Xarray Integration

Cartopy transforms can be passed to xarray!

This creates a very quick path for creating professional looking maps from netCDF data.

Remember we've looked at this SST dataset before in the xarray class. We know xarray can make nice default plots, but now we can pass the `transform=` argument and get proper maps!

In [1]:

```
# import statements: xarray, cartopy.crs, cartopy, matplotlib
import xarray as xr

import cartopy.crs as ccrs

import matplotlib.pyplot as plt

%matplotlib inline
```

load in our SST dataset

```
url =
'http://www.esrl.noaa.gov/psd/thredds/dodsC/Datasets/noaa.ersst.v5/sst.mnmea


ds = xr.___(url, _____)
```

In [9]:

```
url = 'http://www.esrl.noaa.gov/psd/thredds/dodsC/Datasets/noaa.ersst.v5/sst.mnm
ds = xr.open_dataset(url, drop_variables=['time_bnds'])

ds
```

Out[9]:

```
xarray.Dataset

▶ Dimensions:        (lat: 89, lon: 180, time: 2011)
▼ Coordinates:
    lat              (lat)                float32  88.0 86.0 84.0 ... -86.0 -88.0
    lon              (lon)                float32  0.0 2.0 4.0 ... 354.0 356.0 35...
    time             (time)       datetime64[ns]  1854-01-01 ... 2021-07-01
▼ Data variables:
    sst              (time, lat, lon)     float32  ...
▶ Attributes:   (38)
```

make a map of SST

Pick a particular day and create a dataset for it:

In [11]:

```
sst = ds.sst.sel(time='2000-01-01', method='nearest')
```

then create a figure with Robinson projection, coastlines & gridlines:

```
fig = plt.figure(figsize=(9,6))
ax = plt.axes(____=ccrs.Robinson())
ax.coastlines()
ax.gridlines()
```

and plot our xarray dataset using the built in xarray plotting, plus that new special thing we need to do for maps:

```
sst.___(ax=ax, ___=ccrs.PlateCarree(), vmin=2, vmax=30, cbar_kwargs=
{'shrink': 0.4})
```

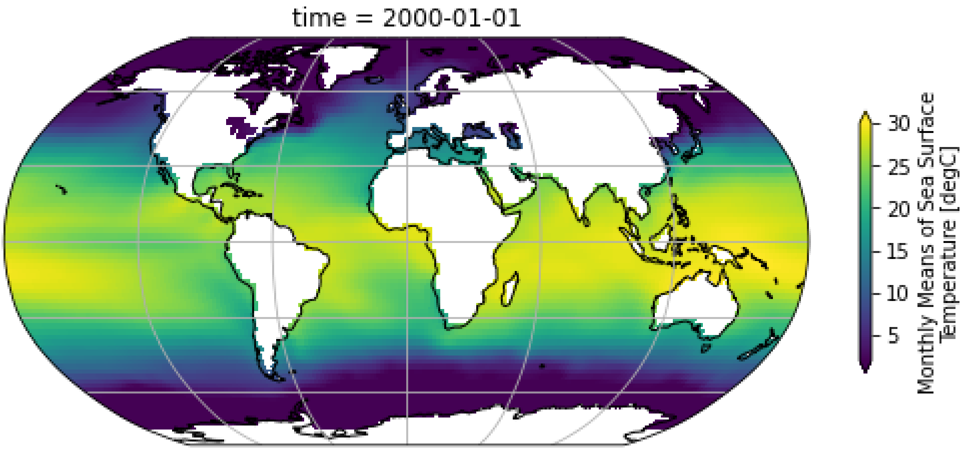
In [16]:

```
fig = plt.figure(figsize=(9,6))
ax2 = plt.axes(projection=ccrs.Robinson())
ax2.coastlines()
ax2.gridlines()

# could do sst.plot(ax=ax2,...) but not needed since only one axis
sst.plot(transform=ccrs.PlateCarree(),vmin=2, vmax=30, cbar_kwargs={'shrink': 0.
```

Out[16]:

```
<matplotlib.collections.QuadMesh at 0x7f9fd0839280>
```



Excercise 02

In [ ]: