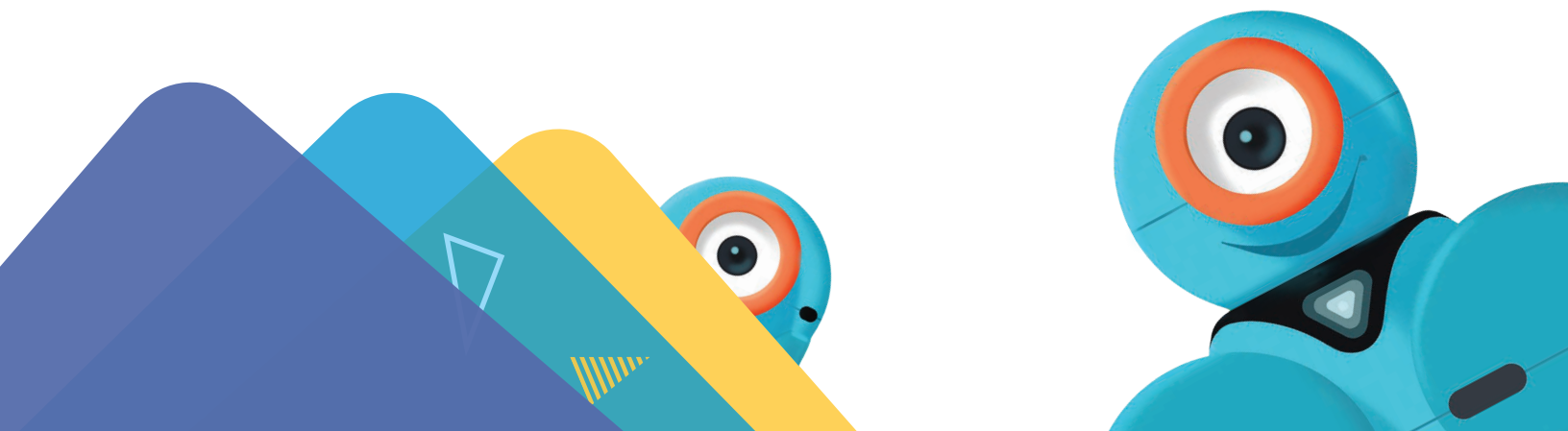


Dash's Road Trip (F 2.1 - F 2.3): Teacher Packet

In this packet, we've included resources that will help you and your students as they independently complete the **Dash's Road Trip 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





Road Trip!

Dash needs to get to the gas station to fill up before a big road trip!



1. To get to the gas station, Dash first needs to **drive 65 cm**. To drive a specific distance, **set** a **variable** to 65.



2. Then have Dash **drive** that distance.



3. Almost there! Now program Dash to make a **45-degree left turn**.



4. Finally, Dash needs to **drive 35 cm**. What blocks should you use?



5. Add some **animations** for when Dash arrives at the gas station!

Road Trip!

Time: 15 minutes

Hints

- You can find the **Set Variable** block and the **Do Drive Variable** block in the **Variables** menu.
- Variables** allow us to store specific information in a program. In the **Blockly** app, the variables are represented by different fruits: orange, banana, apple, cherry, and watermelon.
- Make sure you use the same variable for the **Do Drive Variable** and **Set Variable** blocks.
- You will need 2 **Set Variable** and 2 **Do Drive Variable** blocks for this program because Dash needs to travel 2 distances.



- To get to the gas station, Dash first needs to **drive 65 cm**. To drive a specific distance, **set** a **variable** to 65.



- Then have Dash **drive** that distance.



- Almost there! Now program Dash to make a **45-degree left turn**.

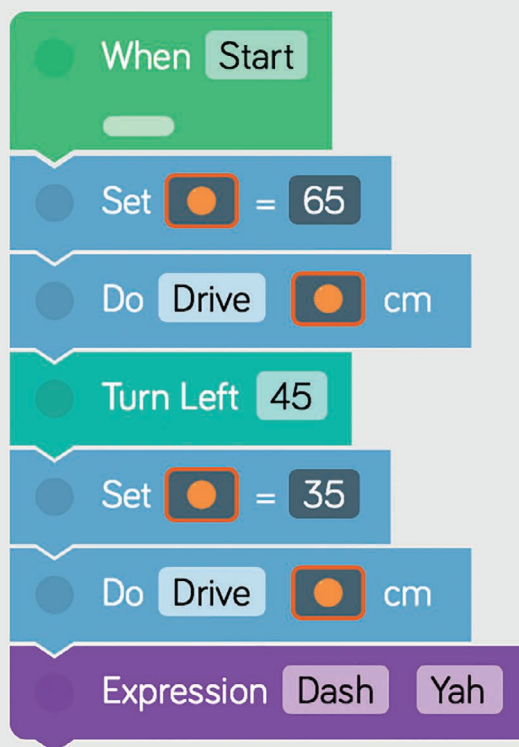


- Finally, Dash needs to **drive 35 cm**. What blocks should you use?



- Add some **animations** for when Dash arrives at the gas station!

Suggested Solution:



Discussion Questions

1. What if you wanted to use a different variable to set the **35 cm distance** in the challenge? How would you change the program to include a second variable?
2. How could variables be used in other programs? For example, how could you use variables to keep track of how often something happens to Dash or Dot?

Cross-Curricular Connections

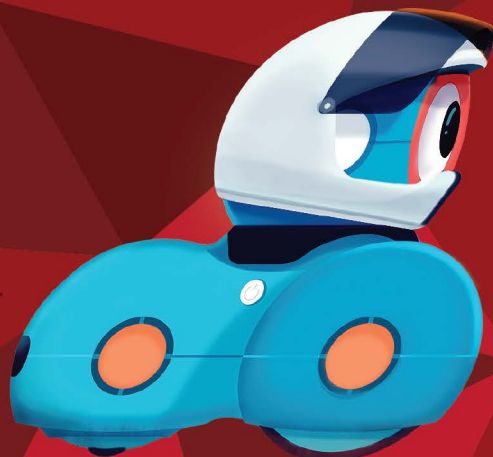
MATH

- Have students use a coordinate grid to mark Dash's starting point and the location of the gas station. Then have them plot out a path from Dash to the gas station and use coordinate numbers to describe the route. (CCSS.MATH.5.G.A.1)
- Have students design a new path to the gas station using at least 2 acute turns and 2 obtuse turns. Then have them calculate the sum of all of the angles Dash turned on the path. (CCSS.MATH.4.MD.C.5, CCSS.MATH.4.MD.C.6, CCSS.MATH.4.MD.C.7)

ELA

- Have students write a narrative that describes why and where Dash is going on the big road trip. Encourage them to use descriptive and sensory details. (CCSS.ELA-W.5.3)

NOTES:



Pump It Up!

Dash needs to fill up the gas tank before heading out on the open road!



1. Let's help Dash get ready for the road trip!
Start with **2 When** blocks:



2. **When the Top Button** is pressed, use a **variable** to help fill up Dash's tank,



and then use the **eye lights** to show how much gas is in the tank.



3. **When Button 3** is pressed, program Dash to make an engine **sound** and **drive** the distance of the **variable**. Then **set** the variable back to **0**.

Now, play the program! Press the **Top Button** several times to fill up Dash's gas tank. Then press **Button 3** to make Dash go!

Pump It Up!

Time: 20 minutes

Hints

- To change the variable, tap the **Change Variable** block and enter **13** into the number pad.
- To select **Eye Light On**, tap on the **Do** block and select the last item. Then Dash's eye lights will show the variable number.
- Make sure you use the same variable for the **Change Variable**, **Do Eye Light On Variable**, **Do Drive Variable**, and **Set Variable** blocks.

F | 2.2 Variables

1. Let's help Dash get ready for the road trip!
Start with **2 When** blocks:

When Dash Top Button

When Dash Button 3

2. **When the Top Button** is pressed, use a **variable** to help fill up Dash's tank,

Change by + 13

and then use the **eye lights** to show how much gas is in the tank.

Do Eye Light On

3. **When Button 3** is pressed, program Dash to make an engine **sound** and **drive** the distance of the **variable**. Then **set** the variable back to **0**.

Now, play the program! Press the **Top Button** several times to fill up Dash's gas tank. Then press **Button 3** to make Dash go!

Suggested Solution:

When Start

When Dash Top Button

When Dash Button 3

Change by + 13

Transport Dash Car Engine

Do Eye Light On

Do Drive cm

Set = 0

Discussion Questions

1. The **Do Eye Light On Variable** block uses Dash's eye lights to show the variable number. Why do you think the challenge has us add **13** to the variable each time the **Top Button** is pressed? (Hint: There are 12 eye lights. Each eye light represents a number. The top light represents the number 0 and then increases clockwise. The **Do Eye Light On Variable** block divides the **variable** by 12 and shows the remainder. For example, when the variable is 13, $13 \div 12 = 1$ with a remainder of 1. Thus, the eye light for 1 is turned on.)
2. What would happen if you changed the variable using a different operator (e.g., subtraction, multiplication, division)? How would it affect the way Dash moved after **Button 3** is pressed?

Cross-Curricular Connections

MATH

- Give students random numbers between 13-900. Then have them predict how the eye light pattern will display each of the numbers, given that the **Do Eye Light On Variable** block divides the variable by 12 and shows the remainder. (CCSS.MATH.5.G.A.1)
- Have students press the **Top Button** 10 times and record the variable number each time. Then have them predict the values of the next 10-20 presses. Students can check their answers by running the program. (CCSS.MATH.5.OA.B.3)

ELA

- In real life, Dash runs on electric power, but in the story, Dash runs on gas. Have students write an opinion piece about the pros and cons of using electric versus gas-powered vehicles. (CCSS.ELA.W.5.1)
- Have students write a narrative that describes a road trip they've taken. If the students haven't been on a road trip, have them describe where they would go and what they would do on a dream road trip. Encourage them to use descriptive and sensory details. (CCSS.ELA.W.5.3)

NOTES:



On the Road!

Dash is on the road but keeps running out of gas! How can you help Dash know when the gas in the tank is getting low?



1. Program Dash so that pressing the **Top Button**:

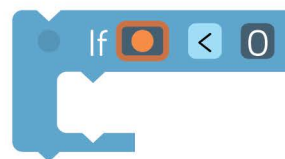


- **adds** gas to the tank and **changes** the **variable** by **+13**.
- uses the **eye lights** to **show** how much gas is in the tank.

2. Then program Dash so that pressing **Button 3**:



- has Dash **drive forward 50 cm**.
- **lowers** the amount of gas in the tank and **changes** the **variable** by **-50**.
- checks to see **if** the **variable** is **<0**.



If the tank is empty, then:

- turn **off** all **lights**.
- have Dash make a warning **sound**.
- **set** the **variable** back to **0**.

Now Dash knows when the gas tank needs to be filled back up! Vroooooom!

On the Road!

Time: 25 minutes

Hints

- Use Challenge Card F 2.2 to help you set up this program.
- To change the variable, tap the **Change Variable** block. Then you can add, subtract, multiply, or divide the variable. To **subtract**, tap the negative sign and then tap the number you want to subtract in the key pad.
- The **If Variable** block checks the value of the variable. Tap the **If Variable** block and change the operator to **<** in the menu. This will check if the variable is **less than 0**.

Suggested Solution:

