Level One: Fundamentals

Youth Guide

PEER REVIEWED

23001DD

Discovering Computer Science & Programming through Scratch

Updated for Scratch 3.0

About this Guide

This guide is an introduction to the basic elements of programming within the Scratch 3.0 environment. It covers the most fundamental principles of programming in any programming language: sequence, iteration, conditionals, variables, and modularization. After completing the activities in this guide you should be able to write simple programs for a variety of purposes. Learning to program is much like learning to play a musical instrument. Only with lots of practice can you improve your skills and create beautiful things. This guide should give you some fundamentals on which to build, but you will want to spend lots of time on your own practicing, experimenting, exploring, and creating. Luckily, doing this is easy and fun with Scratch.

Acknowledgments

Authors:

Lenny Pitt, Professor Emeritus of Computer Science, University of Illinois

Judy Rocke, Curriculum Development Specialist, Office for Mathematics, Science, and Technology Education (MSTE), University of Illinois

Jana Sebestik, Assistant Director STEM Curriculum Design, (MSTE), University of Illinois

Principal Investigator: Lenny Pitt, University of Illinois

Co-Principal Investigators:

Maya Israel, Assistant Professor, College of Education, University of Illinois

George Reese, Director of the Office for Mathematics, Science, and Technology Education, University of Illinois

Layout and Design:

Christina Tran, Graphic Designer, MSTE, University of Illinois, iamchristinatran.com

Support for this guide is provided by the 4-H Computing Connections (CS4H) project funded by the University of Illinois Extension and Outreach Initiative and also by the Department of Energy and the Department of Homeland Security under Award Number DE-OE0000780.

Scratch is a project of the Lifelong Kindergarten Group at the MIT Media Lab (**scratch.mit.edu**). Images of the Scratch cat are used with permission. All other screenshots and images used in this guide are licensed under the Creative Commons Attribution-ShareAlike License.

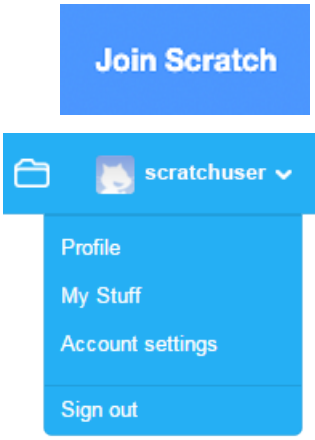
Table of Contents

Creating with Scratch

Scratch is a visual programming language that lets you create and share projects with others. It encourages you to problem solve, think logically, test and evaluate outcomes, and collaborate with others.

Scratch works best with Firefox or Chrome browsers

Join Scratch



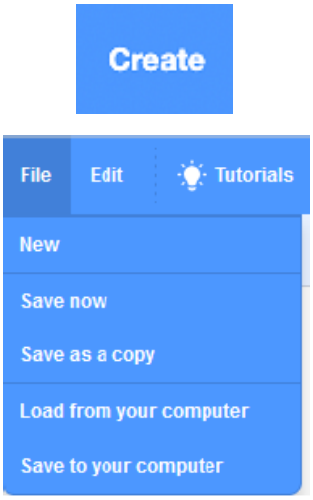
To try out Scratch and see examples of projects, visit:

scratch.mit.edu

To create your own account, click **Join Scratch** at the top right. With an account, you can create and save projects. You can also share your activities on the Scratch website and view other people's projects.

Begin a Project

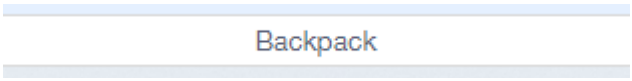
Scratch is typically run while online in a browser. To begin, click on **Create** in the top banner. Give your project a name in the window at the top in the menu bar.



While the work that you do gets saved automatically, you can force a save at any time by clicking on **File** and selecting **Save now**. To find your projects, in the upper right, click on your username and choose **My Stuff** or click on the file folder icon to the left of your username.

You can also save Scratch projects on your computer, or load them from your computer - these are options from the File menu.

An advantage to working online is the **Backpack** at the bottom of the **Script Area**. You can save scripts in your backpack from one project and then pull them out to use in another project.



