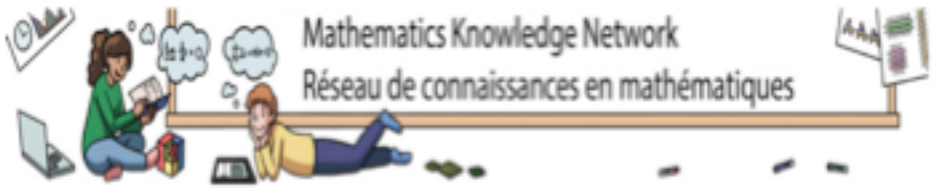


Brock University

Activity Summary*

Manipulating Expressions and Solving Equations (Part 2: Problem Solving)

(Assuming students have completed the Python Basics activity
and the activity: Manipulating Expressions and Solving Equations (Part 1: Intro))

**Activity summary format inspired from the ScratchMaths program*
and their presentation of the learning objectives by using the 5E's pedagogy

Learning Objectives:

- **Explore** how to solve problems that can be modelled with first-degree expressions and equations with the help of Python
- **Envisage** solutions to problems by trying to understand the problems, use mathematical knowledge to model the problems, and represent the models in Python code
- **Exchange** problem solving approaches and solutions with peers

Activity Description:

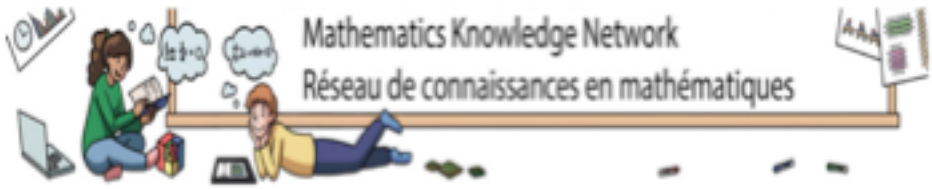
- This activity is about solving word problems using the mathematical and programming concepts introduced in a previous activity: Manipulating Expressions and Solving Equations (Part 1: Intro).
- After importing useful modules and declaring variables they will use later in the activity, students read through two example problems and their solutions.

  Example 1: The first example (about Joan and Chris's combined weekly take-home pay) includes the creation, addition, and use of polynomial expressions in one variable to represent a real-world situation. Students are shown how they can use Python to help them write, simplify, and evaluate expressions.

  Example 2: The second example (about the perimeter of a mini pool table) shows how to use the formula for the perimeter of a rectangle to create a first-degree equation in one variable that is needed to solve the problem. Students are shown how they can use Python to help them solve equations.

- A brief explanation is provided on how to clear a variable using Python's *var()* function, as students will be required to do this when completing the practice questions.

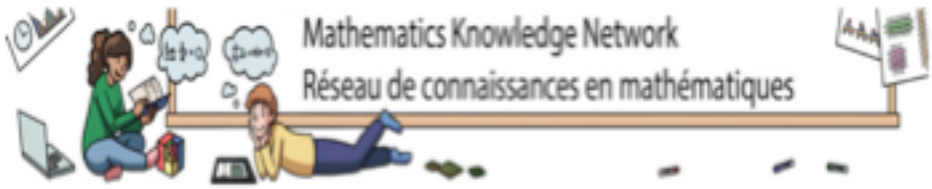
Mathematics Knowledge Network
Fields Institute for Research in Mathematical Sciences
222 College Street, 2nd Floor, Toronto ON, M5T 3J1



- Students are given four questions to practice solving word problems with Python. These questions were carefully chosen to build on the example problems. Students explore questions about perimeter (*#1*), about decay/loss in value (*#2* and *#4*), and about the work hours needed to make a certain wage (*#3*). They use Python to write, simplify, and evaluate expressions (*#1*, *#2*, and *#4*), or to solve equations (*#3*). They encounter some problems (e.g., #4) that would be more difficult to solve solely by hand (due to the arithmetic involved).
- *Practice Question #5* invites students to come up with their own problem and solution! This may give students a chance to get creative and may help them to deepen their understanding of problems that can be modelled with first-degree equations.

Links to Curriculum Expectations:

Curriculum Expectations	Links to Jupyter Notebook
Solve problems that can be modelled with first-degree equations, and compare algebraic methods to other solution methods.	- Students are required to build their knowledge of problems that can be modelled with first-degree equations and how such problems can be solved using algebraic methods, with the help of Python.
Simplify numerical expressions involving integers and rational numbers, with and without the use of technology.	- To solve problems, students are required to use knowledge of how to simplify numerical expressions using Python.
Expand and simplify polynomial expressions involving one variable [e.g., $2x(4x + 1) - 3x(x + 2)$], using a variety of tools (e.g., algebra tiles, computer algebra systems, paper and pencil).	- To solve problems, students are required to use knowledge of how to expand and simplify polynomial expressions using Python.
Solve first-degree equations, including equations with fractional coefficients, using a variety of tools (e.g., computer algebra systems, paper and pencil) and strategies (e.g., the balance analogy, algebraic strategies).	- To solve problems, students are required to use knowledge of how to solve first-degree equations using Python.



Brock University

Discussion Points:

- In *Practice Question #1*, is it *necessary* to determine a simplified expression for the outside perimeter of the frame (i.e., $40x + 8$) in order to determine the outside perimeter when $x = 5$? Notice: We could plug $x = 5$ into the expressions for the length ($10x$) and the width ($10x + 4$) of the frame (to calculate the length and the width) and then we could calculate the perimeter using the formula: $(2 * Length) + (2 * Width)$. Why do you think we were asked to determine and use the simplified expression instead? Is it better to *model* the problem using *only* this simplified expression? What information would be lost if we knew only the simplified expression? The same kinds of questions could be explored for *Practice Question #2* and *Practice Question #4*.
- In *Practice Question #3*, which wage (Steve's, Noah's, or Kayli's) would you want to have and why?
- How can we decide when to use Python to assist us in solving word problems? Should we always use Python? Why (not)?

Things to Note:

- This activity is geared towards students with very little coding knowledge and some basic mathematical knowledge related to the activity. You may want to have students working independently or in groups depending on their skill levels in Python and in mathematics.
- If students are having difficulty getting their code to work, ensure that they are reading the error message from Python and checking for syntax errors (e.g., a missing bracket, comma, colon, etc.).
- This activity relies on the coding concepts that are introduced in the Python Basics activity and in the activity: Manipulating Expressions and Solving Equations (Part 1: Intro). This said, there may also be some new elements of code introduced in this activity. We encourage you and your students to read the comments that are provided throughout the activity (in and around the coding cells), where describe some of the coding elements used. If there is an element of code that you still do not understand, you may need to refer back to the other activities or try looking for information online (e.g., if you need more explanation about Python's *var()* function, trying Googling "var() python").

References

http://www.edu.gov.on.ca/eng/curriculum/secondary/math910curr.pdf
https://scratch.mit.edu/about/

The MKN is funded by the Ontario Ministry of Education. The MKN is a KNAER Project hosted by the Fields Institute for Research in Mathematical Sciences. The views expressed in this document belong to the authors and do not necessarily reflect the opinions of the Ministry of Education nor the Ontario government.

Activity created by Jonathan Dugdale and Angela Wales for MATH 3P41 at Brock University under the ***Math Knowledge Network.*** Coordinated by Dr. Chantal Buteau (Brock University). Amended by Lauren McCann and Tyrell Nurdin. June 2020.

Mathematics Knowledge Network
Fields Institute for Research in Mathematical Sciences
222 College Street, 2nd Floor, Toronto ON, M5T 3J1