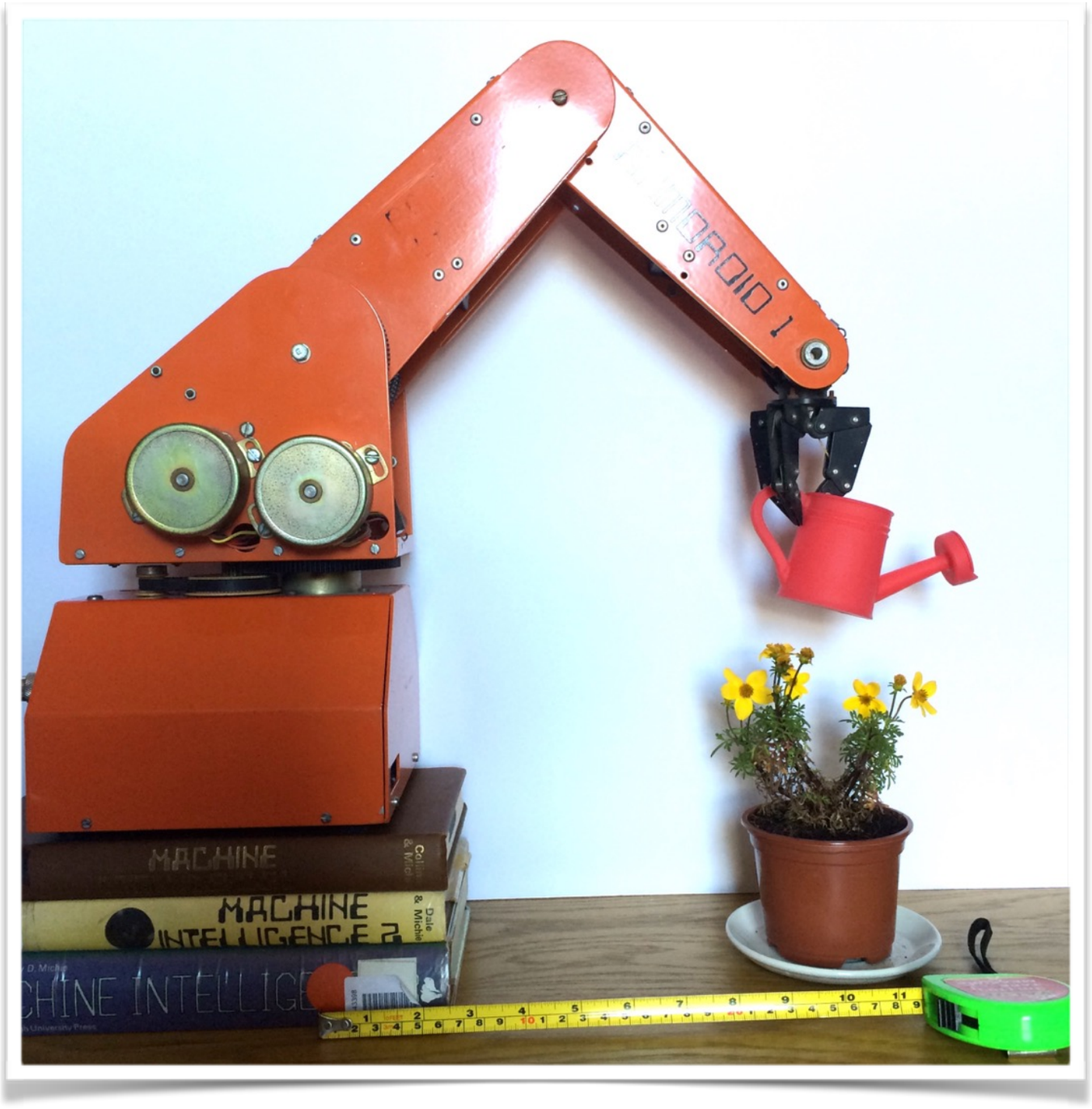
Droid Geometry

@SteveBattle
University of the West of England

Maintenance drones are eco-friendly

Flower Power

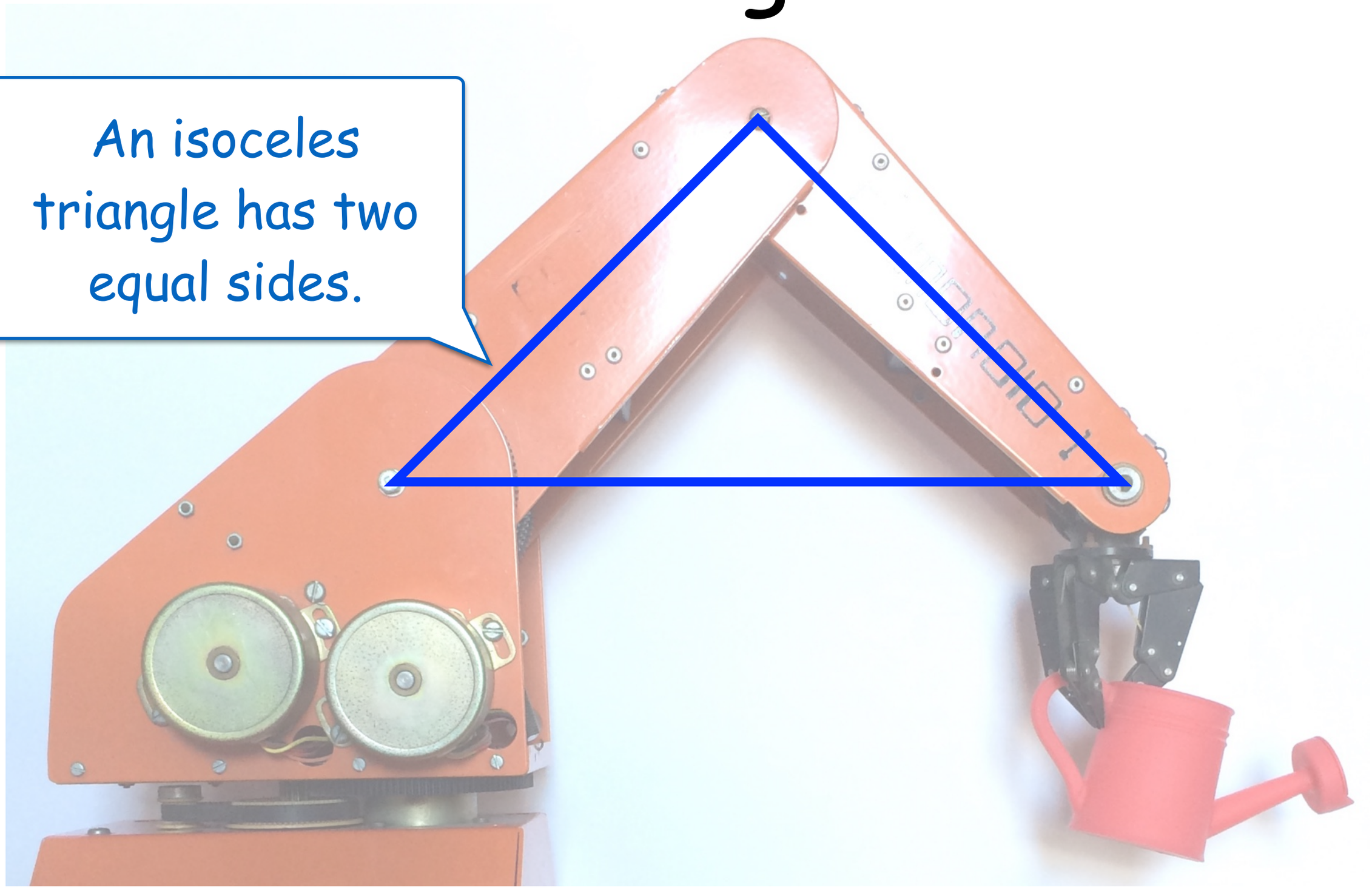


- How can we make a robot arm water a flower.

- This is the 1981 Armdroid 1.

Triangles

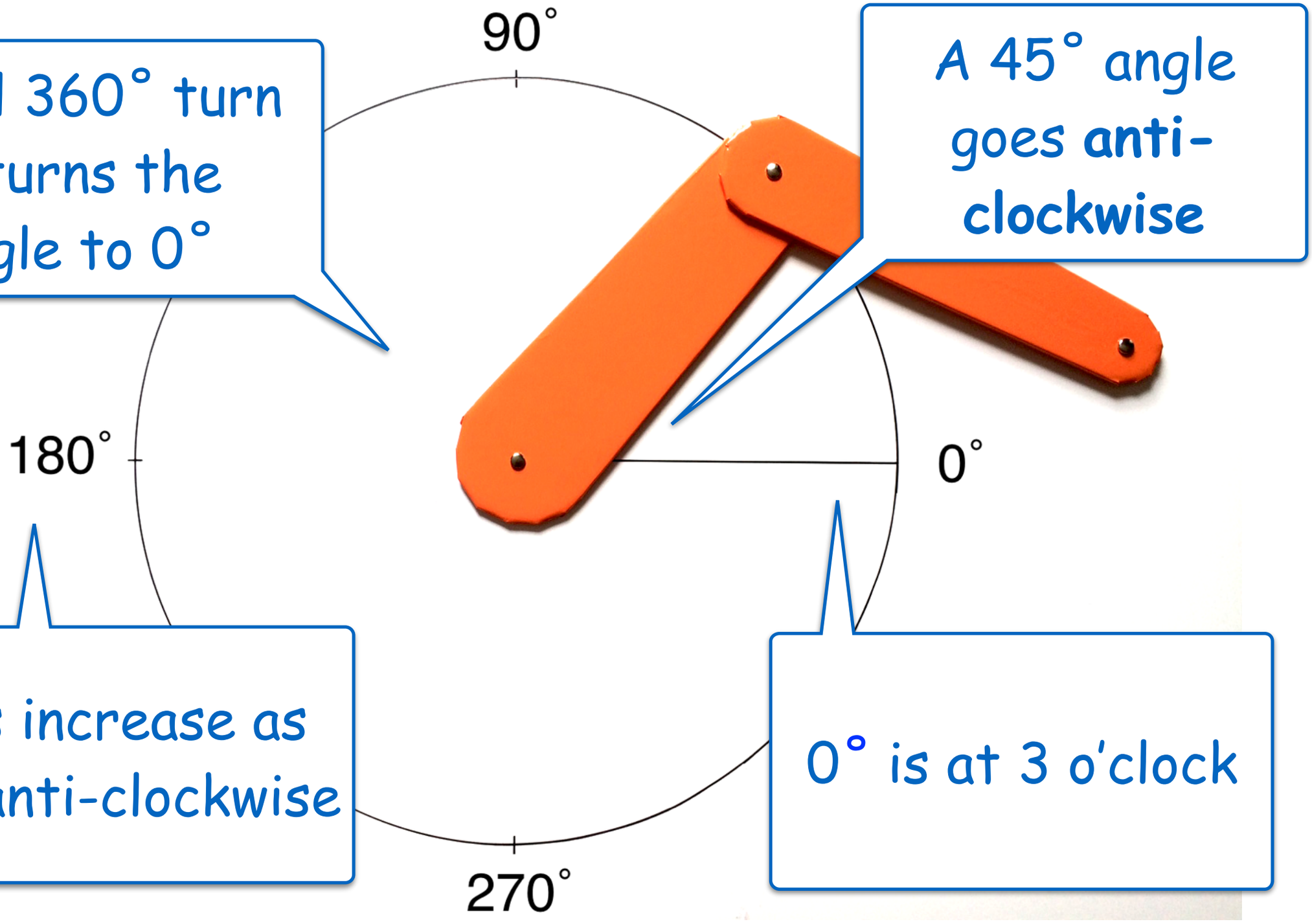
An isoceles triangle has two equal sides.



Circles and Angles

A full 360° turn returns the angle to 0°

A 45° angle goes **anti-clockwise**



