```
In [1]:
```

```
#
#
  File:
#
    NUG_rectilinear_contour_PyNGL.py
#
#
#
    Illustrates creating contours over a map
#
#
  Categories:
#
    contour plots
#
  Author:
#
    Karin Meier-Fleischer
#
  Date of initial publication:
#
#
   June 2015
#
#
  Description:
#
    This example shows how to create contours over
#
    a map using rectilinear data.
#
  Effects illustrated:
#
   o Drawing filled contours over a map
#
    o Defining contour levels based on min/max and spacing
#
  Output:
#
   One visualization is produced.
#
#
#
  Notes: The data for this example can be downloaded from
#
    http://www.ncl.ucar.edu/Document/Manuals/NCL User Guide/Data/
#
 NCL User Guide Python Example:
                                   NUG rectilinear contour PyNGL.py
   - filled contour over map plot
   rectilinear grid (lat/lon)
   - colorbar
 2015-06-04 kmf
from __future__ import print_function
import Ngl, Nio
import os,sys
```

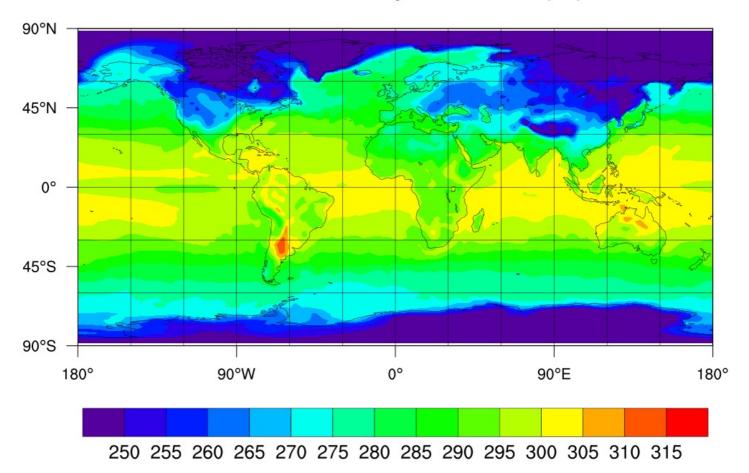
In [4]:

```
#-- define variables
diri = "./"
                                              #-- data directory
fname = "rectilinear_grid 2D.nc"
                                               #-- data file name
ffile = os.path.join(diri, fname)
#---Test if file exists
if(not os.path.exists(ffile)):
    print("You do not have the necessary file ({}) to run this example.".format(ffile))
    print("You can get the files from the NCL website at:")
    print("http://www.ncl.ucar.edu/Document/Manuals/NCL_User_Guide/Data/")
   sys.exit()
minval = 250.
                                               #-- minimum contour level
maxval = 315
                                               #-- maximum contour level
                                               #-- contour level spacing
inc
           5.
```

```
In [6]:
```

```
#-- open file and read variables
                                            #-- open data file
#-- first time step
      = Nio.open_file(ffile, "r")
f
      = f.variables["tsurf"][0,:,:]
temp
      = f.variables["lat"][:]
                                              #-- all latitudes
lat
      = f.variables["lon"][:]
                                               #-- all longitudes
tempac,lon = Ngl.add cyclic(temp,lon)
#-- open a workstation
                     = Ngl.Resources()
                                              #-- generate an res object
wkres
                     = "png"
                                              #-- output type
wks_type
                     = "NUG_rectilinear_contour_PyNGL"
wks name
wks
                      = Ngl.open wks(wks type,wks name,wkres)
#-- set resources
                       = Ngl.Resources()
                                             #-- generate an resource object for plot
if hasattr(f.variables["tsurf"],"long_name") and hasattr(f.variables["tsurf"],"units"):
    res.tiMainString = "{} ({})".format(f.variables["tsurf"].long_name,
                                       f.variables["tsurf"].units)
                                                                         #-- set main title
res.cnFillOn
                          = True
                                                #-- turn on contour fill.
                          = False
res.cnLinesOn
                                               #-- turn off contour lines
                     = False
= False
res.cnLineLabelsOn
                                               #-- turn off line labels.
                                               #-- turn off info label.
res.cnInfoLabelOn
res.cnLevelSelectionMode = "ManualLevels" #-- select manual level selection mode
res.cnMinLevelValF = minval
                                              #-- minimum contour value
                         = maxval
                                              #-- maximum contour value
res.cnMaxLevelValF
                                               #-- contour increment
#-- choose color map
res.cnLevelSpacingF
                         = inc
                         = "rainbow"
res.cnFillPalette
res.mpGridSpacingF
                         = 30
                                               #-- map grid spacing
                                              #-- longitude locations of data
res.sfXArray
                          = lon
                                               #-- latitude locations of data
res.sfYArray
                             lat
res.lbOrientation
                         = "Horizontal"
                                              #-- labelbar orientation
map = Ngl.contour map(wks,tempac,res)
                                             #-- draw contours over a map.
#-- end
Ngl.end()
```

surface temperature (K)



In []: