ozobot bit

OZOBLOCKLY TRAINING

K-1 | Computer Science | 40 Minutes | Beginner

OVERVIEW

Acquaint your students with OzoBlockly using our picture-based blocks that allow them to program Bit, even if they cannot read yet. Using a deconstruction method, students will learn about sequential code and movement while becoming acquainted with Ozobot's block-based editor.

CS TOPICS

Sequential coding, Movement

PREREQUISITES

None

GROUPING

Individuals or pairs

MATERIALS

- Tablet or desktop computer with internet access, one per group
- Printout of map, one per group
- Bit by Ozobot, one per group
- Set of large printout-blocks (attached), cut out
- Tape or magnets

PREPARATION

- You will need to print a copy of the map for each individual or group.
- Print and cut out the large paper blocks that will be used to practice as a class.
- Pre-load http://ozoblockly.com/editor#se47c5 onto the bots and calibrate them to the screen.
- Have the program pre-populated onto the tablets or computers your students will be using by putting navigating to http://ozoblockly.com/editor#se47c5. When students open the browser in class, the program will still be in the editor.
- Have a screen with the Ozoblockly editor (ozoblockly.com/editor), toggled to "Bit" in the top left corner, up to go through each category of blocks (movement, light effects) (optional) and the loading process.
- *Optional: For information on calibrating and flash-loading on a digital device:* http://files.ozobot.com/stem-education/ozoblockly-getting-*started.pdf*

INSTRUCTION

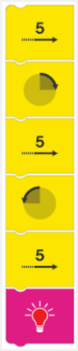
Part 1

EXPLAIN: Today we are going to learn how to tell Ozobot what to do using a computer (or tablet). Telling a robot what to do is called **Programming.** A program gives instructions to Ozobot, kind of like a to-do list that needs to be done in order.
Go Deeper: If you have used Ozobot with markers and paper you can discuss how that was also programming.

DO: Show students the program they are about to run on their bots by displaying the cut-out movement blocks on the board with tape or magnets. Demonstrate how the blocks are moved around, and maybe even have students try to "be the robot" and follow the instructions on the board

Pass out the maps and point out the "directions" Ozobot was given to find the way home. Have them double-click the power button (you may want to take some time to practice double-clicking to run the program) and place their bots on the start. The program as-is will lead Ozobot into the woods.

Original



ASK: Did Bit make it home? Which blocks are telling Bit to do what?

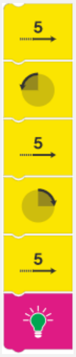
Part 2

DO: Have students take out their tablets (already populated with the program the http://ozoblockly.com/editor#se47c5), select the Bit toggle under the OzoBlockly logo on the left, and select "1."
Optional: Walk through the different block categories in Mode 1 to talk about what each type of block does.

EXPLAIN: Bit was given the wrong directions and needs your help to find the way home! Look at the instructions (program) and make changes to two of the blocks to make sure Bit makes it home!
Optional: Make the program together using the physical blocks and then have students make it on their computer/tablet.

DO: Walk around to monitor progress and help with navigating, moving, and deleting blocks. When students are ready to load their programs, project the OzoBlockly editor so the whole class can see, and demonstrate the loading process by pointing out the flashing button, setting their bots on the white spot, powering it on (click the power button once), and pressing "load." When it is finished flashing and displays a sloid green light, they can run their program. If they are unsuccessful on their first attempt, have them "debut" their program until Bit ends up in the home.

Solution



ACADEMIC STANDARDS

CSTA
2-CS-01 Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.
2-CS-03 Systematically identify and fix problems with computing devices and their components
2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.
2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.
2-AP-17 Systematically test and refine programs using a range of test cases.

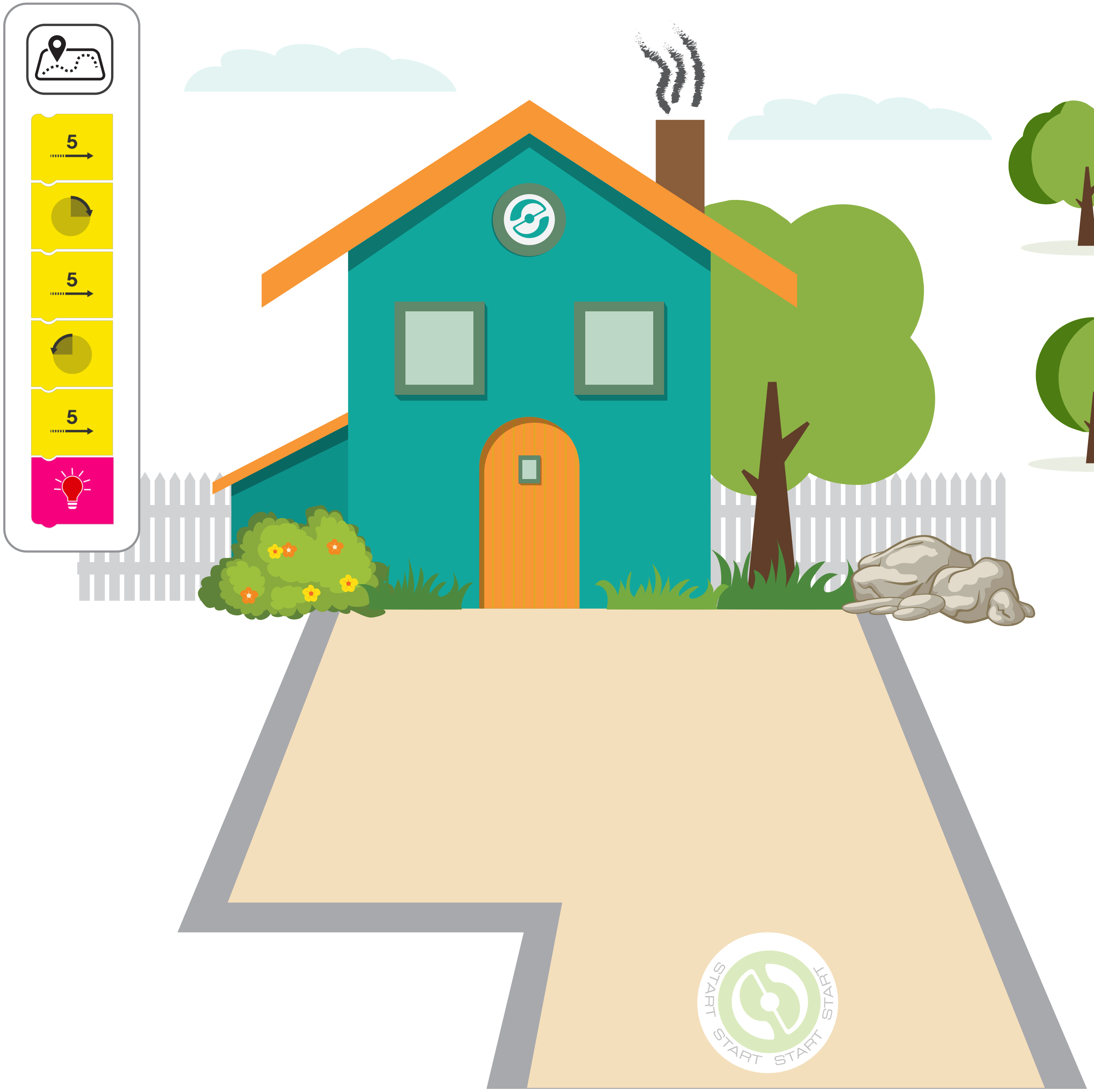
ISTE
1 a. Articulate and set personal learning goals, develop strategies leveraging technology to achieve them and reflect on the learning process itself to improve learning outcomes.
b. Build networks and customize their learning environments in ways that support the learning process.
c. Use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways.
4 a. Know and use a deliberate design process for generating ideas, testing theories, creating innovative artifacts or solving authentic problems.
b. Select and use digital tools to plan and manage a design process that considers design constraints and calculated risks.
d. exhibit a tolerance for ambiguity, perseverance and the capacity to work with open-ended problems.
5 c. Break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving.
6c. Communicate complex ideas clearly and effectively by creating or using a variety of digital objects such as visualizations, models or simulations.
d. Publish or present content that customizes the message and medium for their intended audiences.
7 a. use digital tools to connect with learners from a variety of backgrounds and cultures, engaging with them in ways that broaden mutual understanding and learning.
b. Use collaborative technologies to work with others, including peers, experts or community members, to examine issues and problems from multiple viewpoints.
c. Contribute constructively to project teams, assuming various roles and responsibilities to work effectively toward a common goal.