Angles increase as you go anti-clockwise

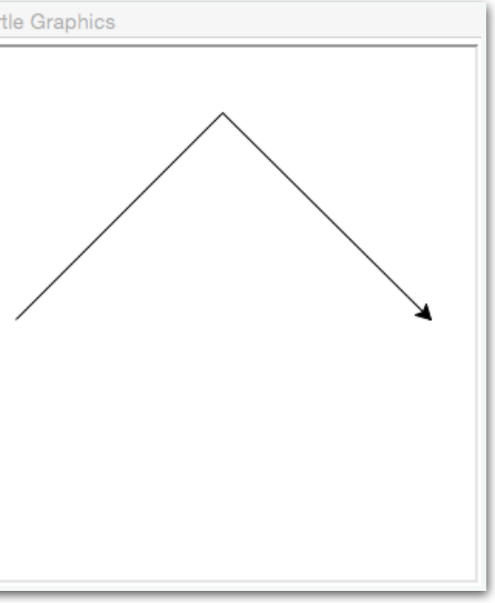
0° is at 3 o'clock

Turtles all the way down

Python 3.4.3 Shell

```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 23 2015, 02:52:03)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> from turtle import *
>>> left(45)
>>> forward(190)
>>> right(90)
>>> forward(190)
>>>
```

Python Turtle Graphics



Ln: 9 Col: 4

Same measurements as the Armdroid

- We can simulate this in Python.
- IDLE is a Python program editor.

