Orientation and Hardcopy Output

```
# Import the modules needed for the tuturial
import vcs, cdms, cdutil, time, os, sys

# Open data file:
filepath = os.path.join(sys.prefix, 'sample_data/clt.nc')
cdmsfile = cdms.open( filepath )

# Extract a 3 dimensional data set
data = cdmsfile('clt')

# Initialize VCS:
v = vcs.init()

# Opening a VCS Canvas - not necessary to do this
# before issuing the plot command!
v.open()

# A quick plot of the data
v.plot( data )

print ""
print "Press the Return key to see next plot."
sys.stdin.readline()

# Changing plot orientation to "Portrait"
v.portrait()

print "Press the Return key to see next plot."
sys.stdin.readline()

# To change the orientation back to "Landscape"
v.landscape()
print "Generating Landscape output..."


###############################################################
#Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â
# Saving "Landscape" orientation graphics to fileÂ Â Â
# "Landscape" is the default output orientation.Â Â  Â
#Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â
###############################################################

# Append to a postscript file
v.postscript('test.ps')

# Overwrite the existing postscript file
v.postscript('test.ps')

# GIF format - append landscape orientation gif image
v.gif('test.gif', merge='a', orientation='l', geometry='800x600')

# CGM format - append to an existing cgm file
v.cgm('test.cgm', 'a')

# Encapsulated Postscript - overwrite an existing eps file
v.eps('test.eps', 'r')

# PDF format
v.pdf('test.pdf')
```

```
##############################################################
# GhostScript (gs) formatÂ  Â  Â  Â  Â  Â  Â  Â  Â  Â
# This routine allows the user to save the VCS canvas in one of the manyÂ Â Â Â Â Â Â Â Â Â Â Â Â
# GhostScript (gs) file types (also known as devices). To view otherÂ Â Â Â  Â  Â  Â  Â  Â  Â
# GhostScript devices, issue the command "gs --help" at the terminalÂ Â  Â  Â  Â  Â  Â  Â  Â
# prompt.Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â
##############################################################

v.gs('example')  # defaults: device='png256', orientation='l' and resolution='792x612'
v.gs(filename='example.tif', device='tiffpack', orientation='l', resolution='800x600')
v.gs(filename='example.pdf', device='pdfwrite', orientation='l', resolution='200x200')


# Changing plot orientation to "Portrait"
v.portrait()
print "Generating Portrait output..."

##############################################################
#Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â
# Saving "Portrait" orientation graphics to fileÂ Â Â Â Â  Â  Â  Â  Â  Â
#Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â
##############################################################

# Append postscript output to an existing file
v.postscript('test.ps','a','p')

# Overwrite existing postscript file with a new postscript file
v.postscript('test.ps','r','p')

# GIF format - overwrite gif image(s) output with portriat gif image
v.gif('test.gif', merge='r', orientation='p', geometry='800x600')

# CGM format - overwrite existing cgm file
v.cgm('test.cgm', 'r')

# Encapsulated Postscript - append portait output to an existing eps file.
v.eps('test.eps', 'a', 'p')

# PDF format
v.pdf ('test.pdf', 'p')


##############################################################
# GhostScript (gs) formatÂ Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â
# This routine allows the user to save the VCS canvas in one of the many Â  Â  Â  Â  Â  Â  Â  Â
# GhostScript (gs) file types (also known as devices). To view otherÂ Â Â Â  Â  Â  Â  Â  Â  Â
# GhostScript devices, issue the command "gs --help" at the terminalÂ Â Â Â Â Â Â Â Â Â Â Â Â Â Â
# prompt.
##############################################################

v.gs(filename='example.jpg', device='jpeg', orientation='p', resolution='1000x1000')
```