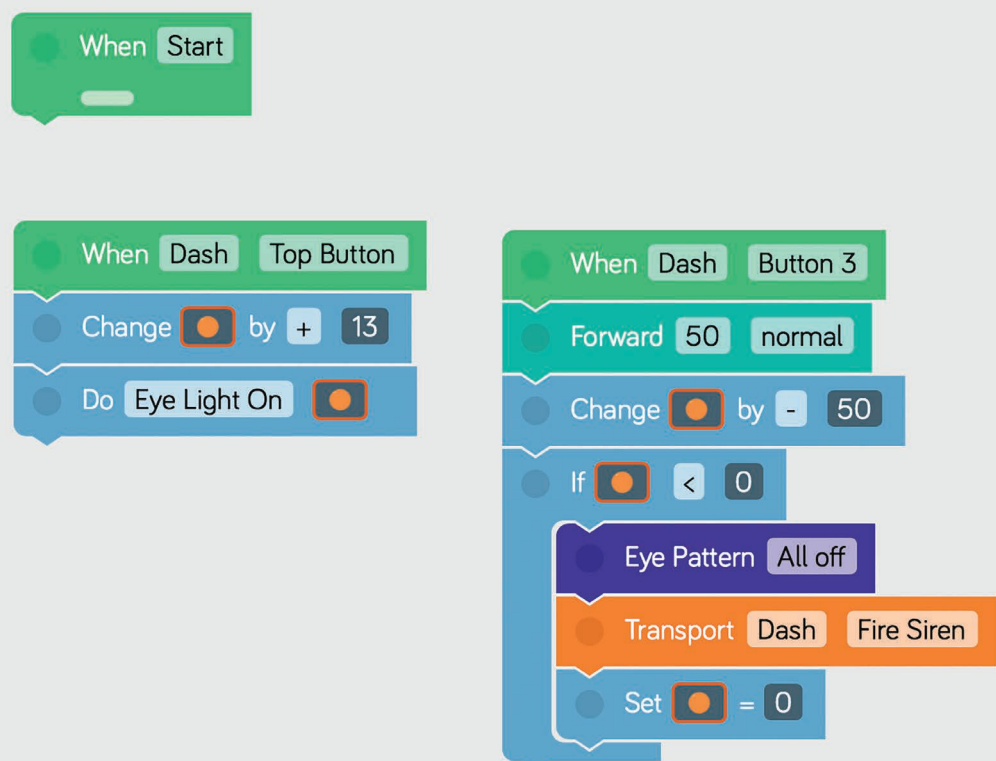
F | 2.3 Variables

1. Program Dash so that pressing the **Top Button**:
 - adds gas to the tank and **changes** the **variable** by **+13**.
 - uses the **eye lights** to **show** how much gas is in the tank.
2. Then program Dash so that pressing **Button 3**:
 - has Dash **drive forward 50 cm**.
 - **lowers** the amount of gas in the tank and **changes** the **variable** by **-50**.
 - checks to see **if** the **variable** is **<0**.

If **<** **0** **If** the tank is empty, then:

 - turn **off** all **lights**.
 - have Dash make a warning **sound**.
 - **set** the **variable** back to **0**.

Now Dash knows when the gas tank needs to be filled back up! Vroooooom!



```

When Start
  Change [gas] by + 13
  Do Eye Light On [gas]

When Dash Top Button
  Change [gas] by + 13
  Do Eye Light On [gas]

When Dash Button 3
  Forward 50 normal
  Change [gas] by - 50
  If [gas] < 0
    Eye Pattern All off
    Transport Dash Fire Siren
    Set [gas] = 0
  
```

Discussion Questions

1. Why do you need to **subtract 50** from the variable each time Dash moves **forward 50 cm**? How does this mimic the way gas is used to power cars?
2. How can you fill up Dash's gas tank faster? How would that change the way the eye lights show how much gas is in the tank?

Cross-Curricular Connections

MATH

- Give students random numbers between 13-900. Then have them predict how the eye light pattern will display each of the numbers, given that the **Do Eye Light On** block divides the variable by 12 and shows the remainder. (CCSS.MATH.5.NBT.B.6)
- Have students predict the value of the variable depending on the number of times Dash's **Top Button** and **Button 3** are pressed. (E.g., Top Button is pressed 9 times = variable of 117, Button 3 is pressed 2 times = variable of -100. Then the final value of the variable is 17.) (CCSS.MATH.5.OA.B.3)

ELA

- Have students research and write about how gas is used to power cars. Then have them describe how they would change Dash's program to better reflect the way gas is used by cars. (CCSS.ELA.W.5.2)
- Have students write a narrative about what they would do if they ran out of gas on a road trip and were far away from any gas stations. Encourage them to use descriptive and sensory details. (CCSS.ELA.W.5.3)