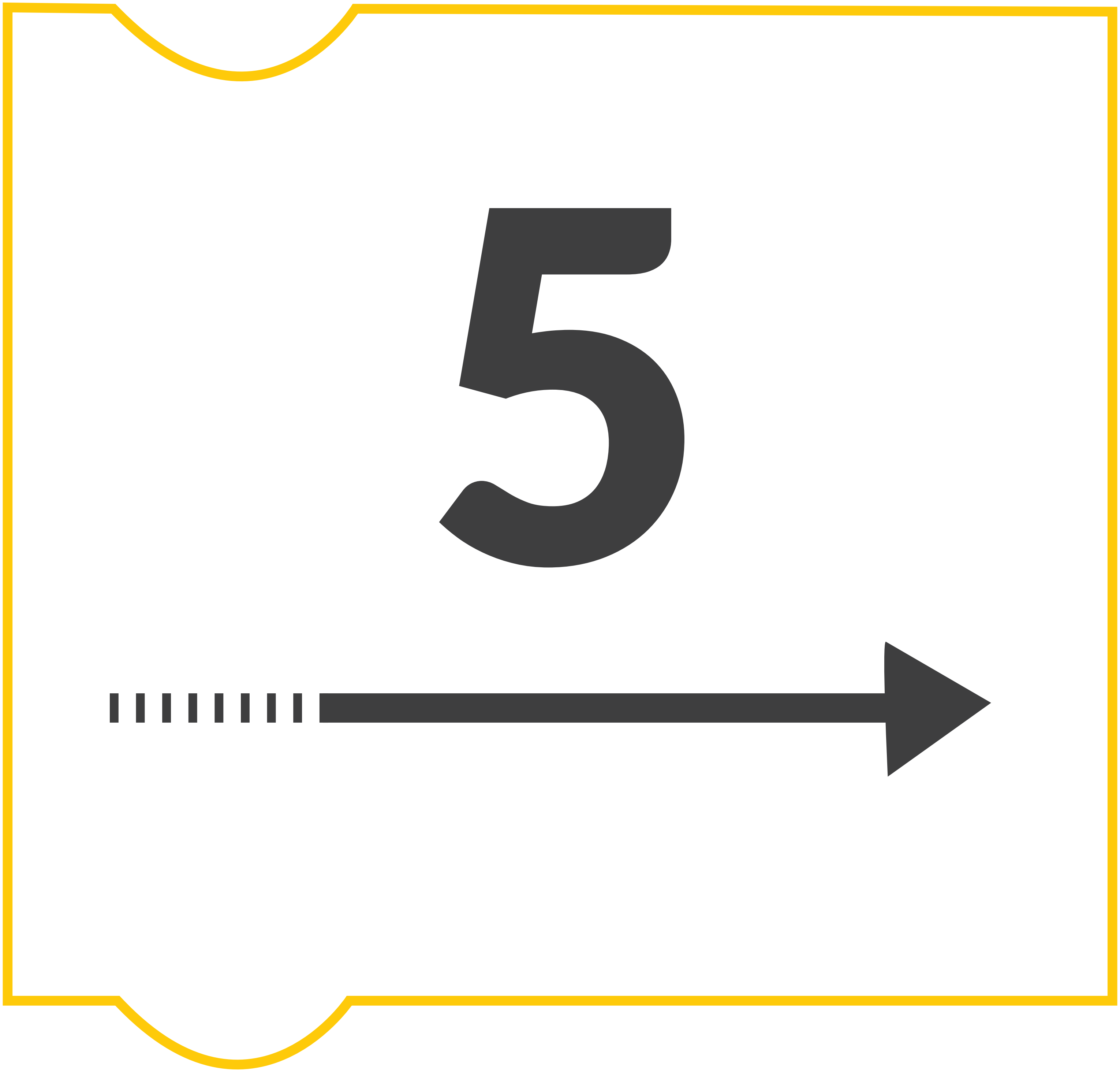
CCSS STANDARDS FOR MATHEMATICAL PRACTICES

