



SCRATCH

Lesson One – Intro to Scratch, Using the Control, Motion, Looks, and Pen Commands

WHAT IS SCRATCH?



National Science Foundation
WHERE DISCOVERIES BEGIN

Microsoft



MACARTHUR
The John D. and Catherine T. MacArthur Foundation

Google

iomega
an EMC company

- Scratch is a programming language that makes it easy to create your own interactive stories, animations, games, music, and art -- and share your creations on the web.
- As young people create and share Scratch projects, they learn important mathematical and computational ideas, while also learning to think creatively, reason systematically, and work collaboratively.
- Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab, with financial support from the National Science Foundation, Microsoft, Intel Foundation, MacArthur Foundation, Google, Iomega and MIT Media Lab research consortia.



WHAT IS SCRATCH?

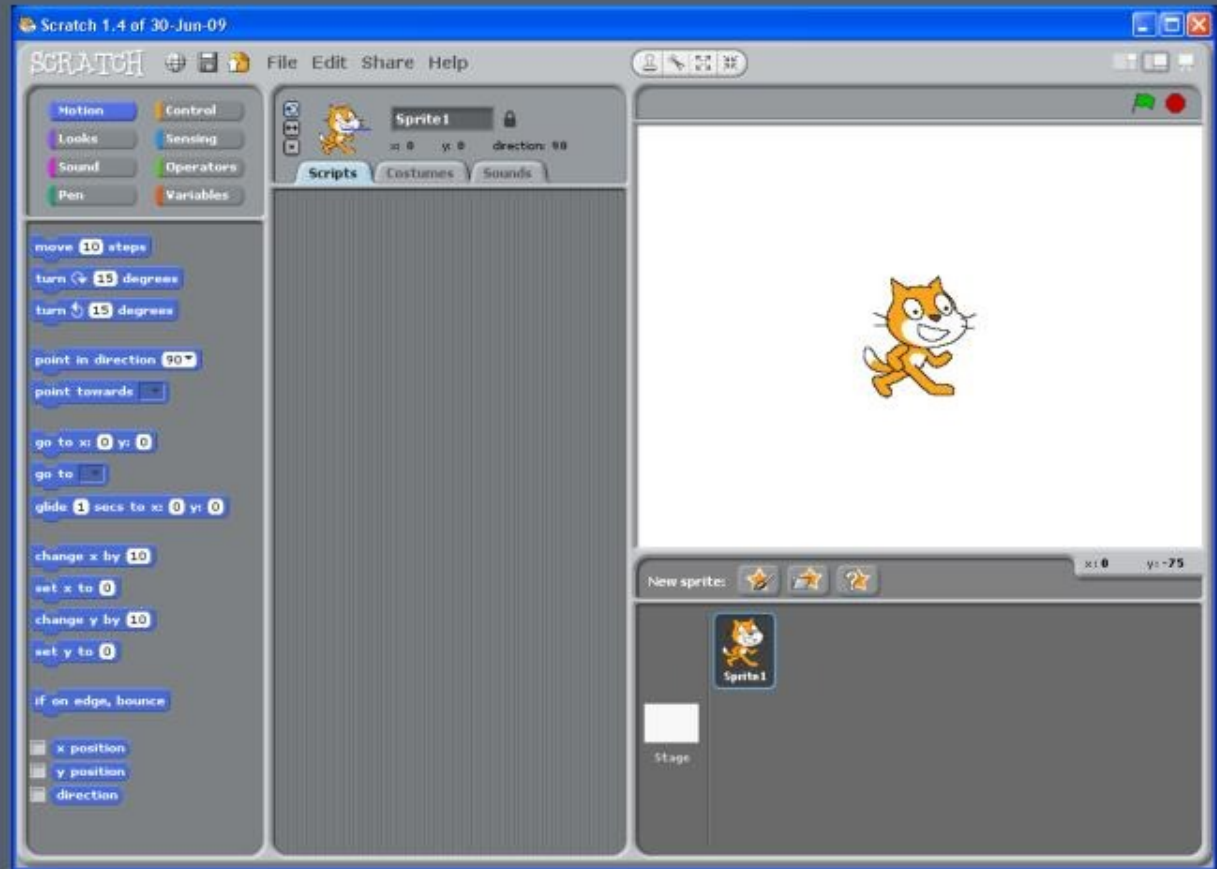


National Science Foundation
WHERE DISCOVERIES BEGIN



- Bottom line...
- ***SCRATCH IS FUN AND EDUCATIONAL***
- *...appropriate for ALL ages, and is even being used in a new college Intro To CS course at several universities around the nation.*
- Now let's get started...





PROGRAMMING ENVIRONMENT

The SCRATCH window



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables



Sprite 1



x: 0

y: 0

direction: 90

Scripts

Costumes

Sounds

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90

point towards

go to x: 0 y: 0

go to

glide 1 secs to x: 0 y: 0

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

☐ x position☐ y position☐ direction

This is the Scratch window,
or programming
environment.

*Say "Hello" to Scratch,
the Cat!*



New sprite:



x: 0

y: -75



Sprite 1



Stage



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables



Sprite 1



x: 0

y: 0

direction: 90

Scripts

Costumes

Sounds

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90

point towards

go to x: 0 y: 0

go to

glide 1 secs to x: 0 y: 0

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

☐ x position☐ y position☐ direction

Right now, go ahead and open a Scratch window on your computer and follow along as we explore this exciting programming tool!



New sprite:



x: 0

y: -75



Sprite 1



Stage



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables



Sprite 1



x: 0

y: 0

direction: 90

Scripts

Costumes

Sounds

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90

point towards

go to x: 0 y: 0

go to

glide 1 secs to x: 0 y: 0

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

x position

y position

direction

There are four main sections...

- The Command Center



New sprite:



x: 0

y: -75



Sprite 1



Stage



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables



Sprite 1



x: 0

y: 0

direction: 90

Scripts

Costumes

Sounds

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90

point towards

go to x: 0 y: 0

go to

glide 1 secs to x: 0 y: 0

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

x position

y position

direction

There are four main sections...

