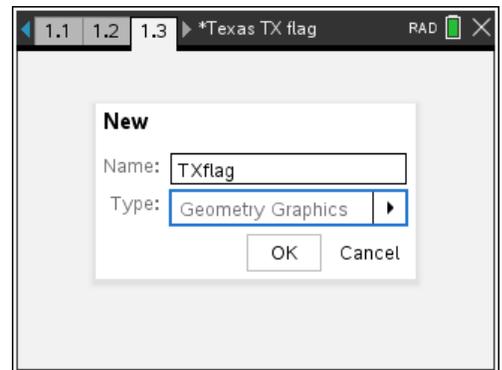
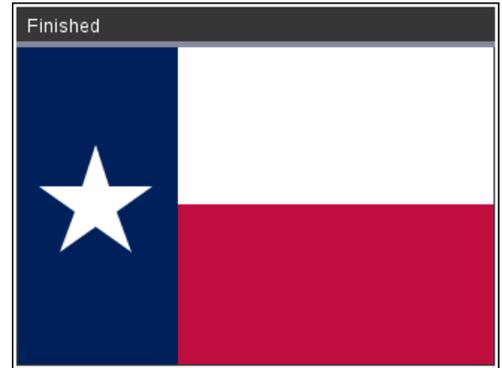


In the first activity using the `ti_draw` module, you were introduced to the rectangle, circle and polygon tools. Here you will apply your drawing skills to make an image of a flag. Our demonstration makes the Texas flag, of course, for the home of Texas Instruments, Inc.

Introduction: Most flags have a nice geometric design. Some flags are a little more complex. In this activity you will make the flag of the state of **Texas**, USA, since this is **Texas** Instruments, Inc. There are only three colors and the challenge is to make the Lone Star in the proper position. Your experience with the **capTlan's Shield** will come in handy!

You can find most flag proportions and design specifications online.

1. Begin a new Python program using the **Geometry Graphics** template.

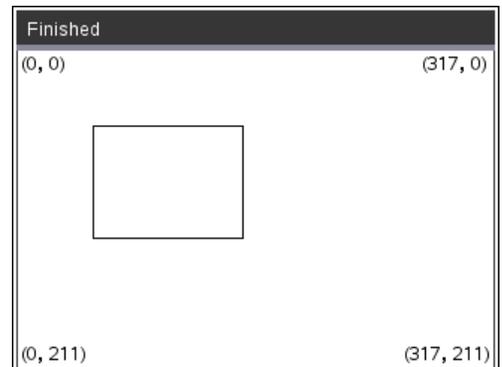


2. We will use the *default* window which has the origin (0,0) in the upper left corner. All values here are *pixels*. The top right corner is (317, 0) and the bottom left corner is (0,211). When drawing a rectangle, the width is measured from left to right and the height is measured from **top-to-bottom**.

The rectangle shown here is made with:

`draw_rect(50, 50, 100, 75)`

(50,50) is the upper left corner. The width is 100 (to the right) and the height is 75 (*down*).



Remember that the screen is 318x212 pixels which means the *aspect ratio* is **3:2** (318 / 212). Many flags use this aspect ratio.



10 MOC: Python Modules

TI-NSPIRE™ CX II PYTHON

TI DRAW: FLAGS

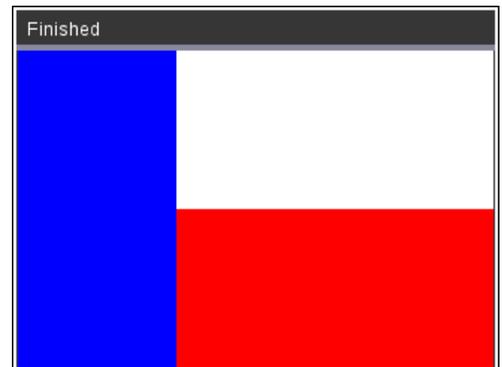
3. Since the screen is white, begin the flag by drawing a red rectangle in the lower half of the screen:

```
set_color(255,0,0)  
fill_rect(0,106,318,106)
```

Note that this is *not* the proper shade of red! It will have to be edited.



