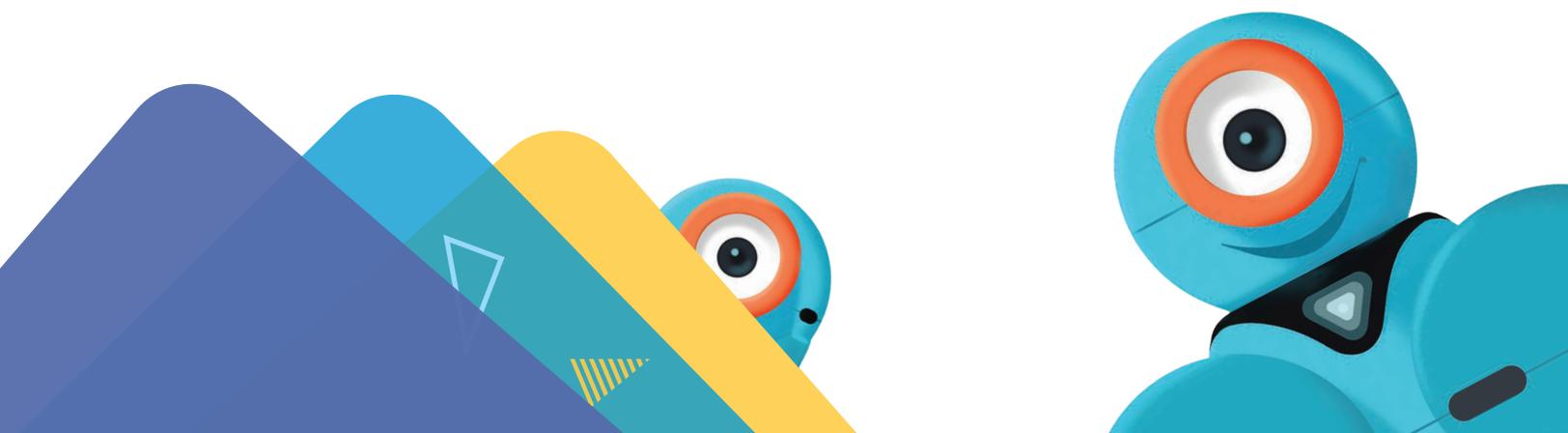


Dot at the Petting Zoo (B 2.1 - B 2.3): Teacher Packet

In this packet, we've included resources that will help you and your students as they independently complete the **Dot at the Petting Zoo Challenge Card** set:

-  **Challenge Cards:** a set of 3 **Challenge Cards** for students to practice coding concepts
-  **Solution Guides:** hints, suggestions, discussion questions, and cross curricular extension activities for each card
-  **Worksheets and Resources:** implementation strategies, planning/reflection worksheets for students, and an evaluation worksheet
-  **Lesson Plan (optional):** a whole-class instruction lesson focusing on the coding concept(s) that students will practice while completing the Challenge Cards

Looking for More? Visit: www.education.makewonder.com





Petting Zoo

Dot is visiting a petting zoo.
There are all kinds of animals.
What animal is Dot petting?



1. Let's start with a **Repeat Forever** block.



2. Now let's have **Dot** make some **animal sounds**. Set the **Animal** sound block to **Random** and put it **inside** the **Repeat Forever** block.



3. Have Dot wait for **4 seconds**. Put the **Wait For** block **inside** the **Repeat Forever** block.



4. Can you guess what animal Dot is petting before Dot makes a new **animal sound**?



Record your own animal sounds to add to the petting zoo.



Petting Zoo

Time: 5-10 minutes

Hints

- Look for the **Repeat Forever** and **Wait For** blocks in the **Control** menu.
- The **Random** selection is the very last option in the **Animal** sound block menu. Don't forget to select **Dot**.
- Place the **Animal** sound and **Wait For** blocks inside the **Repeat Forever** block.

B
2.1
Loops

1. Let's start with a **Repeat Forever** block.

Repeat forever
2. Now let's have **Dot** make some **animal sounds**. Set the **Animal** sound block to **Random** and put it **inside** the **Repeat Forever** block.

Animal Dot (Random)
3. Have Dot wait for **4 seconds**. Put the **Wait For** block **inside** the **Repeat Forever** block.

Wait for 4 seconds
4. Can you guess what animal Dot is petting before Dot makes a new **animal sound**?

★ Record your own animal sounds to add to the petting zoo.

My Sounds

Suggested Solution:

The suggested solution script consists of the following blocks:

- When Start** (Green flag clicked)
- Repeat forever** (Loop block containing):
 - Animal** (Sound block) set to **Dot** and **(Random)**
 - Wait for 4 seconds** (Control block)

Discussion Questions

1. If you change the order of the blocks, are you still able to guess what animal Dot is petting?
2. What if you only wanted to play the game 3 times? How could you change the code to make this happen?

