
COMP 102: Computers and Computing

Lecture 6: Introduction to Scripting

Instructor: Joelle Pineau (jpineau@cs.mcgill.ca)

Class web page: www.cs.mcgill.ca/~jpineau/comp102

Programming basics

- Difference between a programming language and a program.
- Understanding the need to be very precise when giving instructions to a computer.
- Notion of variables, naming conventions, variable types.
- Mathematical operators, comparison operators.
- Loops and functions (more on this today).

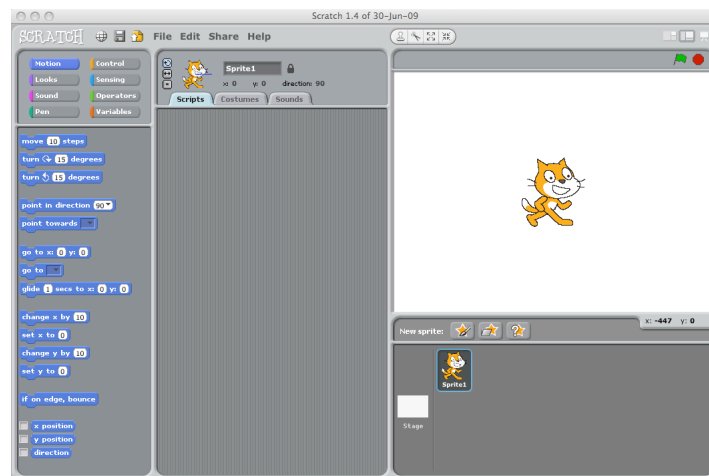
Scratch

- Developed by the “Lifelong Kindergarten” group at MIT.
 - Homepage: <http://scratch.mit.edu/>
 - For online help: <http://info.scratch.mit.edu/Support>
 - To download: <http://scratch.mit.edu/download>

NOTE: You do not have to give out any of your personal info if you don't want to. When you get to the registration screen, just click on the “Continue to Scratch Download” button at the bottom of the screen.

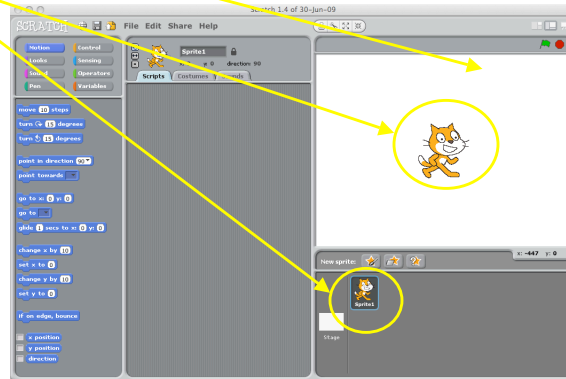
- Scratch allows users to write programs by dragging together blocks.

Programming Window



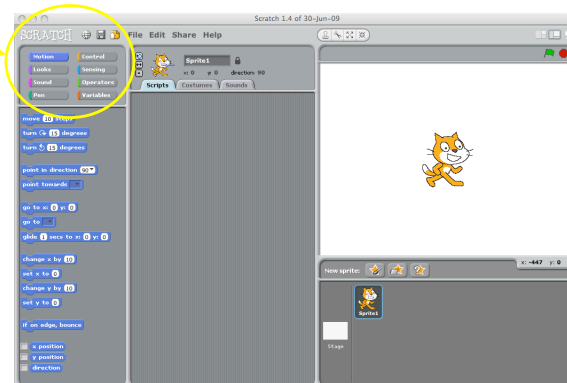
Things: Performance area

- Stage (background)
- Sprites (objects)
 - Scripts (behaviors)
 - Costumes (appearance)
 - Sounds (available sounds)

