

## Command-line arguments

- Each program requires certain **arguments** to define its operation.
- Arguments are explained in the **manual pages** and in the usage messages.
- Each argument specification begins with a **hyphen**, followed by a letter, and sometimes a number or character string immediately after the letter.
- Do not space between the hyphen, letter, and number or string.
- Do space between options.
- Example: `pscoast -R0/20/0/20 -G200 -JM6i -W0.25p  
-B5 -V > map.ps`

## Standardized arguments (1/2)

Among the many possible arguments in GMT, 13 have the same meaning for all programs:

- `-B` Defines tickmarks, annotations, and labels for basemaps and axes
- `-H` Specifies that input tables have header record(s)
- `-J` Selects a map projection or one of several non-map projections
- `-P` Selects Portrait plot orientation [Default is landscape]
- `-R` Defines the min. and max. coordinates of the map/plot region
- `-U` Plots a time-stamp, by default in the lower left corner of page
- `-V` Verbose operation

## Standardized arguments (2/2)

- `-X` Sets the x-coordinate for the plot origin on the page
- `-Y` Sets the y-coordinate for the plot origin on the page
- `-c` Specifies the number of plot copies
- `-:` Input geographic data are (lat,lon) rather than (lon,lat)
- `-K` Allows more plot code to be appended to this plot later
- `-O` Allows this plot code to be appended to an existing plot

## Standardized arguments: exercises

- Verbose:

```
pscoast -R0/360/-70/70 -JM6.5i -Ba60f30 -G0 -Dc  
-P -V > GMT_mercator.ps
```

- Time tag:

```
pscoast -R0/360/-70/70 -JM6.5i -Ba60f30 -G0 -Dc  
-P -U > GMT_mercator.ps
```

- Landscape:

```
pscoast -R0/360/-70/70 -JM6.5i -Ba60f30 -G0 -Dc  
-U > GMT_mercator.ps
```

- Shift by 4 inches verticall:

```
pscoast -R0/360/-70/70 -JM6.5i -Ba60f30 -G0 -Dc  
-P -Y4i > GMT_mercator.ps
```

## Overlay and Continue modes

- If a GMT command is **NOT THE LAST** (in a script for instance), then it must contain `-K`, meaning *“more postscript code will be appended later”*.
- If a GMT command is **NOT THE FIRST**, then it must contain `-O`, meaning *“this postscript code is an overlay on top of previous code.”*
- Therefore, in a GMT script:
  - The first command contains `-K` only
  - The last command contains `-O` only
  - All commands in between contain `-O -K`

## Colors in GMT

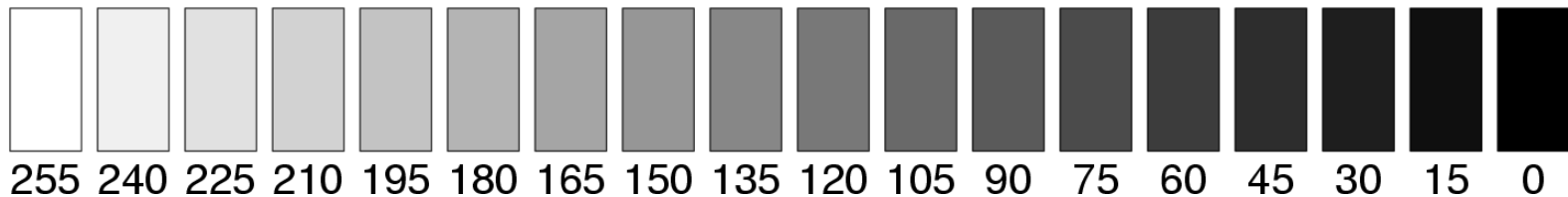
- GMT allows you to use `color` and grey scales for drawing lines or filling polygons (landmasses, symbols, etc.)
- For instance:
  - Many GMT commands use `-G` to define the fill color (e.g., `pscoast` for landmasses)
  - Many GMT commands use `-W` to define the pen color (and width) for coastlines
  - `pscoast` uses `-I` to define the pen color (and width) for rivers
  - `pscoast` uses `-S` to define the fill color for oceans and lakes.
- Colors are defined with numbers from 0 to 255 that give their intensity

## Grey scale in GMT

- White is coded as 0 - Black is coded as 255
- Anything between 0 and 255 is grey
- Example of black landmasses and grey coastlines:

```
pscoast -R0/360/-70/70 -JM6.5i -Ba60f30 -Dc -G0  
-W4/200 -P > GMT_mercator.ps
```

SHADES OF GREY



## Colors in GMT

- Colors in GMT are defined as a mixture of 3 primary colors: **Red** / **Green** / **Blue** (or R/G/B in short).
- The intensity of each of those 3 primary colors ranges from 0 (lowest intensity) to 255 (highest intensity).
- For instance:
  - Red is coded as **255/0/0** (full red, no green, no blue).
  - Yellow is a mixture of red and green, therefore defined as **255/255/0**.
- The web link below provides a wide range of RGB color definitions:  
<http://stommel.tamu.edu/baum/GMT/colors/colors.html>



White (255 255 255)



Red (255 0 0)



Green (0 255 0)



Blue (0 0 255)



Magenta (255 0 255)



Cyan (0 255 255)



Yellow (255 255 0)



Black (0 0 0)



Aquamarine (112 219 147)



Brass (181 166 66)



Cadet Blue (95 159 159)



Copper (184 115 51)



Dark Green (47 79 47)



Dark Orchid (153 50 205)



Dark Purple (135 31 120)



Dark Wood (133 94 66)



Dim Grey (84 84 84)



Firebrick (142 35 35)



Flesh (245 204 176)



Forest Green (35 142 35)



Gold (205 127 50)



Goldenrod (219 219 112)



Green Copper (82 127 118)



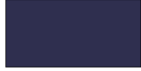
Khaki (159 159 95)



Maroon (142 35 107)



Midnight Blue (47 47 79)



New Tan (235 199 158)



Old Gold (207 181 59)



Orange (255 127 0)



Orchid (219 112 219)



Quartz (217 217 243)



Rich Blue (89 89 171)



Scarlet (140 23 23)



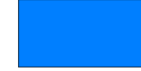
Sea Green (35 142 104)



Sienna (142 107 35)



Slate Blue (0 127 255)



Spring Green (0 255 127)



Steel Blue (35 107 142)



Tan (219 147 112)



Violet (79 47 79)



Blue Violet (159 95 159)



Bright Gold (217 217 25)



Brown (166 42 42)



Bronze (140 120 83)



Cool Copper (217 135 25)



Coral (255 127 0)



Cornflower Blue (66 66 111)



Dark Brown (92 64 51)



Dk. Slate Blue (107 35 142)



Dark Tan (151 105 79)



Dusty Rose (133 99 99)



Feldspar (209 146 117)



Green Yellow (147 219 112)



Hunter Green (33 94 33)



Indian Red (78 47 47)



Light Blue (192 217 217)



Lt. Steel Blue (143 143 189)



Light Wood (233 194 166)



Lime Green (50 205 50)



Mandarin Orange (228 120 51)



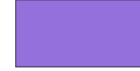
Medium Blue (50 50 205)



M. Forest Green (107 142 35)



Medium Orchid (147 112 219)



M. Sea Green (66 111 66)



M. Slate Blue (127 0 255)



M. Spring Green (127 255 0)



M. Violet Red (219 112 147)



Medium Wood (166 128 100)



Navy Blue (35 35 142)



Neon Blue (77 77 255)



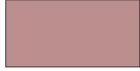
Orange Red (255 36 0)



Pale Green (143 188 143)



Pink (188 143 143)



Plum (234 173 234)



Salmon (111 66 66)



Silver (230 232 250)



Sky Blue (50 153 204)



Spicy Pink (255 28 174)



Violet Red (204 50 153)



Wheat (216 216 191)



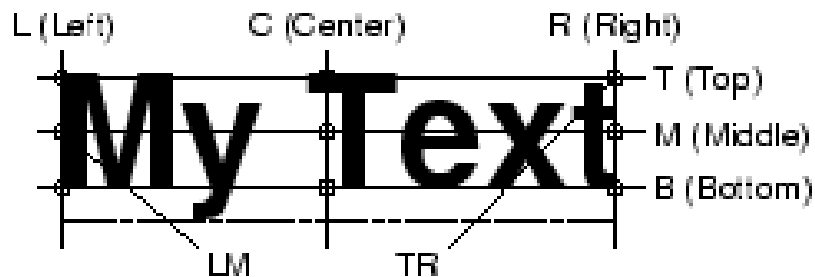
## Other pen attributes

A pen in GMT has three attributes: width, color, and texture, specified for instance with: `-Wwidth[/color][ttexture][p]`

- **Width:** measured in units of the current device resolution. Append `p` to specify pen width in points (1/72 of an inch).
- **Color:** see above, gray shade or RGB.
- **Texture:**
  - `to` = dotted line
  - `ta` = dashed line
  - `tstring:offset = string = length_gap_length_gap`, offset from origin
- **Example:** `pscoast -R-140/-50/20/65 -JM6.5i -Ba20f5  
-N1/2/255/0/0t20_10_5_10:10 -N2/2/0/to  
-W1/0/0/255 -P > GMT_mercator.ps`

## Plotting text strings

- GMT command is `pstext`
- Input data to `pstext` must contain:  
*x y size angle fontno justify text*
  - *x y* = location, in plot units
  - *size* = font size in points (e.g. 12)
  - *angle* = angle CCW from horizontal
  - *fontno* = font number (e.g. 0 for Helvetica)
  - *justify* = combination of LCM and TMB:



## Plotting text strings: input data

As an external file:

