NumPy Set Operations

‹ Previous Next ›

What is a Set

A set in mathematics is a collection of unique elements.

Sets are used for operations involving frequent intersection, union and difference operations.

Create Sets in NumPy

We can use NumPy's `unique()` method to find unique elements from any array. E.g. create a set array, but remember that the set arrays should only be 1-D arrays.

Example

Convert following array with repeated elements to a set:

```
import numpy as np

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

x = np.unique(arr)

print(x)
```

Try it Yourself »

Finding Union

To find the unique values of two arrays, use the `union1d()` method.

Example

Find union of the following two set arrays:

```
import numpy as np

arr1 = np.array([1, 2, 3, 4])
arr2 = np.array([3, 4, 5, 6])

newarr = np.union1d(arr1, arr2)

print(newarr)
```

Try it Yourself »

Finding Intersection

To find only the values that are present in both arrays, use the `intersect1d()` method.

Example

Find intersection of the following two set arrays:

```
import numpy as np

arr1 = np.array([1, 2, 3, 4])
arr2 = np.array([3, 4, 5, 6])

newarr = np.intersect1d(arr1, arr2, assume_unique=True)
```

```
print(newarr)
```

Try it Yourself »

Note: the `intersect1d()` method takes an optional argument `assume_unique`, which if set to True can speed up computation. It should always be set to True when dealing with sets.

Finding Difference

To find only the values in the first set that is NOT present in the seconds set, use the `setdiff1d()` method.

Example

Find the difference of the set1 from set2:

```
import numpy as np

set1 = np.array([1, 2, 3, 4])
set2 = np.array([3, 4, 5, 6])

newarr = np.setdiff1d(set1, set2, assume_unique=True)

print(newarr)
```

Try it Yourself »

Note: the `setdiff1d()` method takes an optional argument `assume_unique`, which if set to True can speed up computation. It should always be set to True when dealing with sets.

Finding Symmetric Difference

To find only the values that are NOT present in BOTH sets, use the `setxor1d()` method.

Example

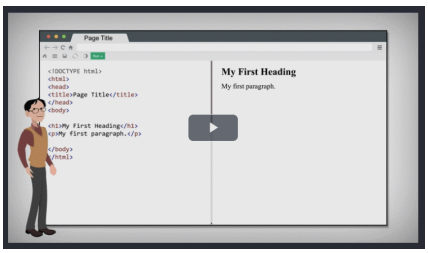
Find the symmetric difference of the set1 and set2:

```
import numpy as np

set1 = np.array([1, 2, 3, 4])
set2 = np.array([3, 4, 5, 6])

newarr = np.setxor1d(set1, set2, assume_unique=True)

print(newarr)
```

Try it Yourself »

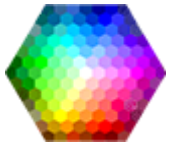
Note: the `setxor1d()` method takes an optional argument `assume_unique`, which if set to True can speed up computation. It should always be set to True when dealing with sets.

We just launched
W3Schools videos



Explore now

COLOR PICKER



Get certified
by completing
a Python
course today!

Get started

CODE GAME

Play Game

Report Error

Spaces

Pro

Get Certified

Top Tutorials

HTML Tutorial
CSS Tutorial
JavaScript Tutorial
How To Tutorial
SQL Tutorial
Python Tutorial
W3.CSS Tutorial
Bootstrap Tutorial
PHP Tutorial
Java Tutorial
C++ Tutorial
jQuery Tutorial

Top References

HTML Reference
CSS Reference
JavaScript Reference

SQL Reference
Python Reference
W3.CSS Reference
Bootstrap Reference
PHP Reference
HTML Colors
Java Reference
Angular Reference
jQuery Reference

Top Examples

HTML Examples
CSS Examples
JavaScript Examples
How To Examples
SQL Examples
Python Examples
W3.CSS Examples
Bootstrap Examples
PHP Examples
Java Examples
XML Examples
jQuery Examples

Get Certified

HTML Certificate
CSS Certificate
JavaScript Certificate
Front End Certificate
SQL Certificate
Python Certificate
PHP Certificate
jQuery Certificate
Java Certificate
C++ Certificate
C# Certificate
XML Certificate