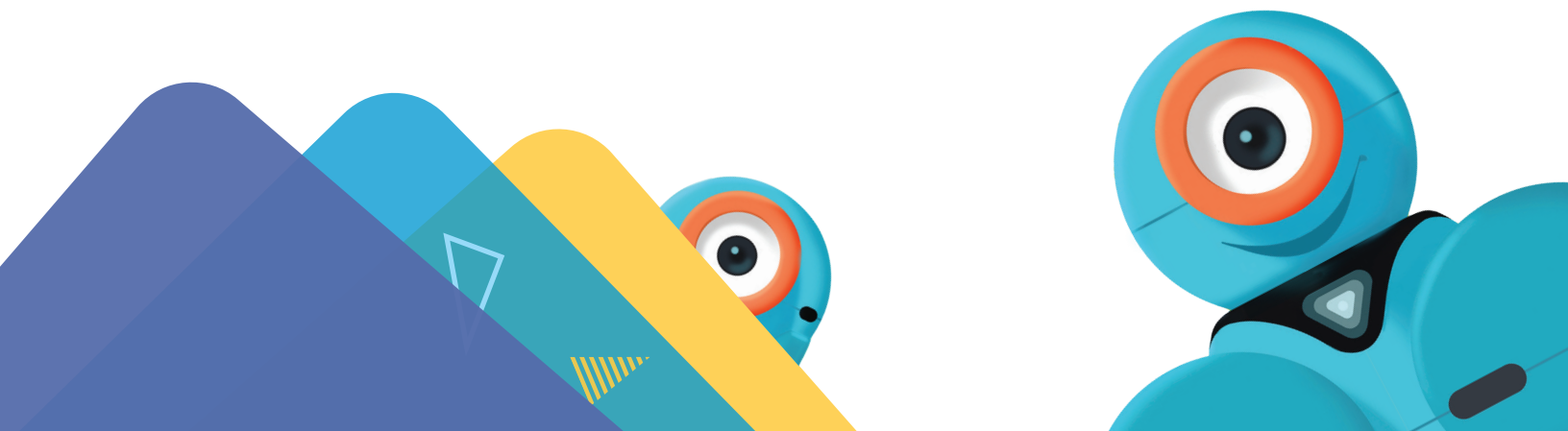
NOTES:

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

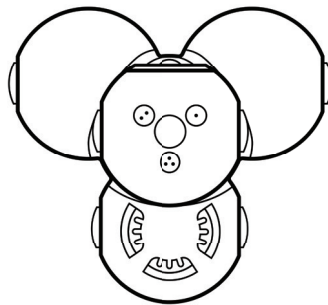
Dash Planning Worksheet

Name(s): _____ Date: _____

Coding Level: _____ Card #: _____

What do you want Dash 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

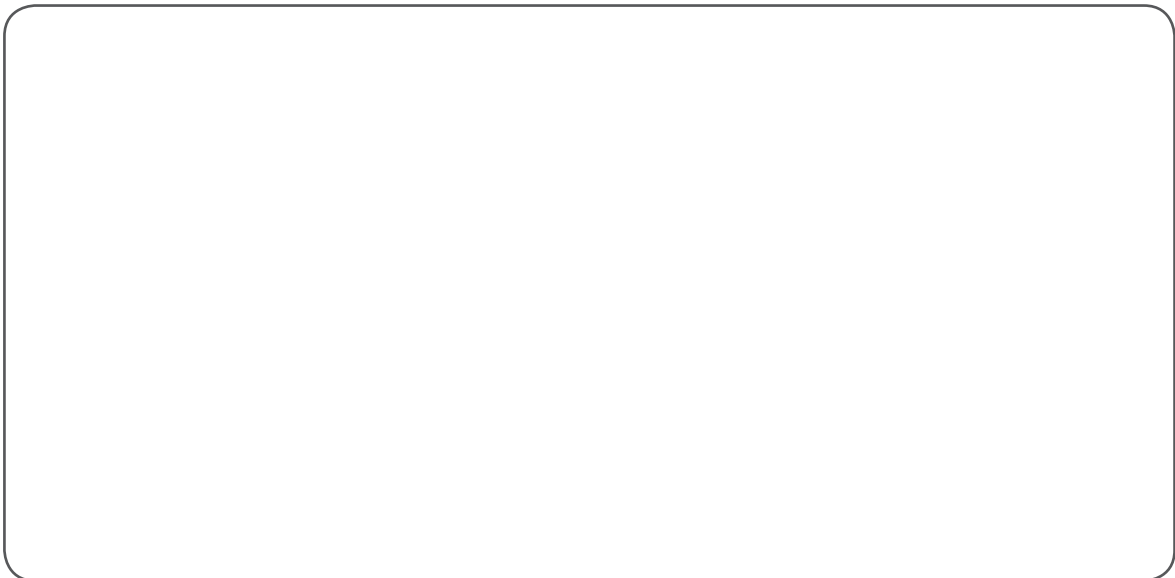
Name(s): _____ Date: _____

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.