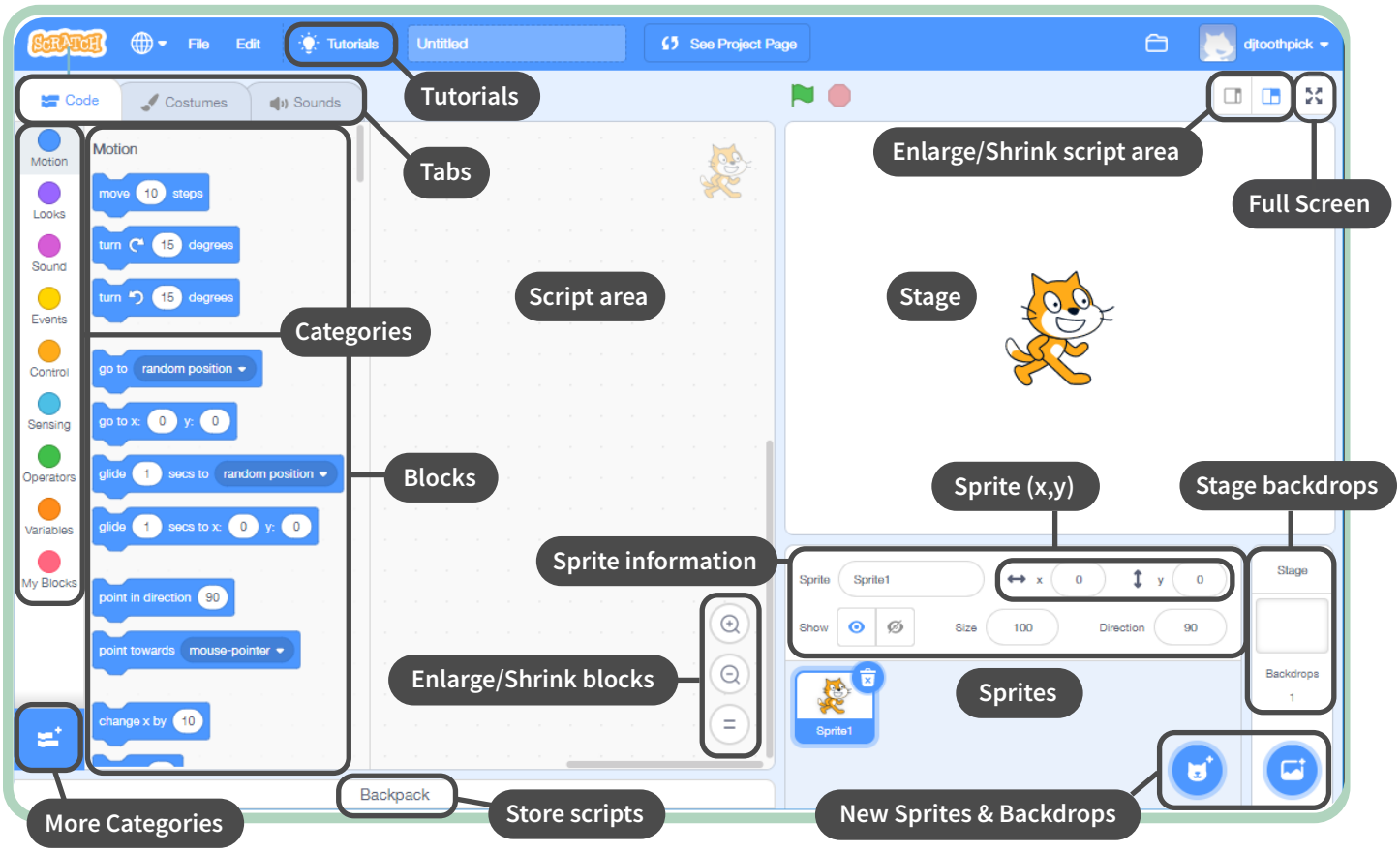
To learn more about the backpack, go to the ScratchEd YouTube Channel and search, "How to use the Backpack in Scratch" or visit:

www.youtube.com/watch?v=AuolBPY8qdc&t=29s

Working Offline

To use Scratch without an internet connection, you can download a version of Scratch onto your computer. See the instructions at the end of this guide.

Scratch Environment



Stage: The area where the Sprites appear

Sprites: Objects that move about the stage

Sprite information: Shows sprite's name, location, visibility, size and directions it is pointing

Sprite (x,y): Shows the (x,y) coordinates of the selected Sprite on the stage

New Sprites & Backdrops: Used to add or create new sprites and backdrops

Blocks: Items or tiles used to command your Sprite to perform an action

Enlarge/Shrink blocks: Enlarge (+) or shrink (-) size of blocks to make them fit easier in Script area

Script area: Area to place blocks or scripts—Right click the background of the Script Area to Undo, Redo, Clean up Blocks, Add Comment, Delete

Tabs: Switch between Scripts, Costumes, and Sound

Categories: Click each to show blocks in that category—Blocks are color coded in each category

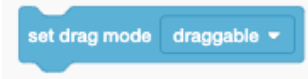
More Categories: Click to show categories that are not listed until you add them to the category list

Stage backdrops: Used to change the background of the stage

Tutorials: Shows helpful tips, frequently asked questions, and activity suggestions

Backpack: Store a script or Sprite here so it can be used in other Scratch projects

Full-Screen: Show only the stage area on the screen–To drag a Sprite around the stage in full-screen, use this block from the **SENSING** category



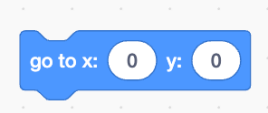
Getting Started with Scratch

Go to the Scratch website at **scratch.mit.edu** and click **Create** to open a new project. Your new project page should look something like the Scratch Environment on the previous page, where we've identified the different elements. There is lots to explore.

Spend time dragging blocks into the Script area, and snapping them together to create a script. The blocks in each category are color coded to match the name of the category. This makes it easier to find a block. Share what you discover!

The Stage

The Scratch cat is the sprite that appears on the stage when you open a new activity. Sprites are the objects that follow instructions to move around the stage. Find out how the stage is organized.



1. Drag this block from the **MOTION** category into the Script Area.
2. Put 0 into the x box and 0 in the y box. Click the block. Where is the cat now?
3. Input ordered pairs into the x and y boxes. Try (150, 150) and then click on the block. Try (-150, -150). What does each ordered pair of numbers tell the cat to do?
4. Predict where the cat will go with (-150, 150) and (150, -150). Test it. Try other ordered pairs. Predict where the cat will go before you test each ordered pair.
5. How is the stage organized?
6. Get rid of the **go to x: y:** block by dropping it back into the blocks area from which it came, or by right-clicking it and then selecting **Delete** from the drop-down menu.

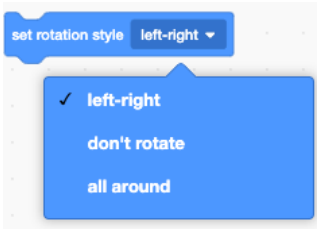
Sprites

Make sprites speak, move, change color, size, and direction by using blocks from the **MOTION** and **LOOKS** categories.

1. Pull out these blocks and drop them into the Script Area.
2. Click on the **move 10 steps** block. What do you notice? The cat should have moved a small amount. A step in Scratch is a single pixel on the screen, so 10 steps is not very far.

   Change the value 10 by clicking it and replacing it with 100 and then click the block again. In Scratch, wherever you see a white area inside of a block you can type in your own values to change the behavior.

3. Snap the blocks together. Click the top block. What does the cat do?
4. Separate the blocks in a script by clicking a block and pulling it away. Put the blocks in different orders or sequences. Run the script again after each change. How does the sequence change the behavior?
5. Change the input numbers in the blocks.
6. Use the **point in direction** block. Input (-90) or use the drop down menu to select (- 90). Run the script again. Why is the cat upside down?
7. A sprite's rotational style is controlled by the **set rotational style** block from the **MOTION** category. Pull this block out. Use the drop-down menu to explore the rotational styles. The first style is left-right, the second is don't rotate, and the third is all around. Try each and run the script again. Think about when it makes sense to use each.

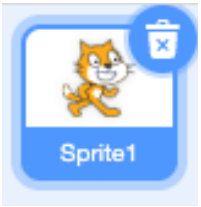


You can duplicate an entire script. Right click on the top block of the script. Then select **Duplicate**.

Delete a script you no longer need by dragging the script into the block area and dropping it there. Undelete a script by right-clicking in the script area, then click **Undo**.

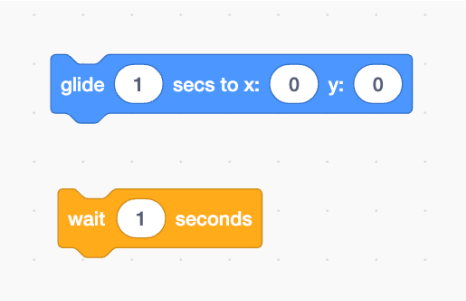
8. The sprite's information is shown in the box below the stage. This box gives the sprite's name, its (x, y) location, if it is shown or hidden on the stage, its size, and the direction it is pointing.

   This is a good time to change the sprite's name to something other than "Sprite1". Your projects may have multiple sprites, and it will be less confusing if you name them something that explains what they are. Click in the area showing the sprite name. Change the name to "Cat" or "Fluffy" or another name for the cat besides "Sprite1".



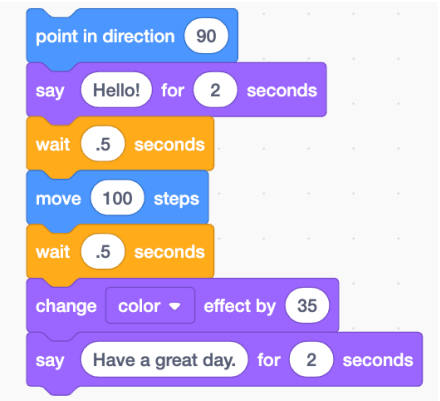
Sometimes the cat moves too quickly to see. Find these new blocks in the **MOTION** and **CONTROL** categories. Add them to your script to make the cat follow the list of instructions below. In what sequence do you place the blocks? What numbers do you input into the **glide** block to make the cat glide, and then return to its starting location?

- Glide across the stage
- Wait one second
- Change color
- Wait a second
- Change size
- Say "What's up?" for 2 seconds
- Glide back across the stage

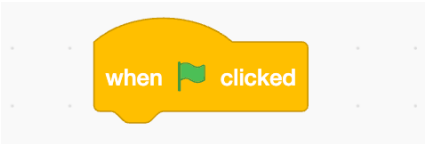


Starting and Stopping Scripts

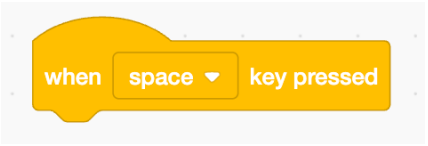
A script is a series of connected blocks that give instructions to a Sprite.



1. Create this script.
2. Click any block to start the script.
3. There are other ways to start a script. Pull out the **when green flag** block. Make it the first block in the script. When you click the green flag above the stage, all scripts with the **when green flag** block will start at the same time.



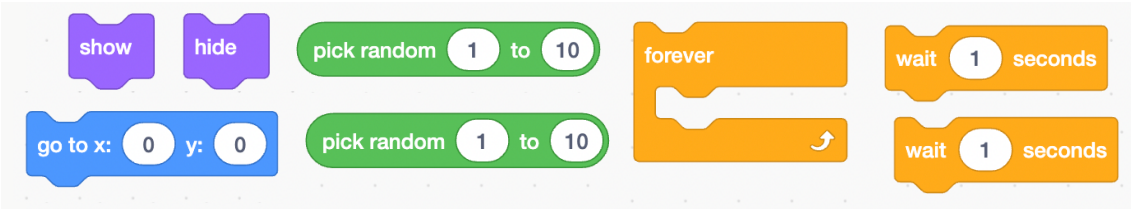
4. Pull out this **when** block. It has a drop down menu. Explore other ways to start a script using the drop down menu. If you want more than one script to run at the same time, use the same **EVENTS** block to start the scripts.



