandi to All Participants (12:24:31 PM):
gc
From Rupesh to All Participants (12:24:31 PM):
err
From Fayaz to All Participants (12:24:32 PM):
why do we reshare 3*3 to 3 *1
From Aravind to All Participants (12:24:40 PM):
3X3 only
From sanjeev to All Participants (12:24:40 PM):
garbage??
From Aravind to All Participants (12:24:42 PM):
the result
From sridhar to All Participants (12:24:42 PM):
we cannot reshape
From srinivas bandi to All Participants (12:24:44 PM):
garbage
From Fayaz to All Participants (12:24:47 PM):
thast not a right call.
From Rupesh to All Participants (12:24:59 PM):
lol
From Kanishk to All Participants (12:25:28 PM):
can you please explain the 3rd part again??

Axis of an array

- axis=0 : access column wise
- axis=1: access row wise




```
multi_array=np.array([(4,1),(2,5),(8,9)])
print(multi_array.sum(axis=0))
```

```
[14 15]
```

```
multi_array=np.array([(4,1),(2,5),(8,9)])
print(multi_array.sum(axis=1))
```

```
[ 5  7 17]
```

Chat

From Shreeja to All Participants (12:47:16 PM):
a*a
From pragnyan to All Participants (12:47:20 PM):
transpose is same
From Aakash to All Participants (12:47:42 PM):
could you show it here
From Mohsin Khan to All Participants (12:47:42 PM):
AT=A
From srinivas bandi to All Participants (12:47:42 PM):
yes
From satish to All Participants (12:47:43 PM):
s
From VSV Narasimha Murthy to All Participants (12:47:43 PM):
S
From Rupesh to All Participants (12:47:46 PM):
m*n=n*m
From Talha Siddiqui to All Participants (12:48:26 PM):
Numpy is awesome :D
From Rashmi to All Participants (12:48:37 PM):
how to do reverse diagonal
From Aakash to All Participants (12:48:56 PM):
So is numpy another language integrated with python or is it part of python library
From PK to All Participants (12:49:17 PM):
3rd party
From Aakash to All Participants (12:49:22 PM):
just like sring in JAVA
From Shirish Joshi to All Participants (12:49:36 PM):
is it pre comlpied?
From Aravind to All Participants (12:49:48 PM):
@sai your laptop battery is draining
From pragnyan to All Participants (12:50:11 PM):
can u plz explain broadcast_to function
From satish to All Participants (12:50:43 PM):
3 2
From Varun Gor to All Participants (12:50:44 PM):
2X3
From Amit to All Participants (12:50:46 PM):
3-22
From srinivas bandi to All Participants (12:50:46 PM):
2*3
From Bhavya to All Participants (12:50:48 PM):
3*2
From shailesh to All Participants (12:50:48 PM):
32
From Aakash to All Participants (12:50:49 PM):
3,2
From Varun Gor to All Participants (12:50:50 PM):
3X2
From Mansi to All Participants (12:50:50 PM):
3*2
From Sijo to All Participants (12:50:51 PM):
3 2
From VSV Narasimha Murthy to All Participants (12:50:54 PM):
3 x 2

Linspace

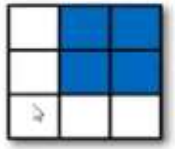
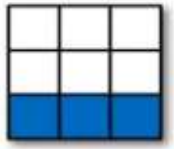
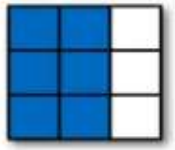
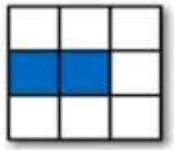
- What would be the output of array(np.linspace(2,4,9))?
- Output -> [2 2.25 2.5 2.75 3.0 3.25 3.50 3.75 4.0]
- Space between values in linspace = (Max value – Initial Value)/[(No.of elements) – 1]

