Documentation for module numpy.lib.arraysetops

gendocs.py

November 28, 2007

Contents

1 Module numpy.lib.arraysetops

Set operations for 1D numeric arrays based on sorting.
Contains: ediff1d, unique1d, intersect1d, intersect1d_nu, setxor1d, setmember1d, union1d, setdiff1d
Notes:
All functions work best with integer numerical arrays on input (e.g. indices). For floating point arrays, innacurate results may appear due to usual round-off and floating point comparison issues.
Except unique1d, union1d and intersect1d_nu, all functions expect inputs with unique elements.
Speed could be gained in some operations by an implementaion of sort(), that can provide directly the permutation vectors, avoiding thus calls to argsort().
Run _test_unique1d_speed() to compare performance of numpy.unique1d() and numpy.unique() - it should be the same.
To do: Optionally return indices analogously to unique1d for all functions.
created: 01.11.2005 last revision: 07.01.2007

1.1 Functions

1.1.1 Function ediff1d

The differences between consecutive elements of an array, possibly with prefixed and/or appended values.
Parameters:

‘ary‘ ... array
This array will be flattened before the difference is taken.

‘to_end‘ ... number, optional
If provided, this number will be tacked onto the end of the returned differences.

‘to_begin‘ ... number, optional
If provided, this number will be taked onto the beginning of the returned differences.

Returns:

‘ed‘ ... array
The differences. Loosely, this will be (ary[1:] - ary[:-1]).

1.1.2 Function unique1d

Find the unique elements of 1D array.
Most of the other array set operations operate on the unique arrays generated by this function.
Parameters:

‘ar1‘ ... array
This array will be flattened if it is not already 1D.

‘return_index‘ ... bool, optional
If True, also return the indices against ar1 that result in the unique array.

Returns:

‘unique‘ ... array
The unique values.

‘unique_indices‘ ... int array, optional
The indices of the unique values. Only provided if return_index is True.

1.1.3 Function intersect1d

Intersection of 1D arrays with unique elements.
Use unique1d() to generate arrays with only unique elements to use as inputs to this function. Alternatively, use intersect1d_nu() which will find the unique values for you.
Parameters:

‘ar1‘ . . . array
‘ar2‘ . . . array

Returns:

‘intersection‘ . . . array

1.1.4 Function intersect1d_nu

Intersection of 1D arrays with any elements.
The input arrays do not have unique elements like intersect1d() requires.
Parameters:

‘ar1‘ . . . array
‘ar2‘ . . . array

Returns:

‘intersection‘ . . . array

1.1.5 Function setxor1d

Set exclusive-or of 1D arrays with unique elements.
Use unique1d() to generate arrays with only unique elements to use as inputs to this function.
Parameters:

‘ar1‘ . . . array
‘ar2‘ . . . array

Returns:

‘xor‘ . . . array
The values that are only in one, but not both, of the input arrays.

1.1.6 Function setmember1d

Return a boolean array of shape of ar1 containing True where the elements of ar1 are in ar2 and False otherwise.
Use unique1d() to generate arrays with only unique elements to use as inputs to this function.
Parameters:

‘ar1‘ . . . array
‘ar2‘ . . . array

Returns:

‘mask‘ . . . bool array
The values ar1[mask] are in ar2.

1.1.7 Function union1d

Union of 1D arrays with unique elements.
Use unique1d() to generate arrays with only unique elements to use as inputs to this function.
Parameters:

‘ar1‘	...	array
‘ar2‘	...	array

Returns:

‘union‘	...	array

1.1.8 Function setdiff1d

Set difference of 1D arrays with unique elements.
Use unique1d() to generate arrays with only unique elements to use as inputs to this function.
Parameters:

‘ar1‘	...	array
‘ar2‘	...	array

Returns:

‘difference‘	...	array The values in ar1 that are not in ar2.