

The Additive Inverse Property

(Also available in [Pyret](#))

Students develop a more nuanced interpretation of the Commutative and Associative Properties as a result of their exploration of the inverse relationship between addition and subtraction.

Lesson Goals	Students will be able to... <ul style="list-style-type: none">• Recognize that subtracting x is the same as adding $-x$.• Acknowledge <i>flexibility</i> in the Order of Operations.
Student-facing Lesson Goals	<ul style="list-style-type: none">• Let's explore the inverse relationship between addition and subtraction.
Prerequisites	<ul style="list-style-type: none">• Translating Between Words and Math• Simple Data Types• Contracts• The Commutative Property• Equivalence• The Associative Property
Materials	<ul style="list-style-type: none">• PDF of all Handouts and Page• Additive Inverse Starter File• Lesson Slides• Printable Lesson Plan (a PDF of this web page)

Key Points For The Facilitator

- This lesson is all about *mathematical flexibility*, and may challenge preexisting ideas about arithmetic. For students who've had a mostly procedural introduction to mathematics - or who have high math anxiety - the flexibility introduced by this lesson can be intimidating!
- Embrace an open mind and expect the same of your students, and be attentive to their emotional response as the lesson progresses. If students are getting anxious, they will not be able to concentrate and focus on the content.
- There is no expectation that students will master operations with integers. Rather, the lesson is designed to get students thinking about how every number has an additive inverse.

Glossary

Additive Inverse Property :: Adding a number and its opposite always produces zero.

Associative Property :: When adding three numbers or multiplying three numbers, it does not matter whether you start with the first pair or the last. The same is true when either adding or multiplying four numbers, five numbers, etc.

comments :: messages in the code, generally ignored by the computer, to help people interacting with the code understand what it is doing

Commutative Property :: For any expression involving only addition or only multiplication, changing the order of the numbers will not change the result.

equivalent :: expressions are equivalent when they simplify to the same value, no matter what value is assigned to their variables (if there are any)

example :: shows the use of a function on specific inputs and the computation the function should perform on those inputs

opposite :: Two numbers are opposites when they are the same distance from zero on the number line.

The Additive Inverse Property

10 minutes

Overview

Students discover the Additive Inverse Property, which tells us that adding a number and its opposite always produces zero.

Launch

We've learned that the **Commutative Property** and **Associative Property** apply to addition... but not division. What if there was a way to rewrite subtraction as addition? Then we could apply the Commutative and Associative Properties to subtraction expressions, too! Let's explore this idea.

I'm going to draw a two-column table on the board. I'm labelling the left-hand side "Starting Value," and I'm labelling the right-hand side "Additive Inverse".

I'm going to write a number in the left-hand column. You are going to tell me what value I should add to that number, to get a sum of zero. I'll record your response in the right-hand column.

@table{

Starting Value	Additive Inverse

}



- The first number is 5. What do I need to add to 5 to get 0?
 - *-5. If this is students' first exposure to negative numbers, consider modeling a few examples.*
- The next number is 20. What do I need to add to 20 to get 0?
- How about $\frac{1}{2}$?
- How about -45 ?
- Can someone offer me another pair of numbers that add up to 0?

Allow a variety of students to share. Record responses on the table.

These number pairs all represent **opposites**.

The **Additive Inverse Property** tells us that adding a number and its opposite always produces zero.

Every number has an additive inverse.

Ensure that students are comfortable with this concept before moving onto the next activity.

Investigate



- Turn to [The Additive Inverse Property](#).
- In the first section, determine the additive inverses and write them in the spaces provided.
- Then, fill in the missing number to complete the equations. Some equations use mathematical notation and some use Circles of Evaluation.

Synthesize

- Can you think of a way to visually represent that adding a number and its opposite always produces zero?
- How would you explain the concept of the Additive Inverse Property to another student?
- Looking forward: Can you predict what effect the Multiplicative Inverse Property might have?

Addition and Subtraction: Inverse Operations *20 minutes*

Overview

Students rewrite addition expressions as subtraction, and subtraction expressions as addition by applying their knowledge of the *Additive Inverse Property*.

Launch

Now that we understand what the additive inverse is, we are ready to think about how we can put it to use... perhaps it can make some computations simpler?



- Complete [Discover Inverse Operations: Addition & Subtraction](#).
- When you're done, complete [Discover Inverse Operations: Addition & Subtraction \(2\)](#).
- What did you observe about the additive inverse?

During this exploration, you should have discovered that:

- Subtraction is equivalent to adding the *opposite*.
- We can rewrite any subtraction expression as addition, and any addition expression as subtraction!

One powerful advantage emerges when we write subtraction as addition: we now can apply the Commutative and Associative Properties to a much broader set of expressions!

Investigate

Now, students are ready to continue their exploration of subtraction as the inverse of addition. Note: To complete this worksheet, students do not need to be fluent at integer addition and subtraction. We just want students thinking about when expressions are *equivalent* based on what they have learned about the additive inverse.



Complete [Which One Doesn't Belong?](#)

Have students share strategies for determining equivalence. What are the different ways that they thought about the Additive Inverse Property?