turtle functions

forward(**length**)	backward(**length**)
left(**angle**)	right(**angle**)
penup()	pendown()
done()	
speed(**s**)	*e.g. ‘slow’, ‘fast’, ‘fastest’*
shape(**name**)	*e.g. ‘turtle’, ‘classic’*
goto(**x**,**y**)	*x,y coordinates*

Robot Simulator

```
from turtle import *

length = 190

def arm(angle):
    left(angle)
    forward(length)

speed('slow')

arm(45)
arm(-45)

done()
```

variable

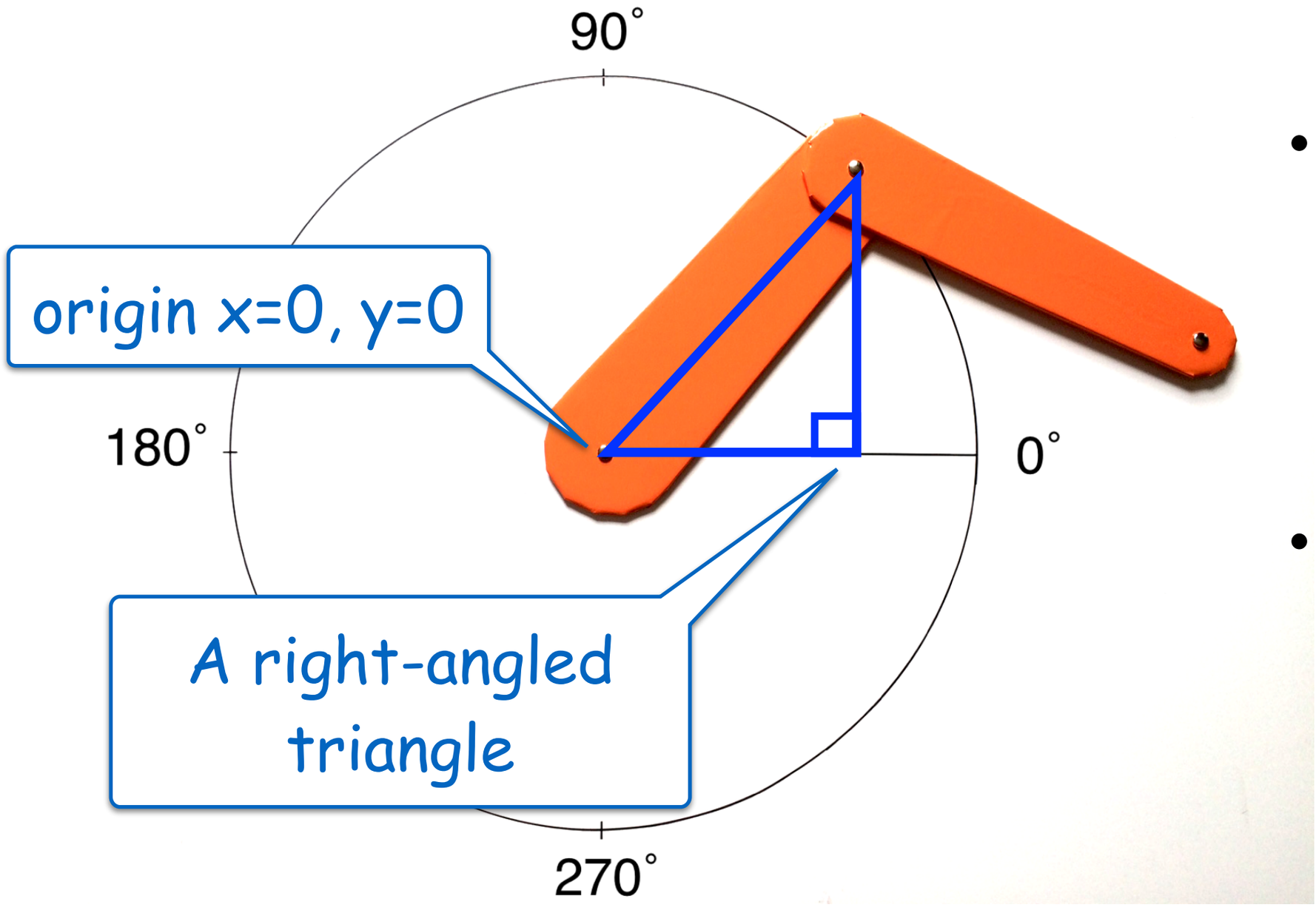
Function definition and parameters

Function call with arguments

try changing the angles

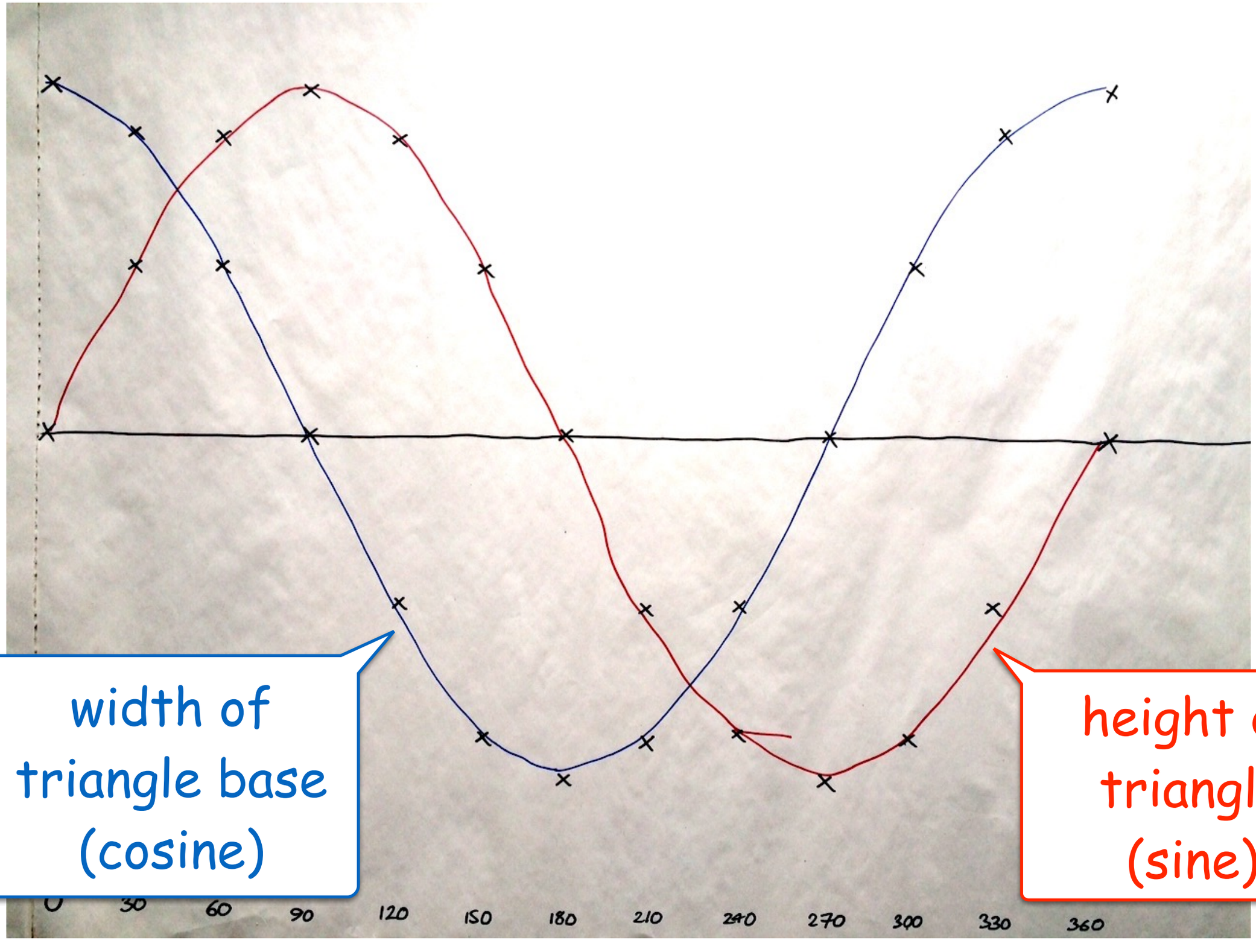
- File > New File
- File > Save
- Run > Run Module