- The Programming Area



New sprite:



x: 0

y: -75



Sprite 1



Stage



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables



Sprite 1



x: 0

y: 0

direction: 90

Scripts

Costumes

Sounds

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90

point towards

go to x: 0 y: 0

go to

glide 1 secs to x: 0 y: 0

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

☐ x position☐ y position☐ direction

There are four main
sections...

- The Costume Design Shop

New sprite:



x: 0

y: -75



Sprite 1



Stage



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables



Sprite 1



x: 0

y: 0

direction: 90

Scripts

Costumes

Sounds

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90

point towards

go to x: 0 y: 0

go to

glide 1 secs to x: 0 y: 0

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

☐ x position☐ y position☐ direction

There are four main
sections...

•And the Display Window



New sprite:



x: 0

y: -75



Sprite 1



Stage



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables



Sprite 1



x: 0

y: 0

direction: 90

Scripts

Costumes

Sounds

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction

point towards

go to x: 0 y: 0

go to

glide 1 secs to x:

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

x position

y position

direction

At the top of the programming area, the current status of the cat “**sprite**” is shown...

- Its position on the screen is x: 0 y: 0, the very center of the screen, the HOME position, just like the origin in a coordinate plane graph in geometry.
- The cat is facing 90 degrees, which you can tell in two ways...it says direction: 90, and there is a small blue line on the sprite that points in the 90 degree direction (to the right).



x: 0

y: -75

Stage



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables



Sprite 1



x: 0

y: 0

direction: 90

Scripts

Costumes

Sounds

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90

point towards

go to x: 0 y: 0

go to

glide 1 secs to x: 0 y: 0

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

☐ x position☐ y position☐ direction

Let's look at the commands
and write a very simple
program to get started...



New sprite:



x: 0

y: -75



Sprite 1



Stage

SCRATCH



File Edit Share Help



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables



Sprite 1



x: 0

y: 0

direction: 90

Scripts

Costumes

Sounds

when green flag clicked

when space key pressed

when Sprite 1 clicked

wait 1 secs

forever

repeat 10

broadcast

broadcast and wait

when I receive

forever if

if

if

First click on the
CONTROL button to switch
command sets.
You always start here.



New sprite:



x: -57

y: -266



Sprite 1



Stage



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables



Sprite1



x: 0

y: 0

direction: 90

Scripts

Costumes

Sounds

when  clickedwhen  key pressed

when Sprite1 clicked

wait 1 secs

forever

repeat 10

broadcast

broadcast and wait

when I receive

forever if

if

if

when  clicked

Click , drag and drop the
“GREEN FLAG HAT”
anywhere into the
programming area. A
Scratch program always
begins with one of the four
available hats. For now we
will always start a program
with the “green flag hat”.



x: 199 y: -267

SCRATCH



File Edit Share Help



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90

point towards

go to x: 0 y: 0

go to

glide 1 secs to x: 0 y: 0

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

x position

y position

direction



Sprite1



x: 0

y: 0

direction: 90

Scripts

Costumes

Sounds

when clicked

move 10 steps



x: -392 y: -12

Now switch to the Motion Command Set, click and drag the "move 10 steps" command underneath the green flag hat until you see a white line appear, then drop it.

SCRATCH



File Edit Share Help



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90

point towards

go to x: 0 y: 0

go to

glide 1 secs to x: 0 y: 0

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

x position

y position

direction



Sprite 1



x: 0

y: 0

direction: 90

Scripts

Costumes

Sounds

when clicked

move 10 steps

Notice how it “snaps” into place at the notch.
This is how you add a command to a program.

Now click on the green flag icon at the top right to run your program.



x: -142 y: -124



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90

point towards

go to x: 0 y: 0

go to

glide 1 secs to x: 0 y: 0

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

x position

y position

direction



Sprite 1



x: 10

y: 0

direction: 90

Scripts

Costumes

Sounds

when green flag clicked

move 10 steps

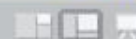
***Your cat moved!...ten steps!***

You can tell by watching the display window, or by looking at the coordinates at the top...the x value is now 10 instead of 0.

Click the green flag several more times and watch your cat move even farther.

x: 10

y: 376



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90

point towards

go to x: 0 y: 0

go to

glide 1 secs to x: 0 y: 0

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

x position

y position

direction



Sprite 1



x: 100

y: 0

direction: 90

Scripts

Costumes

Sounds

when clicked

move 10 steps



The cat is now in position (100,0), which means the green flag was clicked nine more times.

x: 391 y: 208

Stage



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables



Sprite1



x: 100 y: 0 direction: 90

Scripts

Costumes

Sounds

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90

point towards

go to x: 0 y: 0

go to

glide 1 secs to x: 0 y: 0

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

x position

y position

direction

when clicked

move 10 steps

go to x: 0 y: 0



Now, let's make **one program** to cause the cat to move 100 steps, ten steps at a time.

First we need to reset the cat's position to **HOME**.

Click and drag the go to x: 0 y: 0 command and insert it **below** the hat, but **above** the move command.

SCRATCH



File Edit Share Help



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90

point towards

go to x: 0 y: 0

go to

glide 1 secs to x: 0 y: 0

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

x position

y position

direction



Sprite 1



x: 100 y: 0

direction: 90

Scripts

Costumes

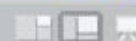
Sounds

when clicked

move 10 go to x: 0 y: 0



Remember to move it slowly into place, watching for the white line to appear where you want to "drop" it.



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables



Sprite 1



x: 100 y: 0 direction: 90

Scripts

Costumes

Sounds

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90

point towards

go to x: 0 y: 0

go to

glide 1 secs to x: 0 y: 0

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

☐ x position☐ y position☐ direction

when clicked

go to x: 0 y: 0

move 10 steps

move 10 steps

move 10 steps

move 10 steps

move 10 steps

move 10 steps

move 10 steps

move 10 steps

move 10 steps

move 10 steps



New sprite:



x: 220 y: 168

Now, add nine more move commands to your program to make a total of ten move 10 steps commands.

Then run your program to see the result.