dash & dot[™] challenge cards

Coding Lessons

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

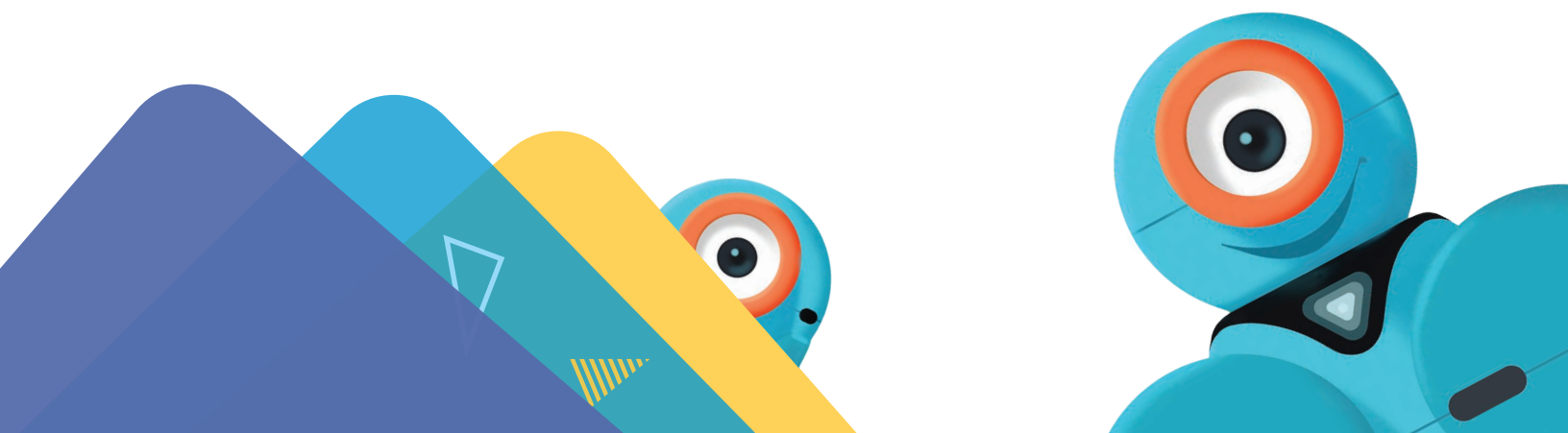
To check out more free coding lessons, please visit:

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 F - Lesson 3

Single Variables

Lesson Information

Overview/Description

Students use **variables** in a program.

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

Coding Level: F

Prior Experience:

Before this lesson, it is recommended that students complete:

- Level A–E: All Lessons
- [Optional] Level A–E **Challenge Cards**: All
- [Optional] Level F **Challenge Cards**: 1.1–1.3

Learning Objectives:

Students will:

- Define a **variable**.
- Understand the benefits of using **variables**.
- Use **variables** to revise code and complete coding challenges in an iterative process.

Target Grade Range: 3–5

Suggested Group Size: 2–3 students per robot

Time Required: 45–60 minutes

Materials:

- 1 Dash robot per group
- 1 tablet per group
- projector or interactive display with mirroring capability
- pencils
- **Challenge Cards**: F 2.1, F 2.2, F 2.3
- [Optional] access to **Twitter** and **Instagram**

Resources/Downloads:

- **Troubleshooting** handout
- **Problem Solving & Debugging** handout
- **Wonder Journal: Dash Planning** and/or **General Planning** worksheets
- **Wonder Journal: Reflection** worksheets
- **Blockly Puzzle Checklist**
- **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

VARIABLES REVIEW

1. Briefly review how students used **variables** in the previous lesson by asking:
 - ""How did we use **variables** to tell a story?" (Sample response: "We used variables to hold places for certain types of words.")
 - "How are **variables** helpful in coding?" (Sample response: Variables store information in our code.")
2. Say, "Today, we will use variables to program Dash."

BLOCKLY IN-APP PUZZLE [Optional]

1. Have students complete the **Troll of Trepidation** puzzle. You can find walkthroughs, solutions, and extensions for the puzzle here: <https://drive.google.com/open?id=0B7m0xQXRpsQIRDhYTkRqOU1wMzQ>.
2. Have students update their **Blockly Puzzle Checklists**.
3. Note: If previous puzzles are locked, show students how to quickly unlock them using this strategy: https://help.makewonder.com/customer/en/portal/articles/2318965-how-can-i-reset-challenges-unlock-free-play-or-unlock-wonder-cloud-?b_id=14312.