Chat

voice very low

From Deekshit to All Participants (11:00:29 AM):
yes
From Kanishk to All Participants (11:00:29 AM):
yes
From Premy to All Participants (11:00:29 AM):
yes
From manaswini acharya to All Participants (11:00:30 AM):
yes
From Sri ASU to All Participants (11:00:31 AM):
Very good
From Dhulipala Somayajulu to All Participants (11:00:32 AM):
yes
From VIKASH to All Participants (11:00:32 AM):
yes
From Kshitiz to All Participants (11:00:33 AM):
yes
From VSV Narasimha Murthy to All Participants (11:00:34 AM):
yes
From Digvijay Singh to All Participants (11:00:51 AM):
no voice
From Aravind to All Participants (11:00:57 AM):
yep..
From Aravind to All Participants (11:00:57 AM):
yep
From Aravind to All Participants (11:00:57 AM):
no screen available
From Aravind to All Participants (11:00:57 AM):
no screen
From Premy to All Participants (11:01:02 AM):
numpy
From Deekshit to All Participants (11:01:39 AM):
2\n4\n9
From Aravind to All Participants (11:01:45 AM):
got it now
From Jayaram to All Participants (11:01:46 AM):
9 items
From Motilal Jadhav to All Participants (11:01:51 AM):
2/8
From satish to All Participants (11:01:56 AM):
2/8
From Aravind to All Participants (11:01:59 AM):
2 to 5 and 9 linespace
From satish to All Participants (11:02:00 AM):
starts from 2
From Digvijay Singh to All Participants (11:02:11 AM):
voice is not coming
From Naresh to All Participants (11:02:20 AM):
4-2/8
From Aravind to All Participants (11:02:24 AM):
4-2/8
From Aravind to All Participants (11:02:28 AM):
4-2/9-1
From satish to All Participants (11:02:35 AM):
4-2/8

Slice arrays

	Expression	Shape
	`arr[:2, 1:]`	`(2, 2)`
	`arr[2]`	`(3,)`
	`arr[2, :]`	`(3,)`
	`arr[2:, :]`	`(1, 3)`
	`arr[:, :2]`	`(3, 2)`
	`arr[1, :2]`	`(2,)`
	`arr[1:2, :2]`	`(1, 2)`

Chat

From aman Nagariya to All Participants (11:03:23 AM):
,
From Sandeep to All Participants (11:03:24 AM):
space and comma
From Aravind to All Participants (11:03:25 AM):
space
From UpX Academy DataScience to All Participants (11:03:43 AM):
@Digvijay: pls logout and log back in
From Bhavya to All Participants (11:03:58 AM):
Rows
From vijay to All Participants (11:04:00 AM):
row
From manaswini acharya to All Participants (11:04:00 AM):
row
From aman Nagariya to All Participants (11:04:01 AM):
row
From Sandeep to All Participants (11:04:01 AM):
rows
From Benny to All Participants (11:04:02 AM):
rows
From vishal bajetha to All Participants (11:04:02 AM):
rows
From satish to All Participants (11:04:03 AM):
rows
From Premy to All Participants (11:04:03 AM):
rows
From Aravind to All Participants (11:04:03 AM):
rows and columns
From VSV Narasimha Murthy to All Participants (11:04:09 AM):
rows
From Rahul Singh to All Participants (11:04:10 AM):
0 to 1
From Bhavya to All Participants (11:04:11 AM):
0 to 1 row
From Aravind to All Participants (11:04:11 AM):
till 2nd index rows
From satish to All Participants (11:04:13 AM):
first element
From aman Nagariya to All Participants (11:04:16 AM):
starting to 2
From lampu to All Participants (11:04:16 AM):
02
From Lokesh to All Participants (11:04:16 AM):
2 rows
From Sandeep to All Participants (11:04:17 AM):
column 2
From jashuva mudusu to All Participants (11:04:18 AM):
before 2 index row
From Premy to All Participants (11:04:20 AM):
0 and 1
From vishal bajetha to All Participants (11:04:21 AM):
0 1
From VIKASH to All Participants (11:04:23 AM):
2nd