Control

Sensing

Operators

Variables

turn 15 degrees

point towards go to

```
set x to 0
```

```
set v to 0
```

if on edge, bounce

y position

 direction


```
xi: 100  yi: 0  direction: 90
```

Scripts

Costumes

Sounds

when clicked

go to x: 0 y: 0

move 10 steps

```
move 10 steps
```

move 10 steps

move 10 steps

move 10 steps

move 10 steps

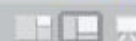
move 10 steps

move 10 steps

move 10 steps

move 10 steps

See if you can find this command, and then add it to your program in the appropriate location(s). Think about it and try it yourself before you look at the next slide!



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables



Sprite 1



x: 100 y: 0

direction: 90

Scripts

Costumes

Sounds

when green flag clicked

when space key pressed

when Sprite 1 clicked

wait 1 secs

forever

repeat 10

broadcast

broadcast and wait

when I receive

forever if

if

if

when green flag clicked

go to x: 0 y: 0

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs



New sprite:



x: 265 y: 139

If you did the right thing, you found the **wait** command in the **CONTROL** group, and inserted it after each **move** command, as shown.



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables

when green flag clicked

when space key pressed

when Sprite 1 clicked

wait 1 secs

forever

repeat 10

broadcast

broadcast and wait

when I receive

forever if

if

if



Sprite 1



x: 100

y: 0

direction: 90

Scripts

Costumes

Sounds

when green flag clicked

go to x: 0 y: 0

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs



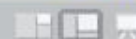
New sprite:



x: 265 y: 139

Now when you run the program, you can actually see the animation. It is a bit slow, but we can change that by using a decimal value smaller than 1.

But let's wait on that...there is a really cool thing we're about to learn, called a **loop**.



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables



Sprite 1



x: 100 y: 0

direction: 90

Scripts

Costumes

Sounds

when green flag clicked

when space key pressed

when Sprite 1 clicked

wait 1 secs

forever

repeat 10

broadcast

broadcast and wait

when I receive

forever if

if

if

when green flag clicked

go to x: 0 y: 0

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs

move 10 steps

wait 1 secs



New sprite:



x: 687 y: 362

First of all, remove all but the first four blocks from the program. To do this, just **click and drag down** the fifth block...all of the others will come with it.

Then drag it back to the command area where it will just “disappear”!



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables



Sprite 1



x: 100 y: 0

direction: 90

Scripts

Costumes

Sounds

when green flag clicked

when space key pressed

when Sprite 1 clicked

wait 1 secs

forever

repeat 10

broadcast

broadcast and wait

when I receive

forever if

if

if

when green flag clicked

go to x: 0 y: 0

move 10 steps

wait 1 secs

repeat 10



Poof! The commands are all gone!

Now find the repeat 10 command and drag it over until it "surrounds" the bottom two command blocks. This command is called a loop.

It will cause everything it surrounds to repeat 10 times.



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables

when green flag clicked

when space key pressed

when Sprite1 clicked

wait 1 secs

forever

repeat 10

broadcast

broadcast and wait

when I receive

forever if

if

if



Sprite1



x: 100 y: 0

direction: 90

Scripts

Costumes

Sounds

when green flag clicked

go to x: 0 y: 0

repeat 10 10 steps

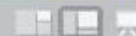
wait 1 secs



The white line needs to appear AFTER the go to x y command and BEFORE the move command, surrounding only the move and wait commands.

x: -493 y: -21

Stage



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables

when green flag clicked

when space key pressed

when Sprite 1 clicked

wait 1 secs

forever

repeat 10

broadcast

broadcast and wait

when I receive

forever if

if

if



Sprite 1



x: 100 y: 0 direction: 90

Scripts

Costumes

Sounds

when green flag clicked

go to x: 0 y: 0

repeat 10

move 10 steps

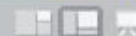
wait 1 secs



Now you have EXACTLY the same program as you did before, but in a much more “elegant” form.

The word “*elegant*” in computer programming means *well-crafted, efficient, short, sweet, and to the point*, as opposed to a “*brute force*” process, which is the way we did it at first.

x: 404 y: 187



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables

when clicked

when space key pressed

when Sprite1 clicked

wait 1 secs

forever

repeat 10

broadcast

broadcast and wait

when I receive

forever if

if

if



Sprite1



x: 100

y: 0

direction: 90

Scripts

Costumes

Sounds

when clicked

go to x: 0 y: 0

repeat 10

move 10 steps

wait 1 secs



These two design analysis concepts , *Elegance* and *brute force*, will be used throughout all of your computer programming career.

As you learn new techniques and refine your skills, hopefully your programming will become less “brute force” and more “*Elegant*”!



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables

when green flag clicked

when space key pressed

when Sprite1 clicked

wait 1 secs

forever

repeat 10

broadcast

broadcast and wait

when I receive

forever if

if

if



Sprite1



x: 100 y: 0 direction: 90

Scripts

Costumes

Sounds

when green flag clicked

go to x: 0 y: 0

repeat 10

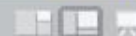
move 10 steps

wait 1 secs



Now let's change the wait time to something smaller than 1. Since it is inside the loop, you only have to change it once, instead of ten times in the previous, "brute force" example.

To change it, simply do a single click inside the white box to highlight the value.



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables

when green flag clicked

when space key pressed

when Sprite1 clicked

wait 1 secs

forever

repeat 10

broadcast

broadcast and wait

when I receive

forever if

if

if



Sprite1



x: 100 y: 0 direction: 90

Scripts

Costumes

Sounds

when green flag clicked

go to x: 0 y: 0

repeat 10

move 10 steps

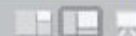
wait 0.5 secs



Then replace it with a new value, say 0.5, which will make the cat move twice as fast.

Try several values until you find a speed you like.

Now the animation is slow enough to be visible, and should look fairly realistic.



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables

when green flag clicked

when space key pressed

when Sprite1 clicked

wait 1 secs

forever

repeat 10

broadcast

broadcast and wait

when I receive

forever if

if

if



Sprite1



x: 100 y: 0 direction: 90

Scripts

Costumes

Sounds

when green flag clicked

go to x: 0 y: 0

repeat 10

move 10 steps

wait 0.5 secs



However, there is one more problem...*most if not all cats cannot move without moving their limbs!*

Fortunately, we can fix that too!

Stage

y: -113

SCRATCH



File Edit Share Help



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables



Sprite1



x: 100 y: 0

direction: 90

Scripts

Costumes

Sounds

when clicked

when space key pressed

when Sprite1 clicked

wait 1 secs

forever

repeat 10

broadcast

broadcast and wait

when I receive

forever if

if

if

when clicked

go to x: 0 y: 0

repeat 10

move 10 steps

wait 0.5 secs

It's time to look at the **costume design** and **editing** aspect of Scratch.

To make any changes to a sprite, you must first select the sprite you wish to adjust.

In this case, the one and only sprite is already selected.

New sprite:



x: 387 y: 113



Sprite1

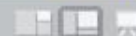


Stage

SCRATCH



File Edit Share Help



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables

when clicked

when space key pressed

when Sprite1 clicked

wait 1 secs

forever

repeat 10

broadcast

broadcast and wait

when I receive

forever if

if

if



Sprite1



x: 100 y: 0 direction: 90

Scripts

Costumes

Sounds

New costume: Paint Import Camera

1



costume1

95x111

3 KB

Edit

Copy



2



costume2

95x111

3 KB

Edit

Copy



To see the “costumes” of this sprite, click on the Costumes tab in the programming area, and the window will change to a costume editing area.

New sprite:



x: 314 y: 224



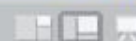
Sprite1

Stage

SCRATCH



File Edit Share Help



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables

when clicked

when space key pressed

when Sprite1 clicked

wait 1 secs

forever

repeat 10

broadcast

broadcast and wait

when I receive

forever if

if

if



Sprite1



x: 100 y: 0 direction: 90

Scripts

Costumes

Sounds

New costume: Paint Import Camera

1



costume1

3 KB

Edit

Copy



2



costume2

3 KB

Edit

Copy



You can see that this sprite has TWO costumes already designed.

You have seen the first one, but can you see any difference in the second one?

New sprite:

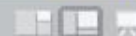


x: 314 y: 224



Sprite1

Stage



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables

when clicked

when space key pressed

when Sprite1 clicked

wait 1 secs

forever

repeat 10

broadcast

broadcast and wait

when I receive

forever if

if

if



Sprite1



x: 100 y: 0

direction: 90

Scripts

Costumes

Sounds

New costume:

Paint

Import

Camera

1



costume1

3 KB

Edit

Copy



2



costume2

3 KB

Edit

Copy



The second one has arms and legs in different positions, which will cause an "animated" effect when alternated back and forth.

Fortunately, there is an easy command to do this...

next costume

New sprite:



x: 314 y: 224



Sprite1

Stage



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables



Sprite 1



x: 100 y: 0

direction: 90

Scripts

Costumes

Sounds

switch to costume costume2

next costume

costume #

say Hello! for 2 secs

say Hello!

think Hmm... for 2 secs

think Hmm...

change color effect by 25

set color effect to 0

clear graphic effects

change size by 10

set size to 100 %

size

show

hide

go to front

go back 1 layers

when clicked

go to x: 0 y: 0

repeat 10

move 10 steps

wait 0.1 secs

To find this command, switch back to the Scripts tab of the programming area, and then select the **Looks** command set.

New sprite:



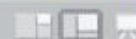
x: -405 y: -175



Sprite 1



Stage



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables



Sprite 1



x: 100 y: 0

direction: 90

Scripts

Costumes

Sounds

switch to costume costume2

next costume

costume #

say Hello! for 2 secs

say Hello!

think Hmm... for 2 secs

think Hmm...

change color effect by 25

set color effect to 0

clear graphic effects

change size by 10

set size to 100 %

size

show

hide

go to front

go back 1 layers

when clicked

go to x: 0 y: 0

repeat 10

move 10 steps

wait 0.1 secs

next costume

Then click and drag the **next costume** command and insert it into your program, inside the loop and after either the move, or the wait command...doesn't matter which.

New sprite:



x: 380 y: 123



Sprite 1

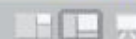


Stage

SCRATCH



File Edit Share Help



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables



Sprite 1



x: 100 y: 0

direction: 90

Scripts

Costumes

Sounds

switch to costume costume2

next costume

costume #

say Hello! for 2 secs

say Hello!

think Hmm... for 2 secs

think Hmm...

change color effect by 25

set color effect to 0

clear graphic effects

change size by 10

set size to 100 %

size

show

hide

go to front

go back 1 layers

when clicked

go to x: 0 y: 0

repeat 10

move 10 steps

wait 0.1 secs

next costume



Now run your program to see the
cat happily skipping across the
screen!



x: 354 y: 227

Stage

Scratch 1.4 of 30-Jun-09

SCRATCH

File Edit Share Help

Motion

Looks

Sound

Pen

Control

Sensing

Operators

Variables

Sprite1

x: 100 y: 0 direction: 90

Scripts

Costumes

Sounds

when green flag clicked

go to x: 0 y: 0

repeat 10

move 10 steps

wait 0.1 secs

next costume

switch to costume costume2

next costume

costume #

say Hello! for 2 secs

say Hello!

think Hmm... for 2 secs

think Hmm...

change color effect by 25

set color effect to 0

clear graphic effects

change size by 10

set size to 100 %

size

show

hide

go to front

go back 1 layers



Now, it is time for you to experiment...

Make the appropriate changes to the current program (no additional commands) to make the cat start on the very left edge (tail barely touching), and end up on the far right of the screen, whiskers barely touching the edge. Go ahead! Try it!

DO NOT LOOK AT THE NEXT SLIDE UNTIL YOU HAVE TRIED IT YOURSELF!

Scratch 1.4 of 30-Jun-09

SCRATCH File Edit Share Help

Motion Looks Sound Pen Control Sensing Operators Variables

Sprite 1 x: -72 y: 0 direction: 90

Scripts Costumes Sounds

when green flag clicked

go to x: -192 y: 0

repeat 38

move 10 steps

wait 0.1 secs

next costume

switch to costume costume2

next costume

costume #

say Hello! for 2 secs

say Hello!

think Hmm... for 2 secs

think Hmm...

change color

set color

clear graphics

change size

set size

size

show

hide

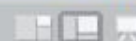
go to front

go back

Here is a possible solution, with the following changes made...

- The starting x value was changed to -192, causing the tail to barely touch the left edge of the screen.
- The number of repetitions for the loop was changed to 38, causing the cat to stop right as the whiskers touched the right edge.

These changes were made after several “**trial and error**” steps, fine tuning the values until the outcome was just right!



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables

Sprite1

x: 150 y: 0 direction: 180

Scripts

Costumes

Sounds

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90

point towards

go to x: 80 y: 0

go to

glide 1 secs to x: 80 y: 0

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

☐ x position☐ y position☐ direction

when clicked

clear

pen up

go to x: 0 y: 0

point in direction 90

pen down

set pen color to

set pen size to 1

glide 1 secs to x: 150 y: 0

turn 90 degrees



Now let's explore some other commands.
Save your first program as "motion1" and then
clear the programming work area.

Then build the new program you see here (call
it "square1") and run it to see what it does.
Hint: the new commands you see are green!

Scratch 1.4 of 30-Jun-09

SCRATCH File Edit Share Help

Motion Looks Sound Pen Control Sensing Operators Variables

Sprite1 x: -150 y: 0 direction: 90

Scripts Costumes Sounds

when green flag clicked

clear

pen up

go to x: 0 y: 0

point in direction 90

pen down

set pen color to blue

set pen size to 1

glide 1 secs to x: 150 y: 0

turn 90 degrees

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90

point towards

go to x: -134 y: 77

go to

glide 1 secs to x: -134 y: 77

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

x position

y position

direction

193 197

Add enough commands to the program so that it draws a square, and then moves the cat out of the way so you can see the square.

Be sure that no extra lines other than the sides of the square are drawn.

Don't look at the next slide until you have tried it yourself!!!



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90

point towards

go to x: -134 y: 77

go to

glide 1 secs to x: -134 y: 77

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

☐ x position☐ y position☐ direction

Sprite 1



x: -150 y: 0

direction: 90

Scripts

Costumes

Sounds

when clicked

clear

pen up

go to x: 0 y: 0

point in direction 90

pen down

set pen color to

set pen size to 1

glide 1 secs to x: 150 y: 0

turn 90 degrees

glide 1 secs to x: 150 y: -150

turn 90 degrees

glide 1 secs to x: 0 y: -150

turn 90 degrees

glide 1 secs to x: 0 y: 0

turn 90 degrees

pen up

go to x: -150 y: 0



Here is a possible solution!
*Some new commands were
introduced! Did you figure out
how they worked?*



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables



Sprite1



x: -150 y: 0

direction: 90

Scripts

Costumes

Sounds

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90

point towards

go to x: -134 y: 77

go to

glide 1 secs to x: -134 y: 77

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

x position

y position

direction

when clicked

clear

pen up

go to x: 0 y: 0

point in direction 90

pen down

set pen color to

set pen size to 1

glide 1 secs to x: 150 y: 0

turn 90 degrees

glide 1 secs to x: 150 y: -150

turn 90 degrees

glide 1 secs to x: 0 y: -150

turn 90 degrees

glide 1 secs to x: 0 y: 0

turn 90 degrees

pen up

go to x: -150 y: 0



First of all, a new group was used, the Pen command group, specifically the clear, pen up, pen down, set pen color, and set pen size.

Stage



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables



Sprite1



x: -150 y: 0

direction: 90

Scripts

Costumes

Sounds

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90

point towards

go to x: -134 y: 77

go to

glide 1 secs to x: -134 y: 77

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

☐ x position☐ y position☐ direction

when clicked

clear

pen up

go to x: 0 y: 0

point in direction 90

pen down

set pen color to

set pen size to 1

glide 1 secs to x: 150 y: 0

turn 90 degrees

glide 1 secs to x: 150 y: -150

turn 90 degrees

glide 1 secs to x: 0 y: -150

turn 90 degrees

glide 1 secs to x: 0 y: 0

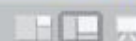
turn 90 degrees

pen up

go to x: -150 y: 0



From the Motion group, the glide and turn commands were used. All of these should be fairly self-explanatory if you observe how they affect the cat in this program.



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90

point towards

go to x: -134 y: 77

go to

glide 1 secs to x: -134 y: 77

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

☐ x position☐ y position☐ direction

Sprite1

x: -150 y: 0

direction: 90

Scripts

Costumes

Sounds

when clicked

clear

pen up

go to x: 0 y: 0

point in direction 90

pen down

set pen color to

set pen size to 1

glide 1 secs to x: 150 y: 0

turn 90 degrees

glide 1 secs to x: 150 y: -150

turn 90 degrees

glide 1 secs to x: 0 y: -150

turn 90 degrees

glide 1 secs to x: 0 y: 0

turn 90 degrees

pen up

go to x: -150 y: 0



There is one more cool technique that will help a lot with the labs you will do for this lesson.

For the last command of this lesson, you had to figure out where to move the cat by thinking about the coordinates, right?

SCRATCH



File Edit Share Help



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables



Sprite1



x: -150 y: 0 direction: 90

Scripts

Costumes

Sounds

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90

point towards

go to x: -134 y: 77

go to

glide 1 secs to x: -134 y: 77

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

☐ x position☐ y position☐ direction

when clicked

clear

pen up

go to x: 0 y: 0

point in direction 90

pen down

set pen color to

set pen size to 1

glide 1 secs to x: 150 y: 0

turn 90 degrees

glide 1 secs to x: 150 y: -150

turn 90 degrees

glide 1 secs to x: 0 y: -150

turn 90 degrees

glide 1 secs to x: 0 y: 0

turn 90 degrees

pen up

go to x: -150 y: 0

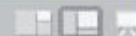


There is an easier way!
*Go to the output area, and click
and drag the cat to a particular
place on the screen, say the top left
hand corner of the screen.*

SCRATCH



File Edit Share Help



Motion

Control

Looks

Sensing

Sound

Operators

Pen

Variables



Sprite1

x: -190 y: 125 direction: 90

Scripts

Costumes

Sounds

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90

point towards

go to x: -190 y: 125

go to

glide 1 secs to x: -190 y: 125

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

x position

y position

direction

when clicked

clear

pen up

go to x: 0 y: 0

point in direction 90

pen down

set pen color to

set pen size to 1

glide 1 secs to x: 150 y: 0

turn 90 degrees

glide 1 secs to x: 150 y: -150

turn 90 degrees

glide 1 secs to x: 0 y: -150

turn 90 degrees

glide 1 secs to x: 0 y: 0

turn 90 degrees

pen up

go to x: -190 y: 125

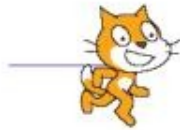
go to x: -150 y: 0



Drag me here
and then double
click me!