1. Create or edit file with text data, for instance:

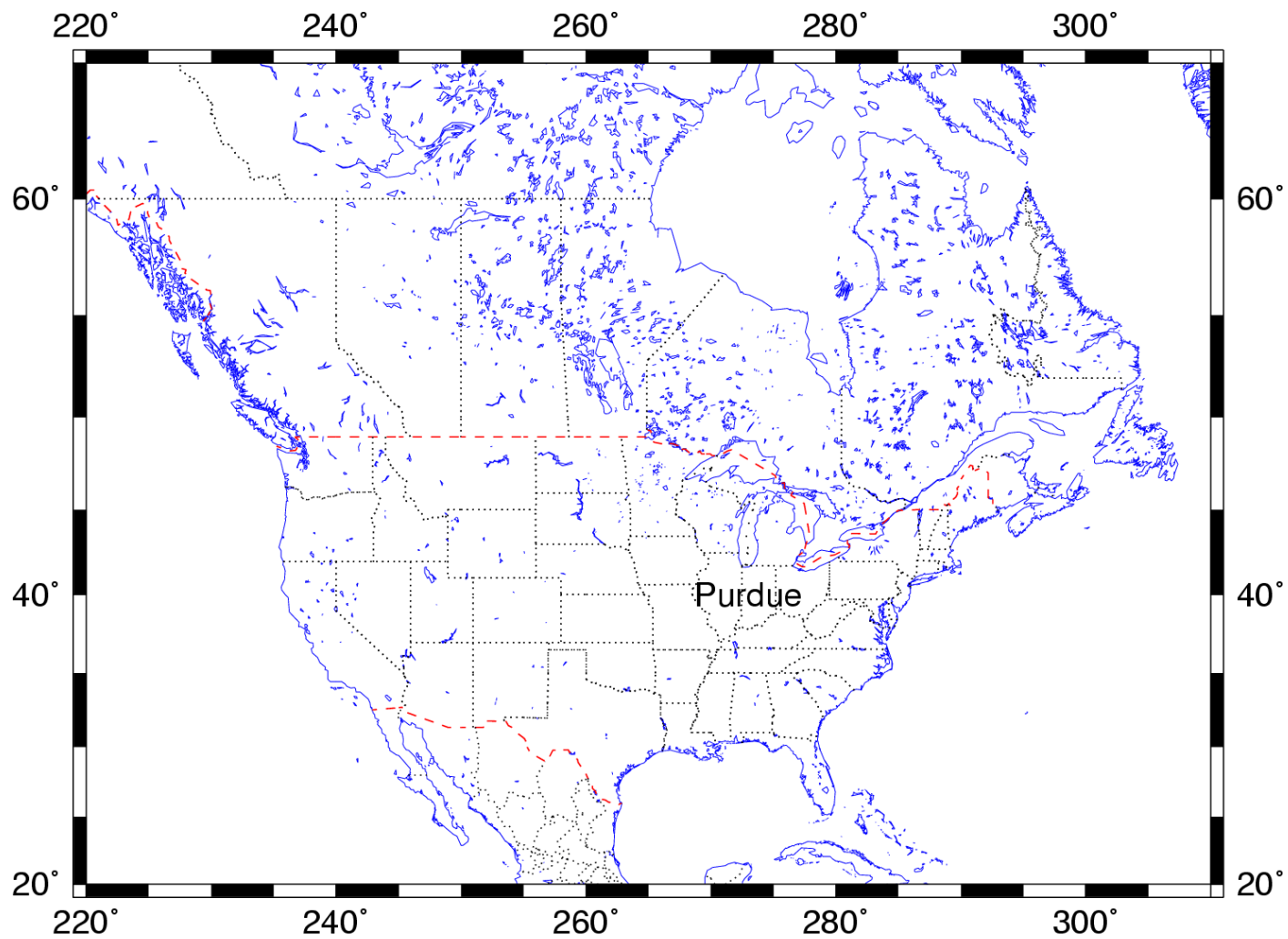
```
echo "-86 40 14 0 0 CM Purdue" >! mytext.txt
```

2. Call `pstext`, for instance after a `pscoast` command:

```
pscoast -R-140/-50/20/65 -JM6.5i -Ba20f5  
        -N1/2/255/0/0ta -N2/2/0/to -W1/0/0/255  
        -P -K > GMT_mercator.ps
```

```
pstext mytext.txt -R -JM -O >> GMT_mercator.ps
```

# Plotting text strings: input data



## Plotting text strings: input data

Directly:

1. By “*piping*” text into `pstext`:

```
echo "-86 40 14 0 0 CM Purdue" | pstext  
mytext.txt -R -JM -O >> GMT_mercator.ps
```

2. Using `<<`:

```
pstext << eof -R -JM -O >> GMT_mercator.ps  
-86 40 14 0 0 CM Purdue  
eof
```

# Exercise

Plot a map of the continental US with the name of each state capital shown at its geographic location. Use file `state_capitals`, `awk`, `pscoast`, and `pstext`.