Cross-Curricular Connections

MATH

- Have students count how many animals Dot has heard. (CCSS.MATH.K.CC.B.4)

ELA

- Have students listen to Dot. For every sound Dot makes, have students say or write an adjective describing that animal. Have students share the adjectives with a partner and have partners guess the animal the adjectives are describing. (CCSS.ELA.L.1.1.F)
- Isolate the initial and final phoneme of each animal's name (e.g., for alligator, a is the initial phoneme and r is the final phoneme). Encourage students to repeat each phoneme. (CCSS.ELA.W.1.7)

NOTES:

B

2.2

Loops



Quick, Hide!

The animals at the zoo have escaped!
Dot needs to hide!



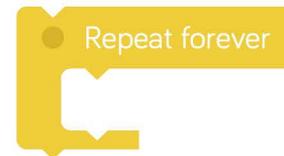
B

2.2

Loops



1. Dot needs to hide from the animals that have escaped! Let's start with a **Repeat Forever** block.



2. Dot hears an animal! Set the **Animal** sound block to **Random** and put it inside the **Repeat Forever** block.



3. Turn all of Dot's lights **off** by moving the **light** blocks **inside** the **Repeat Forever** block.



4. Now wait for **5 seconds** as the escaped animals walk by Dot.



5. It's safe now! Turn all of Dot's **lights** back **on**.



Quick, Hide!

Time: 10-15 minutes

Hints

- To turn off all of Dot's ear lights, choose the color black.
- To turn off all of Dot's eye lights, tap on each light around the eye.
- Place all of the blocks inside the Repeat Forever block. Only the blocks inside the Repeat Forever block will be repeated.

B | 2.2
Loops

1. Dot needs to hide from the animals that have escaped! Let's start with a **Repeat Forever** block.
2. Dot hears an animal! Set the **Animal** sound block to **Random** and put it inside the **Repeat Forever** block.
3. Turn all of Dot's lights **off** by moving the **light** blocks **inside** the **Repeat Forever** block.
4. Now wait for **5 seconds** as the escaped animals walk by Dot.
5. It's safe now! Turn all of Dot's **lights back on**.

Suggested Solution:

```

When Start
  Repeat forever
    Animal Dot (Random)
    All Lights [Black]
    Eye Pattern All off
    Wait for 5 seconds
    All Lights [Blue]
    Eye Pattern All on
  
```

Discussion Questions

1. Look in the **Animation** menu. Is there an animation you could add to the loop that would make the story even more interesting? (Hint: Look under **Play**.) Where in the loop would you add this block?
2. What could you change so that Dot will only hide a certain number of times? (Hint: Choose a different **Repeat** block.)

Cross-Curricular Connections

MATH

- Have students use a missing variable and sum to find the total number of loops that Dot went through in the program. (CCSS.MATH.1.OA.B.4)
- Have students describe the places Dot could hide in the classroom. Students must use correct positional words (e.g., over, under, next to, to the right) to describe the locations. (CCSS.MATH.K.G.A.1)

ELA

- Have students get into groups. Assign each group one or two of the animals Dot encounters at the zoo. Have groups research, write, and present facts about the animals to the class. (CCSS.ELA.W.1.2)

NOTES:

B

2.3

Loops



You Are Getting Sleepy...

The lion and bear from the zoo have found Dot! To escape, Dot needs to make the animals go to sleep.



Copyright © 2017 Wonder Workshop, Inc. All rights reserved.

B

2.3

Loops



1. To make the lion go to sleep, Dot needs to make a sleepy **eye pattern**.



2. Now Dot needs a **different** sleepy **eye pattern** for the bear.



3. **Repeat** each **eye pattern** 5 times.

4. Great job! Now Dot can escape.



Create your own sleepy **eye pattern** to make another animal go to sleep.

You Are Getting Sleepy...

Time: 20-25 minutes

Hints

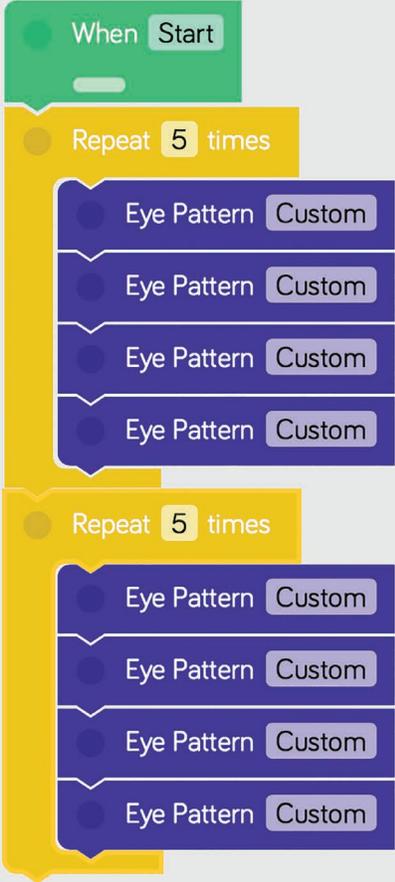
- Change the number in the Repeat block to 5.
- To change the Eye Pattern block, tap on each light to turn it on or off.

B | 2.3
Loops

1. To make the lion go to sleep, Dot needs to make a sleepy **eye pattern**.
2. Now Dot needs a **different** sleepy **eye pattern** for the bear.
3. Repeat each **eye pattern** 5 times.
4. Great job! Now Dot can escape.