When you have the cat where you want it to be, double click on the cat itself.

Now look at the coordinates in the status area, and compare them to the coordinates in the go to x y and glide commands...they should match!



Sprite 1
x: -190 y: 125 direction: 90

Scripts
Costumes
Sounds

```
when clicked  
clear  
pen up  
go to x: 0 y: 0  
point in direction: 90  
pen down  
set pen color to: blue  
set pen size to: 1  
glide 1 secs to x: 150 y: 0  
turn 90 degrees  
glide 1 secs to x: 150 y: -150  
turn 90 degrees  
glide 1 secs to x: 0 y: -150  
turn 90 degrees  
glide 1 secs to x: 0 y: 0  
turn 90 degrees  
pen up  
go to x: -190 y: 125
```



Now just detach and discard the old
go to x y command, insert the one
with the new coordinates, and then
run your program to see that the cat
does indeed end up in the top left
hand corner!

WHAT HAVE YOU LEARNED?

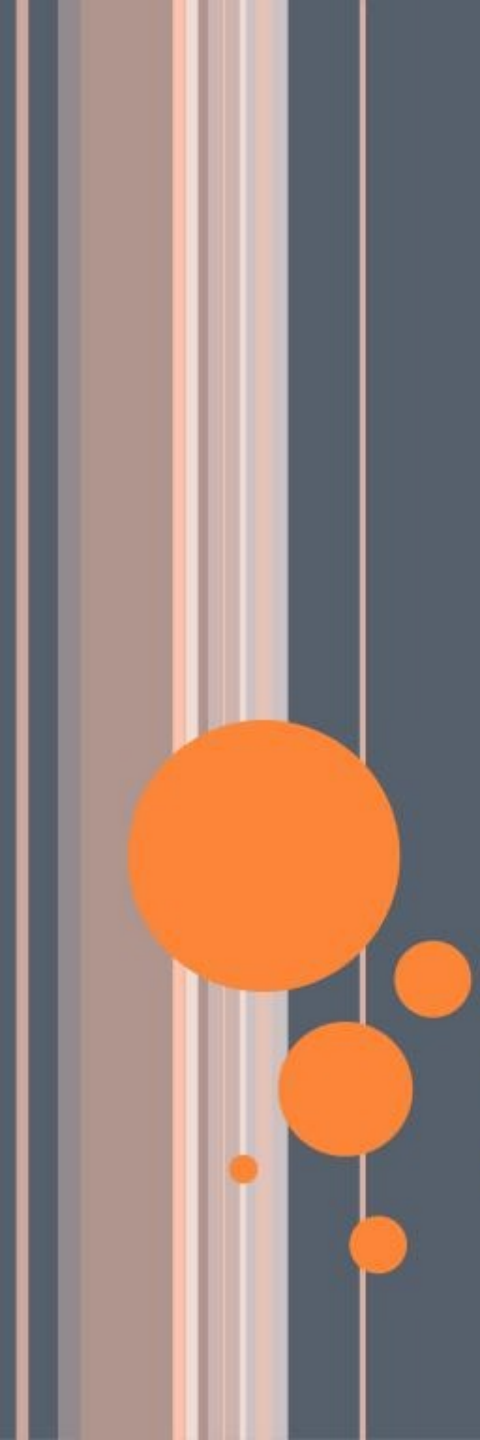


National Science Foundation
WHERE DISCOVERIES BEGIN



- In this lesson you learned the basics of the Scratch window, or programming environment.
- You also learned about some of the many Control, Motion, Looks, and Pen commands to create simple animation and drawing techniques.
- Now it is time to practice your new skills!
- **Have fun!**



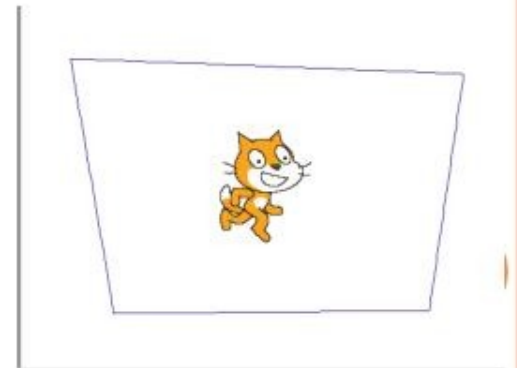


LABS

Do each of the labs in this problem set, and call your instructor over to see it run and grade it.

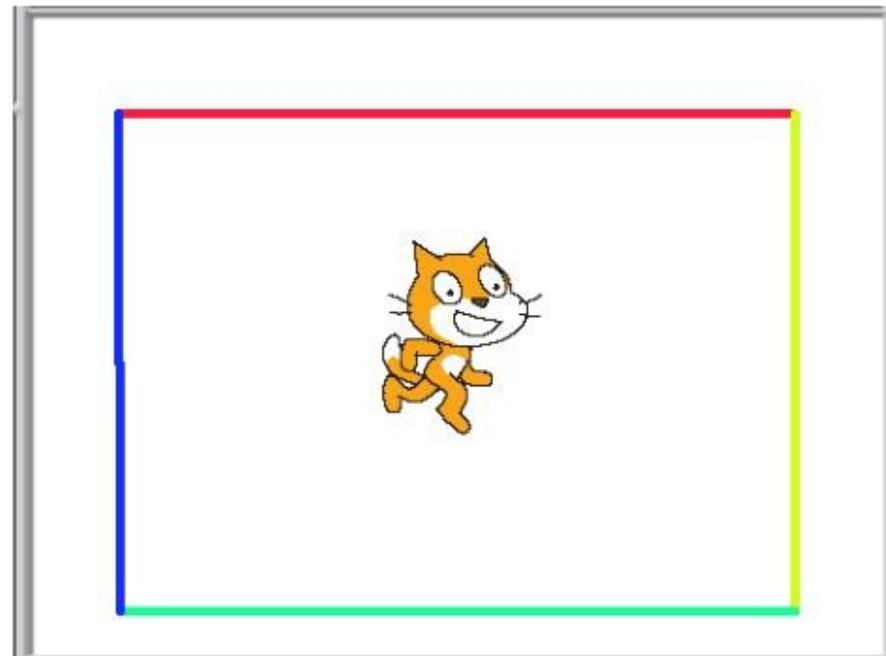
SCRATCH LAB 1A

- WASP (Write A Scratch Program) to start the cat in the home position (0,0), move to a corner of the screen, then **glide and draw** an irregular shaped quadrilateral, roughly around the edge of the screen, and then move back to the home position at the end of the program. Use the original pen color (blue) and size (1). Also, the cat **MUST** face 90 degrees the entire time. The result should look something like this.



