

Broadcasting

Allow operations on arrays with
mismatched shapes

Broadcasting

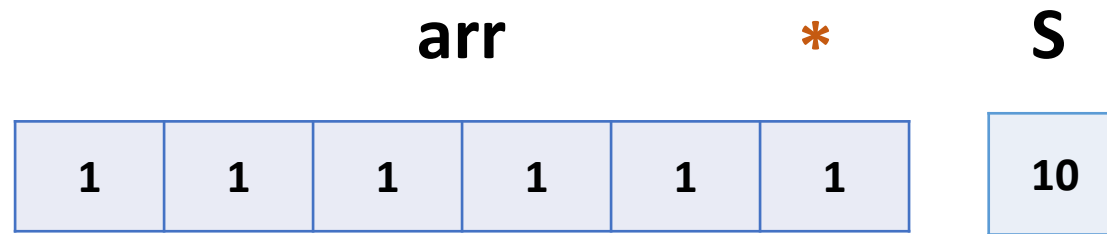
Describe how NumPy treats arrays with different shapes during arithmetic operations. Subject to certain constraints, the smaller array is “broadcast” across the larger array so that they have compatible shapes.

Compatibility in Broadcasting

Broadcasting Scalars

Can always broadcast, independent
of the other array in the operation

Broadcasting Scalars



Broadcasting Scalars

arr

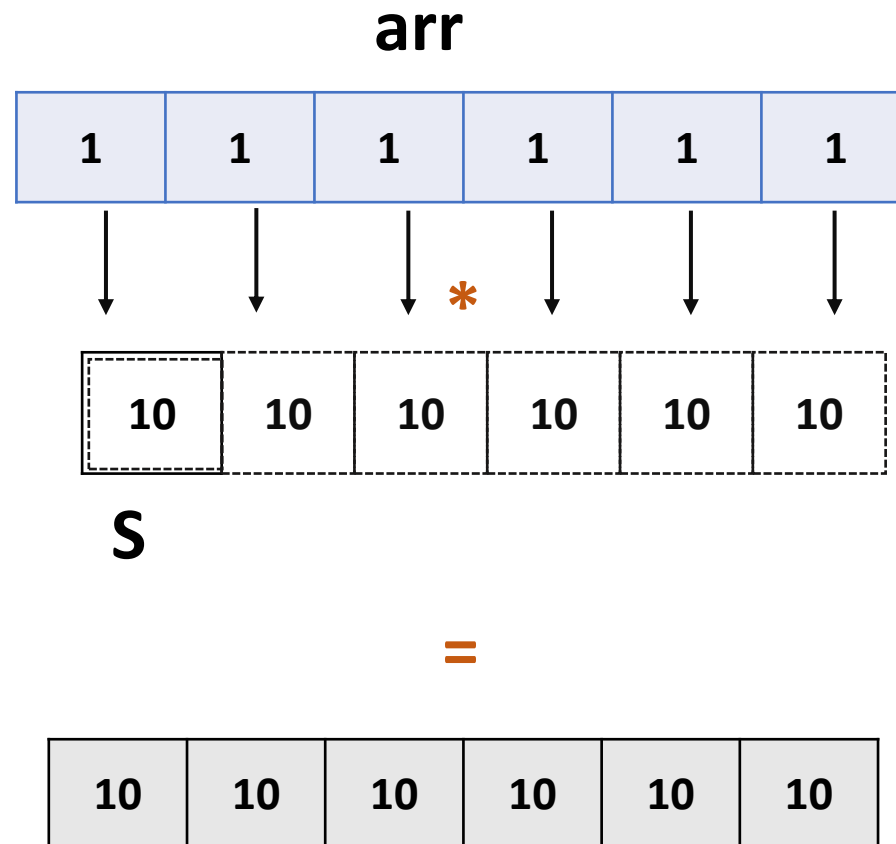
1	1	1	1	1	1
---	---	---	---	---	---

*

10

s

Broadcasting Scalars



Broadcasting Scalars

arr

1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

*

s

10

Broadcasting Scalars

arr

1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

*

s

10	10	10	10	10	10
10	10	10	10	10	10
10	10	10	10	10	10

=

10	10	10	10	10	10
10	10	10	10	10	10
10	10	10	10	10	10

Performed on pairs of arrays on an **element-by-element** basis.

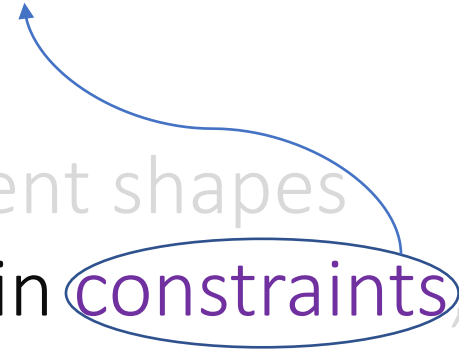
Broadcasting

Describe how NumPy treats arrays with different shapes during arithmetic **operations**. Subject to certain constraints, the smaller array is “broadcast” across the larger array so that they have compatible shapes.

Shapes of the two arrays are compared element-wise

Broadcasting

Describe how NumPy treats arrays with different shapes during arithmetic operations. Subject to certain constraints, the smaller array is “broadcast” across the larger array so that they have compatible shapes.



Compatibility in Broadcasting

Broadcasting Scalars

Can always broadcast,
independent of the other
array in the operation

Broadcasting Arrays

Can only be broadcast if the
shapes of two arrays match

Broadcasting Constraints

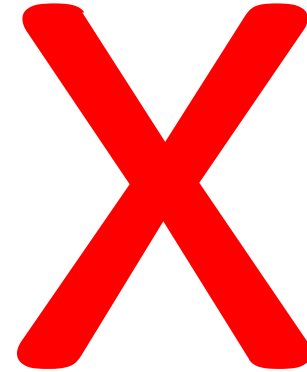
(3, 6)

(3, 1)

Broadcasting Constraints

(3, 6)

(3, 2)



Broadcasting Constraints

(3, 6)

(2, 1)



Dimensions are considered in reverse order, **starting with trailing dimension**, and working forward

Shapes of the two arrays are compared element-wise

Broadcasting

Describe how NumPy treats arrays with different shapes during arithmetic operations. Subject to certain **constraints**, the smaller array is “broadcast” across the larger array so that they have compatible shapes.

Stretch the smaller array by
making copies of its elements

Broadcasting

Describe how NumPy treats arrays with different shapes during arithmetic operations. Subject to certain constraints, the smaller array is “broadcast” across the larger array so that they have compatible shapes.



Broadcasting

No actual copies made;
computationally and
memory-efficient

Describe how NumPy treats arrays with different shapes during arithmetic operations. Subject to certain constraints, the smaller array is “broadcast” across the larger array so that they have compatible shapes.



Either corresponding dimension are equal or one of the two dimension is 1

Corresponding dimensions of arrays must be compatible

Broadcasting

Describe how NumPy treats arrays with different shapes during arithmetic operations. Subject to certain constraints, the smaller array is “broadcast” across the larger array so that they have compatible shapes.

Broadcasting



Powerful: Allows arrays of different shapes to be combined

Memory-efficient: Needless copies avoided

Computationally-efficient: Looping ops in C rather than in Python

Broadcasting Scalars



Scalars are easy to broadcast

Just replicate 1 element

Can always broadcast

Independent of the other array

Broadcasting Arrays



Shapes of the two arrays are compared

Starting from trailing end

Each dimension must be compatible