Name: _____ Date: _____

What is Scratch? Scratch lets you build programs by snapping coloured *blocks* together, like jigsaw pieces. You do not type any code. You drag a block into the *code area*, then join blocks under each other to make a *script*. When you click the green flag, your script runs from the top block down to the bottom block. To start, open `scratch.mit.edu` and click *Create*.

The screen has four main parts.

- The **stage** (top right) is where your *sprite* (the cat) moves and your project plays.
- The **sprite list** (under the stage) shows every character. Click one to give it code.
- The **block list** (middle) holds all the blocks, sorted by colour.
- The **code area** (right) is where you join blocks together.

The block groups you will use. Each block has a colour. The colour tells you which group it is in. Find the colour first, then find the block.

Motion	*(blue)*	Move the sprite: move, turn, go to, point, change x or y.
Looks	*(purple)*	Change how the sprite looks: say, costume, size.
Sound	*(pink)*	Play a sound.
Events	*(yellow)*	Hat blocks that start a script: green flag, key pressed, sprite clicked.
Control	*(orange)*	Choose when and how often blocks run: wait, forever, repeat.

Project 1: Your first script

Click the green flag. The cat says hello and slides across the stage. This is a program: blocks that run one by one, from top to bottom.

Get ready. Open a new Scratch project. The cat is already on the stage.

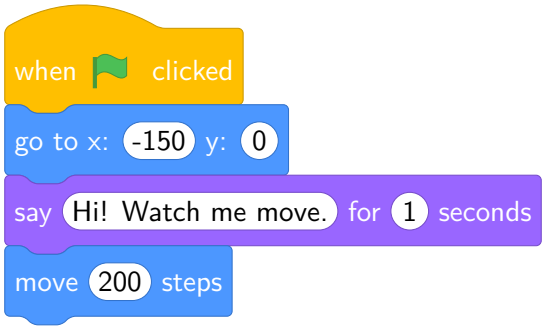
New blocks.

`when green flag clicked`	A yellow Events block. It starts the script when you click the green flag.
`go to x: () y: ()`	Jump to a spot on the stage. The middle is x: 0, y: 0.
`say () for () seconds`	Show a speech bubble for a short time.
`move () steps`	Slide forward by a number of steps.

Build it.

1. From the yellow Events blocks, drag out `when green flag clicked`. Every script starts with a block like this on top.
2. From the blue Motion blocks, join `go to x: -150 y: 0` under it. Now the cat always starts on the left.
3. From the purple Looks blocks, add `say (Hi! Watch me move.) for (1) seconds`.
4. Add `move (200) steps` from the Motion blocks.
5. Click the green flag above the stage. The cat says hello, then slides to the right.

When you are done, your code should look like this:



Your task. Now make your project better. Do these in order. Each one builds on the last.

1. Change the words in the `say` block to your own greeting.
2. Add one more `say` block at the very end, so the cat says something after it stops moving.
3. Make the cat reach the right edge: change `move (200) steps` to a bigger number, like 300.

Project 2: Move with the arrow keys

Make the cat move when you press an arrow key. Each key gets its own small script. A script runs only when you press its key.

New blocks.

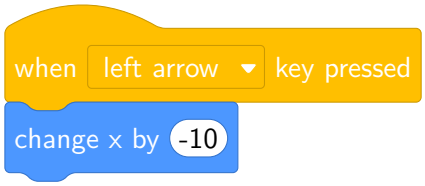
`when [space] key pressed`	A yellow Events block. It runs when you press the key you pick.
`change x by ()`	Move across the stage. Right is a plus number. Left is a minus number.
`change y by ()`	Move up or down. Up is a plus number. Down is a minus number.

Build it.

1. Drag out `when [space] key pressed`. Click the small arrow on the block and pick **right arrow**.
2. Join `change x by (10)` under it. Press the right arrow key. The cat moves right.
3. Make three more in the same way: **left arrow** with `change x by (-10)`, **up arrow** with `change y by (10)`, and **down arrow** with `change y by (-10)`.
4. Now you have four small scripts. Press the arrow keys to move the cat around.

When you are done, your code should look like this:

Move right



Move left

Move up

Move down

Think back to the stage grid. x is how far across. y is how far up. That is why right and up use plus numbers, and left and down use minus numbers.

Your task. Now make your project better. Do these in order. Each one builds on the last.

1. Make the cat move faster: change every 10 to 20, and every -10 to -20.
2. Add `point in direction (90)` to the right script and `point in direction (-90)` to the left script, so the cat faces the way it moves.

Project 3: Make it walk on its own

Use a forever loop. The cat walks across the stage on its own. It moves, swaps its legs, and turns around at the edges. You do not press any key.

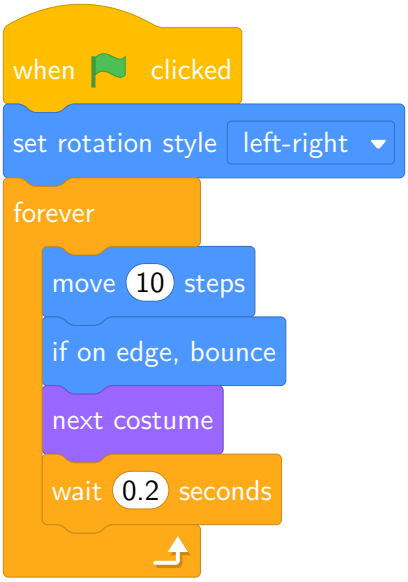
New blocks.

