

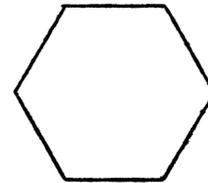
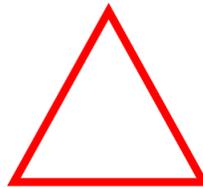
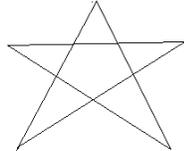
Chapter 2

My First Program

Who Likes To Draw?

Scribblers do! Note that your Scribbler has a hole in the top of it for a pen. You can probably imagine that you control this pen using the function we learned last week, `gamepad()`.

But this time you are not going to control your robot, your robot is going to control itself! In today's class, you will learn to program your robot to draw a square and a star. If you finish early, maybe you can think of another cool thing it can draw (your initials, a hexagon, etc.)



What is a program anyway?

A program is a set of instructions written in a programming language. The computer will follow the instructions *exactly*. Computers might be really fast, but they are surprisingly stupid; they only can do what they are told!

Here is a sample Python program:

```
from myro import *
def yoyo():
    forward(1, 1)
    backward(1, 1)
```

How does this program work? The first line, like we learned last time, lets you use all the robot commands possible.

The second line "`def yoyo() :`" tells the computer that the name of this program is Yoyo.

The final 2 lines, `forward(1,1)` and `backward(1,1)` are examples of robot commands. These functions tell the robot to move forward and backward. They work the same way. The 1st number in the parentheses is the speed of the robot, the 2nd is how many seconds you want it to travel. So they really work like this:

```
forward(SPEED, SECONDS)
backward(SPEED, SECONDS)
```

Now that you know this information, what do you think this program will make the robot do?

Programming People

Here are a few other functions that are helpful for this class:

```
turnLeft(SPEED, SECONDS)
turnRight(SPEED, SECONDS)
```

Everybody find a partner! One of you will be the robot and the other will be the programmer. Once you decide who is who, have the programmer, *only* using the commands `forward`, `backward`, `turnLeft`, and `turnRight`,

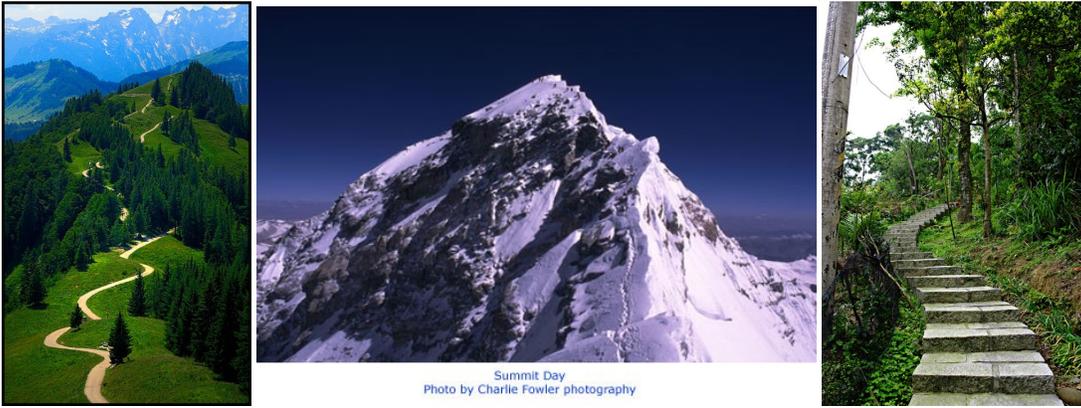
and stop, give your partner *instructions* on how to move in a triangle shape. The robot partner should act this out!

The Python Translation

Okay! Now that you have done the Triangle experiment in person, what would the code on the robot be?

```
from myro import *
def triangle():
    forward(1,1)
    turrrnLeft(1, .4)
    forward(1,1)
    turrrnLeft(1, .4)
    forward(1,1)
```

Fun Fact!



The journey is more important than the destination! Your robot, because of things like friction and battery power, will rarely draw the same shape exactly twice, even if you don't change your code. Don't worry if your shape isn't perfect, it is the code and your thought process that matters most!

Your First Program!

Now that you have looked at several examples of programs, and learned to think like a programmer by giving instructions to your robot friend, it is time to make your first program. Using Python, write a program to have your robot draw a square. If you are successful, go ahead and try drawing a star. Here are a few tips that might help you.

- Plan it out! How would *you* teach someone to draw a square who doesn't know how?
- Keep it simple! What is the easiest way to draw a square? Try writing down several ways to determine which is easiest.
- What worked best during the exercise with your partner? Where did they get confused? Why?
- Remember, your robot can only understand commands like the ones listed above.
- The robot will only do what you tell it to do. If it does something unexpected, go back and read your code. Did you write something you didn't mean to?
- Ask questions! Your instructor around to help you as much as you need it!

Once you have finished your first robot drawings, take them home and hang them on your wall! Be proud of yourself, because you just wrote your first python program!