Robot Kinematics: How far does it reach?



- Break the problem down into triangles.
- We know the arm lengths.

Triangle width & height



width of triangle base (cosine)

height of triangle (sine)

Analogue vs Digital

- To work out the reach of the upper arm read out the width (cosine), w, in the plot for the angle (eg. a=45˚).

arm.py - /Users/Steve/Documents/Python/arm.py (3.4.3)

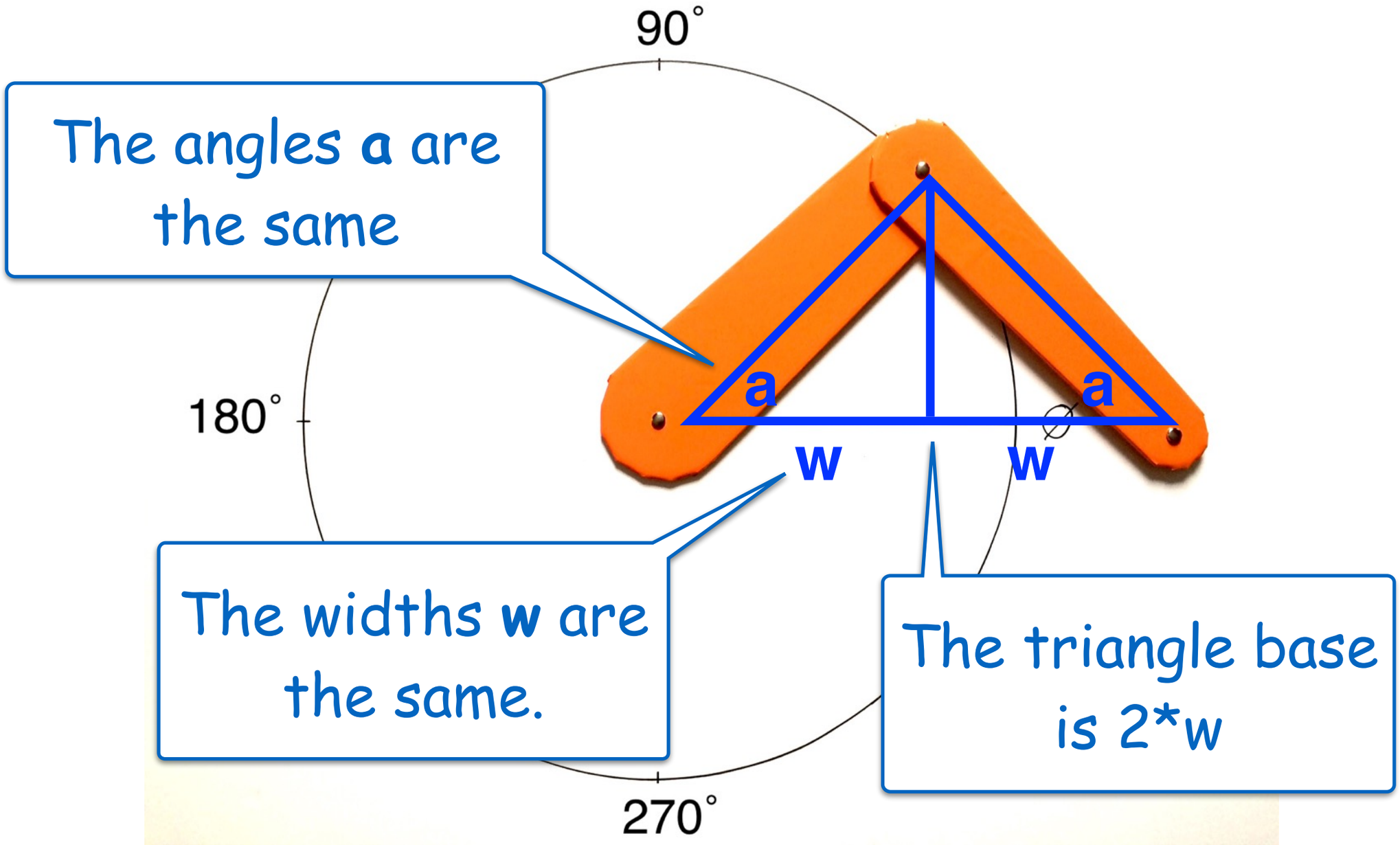
```
a = 45
b = 90

arm(a)
arm(b)

from math import *

w = cos(radians(a))*length
print(w)

done()
```

Python does this with the **cos** function.

Uses degrees **radians**.