`forever`	An orange Control loop. It runs the blocks inside it again and again, without stopping.
`next costume`	Swap to the cat's other picture. Doing this fast makes it look like it is walking.
`if on edge, bounce`	Turn around when the cat reaches the side of the stage.
`wait () seconds`	Pause before the next block. This sets the speed.

Build it.

1. Start a new script with `when green flag clicked`.
2. Add `set rotation style [left-right]`. This stops the cat from turning upside down when it turns around.
3. From the orange Control blocks, drag in `forever`. It is a C shape. The blocks you put inside it run again and again.
4. Inside the forever, put these blocks in order: `move (10) steps`, `if on edge, bounce`, `next costume`, `wait (0.2) seconds`.
5. Click the green flag. The cat walks back and forth on its own.

When you are done, your code should look like this:



A forever loop never stops on its own. Click the red stop sign above the stage to stop it.

Your task. Now make your project better. Do these in order. Each one builds on the last.

1. Change the `wait (0.2) seconds` time so the cat walks at a speed you like.
2. Before the `forever` loop, add `go to x: -200 y: -120` so the cat always starts at the bottom-left.
3. Also before the loop, add `set size to (70) %` so the cat is a bit smaller as it walks.

Project 4: Click the cat

Try a new kind of event. This time the cat does something when you click it, not when you press a key. When you click it, the cat meows and changes its pose.

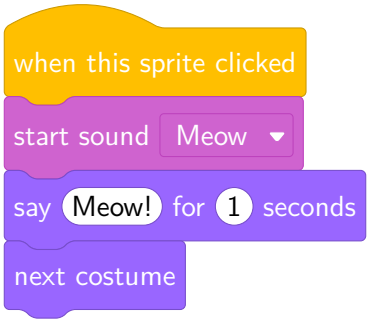
New blocks.

`when this sprite clicked`	A yellow Events block. It runs when you click the sprite on the stage.
`start sound [Meow]`	Play the cat's meow sound.
`change size by ()`	Make the sprite bigger with a plus number, or smaller with a minus number.

Build it.

1. Drag out `when this sprite clicked`.
2. From the pink Sound blocks, add `start sound [Meow]`.
3. From the Looks blocks, add `say (Meow!) for (1) seconds`, then `next costume`.
4. Click the cat on the stage. It meows, talks, and changes its pose.

When you are done, your code should look like this:



Your task. Now make your project better. Do these in order. Each one builds on the last.

1. Add `change size by (10)` at the end, so the cat grows a little each time you click it.
2. After that, add `wait (1) seconds` then `change size by (-10)`, so the cat grows, waits, then goes back to its size.
3. Swap `start sound [Meow]` for a different sound. Click the Sounds tab at the top left to pick one.

Project 5: Mini project: put it all together

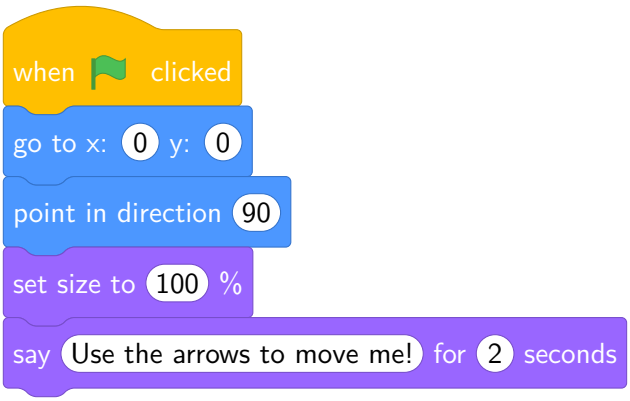
Put your blocks together into one small project. The green flag puts the cat back in the middle and tells the player what to do. The arrow keys move it. Clicking it makes it cheer.

Get ready. Keep your four arrow-key scripts from Project 2. This project uses them again. You will add the two scripts below.

Build it.

1. Add a `when green flag clicked` script that sets up the cat: `go to x: 0 y: 0`, `point in direction 90`, `set size to (100) %`, then `say (Use the arrows to move me!) for (2) seconds`.
2. Add a `when this sprite clicked` script that makes the cat cheer: `say (Yay!) for (1) seconds`, then `next costume`.
3. Click the green flag. Move the cat with the arrow keys, and click it to make it cheer. You have built a small project you can play!

Your two new scripts (the four arrow-key scripts from Project 2 run with these):



Green flag: set up and explain

Click: cheer

Your task. Now make your project better. Do these in order. Each one builds on the last.

1. Give the cat a backdrop. Click *Stage* at the bottom right, then *Choose a Backdrop*.
2. Change the green-flag `say` block to your own instructions for the player.
3. Add one more `when green flag clicked` script with a `forever` loop that spins the cat slowly: `turn (3) degrees`, then `wait (0.1) seconds`.