4. For the blue field, try drawing a **filled rectangle** in the left one-third of the screen. Try it yourself before looking at the next step.

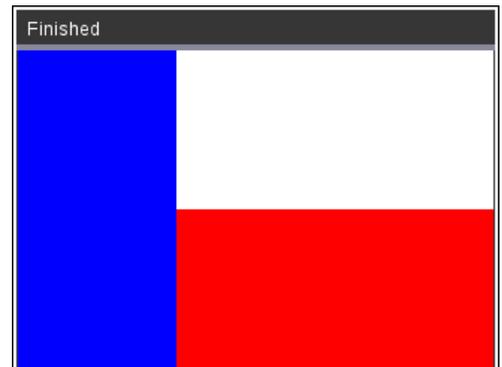


5. We used:

```
set_color(0, 0, 255)  
fill_rect(0, 0, 318/3, 212)
```

*Note: Yes, you can use expressions in the function arguments.
It's OK if you draw 'off the screen'!*

The proportions look right, but the colors are 'off'.





10 MOC: Python Modules

TI-NSPIRE™ CX II PYTHON

6. Now for the fun part: **the star!**

If you completed the previous **capTlan's shield** activity you can use the star-making code here. You will have to adjust the numbers used in the algorithm to place the star in the proper position with the proper dimensions.

Note that we have altered the colors here. You can do that at any time. You can find most flag design details online including dimensions, proportions, and colors.

7. Here is the complete *star* code from the **capTlan's shield** activity:

```

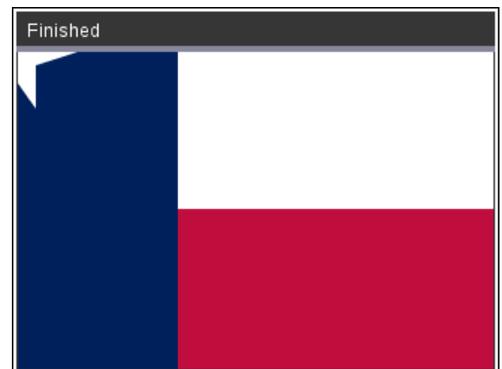
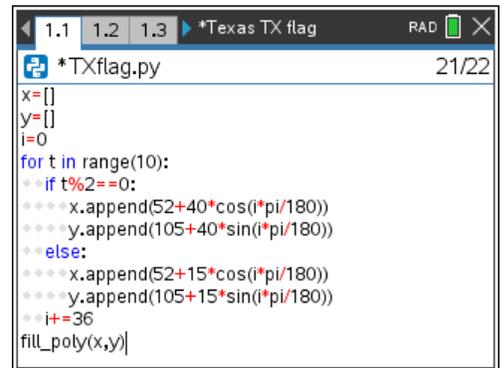
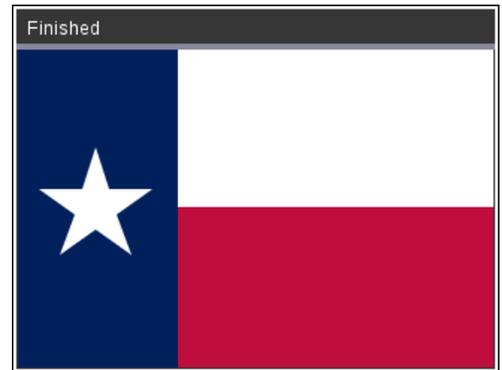
x=[ ]
y=[ ]
i=0
for t in range(10):
  if t % 2 == 0:
    x.append(40 * cos(i * pi / 180))
    y.append(40 * sin(i * pi / 180))
  else:
    x.append(15 * cos(i * pi / 180))
    y.append(15 * sin(i * pi / 180))
  i += 36
fill_poly(x,y)

```

Remember to add **from math import *** to your program to access **sin()**, **cos()** and **pi**.

8. Adding the star code above to the Texas flag produces the image seen here. You can see a piece of the star in the upper left corner. It needs some adjusting to be placed in the proper position.

Translate the star to the middle of the blue rectangle by *adding* values to the x- and y-coordinates.