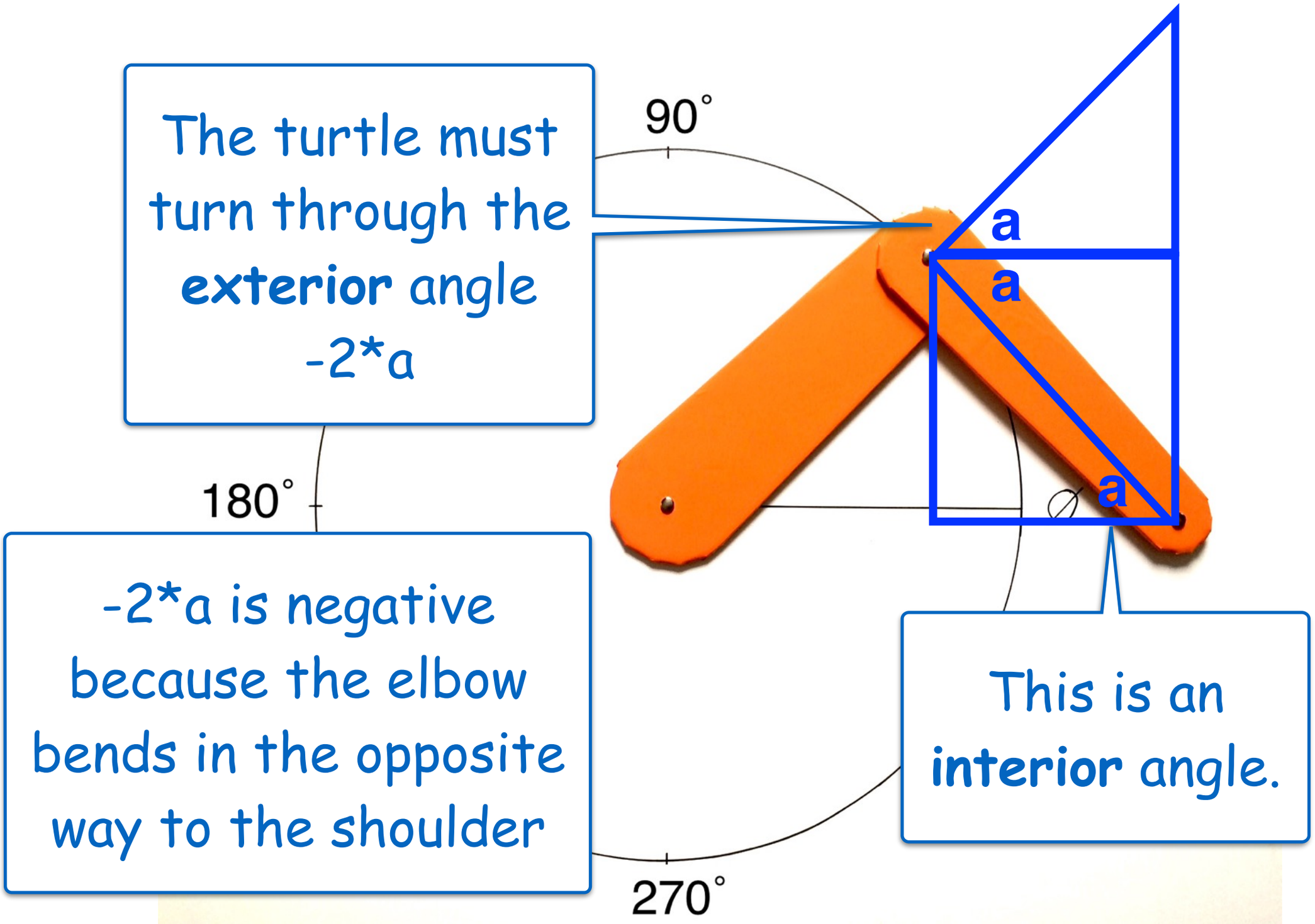
Col: 0

Adding the forearm



Give it some elbow

The turtle must turn through the **exterior** angle -2*a



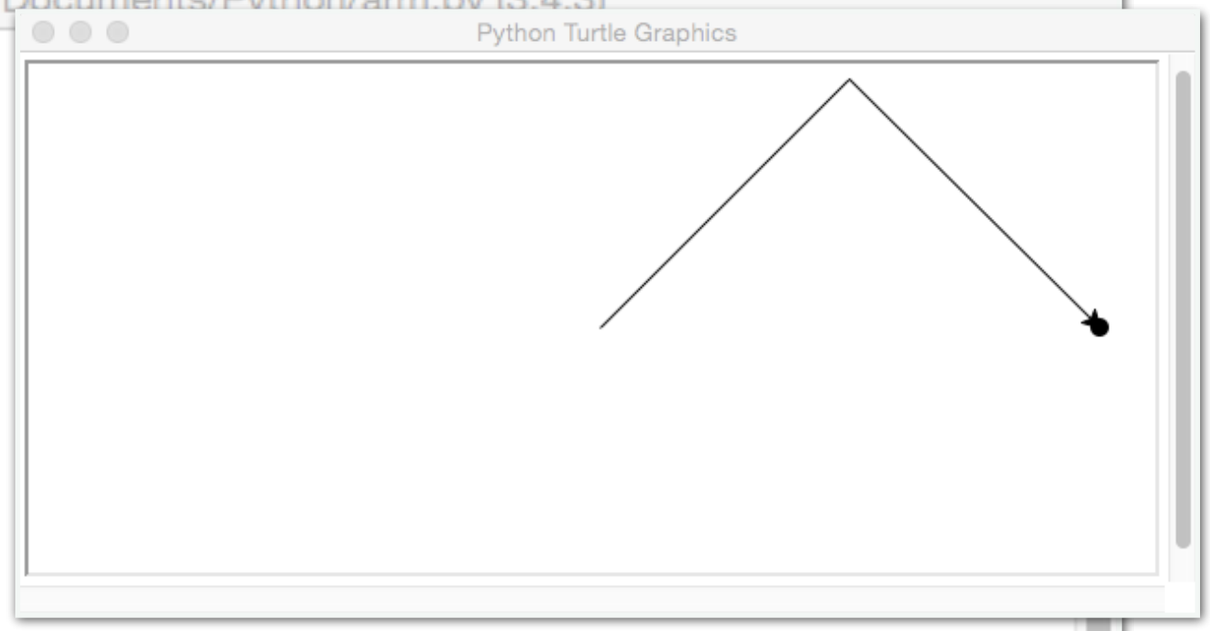
-2*a is negative because the elbow bends in the opposite way to the shoulder

This is an **interior** angle.