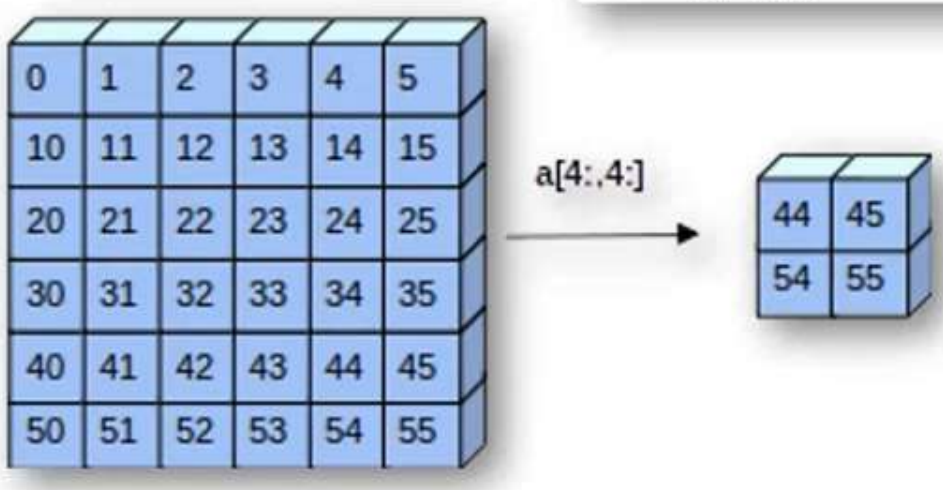
Slice Block of elements

upx Move up in life

a[4:,4:]

```
multi_array=np.array([(0,1,2,3,4,5),(10,11,12,13,14,15),
                      (20,21,22,23,24,25),
                      (30,31,32,33,34,35),
                      (40,41,42,43,44,45),
                      (50,51,52,53,54,55)])
multi_array[4:,4:]
```

```
array([[44, 45],
       [54, 55]])
```



Chat

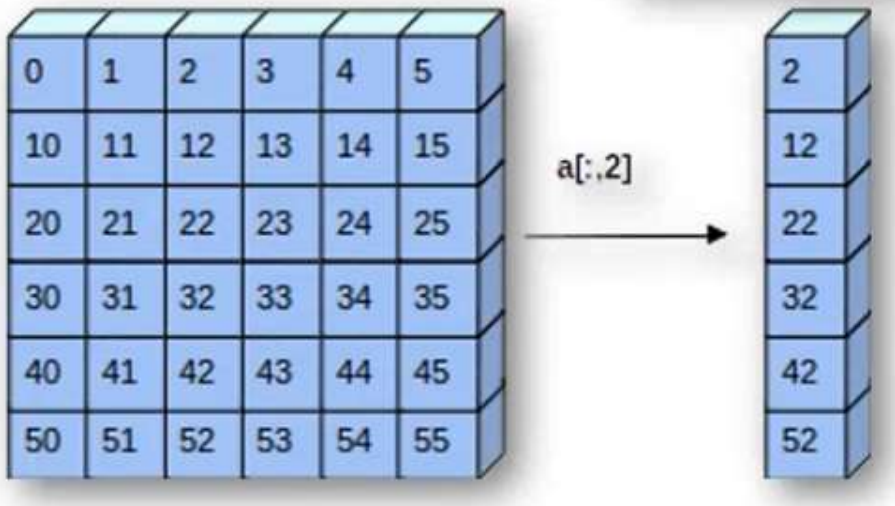
it is vector so (2,)
From Manas to All Participants (11:19:57 AM):
so it shold be 1*2
From Bhavya to All Participants (11:19:58 AM):
row was explicitely mentioned
From Shreyas to All Participants (11:20:09 AM):
1,2
From vishal bajetha to All Participants (11:20:09 AM):
1*2
From venugopal to All Participants (11:20:09 AM):
same
From ajay to All Participants (11:20:10 AM):
2,
From Sivakumar to All Participants (11:20:12 AM):
1, 2
From Sudheer to All Participants (11:20:13 AM):
(1,2)
From Motilal Jadhav to All Participants (11:20:13 AM):
1,2
From Krishna to All Participants (11:20:14 AM):
2
From satish to All Participants (11:20:15 AM):
4,1
From Premy to All Participants (11:20:15 AM):
1,2
From Rupesh to All Participants (11:20:16 AM):
2
From Amit to All Participants (11:20:16 AM):
2x2
From megha to All Participants (11:20:18 AM):
1,2
From Rashmi to All Participants (11:20:18 AM):
1,2
From Sushant to All Participants (11:20:20 AM):
2 rows
From VSV Narasimha Murthy to All Participants (11:20:21 AM):
1x2
From Krishna to All Participants (11:20:37 AM):
1, 2
From venugopal to All Participants (11:20:38 AM):
but elements are same
From Sushant to All Participants (11:20:43 AM):
1x2
From Amit to All Participants (11:21:00 AM):
yes
From Manas to All Participants (11:21:02 AM):
looks like matirx but not following matrix rules
From venugopal to All Participants (11:21:09 AM):
in what context we have to use
From venugopal to All Participants (11:21:21 AM):
ok
From ajay to All Participants (11:21:37 AM):
where the shape will be used ?
From upender to All Participants (11:21:38 AM):
if 0 an 0:1 are same then why shpe differ ?

