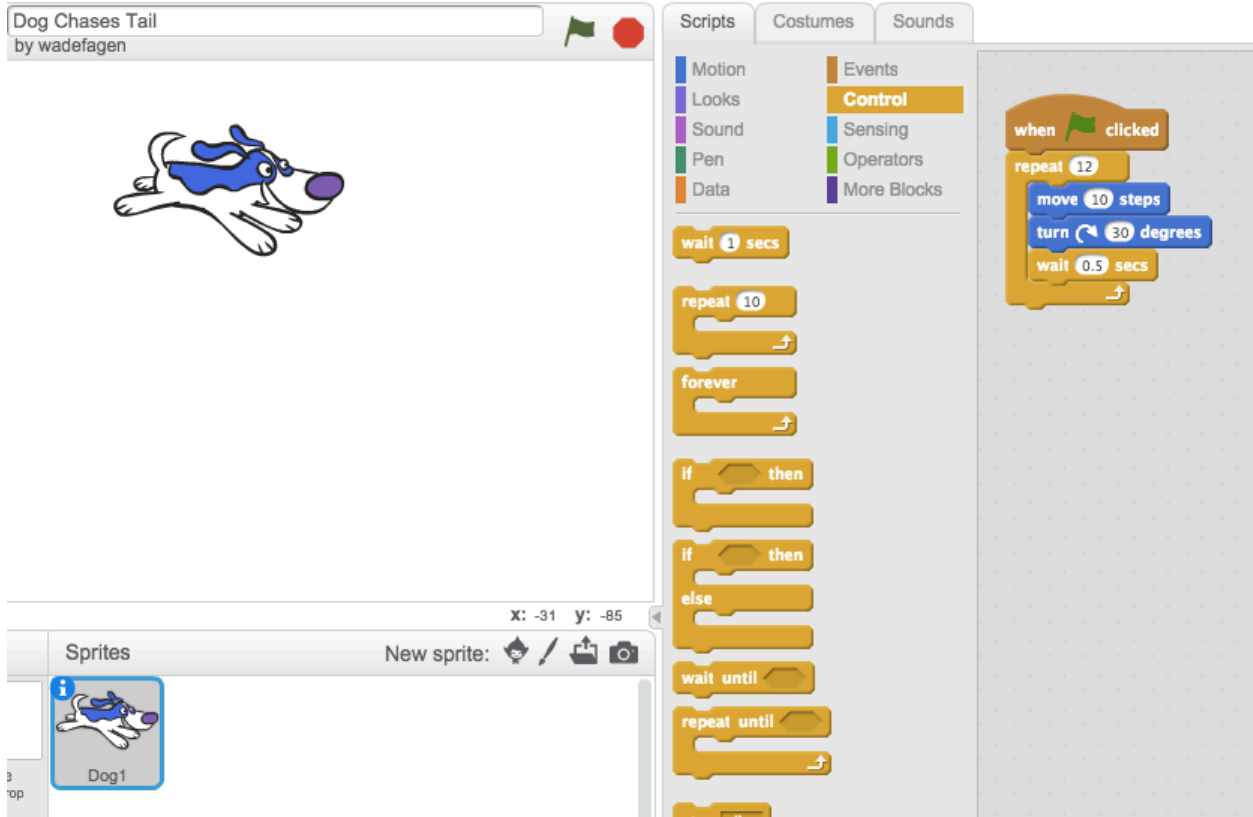


Binary: Everything in computer is based on binary, a system using only zeros and ones to present numbers. For example, for a 3-bit system, we can write 1-7 in the following way:

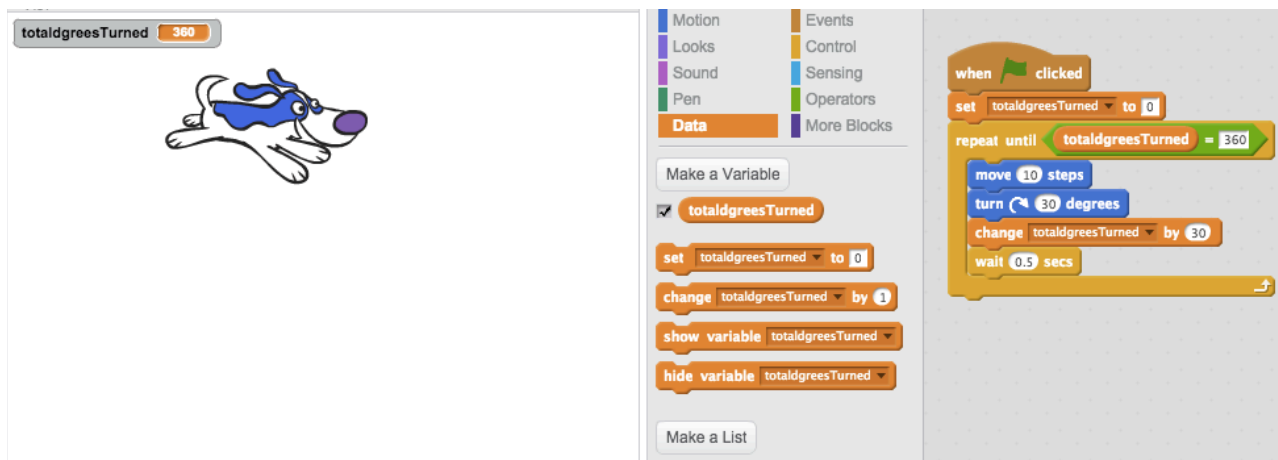
0- 000	1- 001	2- 010	3- 011
4- 100	5- 101	6- 110	7- 111

Loops: (example: Dog chases tail)

1. If we want the little dog chases his tail by running a circle (360°), we do not have to write a lot of lines. Instead, we can let him run and turn 30 degrees each time, and repeat that for 12 times.



2. Also, the condition for the loop can be repeating until the dog has turned 360 degrees. We can create a variable named “totaldegreesTurned”, set it to be 0 at first, and then add 30 every time the dog turns 30 degrees. When totaldegreesTurned=360, the dog stop.



Functions: allow us to group code for easy use.

If we want the dog stop once he catches the ball, we can write the following code:

The image shows the Scratch development environment. On the left, a dog sprite is positioned near an orange ball. The 'Sprites' panel at the bottom left shows 'Dog1' and 'Ball'. The code area on the right contains the following script:

```

when clicked
repeat until touching Ball?
  move 10 steps
  turn 30 degrees
  wait 0.5 secs
  
```

And we can also do it in this way: **DEFINE** a function name “turnDog”, so the dog run and turn 30 degrees for a single movement. Then we **CALL** this function in our main program, which makes the dog repeat that movement until catch the ball. It’s only 3 lines!

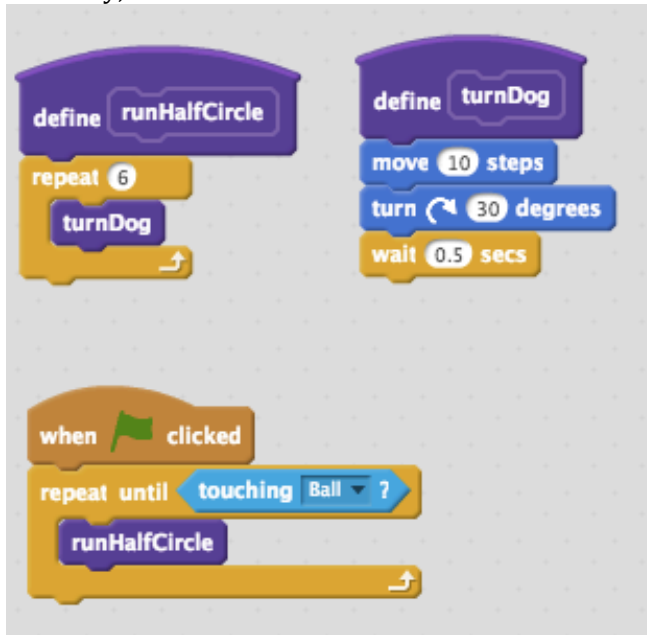
The image shows the Scratch code for defining and calling a function. The code is as follows:

```

define turnDog
  move 10 steps
  turn 30 degrees
  wait 0.5 secs

when clicked
repeat until touching Ball?
  turnDog
  
```

Similarly, we have:



Lab 1: Underwater Maze

Similar to what we learn on class, define “moveForward”, “turnLeft” and other functions, try to call them in your main program helping the fish out!