★ Create your own sleepy eye pattern to make another animal go to sleep.

Suggested Solution:



```

When Start
  Repeat 5 times
    Eye Pattern Custom
    Eye Pattern Custom
    Eye Pattern Custom
    Eye Pattern Custom
  Repeat 5 times
    Eye Pattern Custom
    Eye Pattern Custom
    Eye Pattern Custom
    Eye Pattern Custom
  
```

Discussion Questions

1. Could we use the **Repeat Forever** or **Repeat Until** block for this challenge? Why or why not?
2. Could you add **ear lights** or **sounds** to the loop?

Cross-Curricular Connections

MATH

- Have students count the lit and unlit lights in each of Dot's **Eye Patterns**. (CCSS.MATH.K.CC.B.4.A)

ELA

- Have students write a narrative that tells how the animals escaped. (CCSS.ELA-W.1.3)

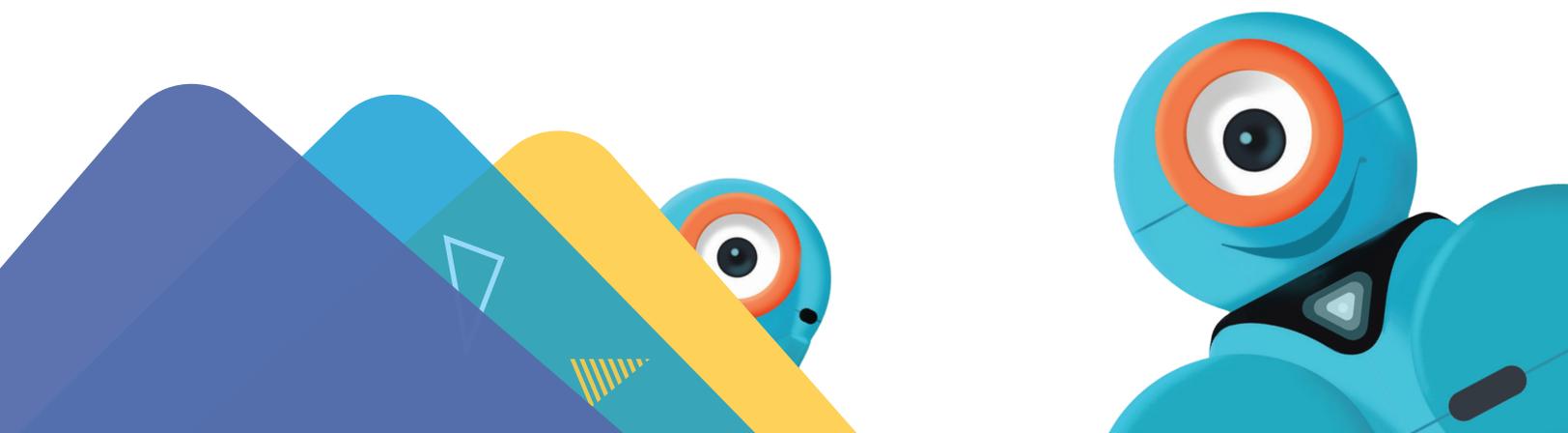
NOTES:

Worksheets & Resources

In this section, you will find the following worksheets/resources:

- Challenge Card Tips & Tricks
- Planning Worksheets
- Reflection Worksheets
- Troubleshooting Strategies
- Problem Solving & Debugging Strategies
- Evaluation Rubric
- Glossary

Looking for More? Visit: www.education.makewonder.com



Challenge Card Tips & Tricks



Determine Team Roles

Swap roles with your teammates for each challenge. Team roles include lead programmer, robot wrangler, and documentarian.



Plan Your Path

Draw out the path you want Dash to follow. Then plan out the blocks you'll need. You can also get up and walk the path that you think Dash should take.



Mark Your Spots

Use tape to mark Dash's starting spot and the location of any obstacles/objects.



Go Back to Start

Always put Dash back at the starting spot before playing a program again.



Use the When Start Block

Place your blocks under the **When Start** block. The **When Start** block should always be on your screen.



Think in Centimeters

Dash moves in centimeters. A centimeter is about the width of your finger.



Check Off the Steps

Use a dry erase marker to check off each step as you complete it. Make sure you erase the marks after you're done.



Help Your Robots Hear You

If the classroom is noisy, use the **Hear Clap** cue instead of the **Hear Voice** cue. Ask the teacher if you may try out your program with Dash and/or Dot outside or in the hallway.