10 MOC: Python Modules

TI-NSPIRE™ CX II PYTHON

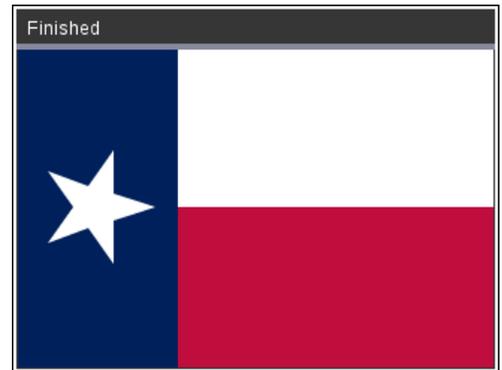
TI DRAW: FLAGS

- 9. We added 53 (half the width of the blue rectangle) to the x-coordinates and 106 (one half the height of the flag) to the y-coordinates:

```

if t % 2 == 0:
  x.append(53+ 40 * cos(i * pi / 180))
  y.append(106+ 40 * sin(i * pi / 180))
else:
  x.append(53+ 15 * cos(i * pi / 180))
  y.append(106+ 15 * sin(i * pi / 180))

```



- 10. To *rotate* the star so that it points straight up, change the initial value of the variable *i* as you did for the **capTlan's shield**. Try it now.

```

1.1 1.2 1.3 *Texas TX flag RAD 13/22
*TXflag.py
x=[]
y=[]
i=0
for t in range(10):
  if t%2==0:
    x.append(52+40*cos(i*pi/180))
    y.append(105+40*sin(i*pi/180))
  else:
    x.append(52+15*cos(i*pi/180))
    y.append(105+15*sin(i*pi/180))
  i+=36

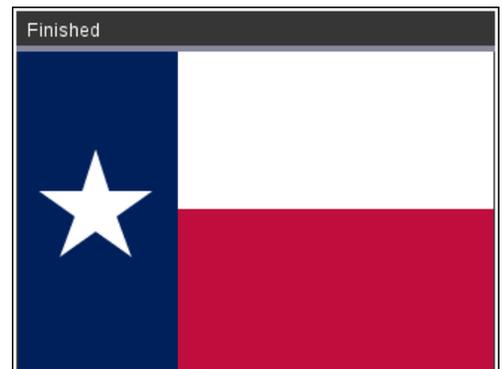
```

- 11. We used

$$i = -90$$

since the screen is coordinated 'upside down'! The y-values *increase* from top to bottom in this 'default' window.

Your task is to find (*online*) the appropriate shades of red and blue for this Texas ('The Lone Star State') flag.





10 MOC: Python Modules

TI-NSPIRE™ CX II PYTHON

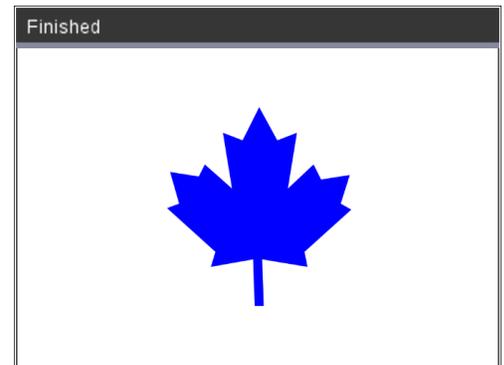
TI DRAW: FLAGS

Challenge: select a state, province, or country flag to design using your TI-Nspire CX II and your Python drawing tools.

For the flag of **Canada**, you will have to determine the coordinates of the *vertices* of the maple leaf *polygon* and then use `fill_poly(x, y)` to make the leaf.

The lists of the x- and y-coordinates are hidden in this image.

```
1.1 1.2 1.3 ▶ *Canada ...nna RAD [171, 25/36]
*canada.py
# Canada Flag...
from ti_draw import *
xs= [157,
ys= [171,
set_color(0,0,255)
fill_poly(xs,ys)
```



Note the aspect ratio of the **Canada** flag
And the color is wrong here
And the vertical *red* stripes are missing
but which shade of *red*?
So... the rest of the job is left to you.
Bonne chance!