Test the results

arm.py - /Users/Steve/Documents/Python/arm.py (3.4.3)

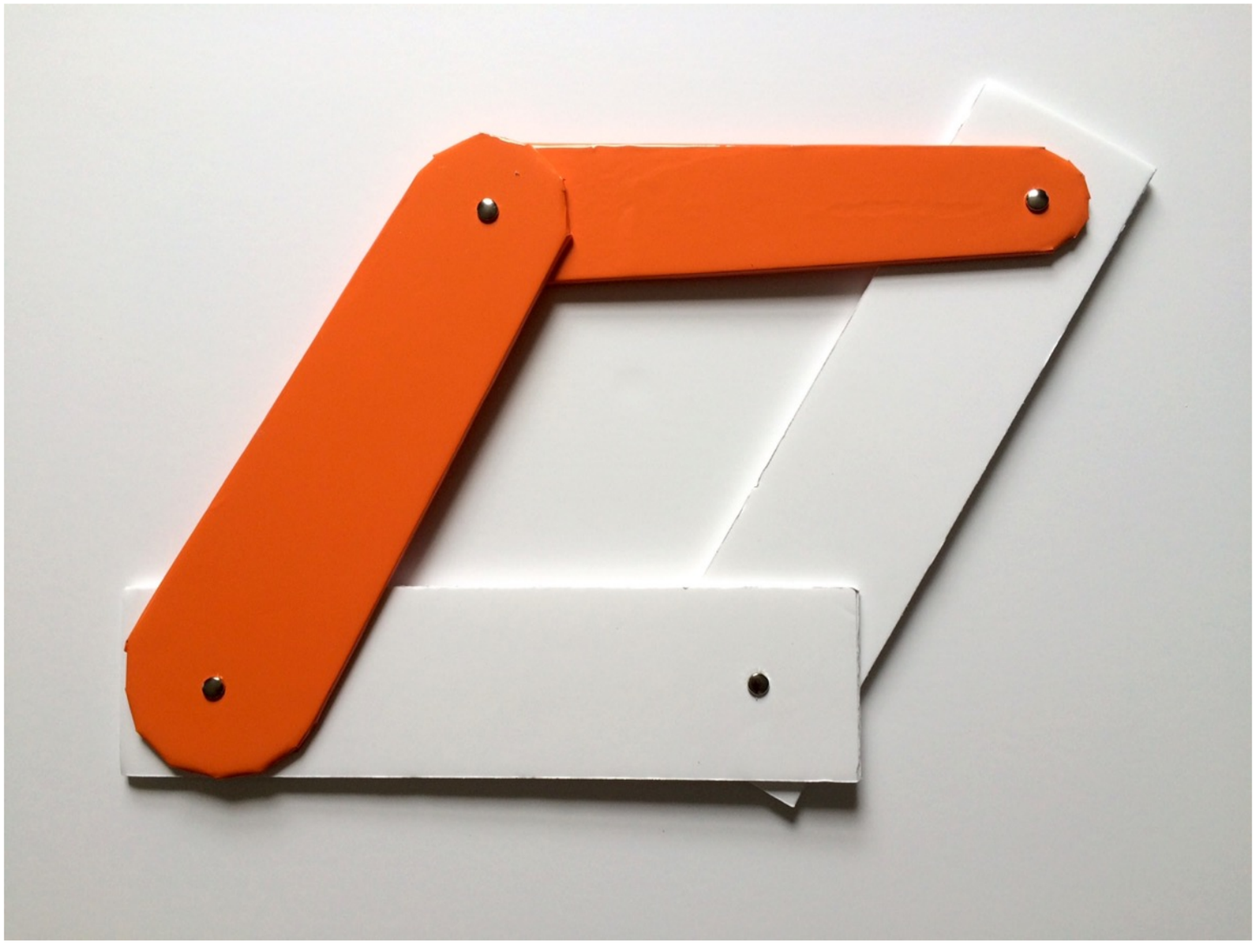
```
a = 45
b = -2*a

arm(a)
arm(b)

from math import *

w = cos(radians(a))*length
reach = 2*w
print(reach)

penup()
goto(reach,0)
pendown()
dot(10)

done()
```



Check the results by placing a **dot** at the predicted position.