Set a Time Limit

Give yourself or your team a set amount of time in which to complete the challenge

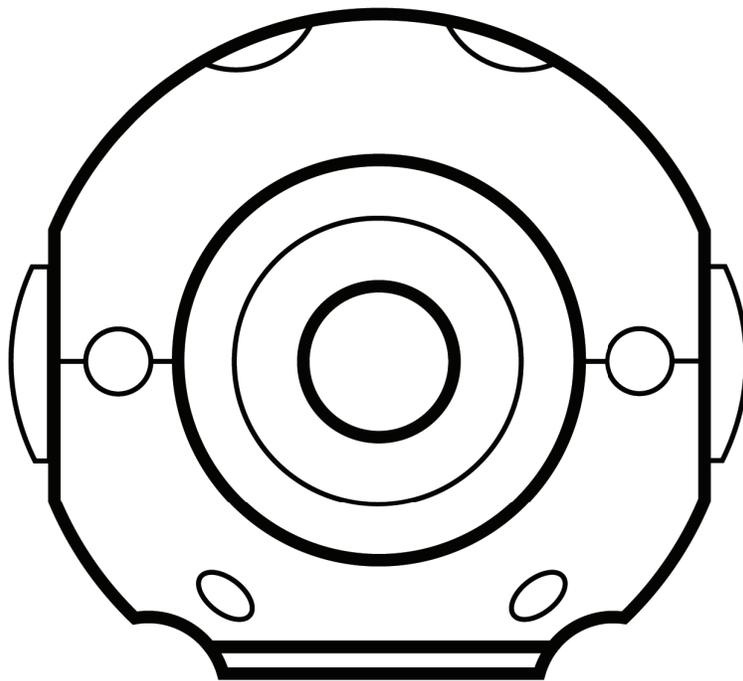
Dot Planning Worksheet

Name(s): _____ Date: _____

Coding Level: _____ Card #: _____

What do you want Dot to do?

Draw out the steps of the challenge or write a few sentences describing your goal.



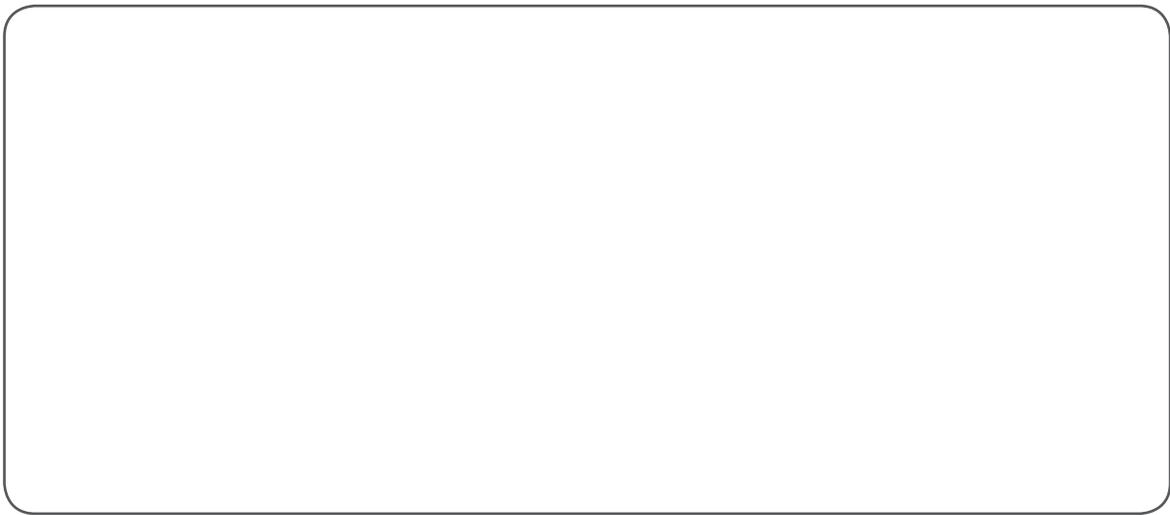
General Planning Worksheet

Name(s): _____ Date: _____

Coding Level: _____ Card #: _____

1. What do you want Dash or Dot to do?

Draw out the steps of the challenge or write a few sentences describing your goal.



2. What will you do to achieve your solution?

What will each team member do? What steps will you need to take? What blocks will you use?