Slice a column

a[:,2]

```
multi_array=np.array([(0,1,2,3,4,5),(10,11,12,13,14,15),
                      (20,21,22,23,24,25),
                      (30,31,32,33,34,35),
                      (40,41,42,43,44,45),
                      (50,51,52,53,54,55)])
multi_array[:,2]
```

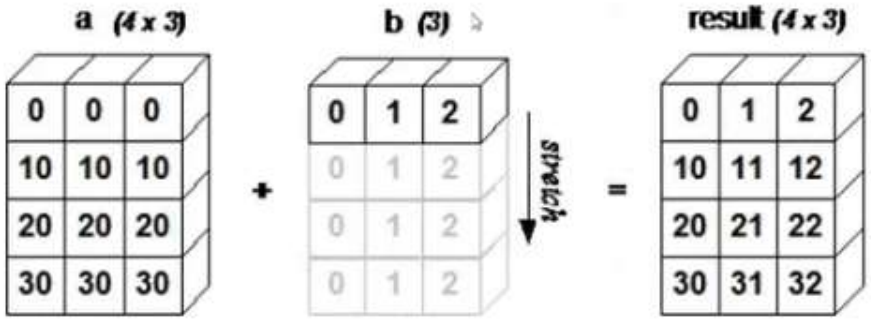
```
array([ 2, 12, 22, 32, 42, 52])
```



Chat

From Rahul Singh to All Participants (11:22:26 AM):
44,45,54,55
From Narendra to All Participants (11:22:28 AM):
2,2
From VSV Narasimha Murthy to All Participants (11:22:28 AM):
44,45
From Spandana Musku to All Participants (11:22:29 AM):
clear
From VSV Narasimha Murthy to All Participants (11:22:31 AM):
54,55
From Rahul Singh to All Participants (11:22:35 AM):
2,2
From Aakash to All Participants (11:22:35 AM):
should consider only 5th row
From Rashmi to All Participants (11:22:38 AM):
2x2
From megha to All Participants (11:22:38 AM):
2,2
From Sushant to All Participants (11:22:39 AM):
got it
From Kshitiz to All Participants (11:22:41 AM):
yes
From manaswini acharya to All Participants (11:22:49 AM):
all
From Aakash to All Participants (11:22:50 AM):
ok
From Sushant to All Participants (11:22:53 AM):
range 4 till 5
From PK to All Participants (11:22:58 AM):
this only i got
From Sudheer to All Participants (11:23:00 AM):
5
From vishal bajetha to All Participants (11:23:00 AM):
all
From Jayaram to All Participants (11:23:01 AM):
all rows
From Premy to All Participants (11:23:02 AM):
all rows
From megha to All Participants (11:23:02 AM):
all
From Narendra to All Participants (11:23:02 AM):
all
From vijay to All Participants (11:23:02 AM):
all
From Rupesh to All Participants (11:23:02 AM):
0
From Manas to All Participants (11:23:02 AM):
all
From Motilal Jadhav to All Participants (11:23:02 AM):
all
From Benny to All Participants (11:23:03 AM):
all
From Spandana Musku to All Participants (11:23:03 AM):
all