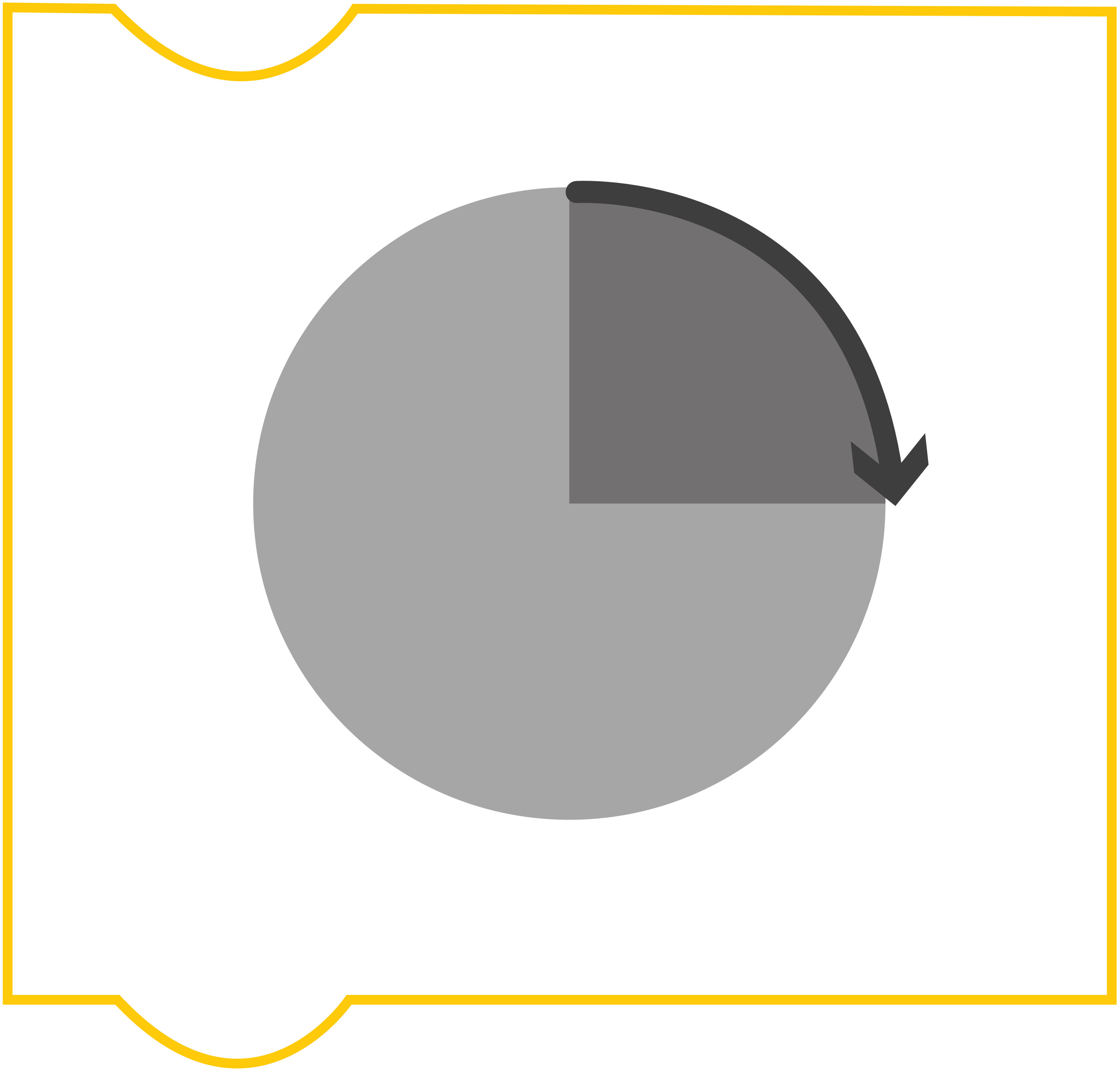
1. Make sense of problems and persevere through solving them
2. Reason abstractly and quantitatively
3. Construct viable arguments and critique the reasoning of others
4. Model with mathematics
5. Use appropriate tools strategically
6. Attend to precision
7. Look for and make use of structure
8. Look for and express regularity in repeated reasoning