Reflection Worksheet

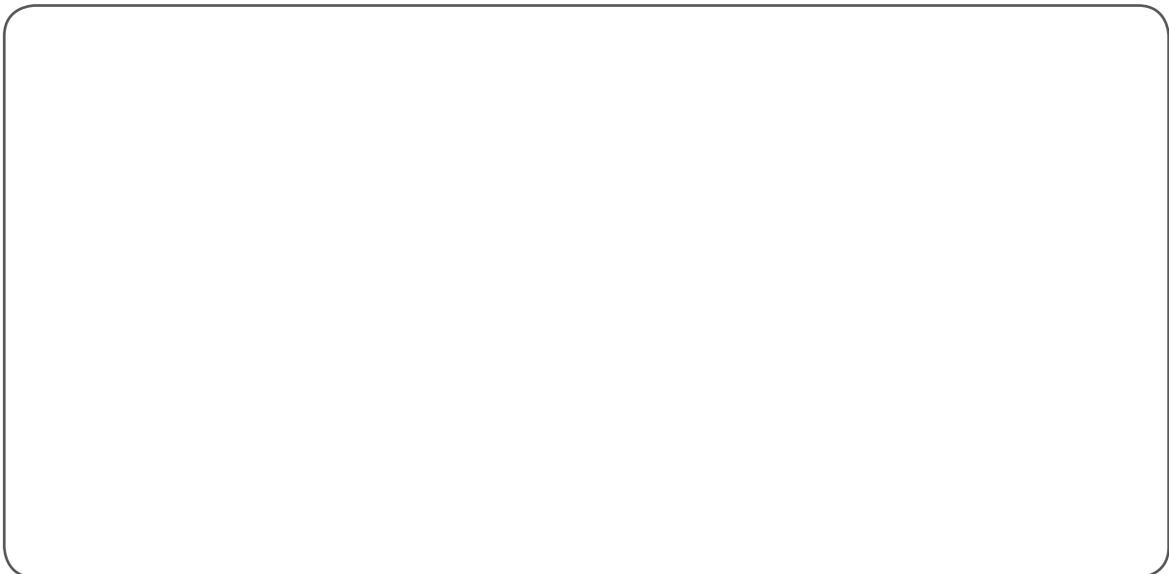
Name(s): _____ Date: _____

Coding Level: _____ Card #: _____

1. What did Dash and/or Dot do when you ran your program?



2. Did you make any mistakes? If so, how did you fix them?



Advanced Reflection Worksheet

Write a reflection entry in your Wonder Journal. Try to answer these questions as part of your reflection:

Results

- What did Dash and Dot do when you ran your program?
- Did you make any mistakes? If so, how did you fix them?

Connections

- What did you like the most about this challenge? Why?
- What was the most difficult part of the challenge? What did you learn from it?

Next Steps

- If you had more time, how would you change or add to your code?
- What are you planning to do next? Will you try another Challenge Card or start a new coding project?

Troubleshooting

If your program is not running correctly . . .

- Check if Dash and/or Dot are turned on.
- Make sure Dash and/or Dot are connected to the app.
- Make sure your blocks are connected to the **When Start** block.
- Try restarting the app.

If Dash and/or Dot are disconnecting . . .

- Turn off the robots and turn them on again. Then reconnect the robots to the app.
- Press play and then press stop to make the robots reset.
- Try charging the robots.

Three, then me!

- Ask or get help from three of your classmates. If you still need help, then ask the teacher.

Problem Solving & Debugging

Break down the challenge

- What do you need for the challenge? Which robots? Which materials and/or accessories?
- What are Dash and/or Dot supposed to do?
- Have you solved similar challenges to this one?
- Focus on one step at a time.

Plan your solution

- Draw a picture or make a list of what you want Dash or Dot to do.
- What blocks will you need to complete the challenge?
- Are there any hints on the card that can help?
- Use tape to mark Dash's starting point.
- Use tape to mark each obstacle's location.

Test Your Code

- Does your code complete the challenge?
- If not, play your code again. Watch as the program goes through each block. Do you notice any mistakes?
- Do you need to change, delete, or add more blocks?
- Are your blocks telling Dash to do something when you actually want Dot to do something?

Improve your work

- Ask another student or group to check your program.
- Is there an easier way to complete the challenge? Can you use fewer blocks?
- How can you improve your program? Could you add more lights, sounds, or other customizations?

Evaluation Rubric

	Programming	Reflection & Documentation	Collaboration & Communication	Creativity
1 Novice	Completed part of the activity and needed assistance throughout the process.	Use a journal, worksheets, and/or multimedia tools (such as video and images) to document some of the activity results.	Participated little or not at all in classroom discussions. Demonstrated little to no cooperation with group members during the activity.	Demonstrated limited creativity in developing ways to complete the activity.
2 Developing	Used the targeted coding concept(s) to complete the activity with some assistance.	Incorporated some target vocabulary and some thoughtful reflection on the coding process while documenting activity results using journal entries and multimedia tools.	Occasionally participated in classroom discussions and cooperated somewhat with group members.	Developed a few different ways to complete the activity, but the solution was not particularly creative.
3 Proficient	Used the targeted coding concept(s) to complete the activity without assistance.	Incorporated target vocabulary and reflection on the coding process. Clearly documented activity results using journal entries and multimedia tools.	Actively participated in classroom discussions. Answered questions and cooperated with group members during the activity.	Applied the iterative process to develop creative and unexpected solutions for the activity.
4 Exemplary	Used the targeted coding concept(s) to complete the activity without assistance. Enhanced the solution with more efficient (e.g., fewer blocks) and/or advanced features (e.g., lights, sounds) in the code.	Incorporated advanced target vocabulary and in-depth reflection on the coding process. Thoroughly and clearly documented and presented activity results.	Actively participated in classroom discussions and cooperated with group members. Gave constructive feedback to others and effectively incorporated feedback from others.	Went above and beyond to develop, revise, and execute imaginative solutions for the activity.

wonder
workshop

dash & dot[™]
challenge
cards

Coding Lessons

The following lesson plan is a part of
our **Learn to Code Curriculum**.