Broadcasting Arrays




```
array_a=np.array([[0,0,0],[10,10,10],
                  [20,20,20],[30,30,30]])
array_b=np.array([0,1,2])
array_broadcast=array_a+array_b
array_broadcast
```

```
array([[ 0,  1,  2],
       [10, 11, 12],
       [20, 21, 22],
       [30, 31, 32]])
```

Chat

range 4 till 5
From PK to All Participants (11:22:58 AM):
this only i got
From Sudheer to All Participants (11:23:00 AM):
5
From vishal bajetha to All Participants (11:23:00 AM):
all
From Jayaram to All Participants (11:23:01 AM):
all rows
From Premy to All Participants (11:23:02 AM):
all rows
From megha to All Participants (11:23:02 AM):
all
From Narendra to All Participants (11:23:02 AM):
all
From vijay to All Participants (11:23:02 AM):
all
From Rupesh to All Participants (11:23:02 AM):
0
From Manas to All Participants (11:23:02 AM):
all
From Motilal Jadhav to All Participants (11:23:02 AM):
all
From Benny to All Participants (11:23:03 AM):
all
From Spandana Musku to All Participants (11:23:03 AM):
all
From sundar to All Participants (11:23:03 AM):
all
From Rupesh to All Participants (11:23:04 AM):
1
From jashuva mudusu to All Participants (11:23:05 AM):
6
From Sudheer to All Participants (11:23:05 AM):
6 rows
From satish to All Participants (11:23:05 AM):
all
From VSV Narasimha Murthy to All Participants (11:23:06 AM):
all rpws
From Sivakumar to All Participants (11:23:07 AM):
all rows
From ilango kandasamy to All Participants (11:23:07 AM):
a;ll
From shanmugakannan to All Participants (11:23:08 AM):
all rows
From Manas to All Participants (11:23:11 AM):
2
From Motilal Jadhav to All Participants (11:23:12 AM):
2nd
From satish to All Participants (11:23:12 AM):
2
From Narendra to All Participants (11:23:14 AM):
3
From Pinky to All Participants (11:23:14 AM):
2nd

Other common NumPy commands

upx Move up in life

- random : returns random values of particular row size and column size

```
a=np.random.randn(3,4)
a
```

```
array([[ 0.13264159,  0.44562426,  0.92873888, -0.18219814],
       [-0.00654966,  0.7738263 , -0.4006306 ,  1.42971436],
       [-0.42105451,  0.46620049,  0.65935745, -0.73729805]])
```

```
a=np.random.rand(3,2)
a
```

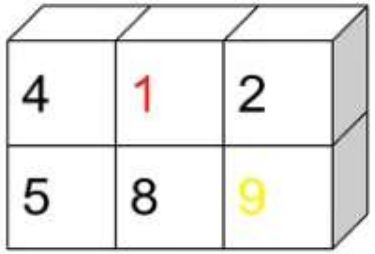
```
array([[ 0.50268006,  0.90024754],
       [ 0.43525091,  0.69126062],
       [ 0.63525222,  0.53072994]])
```