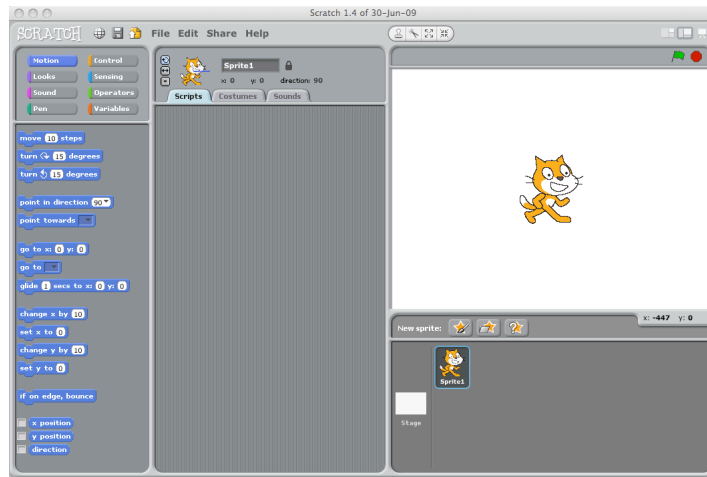


More things






- Run / Stop
- Script inventory
 - Motion
 - Looks
 - Sound
 - Pen
 - Control
 - Sensing
 - Numbers
 - Variables