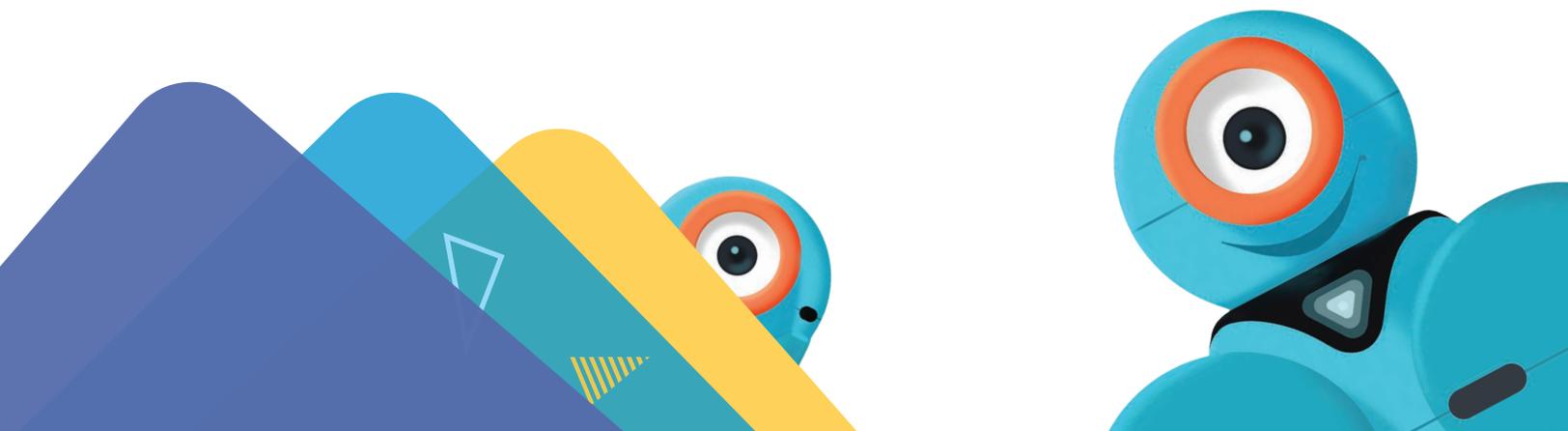
To check out more free coding lessons, please visit:

www.education.makewonder.com/curriculum/learn_to_code

Cross Curricular Lessons

Looking to go beyond the **Hour of Code**? Your students can **code to learn** via our project-based cross curricular lessons!

www.education.makewonder.com/curriculum/code_to_learn



Level B: Lesson 3

Loops: Repeat X with Dot

Lesson Information

Overview/Description

Students will review their previous programming with **sequencing, loops,** and the **Repeat Forever** block. They will also be introduced to using the **Repeat** block to define a specific number of loops.

They will use *Blockly* preset programs and **Challenge Cards** to practice their new skills.

Coding Level: B

Prior Experience:

Before this lesson, it is recommended that students complete:

- Level A Lessons: All
- Level B: Lessons 1–2
- [Optional] Level A **Challenge Cards**: All
- [Optional] Level B **Challenge Cards**: 1.1–1.3

Learning Objectives:

Students will:

- Understand the purpose of using the **Repeat** block.
- Use the **Repeat** block to revise code and complete coding challenges in an iterative process.

Target Grade Range: K–1

Suggested Group Size: 2–3 students per robot

Time Required: 45–60 minutes

Materials:

- 1 Dot robot per group
- 1 tablet per group
- projector or interactive display with mirroring capability
- Pencils

- [Optional] **Challenge Cards:** B 2.1, B 2.2, B 2.3
- [Optional] 1 dry erase marker per group
- [Optional] access to **Twitter** and **Instagram**

Resources/Downloads:

- **Troubleshooting** handout
- **Problem Solving & Debugging** handout
- **Wonder Journal: Dot Planning** and/or **General Planning** worksheets
- **Wonder Journal: Reflection** worksheets
- **Evaluation Rubric**
- [Optional] **Challenge Card Checklist**
- [Optional] **Challenge Card Tips & Tricks** handout

Preparation:

- Fully charge the tablets and robots.
- Install the **Wonder Workshop Blockly** app on each tablet.

Review

LOOPS REVIEW

1. Briefly review how students have used **loops** with Dash before.
 - Say, "We created a dance for Dash using loops. We told Dash what dance moves to do. Then we told Dash to repeat the moves again and again."
2. Say, "Before we used loops, we programmed a dance for Dash. What was the code for Dash's dance like *without* loops?"
 - Sample response: "When we created the code for the dance without loops, it was very long."
3. Ask, "What did the code for Dash's dance look like when you used loops?"
 - Sample response: "The dance was much shorter and easier to create when we used loops."
4. Ask, "Why do we use loops?"
 - Sample response: "We use loops when we want to repeat the same sequence of code over and over again. It makes our code easier to write."

CODE.ORG CONNECTIONS [Optional]

1. Review the *Code.org's* lesson focusing on **loops**: <https://studio.code.org/s/courseb-draft/stage/9/puzzle/4>.
2. Then ask: "How was using loops helpful in solving this puzzle?"

- Sample response: “Using loops meant I could use fewer blocks in my code, which also meant it took me less time to create it.”