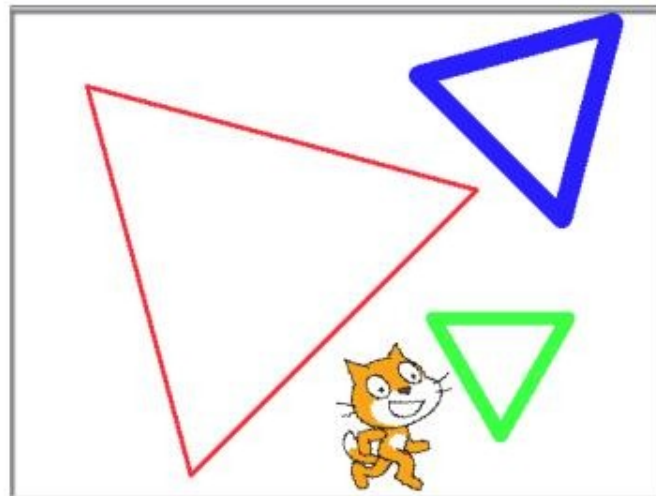
SCRATCH LAB 1B

- WASP so that the cat draws a rectangle around the edge of the screen, in a larger pen size (greater than 1), with a different color for each side. Also, the cat must always face in the direction traveled. The final screen should look something like this.



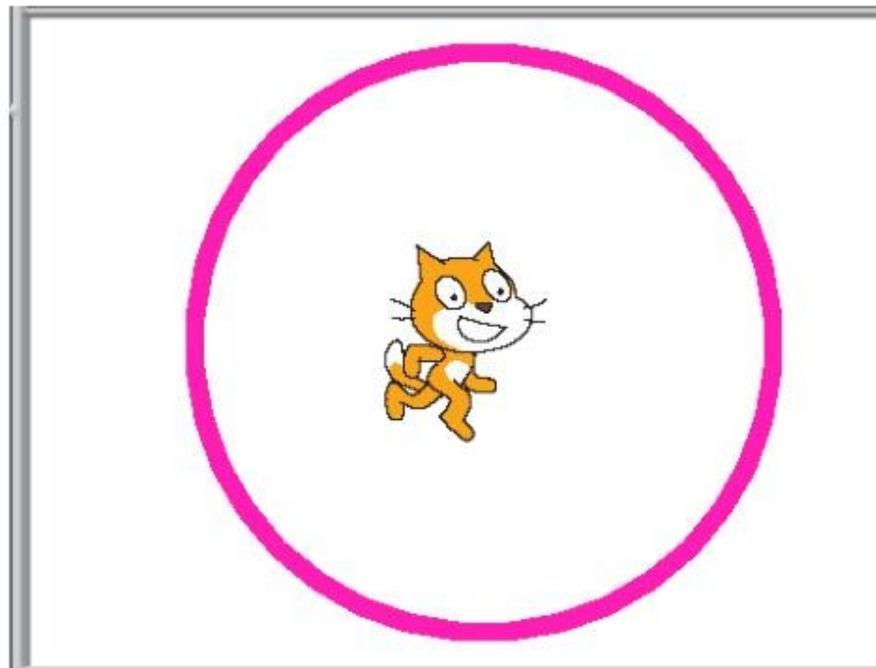
SCRATCH LAB 1C

- WASP to have the cat draw three equilateral triangles to fill up as much of the screen as possible, all of different sizes, colors, pen size, tilted different ways (no two the same), and none overlapping. The cat should end up somewhere on the screen NOT touching any of the triangles...something like this, but unique for you!



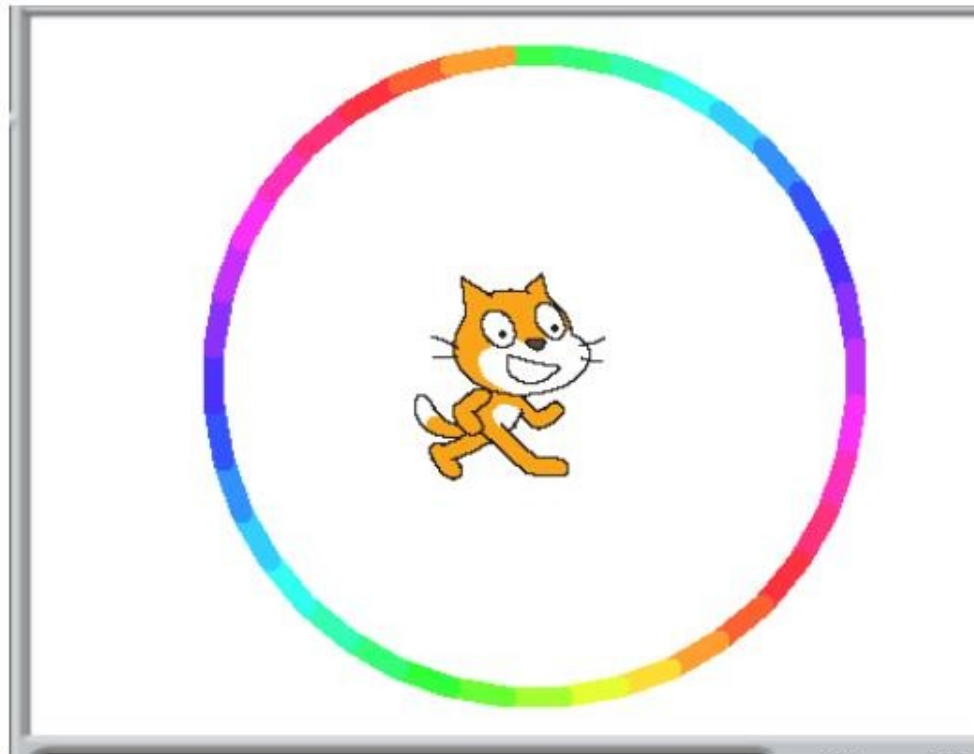
SCRATCH LAB 1D

- WASP so that the cat draws a one-color circle (whatever color you like), that fills up the screen as much as possible, pen size no larger than 10. Again, the cat ends up back home at the end.
Hint: 36 turns, 10 degrees each time.