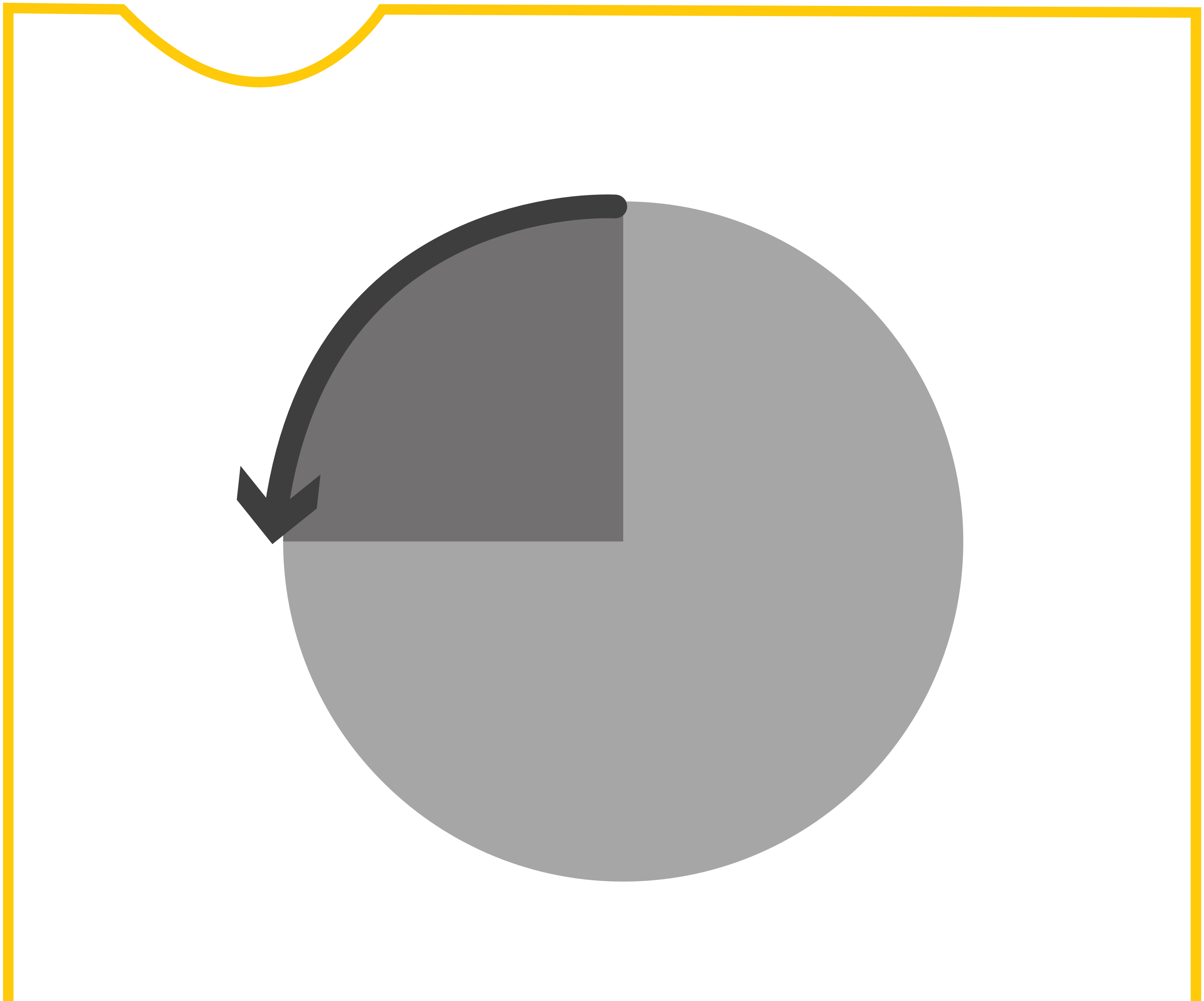
QUESTIONS ABOUT THIS LESSON?