Direct Instruction

INTRODUCTION

1. Project your tablet screen and open the *Blockly* app.
2. Say, “You have been using the **Repeat Forever** block when you want Dash or Dot to make loops. When you use the **Repeat Forever** block, the loop never stops. What if you want a loop to stop after a specific number of times?”
3. Say, “In this lesson, you will use a **Repeat** block. When you use a **Repeat Forever** block, the sequence inside the block continues to repeat until you tell it to stop. However, a **Repeat** block repeats the code sequence a specific number of times.”
4. Review a real-life example where you would want something repeated a specific number of times, such as:
 - A drummer repeats a beat a specific number of times until the end of a song.
 - A video game player repeats movements a specific number of times to get to the end of a level.
5. Show students where they can find the **Repeat** block in the **Control** menu. (Note: This is the block labeled **Repeat [X] times**.)
[Art Notes: Screenshot of Control menu open highlighting the “Repeat” block]

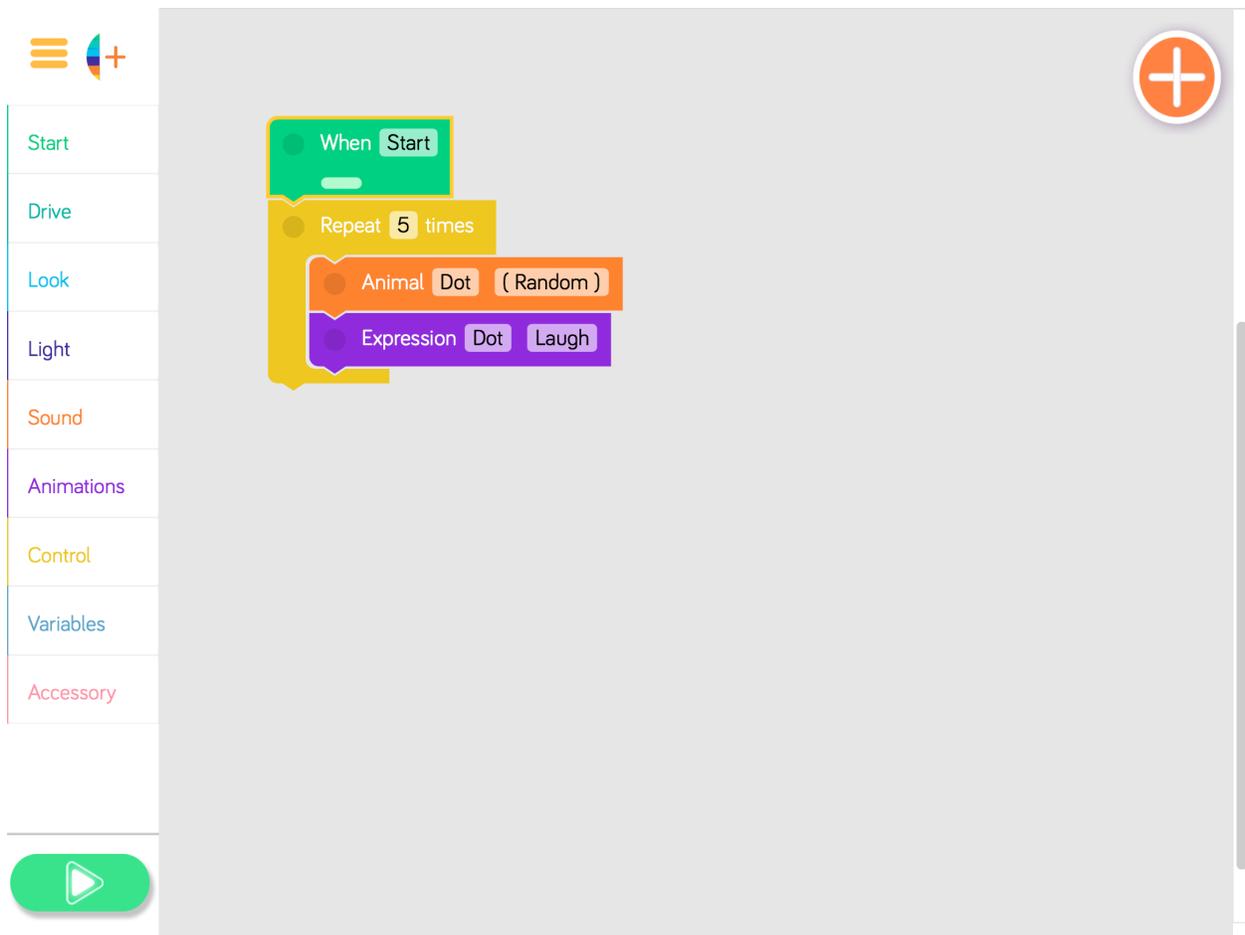
Quick Check

- What is a **loop**?
- When would you like to repeat something a specific number of times?
- Where do you find the **Repeat** block in *Blockly*?

Guided Practice

ACTIVITY: ZOO WHO

1. In *Blockly*, go to the **Create New Project** menu and select the **Zoo Who** preset program. Then tap “create.”



2. Say, "Dot wants to practice repeating a task."

- Play the program and have students share what they notice. (Sample response: "Dot repeated the steps 5 times.")
- Select a student to point out the **Repeat** block.

3. Ask, "What would happen if we changed the number in the **Repeat** block?"

- Have students take turns sharing their guesses. (Sample response: "Dot would make more animal sounds!")
- Select student volunteers to suggest what number to use in the **Repeat** block.
- Tap the **Repeat** block and then use the pop up menu to change the number of times the loop repeats. Then play the program.