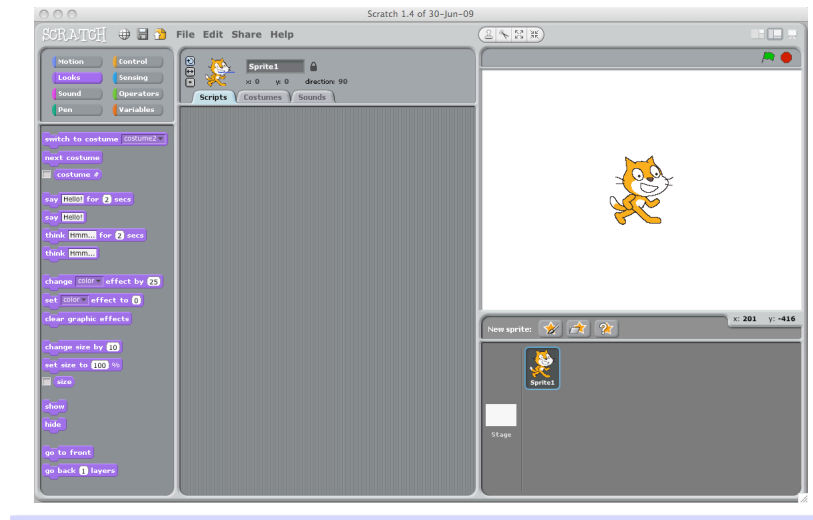
Motion Inventory



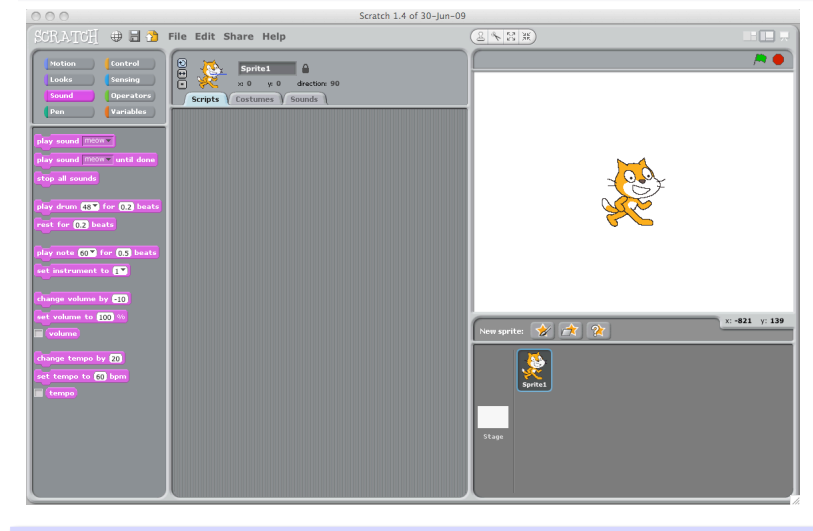
Shapes

- Trigger 
- Statement 
- Ending statement 
- Boolean value 
- Numeric value 

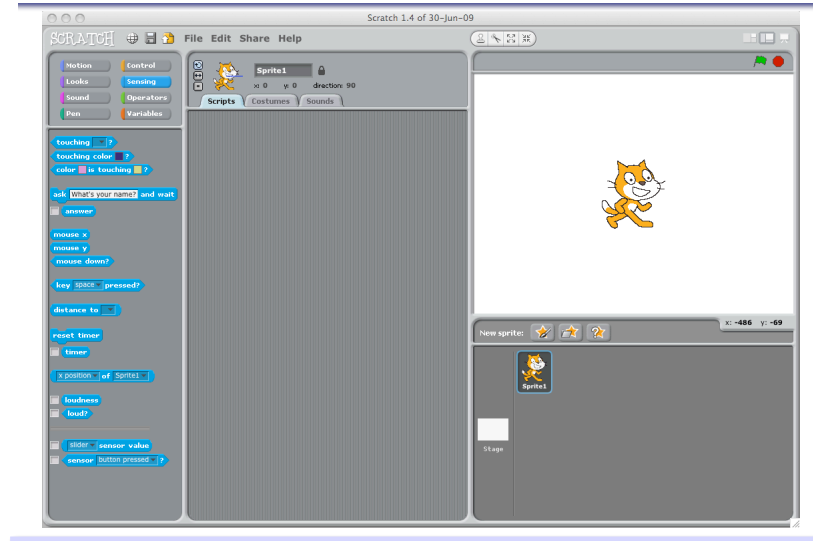
Looks Inventory



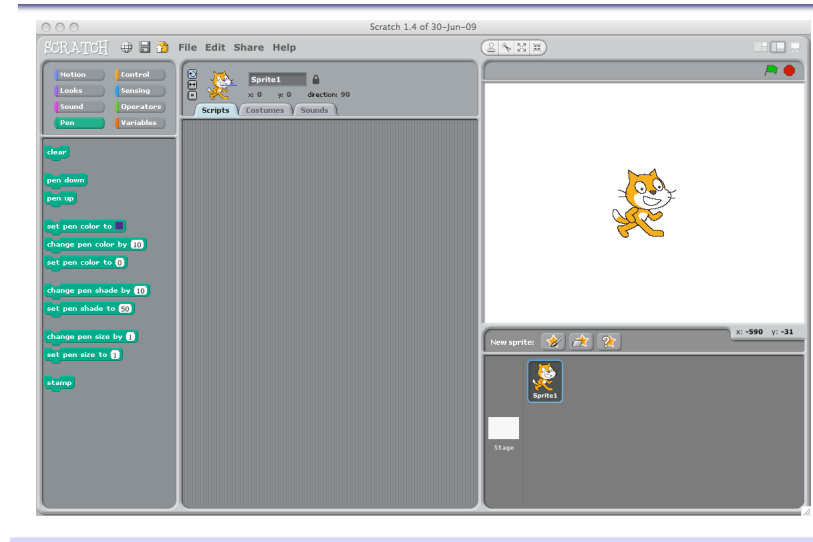
Sound Inventory



Sensing Inventory

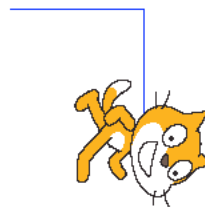


Pen Inventory



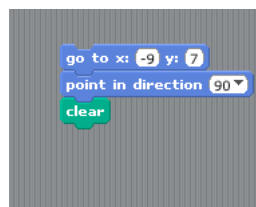
Drawing Commands

- To build a program, drag components from the menus to the workspace.
 - You can choose from any of the sub-menus (Motion, Pen, Control, ...)
- To run your program, simply click on the code.