Ln: 20 Col: 11

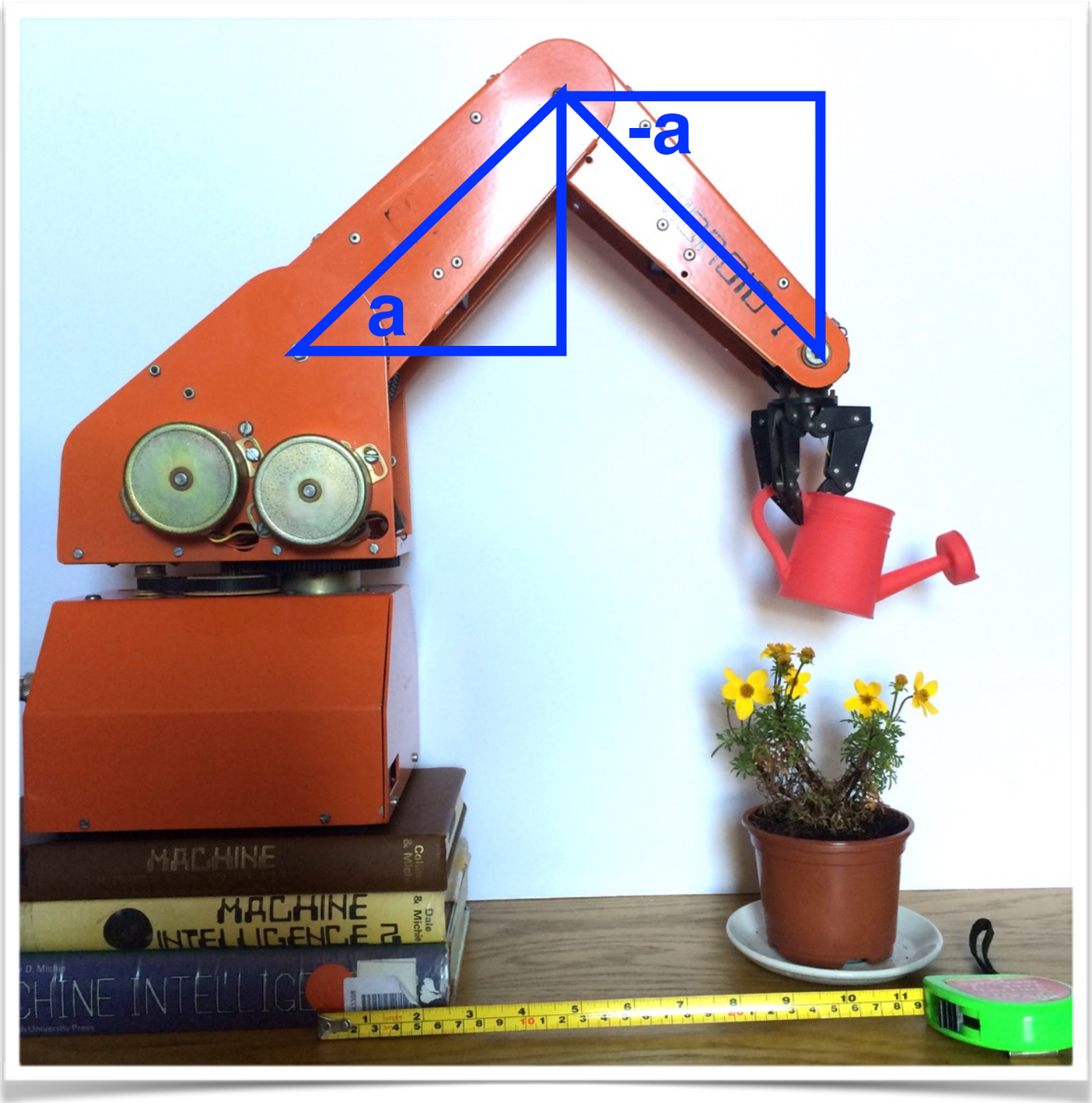
Parallelograms

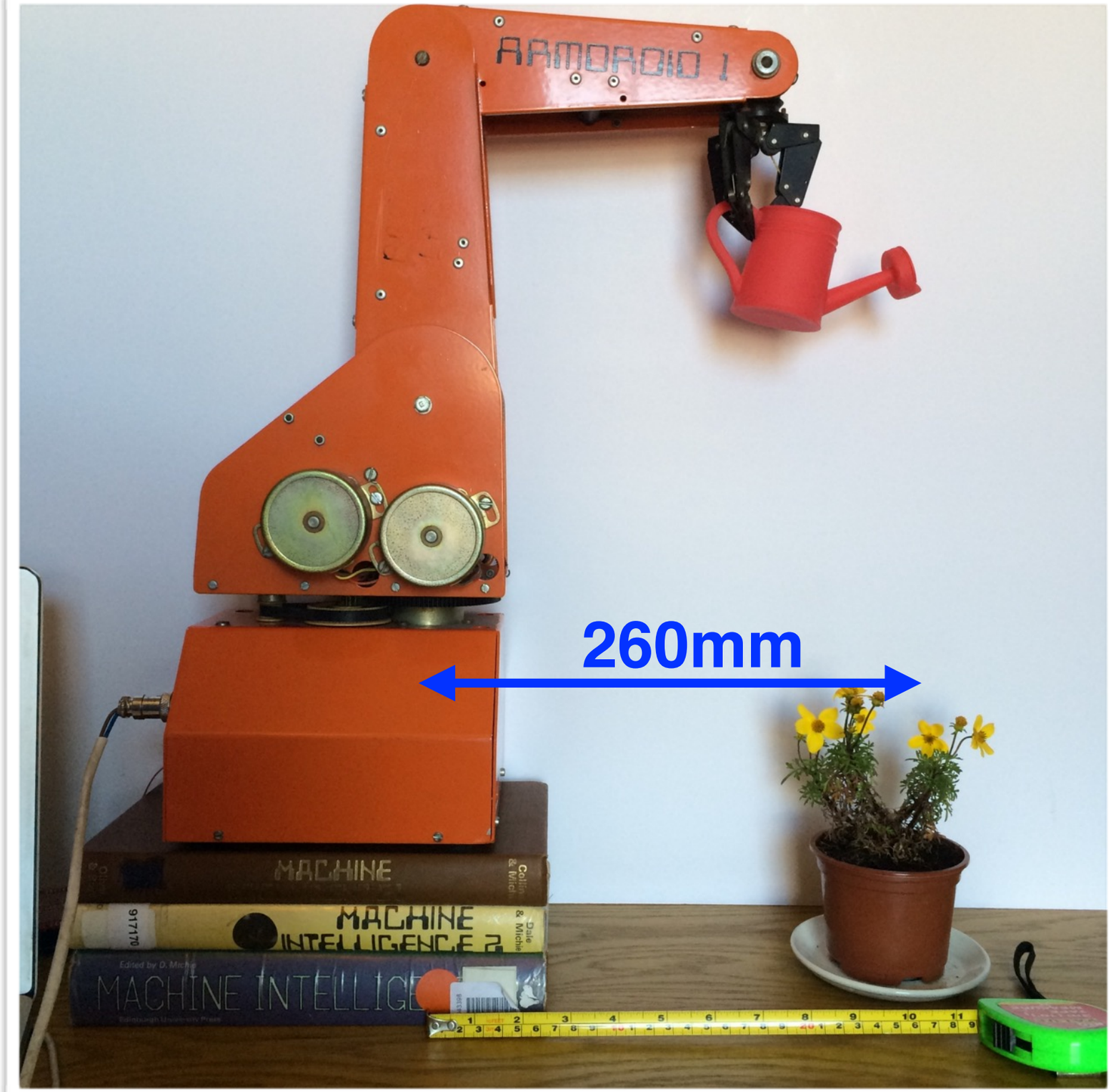


- The **Armdroid** has pulleys so that the forearm maintains its angle.