5. To stop all scripts that are running, click the stop sign on the top of the stage.

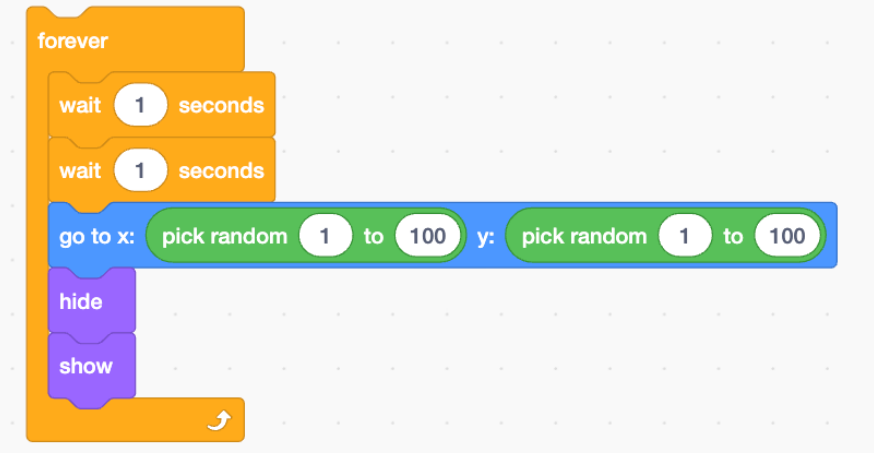
Building a Simple Script

1. Pull out these blocks.



2. Drop the **random 1 to 10** blocks from the **OPERATORS** category into the **go to x:, y:** block, and change the numbers 10 to 100. Click on the block repeatedly to see what happens.

3. Build this script. The **forever** block will expand to let you put other blocks inside. The blocks inside the **forever** block repeat forever in a loop.



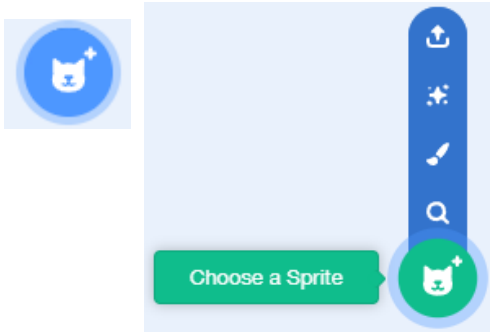
4. Change the sequence of the blocks inside the **forever** block. The new sequence should make the cat do the following repeatedly:
   - Appear for a second
   - Disappear for a second,
   - Move to a random (x, y) location on the stage
   - Then the loop should start over

> It is a good idea to test a script as you create it. As you add blocks to a script, run the partial script to see if it is doing what you expect it to do. It is easier to find problems with the script if you test it as you add blocks.

5. Change the input numbers on the **random** block so that the cat will be able to appear anywhere on the stage. What numbers should you use? Test the script.
6. Change the number in the **wait** block. Try using a decimal.
7. Does the script make the cat appear, disappear, and reappear at a random location?

New Sprites

Get new sprites by clicking the "Choose a Sprite" icon in the bottom right corner below the sprites. There are four different ways to get new sprites.



1. Explore each way to get new sprites.
2. To delete a sprite, right click the sprite and select delete under the stage.

Full Screen Mode

1. Select a new sprite. Write a script to make it move, talk or change color. Start your script with the **When green flag clicked** block from the **EVENTS** category.
2. Click the "Full Screen" button shown in the upper right corner above the stage.
3. Click the green flag shown in the upper left corner to start your script. Test your script.
4. Now click on your sprite and try to drag it around the stage. Notice you cannot drag the sprite. It is not dragable.
5. Click the "Full Screen Mode" again. Then click this "set drag mode" block in the **SENSING** category. You don't have to drag it into the script area, you just need to click it in the list.
6. Now click back to the "Full Screen Mode" again, and try dragging the sprite. What happens now? How could this be useful?

Sequence

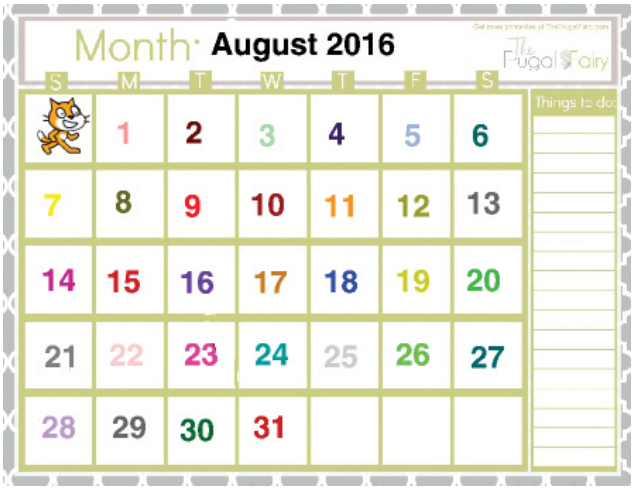
Computers will do exactly what you tell them to do. They will follow instructions one step at a time, in exactly the order you give them. It is important to understand this, because oftentimes when the program is not behaving the way you think it should, it is because it is doing exactly what you said, not what you meant. To "debug" a program it is useful to think like the computer, and carefully examine the instructions one step at a time. In this section you will learn some basic commands, and see how simply assembling them in the right order can result in interesting and useful behavior.

