# numpy unique without sort [duplicate]

**This question already has an answer here:**
numpy.unique with order preserved *5 answers*

How can I use numpy unique without sorting the result but just in the order they appear in the sequence? Something like this?

```
a = [4,2,1,3,1,2,3,4]

np.unique(a) = [4,2,1,3]
```

rather than

```
np.unique(a) = [1,2,3,4]
```

Use naive solution should be fine to write a simple function. But as I need to do this multiple times, are there any fast and neat way to do this?

python numpy

asked Oct 17 '12 at 3:58

kuantkid
**59** 1 5

**marked** as duplicate by Bhargav Rao ♦ python Nov 6 at 17:09

This question has been asked before and already has an answer. If those answers do not fully address your question, please ask a new question.

## 2 Answers

You can do this with the `return_index` parameter:

```
>>> import numpy as np
>>> a = [4,2,1,3,1,2,3,4]
>>> np.unique(a)
array([1, 2, 3, 4])
>>> indexes = np.unique(a, return_index=True)[1]
>>> [a[index] for index in sorted(indexes)]
[4, 2, 1, 3]
```

answered Oct 17 '12 at 4:11

del
**2,258** 4 27 39

---

You could do this using numpy by doing something like this, the mergsort is stable so it'll let you pick out the first or last occurrence of each value:

```
def unique(array, orderby='first'):
    array = np.asarray(array)
    order = array.argsort(kind='mergesort')
    array = array[order]
    diff = array[1:] != array[:-1]
    if orderby == 'first':
        diff = np.concatenate([[True], diff])
    elif orderby == 'last':
        diff = np.concatenate([diff, [True]])
    else:
        raise ValueError
    uniq = array[diff]
    index = order[diff]
    return uniq[index.argsort()]
```

This answer is very similar to:

```
def unique(array):
    uniq, index = np.unique(array, return_index=True)
    return uniq[index.argsort()]
```

But, numpy.unique uses an unstable sort internally so you're not guaranteed to get any specific index, ie first or last.

I think an ordered dict might also work:

```
def unique(array):
    uniq = OrderedDict()
    for i in array:
         uniq[i] = 1
    return uniq.keys()
```

edited Oct 17 '12 at 5:41

answered Oct 17 '12 at 4:13

Bi Rico

**14.7k** 1 20 49

Thanks for your quick reply. I have thought about the first one, but I am not sure whether it is the fastest. The second one should suffer from putting a numpy object into a python object implicitly :) – kuantkid Oct 17 '12 at 4:21

Is the problem with the second `unique` using `np.unique`'s `return_index` argument that it might produce incorrect results? That this `unique` might return a sequence with some elements not respecting the order imposed by the original sequence, e.g., (purely for demonstration) `unique([1,0,1]) --> [0, 1]` ? – Ahmed Fasih Oct 6 '13 at 12:29

`np.unique`'s documentation (docs.scipy.org/doc/numpy/reference/generated/numpy.unique.html) states that the indexes returned with `return_index=True` will indicate the *first* occurrences, so your second `unique` should be safe and correct, right? – Ahmed Fasih Oct 6 '13 at 12:36