



Activity Summary*

Operating with Exponents

(Assuming students have completed the Python Basics activity)

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

Learning Objectives:

- **Explore** the meaning of an exponent, the various rules for operating with exponents, and how Python can be used as tool for thinking about, performing, and verifying these operations
- **bridge** some of the processes of by-hand computations and programming-based computations

Activity Description:

Basics of Exponents

- The activity begins with two questions aimed at prompting students to think about the basic meaning of an exponent: namely, multiplying a number by itself n times can be represented by raising the number to the power (or the exponent) of n .
- Students are provided with code that illustrates this meaning and reminds them how to enter multiplication and exponentiation into Python.
- Students are also introduced to some basic terminology, including “base” and “exponent,” and what it means to “square” or “cube” a number.
- The first subsection (*What is an Exponent?*) only includes positive bases and natural number exponents. The second and third subsections gradually introduce some other possibilities:

Using Exponents with Negative Numbers: Students are shown that if the base is a negative number, the process of exponentiation is the same as for positive bases, but the answer may be positive or negative. They are invited to run code that illustrates the property: if a negative number is raised to an even (or odd) exponent, the answer is positive (or negative).

Replacing Square Roots with Exponents: Students are told that the square root of a number is the same as raising the number to the exponent of $\frac{1}{2}$ and they are invited to verify this on an example using Python. Although it is not explicitly stated in the activity, students may be introduced to the idea that an exponent does not always need to be a natural number.

- Students practice creating their own code. They are asked to calculate the square root of some numbers and to verify their answers through squaring, which may also help them gain some understanding of the relationship between these two inverse operations.

Using Exponents with Fractions

- The meaning of an exponent is extended to the context of a fractional base: i.e., students are told that they must raise both the numerator and the denominator to the exponent.
- The by-hand process of calculating a fraction raised to an exponent is exemplified and then translated into Python code. This includes the use of Python's *Fraction()* tool.
- A practice question invites students to try out this translation themselves.

Multiplying Numbers with Exponents, Dividing Numbers with Exponents, and Power of a Power

- Students are introduced to three exponent rules:
 - When you multiply numbers with the same base, you add their exponents.
 - When you divide numbers with the same base, you subtract their exponents.
 - When you take the power of a power, you multiply the exponents.
- Each rule is illustrated in at least one by-hand example. Then students are shown how the rule can be verified in specific examples by writing and simplifying expressions in Python.
- Several practice questions are provided. Students must apply the exponent rules in different ways to construct Python code that contains two expressions – an original expression and its simplified version – that should lead to the same answers. Students run the code to check their work.

Links to Curriculum Expectations:

Curriculum Expectations	Links to Jupyter Notebook
Substitute into and evaluate algebraic expressions involving exponents (i.e., evaluate expressions involving natural-number exponents with rational-number bases [e.g., evaluate $(\frac{3}{2})^3$ by hand and 9.8^3 by using a calculator]).	<ul style="list-style-type: none"> - Students are required to use knowledge of how to evaluate an expression involving a natural-number exponent with a rational-number base when trying to understand and write Python code. - Students are required to build knowledge of how to use Python to substitute into and evaluate algebraic expressions involving exponents.

Derive, through the investigation and examination of patterns, the exponent rules for multiplying and dividing monomials, and apply these rules in expressions involving one and two variables with positive exponents.

Extend the multiplication rule to derive and understand the power of a power rule, and apply it to simplify expressions involving one and two variables with positive exponents.

- Students are required to build knowledge of the three exponent rules through numerous practice questions. Working in the familiar context of numbers may set a foundation for applying the rules in expressions involving variables.

Discussion Points:

- What does raising a number to an exponent mean? What is another, more understandable, way to express an exponent? (i.e., $2^4 = 2 \times 2 \times 2 \times 2$)
- When a number has the exponent 2 (or 3), we say that the number is “squared” (or “cubed”). Why do you think there are no special words for larger exponents?
- Without using a calculator, how could we figure out the value of $(-1)^{1000007}$?
- What if we want to take the square root of a fraction? For instance, how would we go about calculating a number like $\sqrt{\frac{81}{121}}$?
- *Example 6* attempts to illustrate the reasoning behind the exponent rule for *multiplying* numbers with the same base. Take a closer look at that example. Could we construct similar examples that would illustrate the reasoning behind the other two exponent rules we learned?
- In the practice questions, did you ever find that the expressions ‘a’ and ‘b’ did not have the same value when they should have? How did you go about figuring out where you made an error?



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. 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 we 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 want to find some more explanation about Python's *Fraction()* tool, trying Googling "fraction() 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 Cassandra Biscardi and Kaileigh DiMichele 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