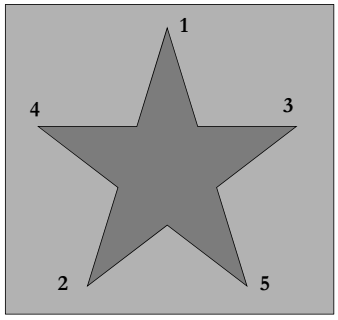
ECS 10

10/29

Assignment

- TAs tell me grades on Program 3 were not very good. Start earlier. It is supposed to be hard. You need to work on it over the course of a few days.
- Current assignment is also hard. Plan on getting help if you need it. Get to lab hours EARLY this week.
- Show up in section with questions. Ask them in front of the class.

Print out list of points

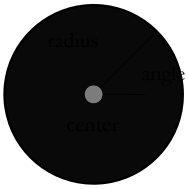


Each point lies on a circle…

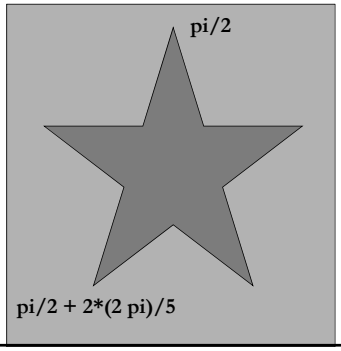
Points on a circle

- (x,y) is a point on the circle.
- (cx,cy) is the center.
- a is the angle (0-2 Pi)
- Counter-clockwise, in radians, from x direction.

x = cx + radius * cos(a)
y = cy + radius * sin(a)



Some Angles



The first two angles for a five-pointed star.

Point from Angles

```
radius = 200
center = [250,250]
angle = math.pi/2.0
x = center[0] + radius*math.cos(angle)
y = center[1] + radius*math.sin(angle)
point = [x,y]
```

- A point is a list, containing an x and a y.
- A list of points will be a list of lists.
- Making a list of lists is a bit tricky.

The Wrong Way

```
>>> print pointList
[[25, 25], [475, 475]]
>>> point3 = [250,250]
>>> pointList + point3
[[25, 25], [475, 475], 250, 250]
```

- + concatenates two lists, pointList with two elements (each a point), and point3 with two elements (each an int), to get a list with four elements.

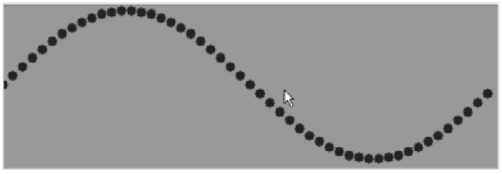
The Right Way

```
>>> pointList + [point3]
[[25, 25], [475, 475], [250, 250]]
```

- + concatenates two lists, pointList with two elements (each a point) and [point3], with one element (a point), to make a list with three elements (all points).

But enough about your project…

- I'm going to do a project with some similarities.



- First, plot the sin() function, from 0 to 2 Pi.

Notice

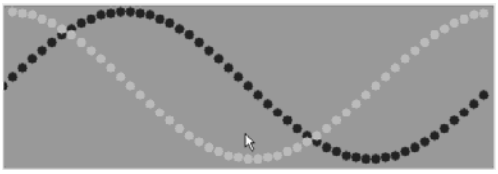
- …how nice it is to have a list of points. I don't have to change the for loop at all, it will draw whatever list I give it.
- Now I'm going to look at the differences between each sin () and the previous one.
- Need to save list of angles…

The value None

- Sometimes we want a variable to be defined, but we don't have a value to give it.
- We can always use the value None.
- None is it's own data type; there is only one possible value for a variable of type None.

```
lastAngle = None
for angle in angleList:
    if lastAngle != None:
        …
    lastAngle = angle
```

Differences look like cos(angle)



- Why?
- Derivative of sin(angle) is cos(angle)

Notice…

- How I can look at a point, and the previous point, in a **for** loop, by always saving the last value in a variable.
- This can be useful for drawing a line between one point and the last one, hmmm?
- There are lots of other ways you could do this.

Take project in small steps

- Draw points of a 5 pointed star as little circles.
- Store the points in a list, then draw them as circles.
- Draw lines connecting each pair of consecutive points in the list (make a pentagon).
- Change how you make the points so that the list is in the order you want for the star.
- Get user input and handle other numbers of points.