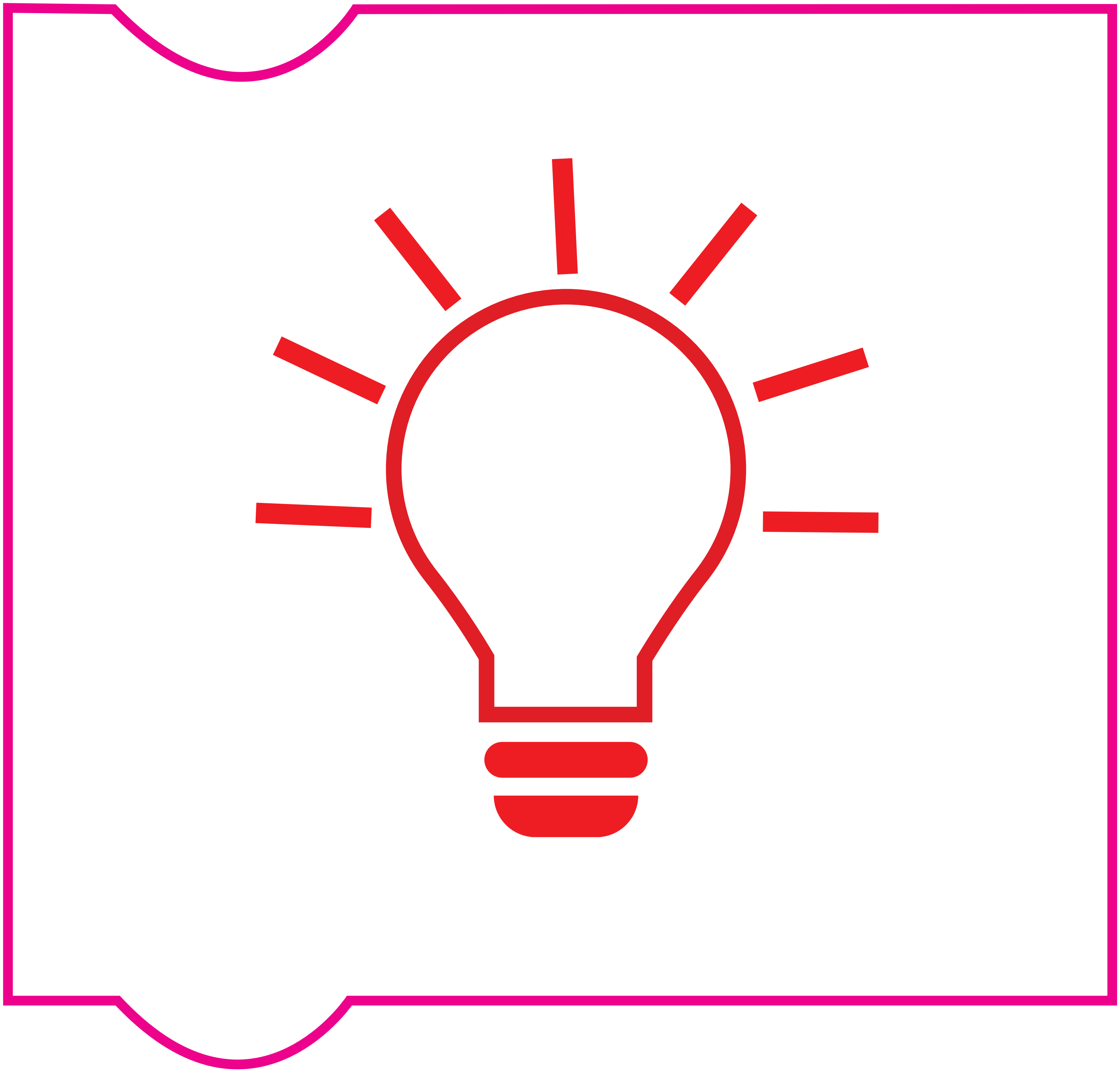
Please contact us at ozoedu@ozobot.com

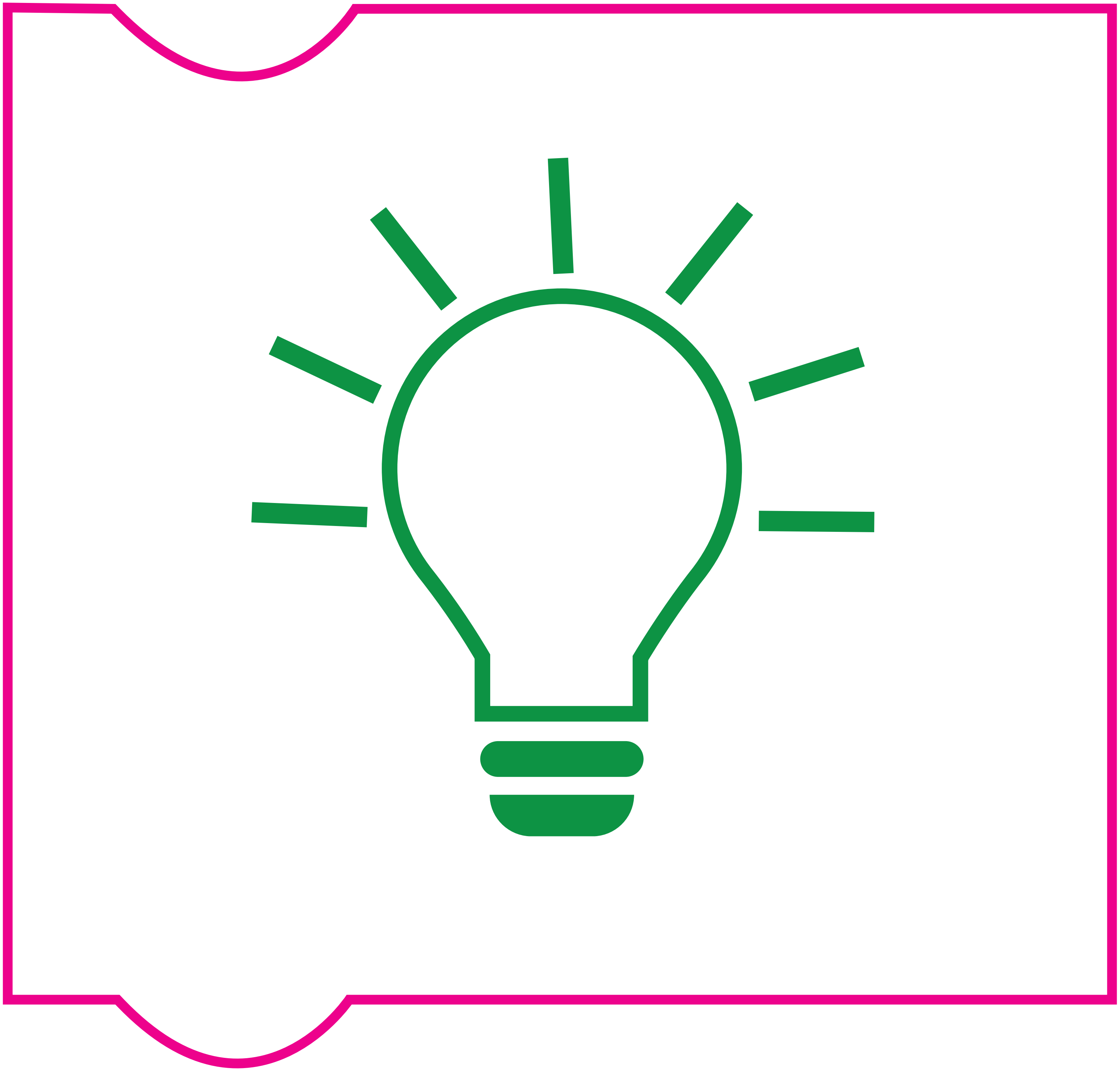

5
5
5
START START START


5









ozobot evo

OZOBLOCKLY TRAINING

K-1 | Computer Science | 40 Minutes | Beginner

OVERVIEW

Acquaint your students with OzoBlockly using our picture-based blocks that allow them to program Evo, even if they cannot read yet. Using a deconstruction method, students will learn about sequential code and movement while becoming acquainted with Ozobot's block-based editor.

CS TOPICS

Sequential coding, Movement

PREREQUISITES

None

GROUPING

Individuals or pairs

MATERIALS

- Tablet or desktop computer with internet access, one per group
- Printout of map, one per group
- Evo by Ozobot, one per group
- Set of large printout-blocks (attached), cut out
- Tape or magnets

PREPARATION

- You will need to print a copy of the map for each individual or group.
- Print and cut out the large paper blocks that will be used to practice as a class.
- Pre-load http://ozoblockly.com/editor#4w3nej onto the bots and calibrate them to the screen.
- Have the program pre-populated onto the tablets or computers your students will be using by putting navigating to http://ozoblockly.com/editor#4w3nej. When students open the browser in class, the program will still be in the editor.
- Have a screen with the Ozoblockly editor (ozoblockly.com/editor), toggled to "Evo" in the top left corner, up to go through each category of blocks (movement, light effects) (optional) and the loading process.
- *Optional: For additional information on calibrating and flash-loading on a digital device:* http://files.ozobot.com/stem-education/ozoblockly-getting-*started.pdf*

INSTRUCTION

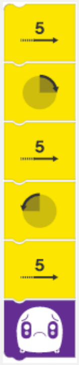
Part 1

EXPLAIN: Today we are going to learn how to tell Ozobot what to do using a computer (or tablet). Telling a robot what to do is called **Programming.** A program gives instructions to Ozobot, kind of like a to-do list that needs to be done in order.
Go Deeper: If you have used Ozobot with markers and paper you can discuss how that was also programming.

DO: Show students the program they are about to run on their bots by displaying the cut-out movement blocks on the board with tape or magnets. Demonstrate how the blocks are moved around, and maybe even have students try to "be the robot" and follow the instructions on the board

Pass out the maps and point out the "directions" Ozobot was given to find the way home. Have them double-click the power button (you may want to take some time to practice double-clicking to run the program) and place their bots on the start. The program as-is will lead Ozobot into the woods.