4. Continue selecting student volunteers to change the number in the **Repeat** block and play the program. Point out to students how Dot's behavior changes each time.

QUICK CHECK

- What is a **loop**? (Sample response: "A loop repeats the bits of code placed inside it.")

- What is the difference between the **Repeat Forever** and the **Repeat** blocks? (Sample response: “The **Repeat** block lets us choose how many times the code is repeated.”)
- How do you change the number of times you want a code to repeat in *Blockly*? (Sample response: “You tap on the **Repeat** block and then change the number.”)

Independent Practice

Have students work on the following activities in small groups (ideally 2–3 students per robot).

Encourage students to share tablet and robot time. Have them establish and rotate through roles such as:

- **Lead Programmer:** Holds the tablet and manipulates the code.
- **Robot Wrangler:** Retrieves and resets the robot after every program attempt.
- **Documentarian:** Records group results, thoughts, and progress. Illustrates group designs and ideas.

When students work together while coding, they’re able to help each other identify mistakes and develop creative solutions!

ZOO WHO EXTENSION

1. Have students use the **Wonder Journal: Dot Planning** and/or **General Planning** worksheet to design ways to add to or alter the **Zoo Who** preset program. They can:

- Change the **sequence**.
- Change the number in the **Repeat** block.
- Add another **Repeat** block.
- Put **lights** and more **animations** or **sounds** in the **Repeat** block.

2. After they finish revising the program, have students:

- Complete a **Wonder Journal: Reflection** worksheet.
- Take a screenshot of their *Blockly* code.
- Take a video of Dot while the code is running.

CHALLENGE CARDS [Optional]

1. You can purchase our **Learn to Code Challenge Card** sets and **Curriculum Guide** here: <https://store.makewonder.com/#/education>

2. Have students complete the following **Challenge Cards**:

- **B 2.1: Petting Zoo**
- **B 2.2: Quick, Hide!**
- **B 2.3: You Are Getting Sleepy . . .**

3. For each challenge, encourage students to:

- Use the **Wonder Journal: Dot Planning** and/or **General Planning** worksheet to discuss how they can complete the challenge.
 - Review the **Troubleshooting** and **Problem Solving & Debugging** worksheets if they run into any problems with their code.
4. After they finish each challenge, have students:
- Complete a **Wonder Journal: Reflection** worksheet.
 - Take a screenshot of their *Blockly* code.
 - Take a video of Dot while the code is running.

Wrap Up

STUDENT PRESENTATIONS

1. Have student groups take turns sharing one of their programs with the class. Encourage them to:
 - Explain their design thinking. (E.g., “We chose to change the number of times the algorithm repeated so we could see how many animal sounds Dot would make.”)
 - Share any obstacles and difficulties they overcame during the activity. (E.g., “We weren’t sure how to change the number of times something repeats, but then we learned that if we tap on the number on the block, a menu pops up where we can choose a different number.”)
2. Encourage students to ask each other how they accomplished different objectives and give each other feedback on their programs. Possible questions/feedback includes:
 - “How did you (program Dot to **repeat** two different sounds)?”
 - “I like how you (changed Dot’s **sequence** and then added another **Repeat Forever** block).”
 - “What if you (had Dot **repeat** more **sequences**)?”

FOLLOW-UP QUESTIONS/DISCUSSION

- Describe a way you use **loops** in your life. (Sample response: “I use loops when I chew my food. I open and close my mouth over and over until I swallow.”)
- How was the **Repeat** block helpful in this code? (Sample response: “Using the **Repeat** block helped me include all the animal sounds without having to add a separate block for each one.”)

ASSESSMENT

- Use our **Evaluation Rubric** to review students’ work and presentations.
- [Optional] Share your students’ work with the world using @wonderworkshop and #dashanddot!

Standards

CSTA

- Develop programs with sequences and simple loops, to express ideas or address a problem.
- Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.
- Develop a plan that describes a program's sequence of events, goals, and expected outcomes.
- Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops.
- Using correct terminology, describe steps taken and choices made during the iterative process of program development.

ISTE

4d: Exhibit a tolerance for ambiguity, perseverance, and the capacity to work with open-ended problems.

5a: Formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions.

6b: Create original works or responsibly repurpose or remix digital resources into new creations.

7c: Contribute constructively to project teams, assuming various roles and responsibilities to work effectively toward a common goal.

NGSS

3-5-ETS1-2: Generate and compare multiple solutions to a problem based on how well they meet the criteria and constraints of the design problem.

Common Core

CCSS.ELA-LITERACY.W.2.8: Recall information from experiences or gather information from provided sources to answer a question.

CCSS.ELA-LITERACY.SL.3.1: Engage effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with diverse partners on grade 3 topics and texts, building on others' ideas and expressing their own clearly.

CCSS.ELA-LITERACY.SL.3.3: Ask and answer questions about information from a speaker, offering appropriate elaboration and detail.

CCSS.ELA-LITERACY.SL.3.4: Report on a topic or text, tell a story, or recount an experience with appropriate facts and relevant, descriptive details, speaking clearly at an understandable pace.