Chat

this only i got
From Sudheer to All Participants (11:23:00 AM):
5
From vishal bajetha to All Participants (11:23:00 AM):
all
From Jayaram to All Participants (11:23:01 AM):
all rows
From Premy to All Participants (11:23:02 AM):
all rows
From megha to All Participants (11:23:02 AM):
all
From Narendra to All Participants (11:23:02 AM):
all
From vijay to All Participants (11:23:02 AM):
all
From Rupesh to All Participants (11:23:02 AM):
0
From Manas to All Participants (11:23:02 AM):
all
From Motilal Jadhav to All Participants (11:23:02 AM):
all
From Benny to All Participants (11:23:03 AM):
all
From Spandana Musku to All Participants (11:23:03 AM):
all
From sundar to All Participants (11:23:03 AM):
all
From Rupesh to All Participants (11:23:04 AM):
1
From jashuva mudusu to All Participants (11:23:05 AM):
6
From Sudheer to All Participants (11:23:05 AM):
6 rows
From satish to All Participants (11:23:05 AM):
all
From VSV Narasimha Murthy to All Participants (11:23:06 AM):
all rpws
From Sivakumar to All Participants (11:23:07 AM):
all rows
From ilango kandasamy to All Participants (11:23:07 AM):
a;ll
From shanmugakannan to All Participants (11:23:08 AM):
all rows
From Manas to All Participants (11:23:11 AM):
2
From Motilal Jadhav to All Participants (11:23:12 AM):
2nd
From satish to All Participants (11:23:12 AM):
2
From Narendra to All Participants (11:23:14 AM):
3
From Pinky to All Participants (11:23:14 AM):
2nd
From RajeshJ to All Participants (11:24:08 AM):
@Sai - Where exactly we will use this slicing and broadcasting ?

Max and Min of an array

- Max: Returns maximum value in an array
- Min: Returns minimum value in an array




```
multi_array=np.array([(4,1,2),(5,8,9)])
print(np.max(multi_array))
print(np.min(multi_array))
```

```
9
1
```

Chat

sd 1
From ilango kandasamy to All Participants (11:25:45 AM):
std 1
From Sandeep to All Participants (11:25:45 AM):
equal from mean
From pragnyan to All Participants (11:25:46 AM):
0-1
From sridhar to All Participants (11:25:46 AM):
mean=0 and sd=1
From vishal bajetha to All Participants (11:25:46 AM):
mean =0
From vijay to All Participants (11:25:47 AM):
bell curve
From Bhavya to All Participants (11:25:48 AM):
mean 0 and sd 1
From Alok to All Participants (11:25:49 AM):
mean 0 std=1
From vishal bajetha to All Participants (11:25:50 AM):
st dev = 1
From Ravish to All Participants (11:25:52 AM):
left and right are same m=m=m
From Sandeep to All Participants (11:26:01 AM):
mean=0, std=1
From manaswini acharya to All Participants (11:26:16 AM):
yes
From bandi srinivas to All Participants (11:26:33 AM):
only positive
From Rahul Singh to All Participants (11:26:34 AM):
no
From VSV Narasimha Murthy to All Participants (11:26:35 AM):
no
From Rupesh to All Participants (11:26:39 AM):
Why row contains[]
From Rupesh to All Participants (11:27:31 AM):
yes
From Rupesh to All Participants (11:27:47 AM):
Ok
From Bhavya to All Participants (11:27:50 AM):
[] separetes rows, ',' separates columns
From ajay to All Participants (11:28:05 AM):
any example where shape is used?
From bandi srinivas to All Participants (11:28:13 AM):
3 rows and 2 elements
From Rupesh to All Participants (11:28:24 AM):
yes
From venugopal to All Participants (11:28:25 AM):
yes
From bandi srinivas to All Participants (11:28:26 AM):
yes
From Lokesh to All Participants (11:28:30 AM):
yes
From pragnyan to All Participants (11:28:48 AM):
whats difference bet max and amax
From bandi srinivas to All Participants (11:28:48 AM):
write a sql query