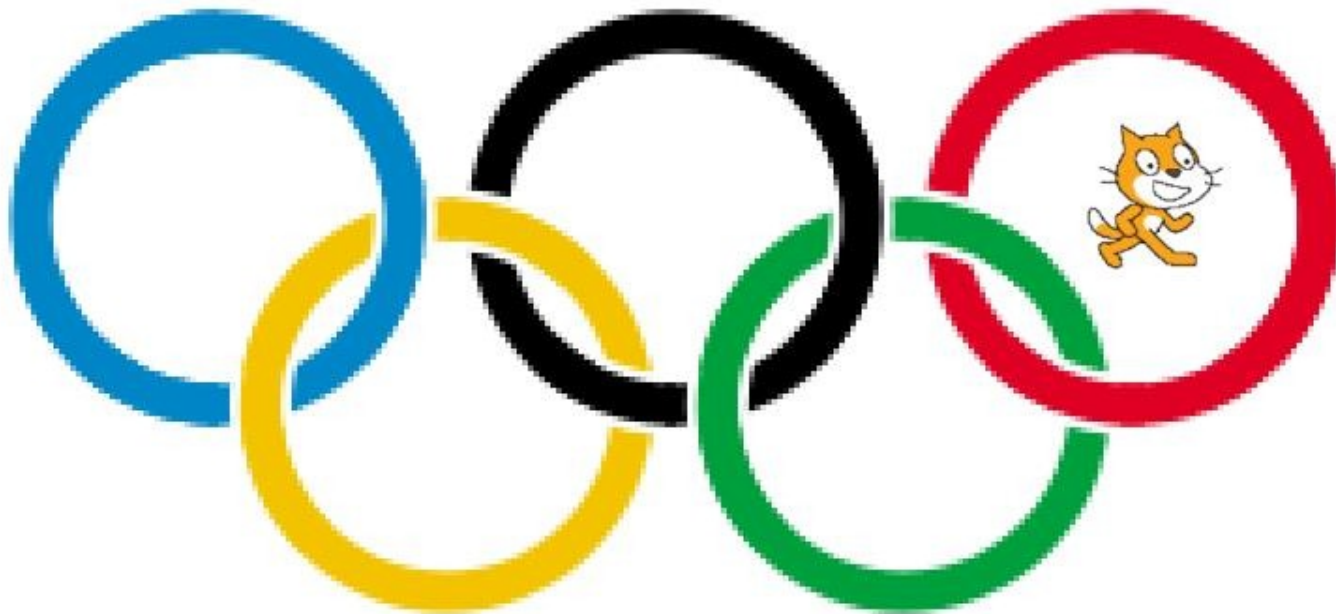
SCRATCH LAB 1E

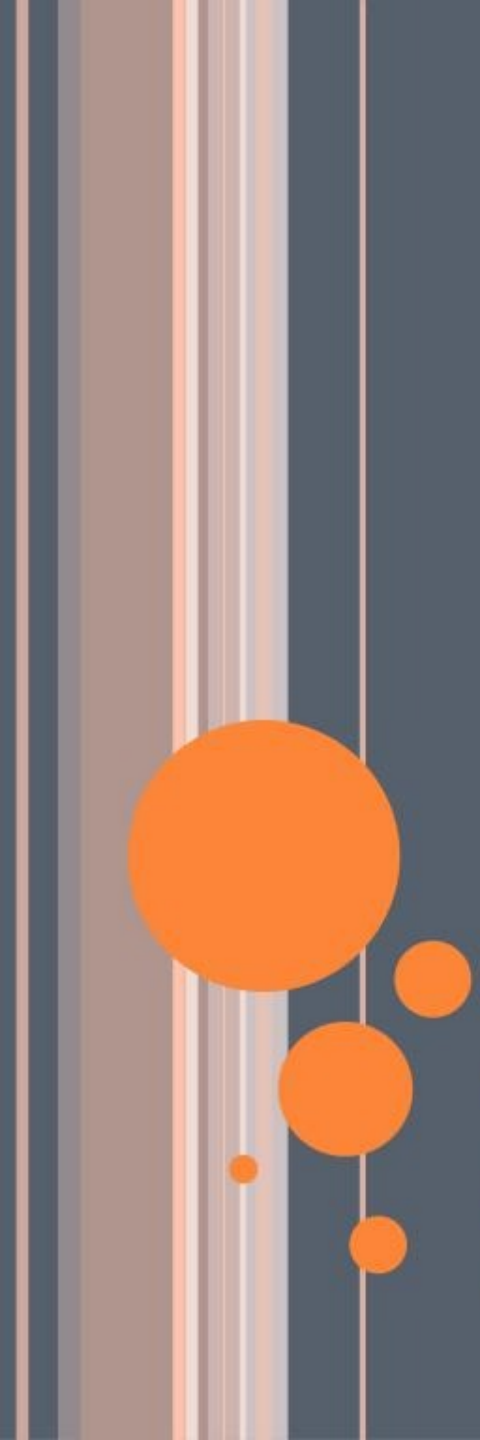
- WASP to modify the previous program so that the cat “skips” along the way, and creates a rainbow circle.



SCRATCH LAB 1F

- WASP to have the cat draw the five Olympic Rings as shown. The cat should end up in one of the five rings, whichever you choose. *Bonus credit if you can produce the interlocking effect.*





CONGRATULATIONS!

Your are now a bona fide Scratch beginning programmer! There is a lot more to learn, and hopefully you had fun with this first lesson.

Now go on to Lesson 2 for more cool stuff!