Original



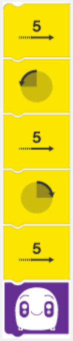
ASK: Did Evo make it home? Which blocks are telling Evo to do what?

Part 2

DO: Have students take out their tablets (already populated with the program the http://ozoblockly.com/editor#4w3nej), select the Evo toggle under the OzoBlockly logo on the left, and select "1."
Optional: Walk through the different block categories in Mode 1 to talk about what each type of block does.

EXPLAIN: Evo was given the wrong directions and needs your help to find the way home! Look at the instructions (program) and make changes to two of the blocks to make sure Evo makes it home!
Optional: Make the program together using the physical blocks and then have students make it on their computer/tablet.

Solution



DO: Walk around to monitor progress and help with navigating, moving, and deleting blocks. When students are ready to load their programs, project the OzoBlockly editor so the whole class can see, and demonstrate the loading process by pointing out the flashing button, setting their bots on the white spot, powering it on (click the power button once), and pressing "load." When it is finished flashing and displays a sloid green light, they can run their program. If they are unsuccessful on their first attempt, have them "debut" their program until Evo ends up in the home.

ACADEMIC STANDARDS

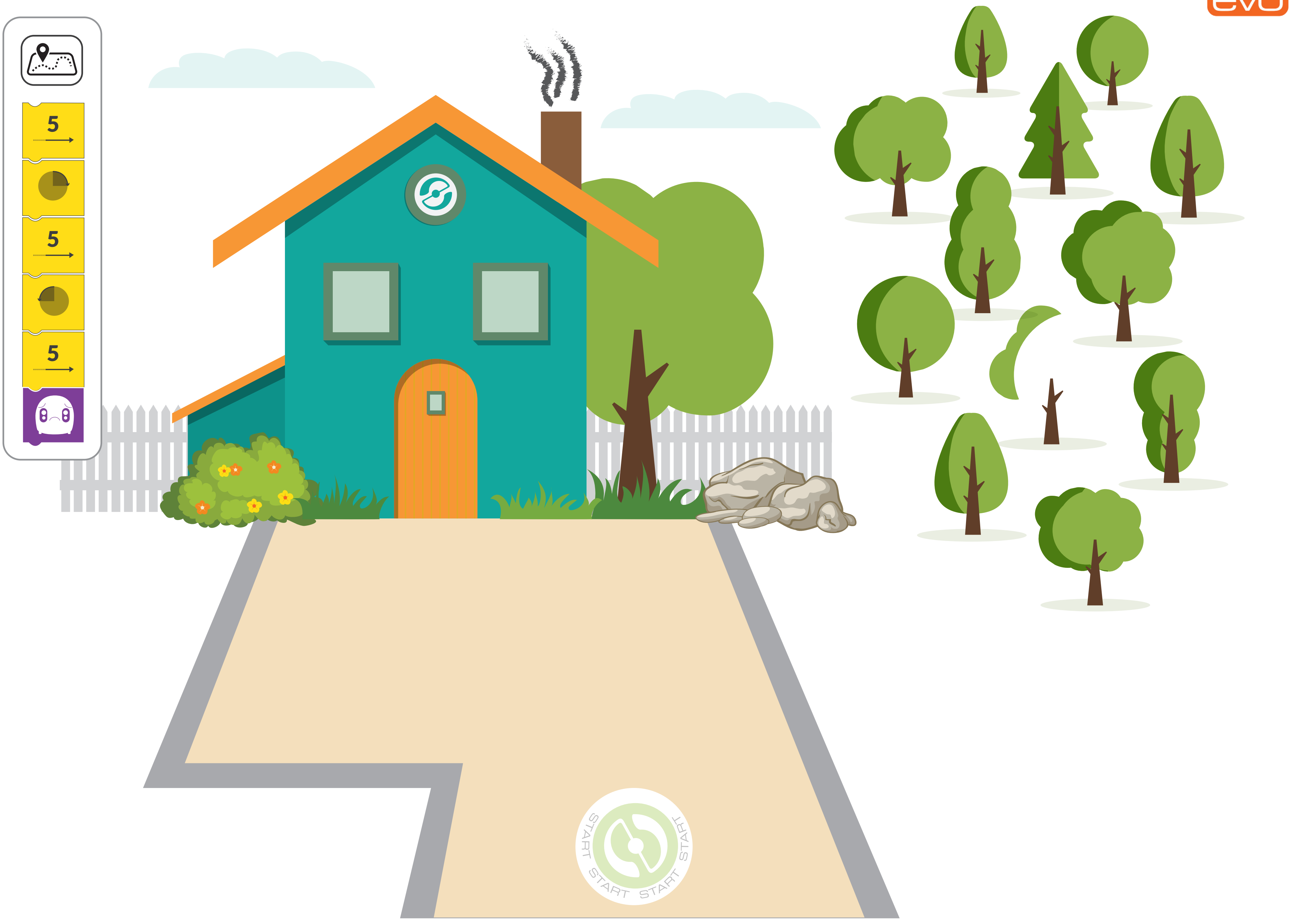
CSTA
2-CS-01 Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.
2-CS-03 Systematically identify and fix problems with computing devices and their components
2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.
2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.
2-AP-17 Systematically test and refine programs using a range of test cases.
ISTE
1 a. Articulate and set personal learning goals, develop strategies leveraging technology to achieve them and reflect on the learning process itself to improve learning outcomes.
b. Build networks and customize their learning environments in ways that support the learning process.
c. Use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways.
4 a. Know and use a deliberate design process for generating ideas, testing theories, creating innovative artifacts or solving authentic problems.
b. Select and use digital tools to plan and manage a design process that considers design constraints and calculated risks.
d. exhibit a tolerance for ambiguity, perseverance and the capacity to work with open-ended problems.
5 c. Break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving.
6c. Communicate complex ideas clearly and effectively by creating or using a variety of digital objects such as visualizations, models or simulations.
d. Publish or present content that customizes the message and medium for their intended audiences.
7 a. use digital tools to connect with learners from a variety of backgrounds and cultures, engaging with them in ways that broaden mutual understanding and learning.
b. Use collaborative technologies to work with others, including peers, experts or community members, to examine issues and problems from multiple viewpoints.
c. Contribute constructively to project teams, assuming various roles and responsibilities to work effectively toward a common goal.
CCSS STANDARDS FOR MATHEMATICAL PRACTICES