CODE.ORG CONNECTIONS [Optional]

1. Review *Code.org's* lesson focusing on **variables**: <https://studio.code.org/s/coursef-draft/stage/6/puzzle/3>. Then ask:
 - "How were **variables** used in these puzzles?" (Sample Response: "The **variables** held a specific measurement so we could create an equal-sided shape.")
 - "Using the same program, could we have used the same **variable** to create a rectangle?" (Sample response: "No, we could not have used a variable to create a rectangle in this program as a rectangle has sides of two different lengths, and a

variable can only hold one piece of information. We would have needed two different variables.”)

Direct Instruction

INTRODUCTION

1. Say, “A **variable** is like a bag; it keeps information stored inside of it. However, there are many ways variables can change. You can add, subtract, multiply, or divide them. You can even use them to help programs make decisions. For example, you can tell the program that **if** the **variable** is greater than a certain number, **then** something will happen.”
2. Brainstorm with students to come up with real life examples of variables. For example:
 - “My wallet can have a variable: the amount of money that’s inside it. First, my wallet has 20 dollars inside it, so the variable is 20 dollars. Then I buy a bag of lollipops and take out 5 dollars. Now the value of my variable is 15 dollars!”
3. Say, “For this activity we are going to use **variables** inside **conditionals**.”

QUICK CHECK

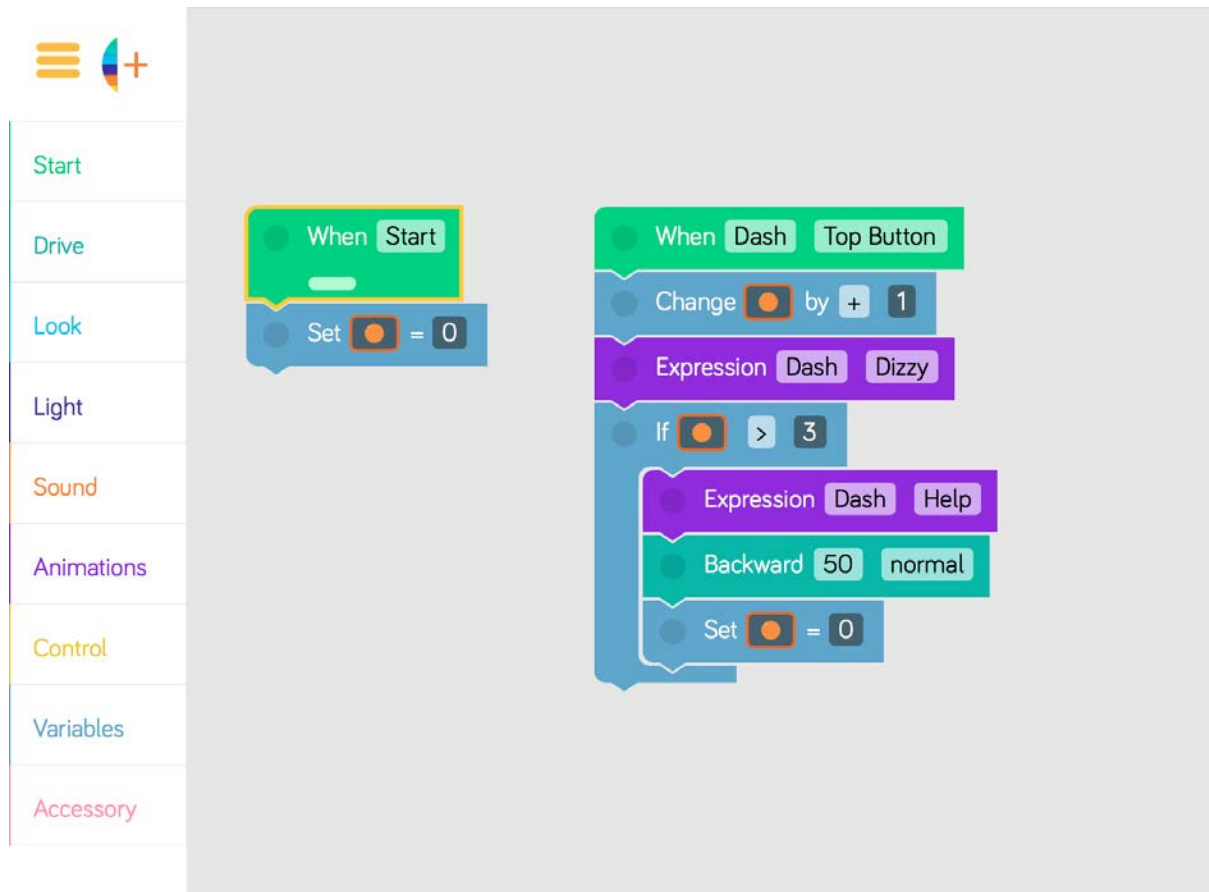
- What are **conditionals**? (Possible response: “Conditionals are statements that help computers make decisions.”)
- What are **variables**? (Possible response: “Variables hold numbers that can be changed.”)

