CIS 122 Fall 2015 Project 3
Due Monday, October 19, 11:59 PM

Briefly

Submit four Python 3 programs
Your programs are worth a total of 25 points
Test your programs -- did they work right? -- before uploading to Canvas. Your programs should contain code that tests your functions.

Project3-jump.py 5 points:

Learning objective
Short simple single purpose functions simplify your programs and reduce errors.

Define a **jump(distance)** function that **moves the turtle without making a mark on the screen**.
Before returning, jump should set the pen down so the turtle can once again make marks.

return None

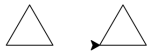
Your code must **test your jump function**:

Draw a triangle

Ask for a distance, use the **input( )** function to accept the distance, then convert it to a **float** (decimal number),

jump the distance requested,

then draw another triangle.



Rubric
3 points Correctly working jump function
1 point Draw triangle function
1 point Test via input and float

Bonus 1 point Use a docstring as first statement after the def in the jump function

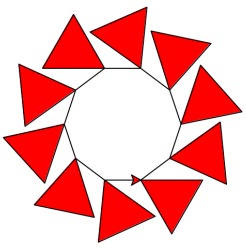
Project3-colored-triangles.py 5 points

Learning objectives
A function can call other functions you have written

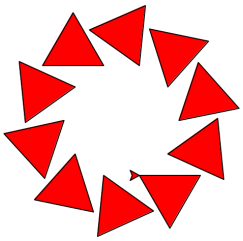
Define a function **draw_triangle(size)**
When called, it draws a triangle, each side **size** long.

Define another function **color_triangle(size, color)**
C**all your draw_triangle function to draw the triangle.**

Use a for loop to draw a series of 10 to 12 triangles each with a color



or



Rubric

5 points total

2 points define color_triangle(size, color)
it fills the triangle with the given color

1 point color_triangle does not duplicate triangle code; instead, it calls draw_triangle(size)

2 points nice display of several colored triangles similar to the display shown above

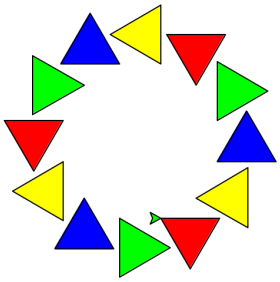
Project3-manycolors.py 7 points

Learning objective

Select one item after another from a list of items

Define a list of 3 to 6 colors

Make a display like that in Project3-colored-triangles but each item should get the next item from your color list.



Rubric

5 points Use colors in order from the color list

2 Individual items display without connecting lines using the jump function.

Project3-colors-from-input.py 8 points

Learning objective

Update items in a list using input() function

Before drawing ask user for input

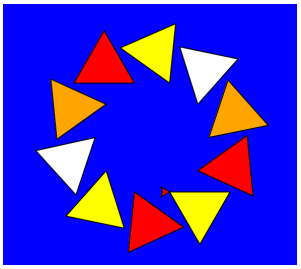
Get at least 3 colors from user
Update your color list entries with the user input

Ask user for background color

As with Project3-manycolors, make a similar display

Example

```
Type color to use, such as blue yellow
Type color to use, such as blue red
Type color to use, such as blue orange
Type color to use, such as blue white
Type background color to use, such as black blue
>>>
```



Rubric 5 points Interaction with user for colors

2 points Use colors correctly in display

1 point Triangles "jump" – no connecting lines

Bonus Challenge

3 points

There's a similarity between draw_triangle and draw_square and draw_pentagon.

Write and test a more general purpose

draw_poly(size, sides)

Do the same for colored polygons

color_poly(size, sides, color)

Instead of using draw_triangle(50) you could do this:

draw_poly(50, 3)

There is a readability issue here; you can't really look at the call to draw_poly and be sure what role the two numbers play.

It is possible to use a "keyword" type of call.
draw_poly(50, **sides=3**)

color_poly(50, **sides=3, color='red'**)

Note: it's possible to use t.circle to do this. Don't.

The point is to learn how to write a more general purpose function.

```
Type color to use, such as blue yellow
Type color to use, such as blue green
Type color to use, such as blue light blue
Type color to use, such as blue orange
Type background color to use, such as black gray
Type number of sides for polygons such as 4 for a square 5
Type number of polygons to draw (1-60) 33
>>>
```