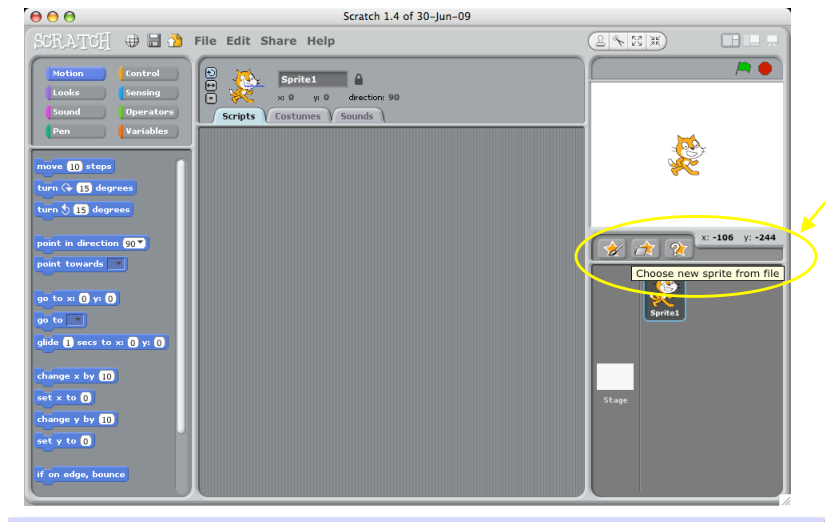


Resetting the Sprite

- To reset position of the Sprite, use commands from menu.