Synthesize

- Claire and Soraya want to write an equivalent expression for $22 - 30$. Claire studies the expression and announces that, because it involves subtraction, the Commutative Property cannot be applied. Is she correct?
- Soraya grabs a pencil and writes the following: $22 + -30$. She says, "There! I fixed it. Now we can apply the Commutative Property." Explain what Soraya did. Is she correct?
- Use the Additive Inverse Property to simplify this expression using mental computation: $3 + 96.8 - 42.74 - 96.8 + 7 - 3 + 42.74$

The "Left-to-Right" Rule

25 minutes

Overview

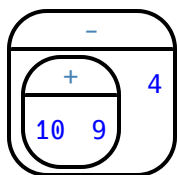
Students examine whether rigid adherence to the "left-to-right" rule is needed when adding and subtracting.

Launch



- Consider this expression: $10 + 9 - 4$. What do we get when we simplify it to a single value?
 - 15
- How did you arrive at your answer?

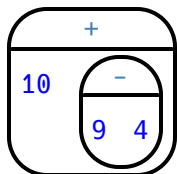
Did you work from left to right to arrive your result? This solving strategy can be represented by the Circle of Evaluation, below.



Is it essential to solve from left to right?

Ask if anyone opted to subtract *before* adding. If so, invite them to share their method and then invite other students to weigh in.

Evaluate the Circle of Evaluation below. Is it equivalent to the previous Circle of Evaluation?



We've learned that the Associative Property applies for expressions with only addition... not addition *and* subtraction. Many of us have also learned that when an expression includes addition and subtraction, we must work from left to right. **So... what's going on!?** It appears that we get the same result regardless of how we simplify this expression.

Investigate

Does subtracting *first* work every time? Can we rearrange the groupings of any expression with both addition and subtraction? Let's investigate.



- Turn to [Subtract First... or Solve Left-to-Right?](#)
- There, you will test out the this algorithm on several different expressions to see if subtracting and then adding produces the correct result every time.
- What do you Notice? What do you Wonder?
- Why are we able to change the groupings for an expression like $10 + 9 - 4$... but *not* for an expression like $10 - 9 - 4$?
- Describe why the this "subtraction first" algorithm works. (Hint: Think about the **Additive Inverse Property!**)

Encourage students to think deeply about why this algorithm works – and if you'd like, invite them to consider and discuss why students all across the country are typically taught just one algorithm when, typically, there are an abundance to choose from!

Let's put our new knowledge to use!

Project the problems below one at a time, and invite students to solve using mental math.



Scan each problem to identify any additive inverses, then solve using mental computation.

- $4 + 5 + 97 - 4 + 3$
 - 105
- $9 + 17 + 41 - 17$
 - 50
- $67 - 104 + 937 - 67 + 104$
 - 937

Synthesize

- How did it feel to scan the problem, find any additive inverses, and then solve mentally?
- Did you like this new approach, or do you prefer solving left to right?
- How would you explain to another student why they do *not* always need to solve from left to right when evaluating expressions with addition and subtraction.
- What are some advantages of solving left to right? What are some disadvantages?

Programming Exploration: The Additive Inverse

Overview

Students learn about examples in WeScheme, and use their new knowledge to think about addition and subtraction as inverse operations.

Launch



- Open the [Additive Inverse Starter File](#), but **do not click "Run" yet!**
- On [Introduction to Examples \(Additive Inverse\)](#), record what you Notice and what you Wonder about the starter file.
- Let's share out some of our Noticings and Wonderings.

Here are some common Noticings. Is there anything on this list that you *didn't* notice?

- Sections 1 and 2 each include five examples.
- Some words are in bold.
- Some words are in different colors and have semicolons in front of them.

Comments let programmers write notes.

Lines and lines of code can be difficult to read! Sometimes programmers want to write down their thinking, or leave notes to help others understand what the code is doing. That's what *comments* are for: a semicolon at the start of a line tells the computer that the rest of the line is a *comment* instead of code. To make comments stand out, they are colored purple.

The comments in lines 2, 14, and 26 are used to break up the code into sections that make the starter file easier for users to interpret.

Examples let programmers test their code.

Sometimes a comment isn't enough. A programmer might want to write down their thoughts so *that the computer can test their thinking*. These are called *examples*.

Take a look at the first *examples block* (lines 4-12). We start writing one or more *examples* of how we want some code to work. When we click "Run", WeScheme will test each of our examples, and report back which ones are correct and which ones are not.



- Return to your starter file and click "Run".

- With your partner, respond to questions 2 through 3 on [Introduction to Examples \(Additive Inverse\)](#).

Debrief with students to ensure that they are looking at the messages that appear in WeScheme. This activity not only provides practice thinking about the additive inverse; it also gives students exposure to tests - bits of code used to verify that code is working as we would expect. Examples and tests are widely used in programming! We explore examples in greater depth in [Functions: Contracts, Examples & Definitions](#).

Investigate

For the remainder of the activity, we will examine examples blocks. The first one includes only examples that pass. The second one has some errors! And the third one includes just one examples... **you** will be responsible for providing additional examples!



With your partner, complete [Introduction to Examples \(Additive Inverse\)](#).

Optional: If you would like to offer students additional practice with the additive inverse in WeScheme, two optional activities include [Are They Identical? \(Additive Inverse\)](#) and [Writing Equivalent Code](#).

Synthesize

- Using WeScheme allows us to evaluate arithmetic expressions easily and efficiently. In your own words, describe *how* you tested the equivalence of different expression - and how you interpreted the results that WeScheme produced.
- In WeScheme, what does it mean for a test to pass? What does it mean for a test to fail?
- What did this programming exploration teach you about the additive inverse?