Guided Practice

ACTIVITY: DIZZY DASH

1. Project your tablet screen, open the *Blockly* app, and go to the menu at the top left of the screen.

2. Go to the **Create New Project** menu and select the **Dizzy Dash** preset program. Then tap “create.”



3. Ask the students to look at the program and predict what they think will happen when it is run.

- Run the program to see if the students' predictions were correct.

4. Ask, “What happens when we start the program?”

- Sample response: “The **variable** is set to 0.”

5. Ask, “What happens when Dash’s **Top Button** is pressed?”

- Sample response: “The variable increases by 1 and Dash turns in a circle.”)

6. Ask, “What happens after you press Dash’s **Top Button** more than 3 times?”

- Sample response: “Dash says ‘Help,’ moves backward, and the variable resets to 0.”

7. Ask, “How can you change the code so that the **variable** increases by 2 each time you press the Top Button?”

- Sample response: “You can tap the **Change Variable** block to change how much the variable increases.”

QUICK CHECK

- How are **variables** used in coding? (Sample response: "Variables save pieces of information.")
- How are **variables** used in your life? (Sample response: "I use a variable to keep track of my score in a video game or board game.")

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!

DIZZY DASH EXTENSION

1. Have students use the **Wonder Journal: Dash Planning** and/or **General Planning** worksheets to design ways to add to or alter the **Dizzy Dash** preset program. They can:

- Change the **animations** and **movements**.
- Add **lights** and **sounds**.
- Change how much the **variable** increases or decreases.
- Add an **event handler**.
- Change the **cues**.

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 Dash 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**:

- **F 2.1: Road Trip**
- **F 2.2: Pump It Up!**
- **F 2.3: On the Road!**

3. For each challenge, encourage students to:

- Use the **Wonder Journal: Dash Planning** and/or **General Planning** worksheets 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 Dash while running their code.

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 changed the **variable** amounts because we wanted the program to skip count using larger numbers.")
 - Share any obstacles and difficulties they overcame during the activity. (E.g., "At first, we didn't know how to increase or decrease the **variable**, but then we tapped on the **Change Variable** block and found where to change the amount.")
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 (add and/or change a **variable**)?"
 - "I like how you (changed the **animations**)."
 - "What if you (changed the **conditional** block)?"

Follow-Up Questions/Discussion

- Could we have written the same program *without* using **variables**? (Sample Response: "No, because **variables** are the only way we can store information in a code.")
- How was the **conditional** block used in this program? (Sample response: "The **conditional** block said that if the **variable** was greater than 3, **then** Dash would say, 'Help!' and move backwards.")
- What would happen if you programmed the **variable** to change by 3 instead of 1? Then what would you have to do to make Dash say, "Help!?" (Sample response: "If you programmed the **variable** to increase by 3, then Dash would say, 'Help!' after hitting the **Top Button** only twice.")

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

- Create programs that include sequences, events, loops, and conditionals.
- Create programs that use variables that store and modify data.
- Modify, mix and incorporate portions of an existing program into one's own work to develop something new or add more advanced features.
- Take on varying roles with teacher guidance, when collaborating with peers during the design, implementation, and review of program development.

ISTE

- 4d: Exhibit a tolerance for ambiguity, perseverance and the capacity to work with open-ended problems.
- 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: At whatever stage, communicating with peers about proposed solutions is an important part of the design process, and shared ideas can lead to improved designs.

Common Core

- CCSS.ELA-LITERACY.W.2.6: With guidance and support from adults, use a variety of digital tools to produce and publish writing, including in collaboration with peers.
- 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.