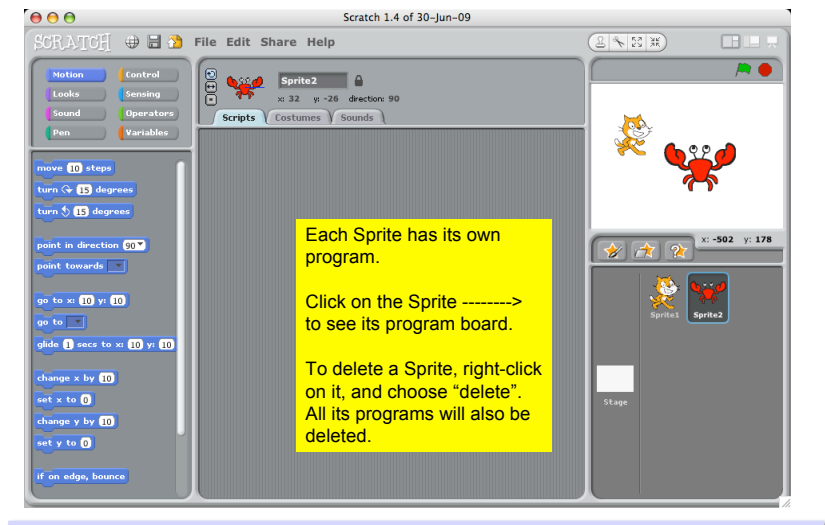
Creating new Sprites



15

Joelle Pineau

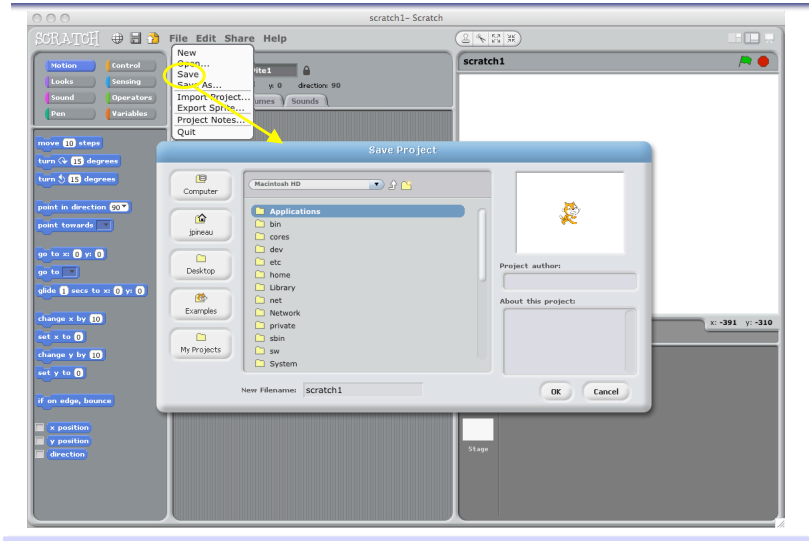
Creating new Sprites



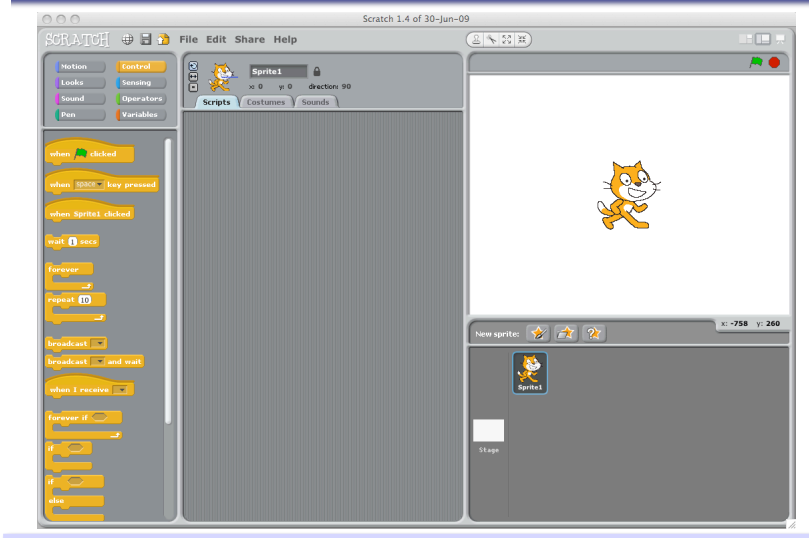
16

Joelle Pineau

Saving your project

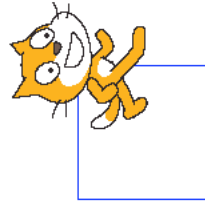


Control Inventory



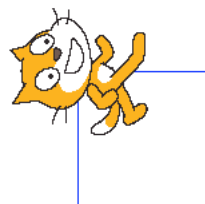
Creating a new command: “square”

```
when I receive square  
pen down  
move 100 steps  
turn 90 degrees  
move 100 steps  
turn 90 degrees  
move 100 steps  
turn 90 degrees  
move 100 steps
```

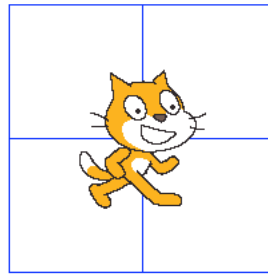
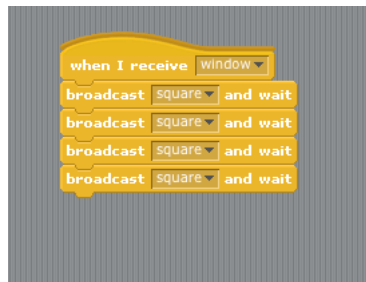


Calling this command

```
broadcast square and wait
```



Use your command!