1. Make sense of problems and persevere through solving them
2. Reason abstractly and quantitatively
3. Construct viable arguments and critique the reasoning of others
4. Model with mathematics
5. Use appropriate tools strategically
6. Attend to precision
7. Look for and make use of structure
8. Look for and express regularity in repeated reasoning

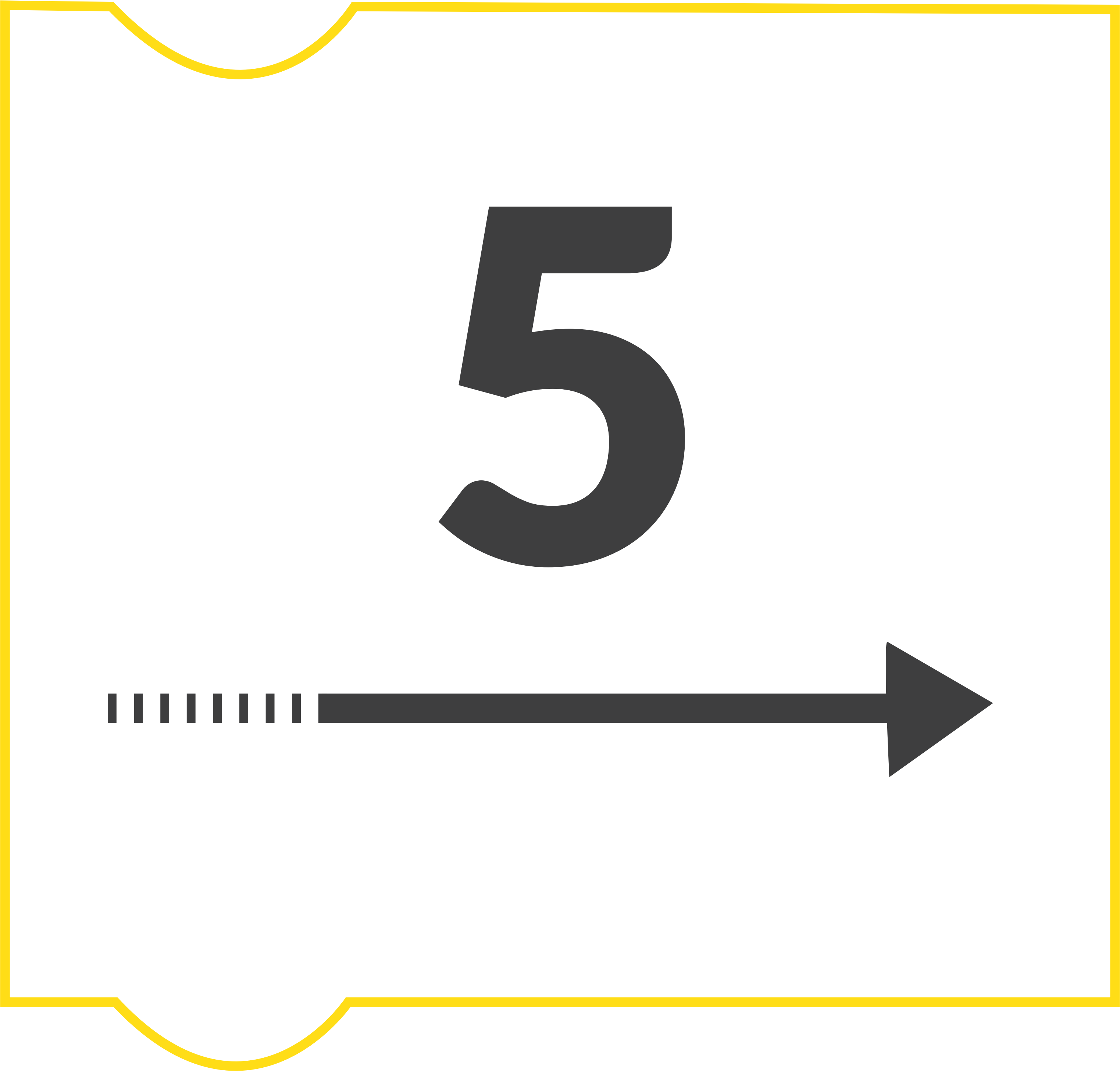
QUESTIONS ABOUT THIS LESSON?