Calendar Activity

Open a new tab. Then open the calendar activity file at:
scratch.mit.edu/projects/52455886/

Select **See Inside** to see the Scratch editor for this activity.

The Calendar Activity is a Scratch file created by the CS4H Team at University of Illinois. When you add your own work to a file, you are creating a **Remix**. Click the Remix button to begin.



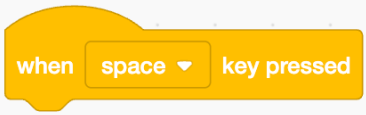
Hint: Check the cat's rotational style. Be sure it is set to all around.

The cat's starting position for this activity is in the first calendar square before **August 1**. Create a reset script which makes the cat return to this position whenever needed.

1. To do this, pull out the **go to** block.

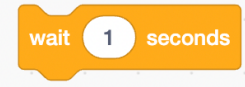
2. Look at the Sprite's coordinates found below the stage in the Sprite's information block. Find the (x, y) coordinates for the cat while it is at this starting location. Type the cat's coordinates into the **go to** block.

3. To start this reset script, use this **when** block.



4. Drag the cat to a different location. Press the space bar. Does the cat return to its starting location?

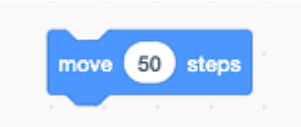
5. The cat moves so quickly that you don't see him move. Use this **wait** block to make it wait at any location. Run your script.



6. Drag your reset script to a corner of the Script Area.

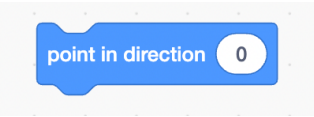
Begin a new script.

7. Use the **move** block to make a new, separate script that makes the cat move from the starting position to **August 1**. Then use the reset script to return the cat to the starting position.

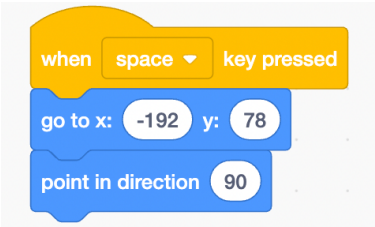


8. When the cat is at the starting position, how many steps would it need to move for it to land in the middle of the date **August 5**? Test it.

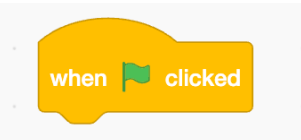
9. When in the starting position, the cat is pointing right. Pull out this **point in direction** block. Explore its drop down menu. What does this block do? Make the cat point down, left, right, and up.



10. Add a **point in direction (90)** block to the reset script so that the cat will always start in the first box, pointing right. Now your reset script looks like the script at the right.



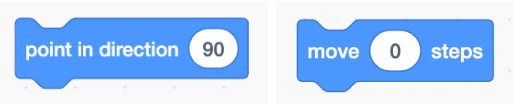
11. Next create a new script. Use only the **move** and **point** blocks to make the cat move to **August 9**. Start the script with the **when green flag** block.



12. Change the order of the blocks in your script. Does the script still work? Why is order important when you are writing scripts?

13. Use only the **move** and **point** blocks to make the cat land on your birthday date when the green flag is clicked. (For example, if your birthday is May 17, make the cat go to August 17.)

    Is the order important? How will you start this script?



14. Use only the **move**, **point**, and **wait** blocks to make the cat first go to your birth date, wait there for two seconds, and then go to your partner's birth date.

I pledge my head to clearer thinking,

my heart to greater loyalty,

my hands to larger service, and

my health to better living,

for my club, my community,
my country and my world.