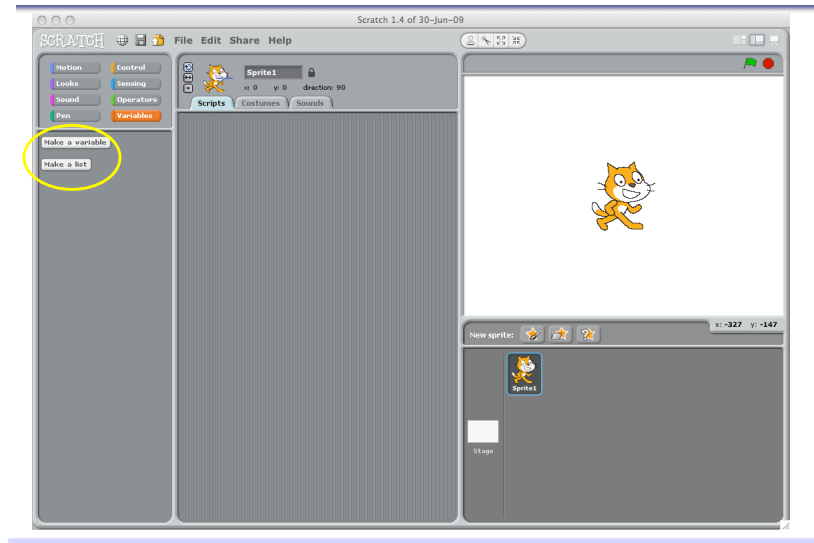


Using variables

- **Declare** a variable called **length**.
- **Assign** is a value of **100**.
- **Access** this variable whenever you **move**.



Variables Inventory

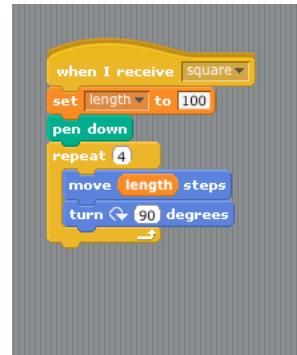


Variable Types in Scratch

- Don't need to specify variable type in Scratch.
 - Decision by designers of the language, to keep things simple.
- How can the machine interpret variable (e.g. know how many bytes to read)?
 - Type of variable is completely determined by the context.
- Why don't other languages do this?

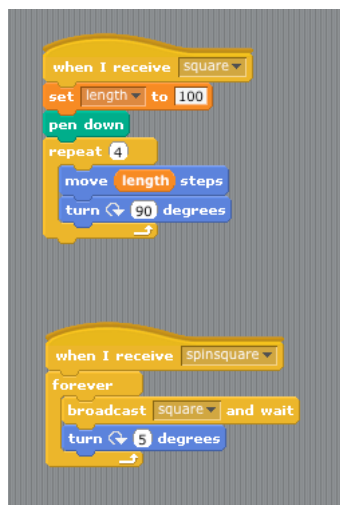
Loops

- Repeat the **move** command 4 times.
- Same output. Shorter program.



```
when I receive square
set length to 100
pen down
repeat 4
  move length steps
  turn 90 degrees
```