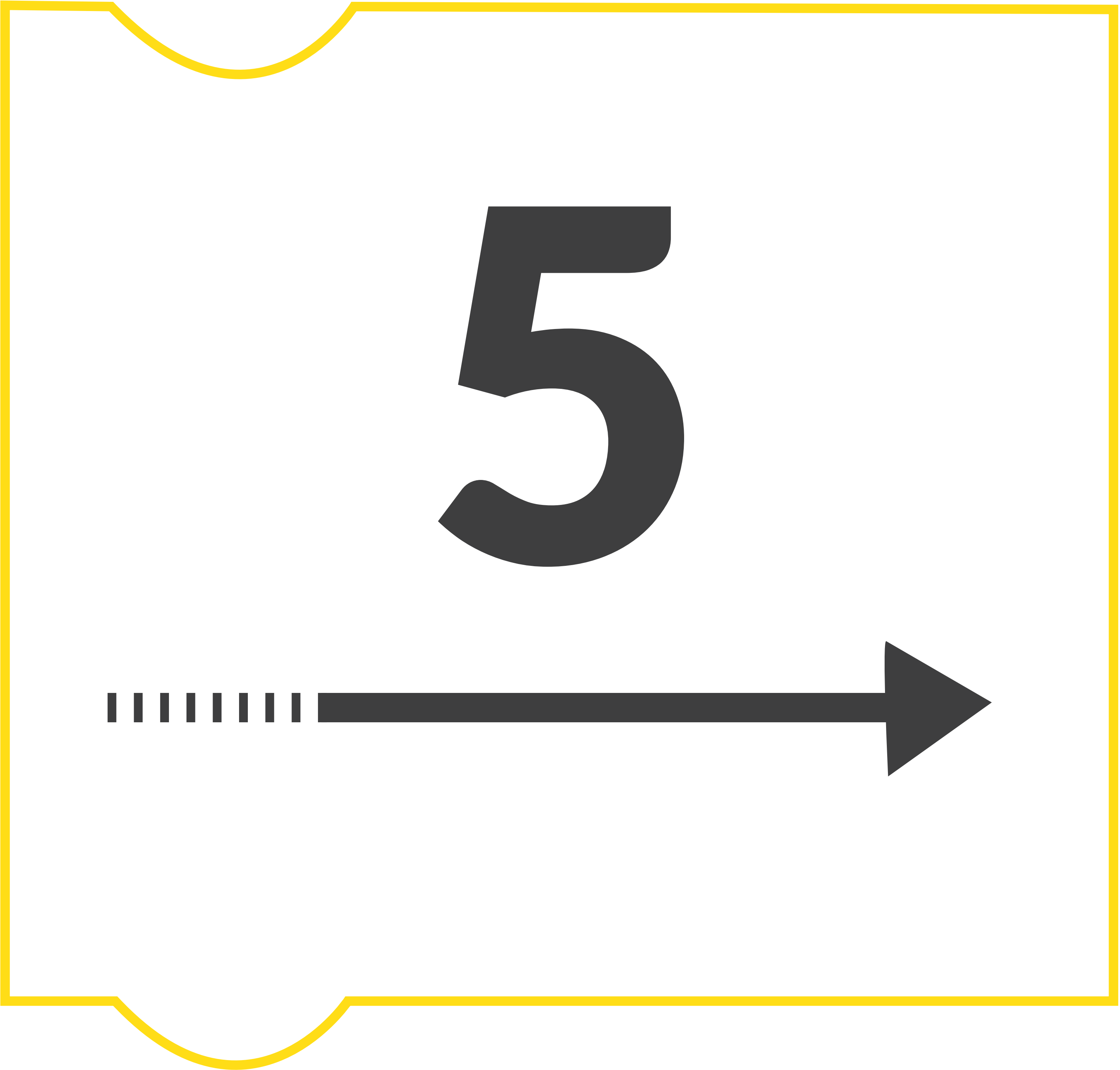
Please contact us at ozoedu@ozobot.com

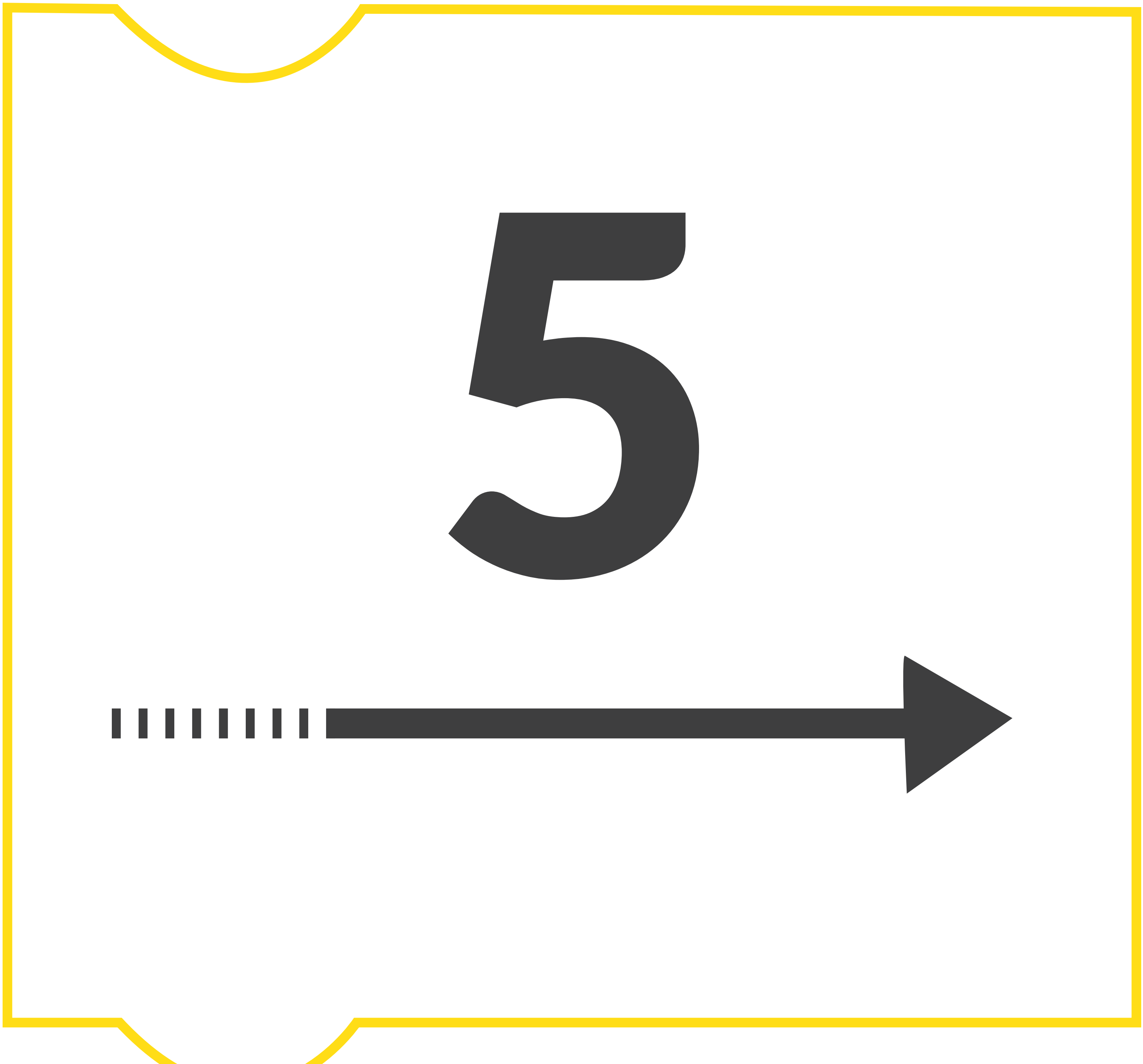
evo



ozobot edu


5


5


5