< (/nDarrays/02-xarray-architecture/)

label-based indexing

> (/nDarrays/04-plotting/)

Overview

Teaching: 5 min
Exercises: 5 min
Questions

- How does the labeling of dimensions enhance the xarray workflow?

Objectives

- learning to locate data in xarray using integer- and label-based lookups
- using named dimensions to located data

Indexing

Indexing is used to select specific elements from xarray files. Let's select some data from the 2-meter temperature `DataArray`. We know from the previous lesson that this `DataArray` has dimensions of time and two dimensional space (latitude and longitude).

You are probably already used to conventional ways of indexing an array. You will know in advance that the first array index is time, the second is latitude, and so on. You would then use positional indexing (http://xarray.pydata.org/en/stable/indexing.html)):

```
ds['t2m'][0, 0, 0]
```

This method of handling arrays should be familiar to anyone who has worked with arrays in MATLAB or NumPy. One challenge with this approach: it is not simple to associate an integer index position with something meaningful in our data. For example, we would have to write some function to map a specific date in the time dimension to its associated integer. Therefore, xarray lets us perform positional indexing using labels instead of integers:

```
ds['t2m'].loc['1979-01-01T06:00:00', :, :]
```

This is great, but we still need to be keeping track of the fact that our index position 1 is the time dimension, position 2 is latitude, etc. So rather than looking up our dimension by position, xarray enables us to use the dimension name instead:

```
ds['t2m'].isel(time=0, latitude=0, longitude=0)
```

Here, the `isel` method refers to a selection by integer position. Finally, we can combine the benefits of both a labeled index and a named dimension as follows:

```
ds['t2m'].sel(time='1979-01-01T06:00:00', latitude=75.0, longitude=180.0)
```

Slicing the data

Key Points

- xarray's labeled dimensions free the user from having to track positional ordering of dimensions when accessing data, creating a more simplified workflow