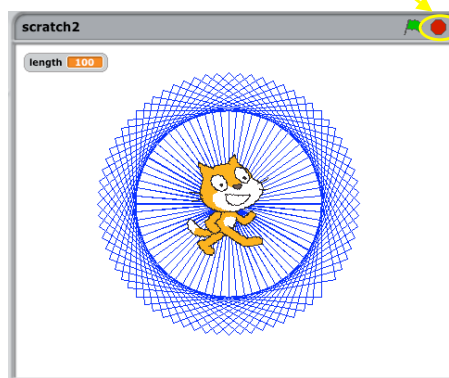
Infinite looping



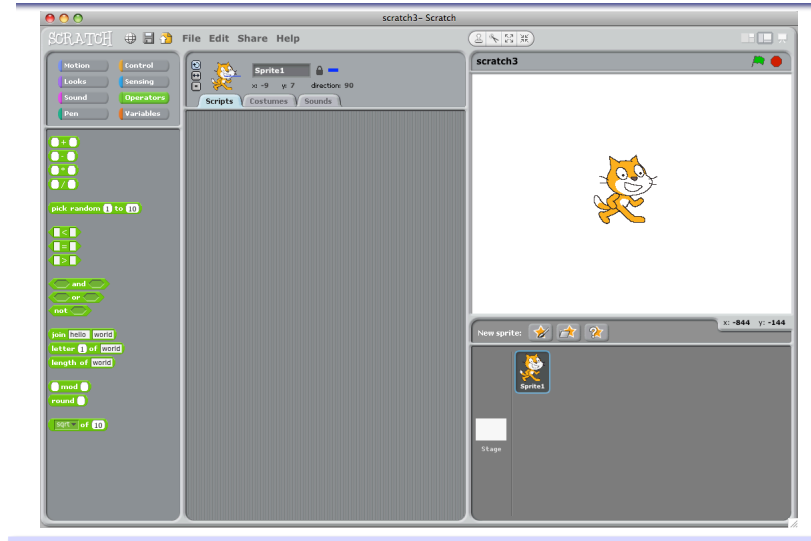
```
when I receive square
set length to 100
pen down
repeat 4
  move length steps
  turn 90 degrees

when I receive spinspace
forever
  broadcast square and wait
  turn 5 degrees
```

To stop, hit the **Stop** button.

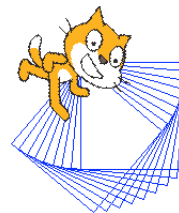
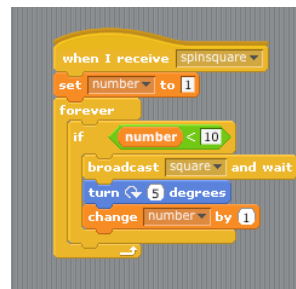


Operators Inventory



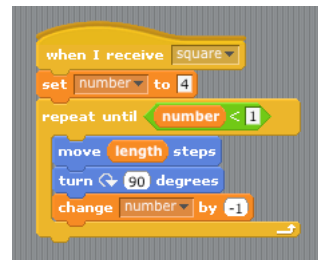
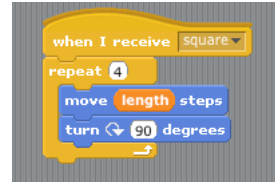
Conditionals

- Denoted by **if** keyword.
- Block of code inside the conditional executes only if **condition is true**.
- Condition expressed as a **logical expression**: `number < 10`
- Condition depends on the changing value of the **number** variable.

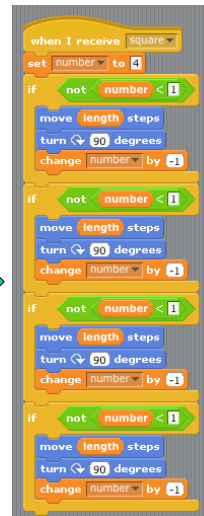
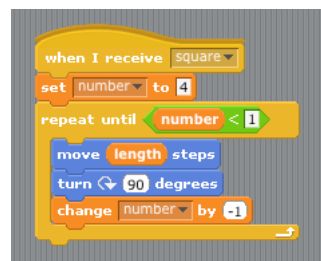


Kinds of Loops

- **For** loop:
 - Repeat a fixed number of times.
 - Flexible
- **While** loop (called “repeat until” in Scratch):
 - Repeat while a (boolean) condition is true.
 - More Flexible



Unrolling, With Conditions



Recursion

Idea: A function that repeats itself in a self-similar way.

First try:

```
set length to 100
set number to 4
broadcast line

when I receive line
  move length steps
  turn 90 degrees
  change length by -1
  broadcast line
```

(infinite recursion)

Try again!

```
set length to 100
set number to 4
broadcast line

when I receive line
  if not number < 1
    move length steps
    turn 90 degrees
    change number by -1
    broadcast line
```

(finite recursion)



Take-home message

- Have fun with Scratch!