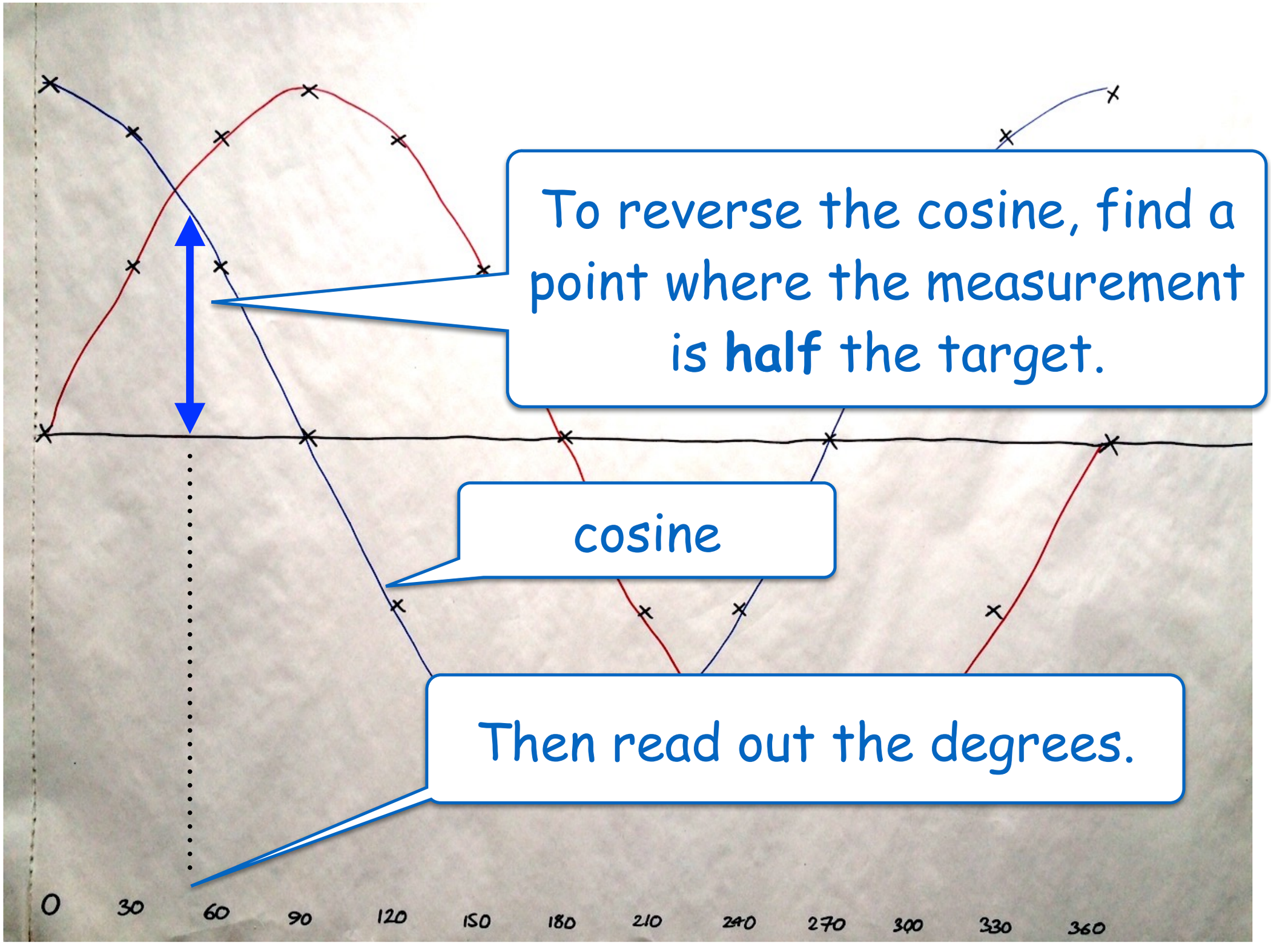
Inverse Kinematics

- If we know where the flower is, how do we work out the angle **a**?




ARMDROID 1
260mm
MACHINE
MACHINE
INTELLIGENCE 2
91710
Edited by D. Michie
MACHINE INTELLIGE

Analogue computer



Going Digital

inverse.py - /Users/Steve/Documents/Python/inverse.py (3.4.3)

```
from turtle import *
from math import *

length = 190
target = 260

def arm(angle):
    left(angle)
    forward(length)

speed('slow')

a = degrees(acos(0.5*target/length))
b = -2*a
print(a)

arm(a)
arm(b)

penup()
goto(target,0)
pendown()
dot(10)

done()
```

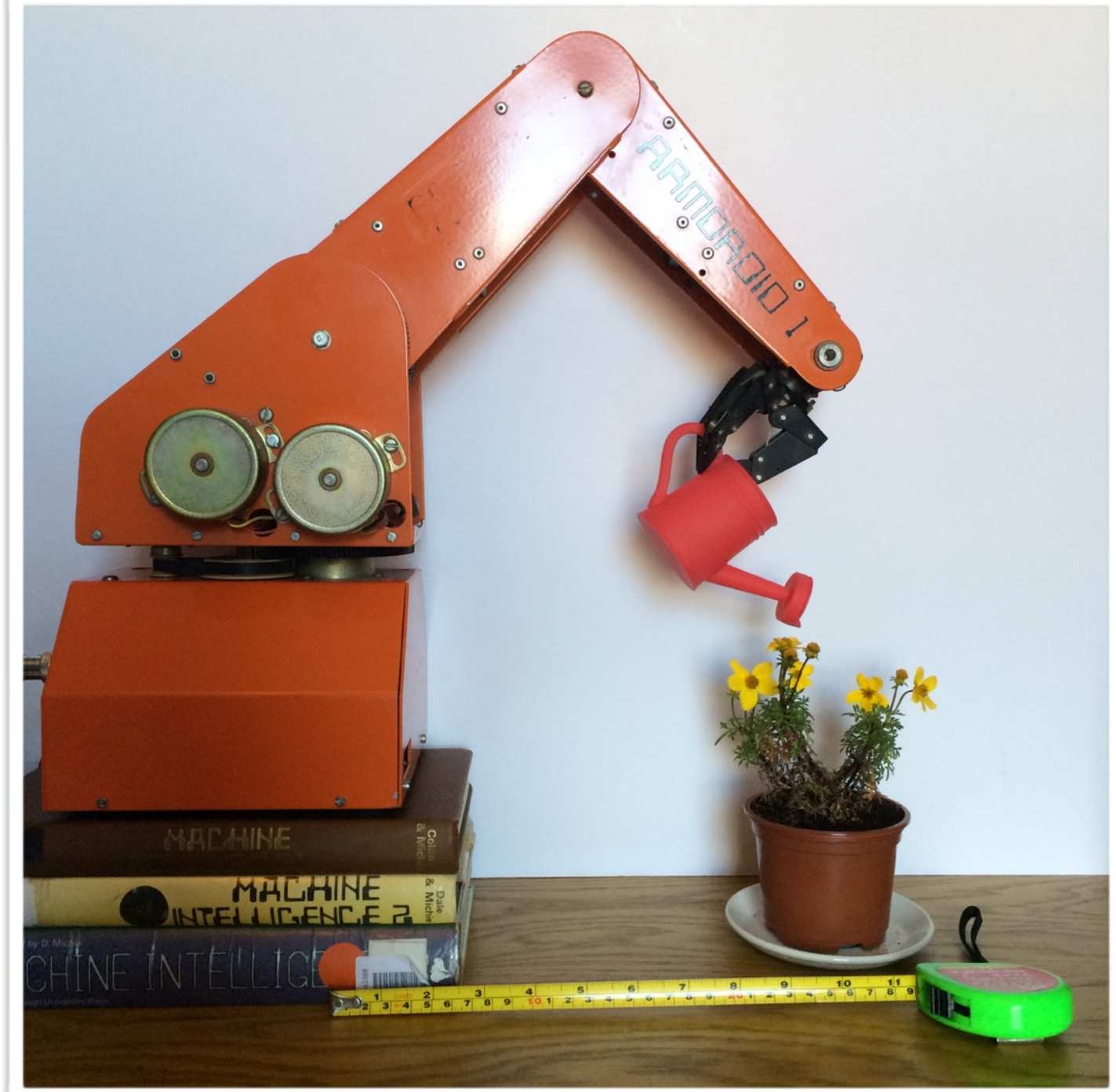
Ln: 8 Col: 15

The flower is 260mm away

Use the inverse (arc) cosine to calculate **a**

Convert back from radians to degrees

Place a dot at the target



Maintenance Drones are eco-friendly.