Arithmetic Expressions

Now that you've written some code, let's take a step back and look at some common arithmetic operators. The behavior of Python operators (+, -, *, /) depends on what type of data you have.

Manager: Recorder:

Presenter: Reflector:

Content Learning Objectives

After completing this activity, students should be able to:

- Summarize the benefits of working as a team.
- Execute mathematical expressions similar to a calculator.
- Identify and justify the precedence of arithmetic operators.
- Describe the function of the three Python division operators.

Process Skill Goals

During the activity, students should make progress toward:

- Recognizing mathematical operations based on tables. (Information Processing)

Meta Activity: Group vs Team

Throughout the course, you will need to examine and process information, ask and answer questions, construct your own understanding, and develop new problem-solving skills.

Questions (8 min) **Start time:**

1. What are some advantages to working in groups/teams?

2. What are some disadvantages to working in groups/teams?

3. Based on the images above, what is the difference between a group and a team? Come up with a precise answer.

4. How can working as a team help you accomplish the tasks described above? Give at least two specific examples.

Model 1 Python Calculator

In a Python Shell window, ">>>" is a ***prompt*** indicating that the interpreter is waiting for input. All text entered after the prompt will be executed immediately as Python code.

If you type a Python ***expression*** (code that results in a value) after the prompt, Python will show the value of that expression, similar to a calculator. You can use Python's `math` module to perform more complex mathematical operations like logarithms and trigonometric operations.

Do not type anything yet! Read the questions first!

Python code	Predicted output	Actual output
`2 + 3`		
`3 * 4 + 2`		
`3 * 4 + 2.0`		
`3(4 + 2)`		
`3 * (4 + 2)`		
`5 / 10`		
`5 / 10.0`		
`5 / 9`		
`2 ** 4`		
`abs(-2) ** 4`		
`math.pow(2, 4)`		
`import math`		
`math.pow(2, 4)`		
`sqrt(4)`		
`math.sqrt(4)`		
`math.cos(0)`		
`math.pi`		
`math.sin(math.pi / 2)`		

Questions (15 min) Start time:

5. In the middle "Predicted output" column, write what value you expect will be displayed, based on your team's experience using a calculator. If there are any lines you are not confident about, place an asterisk next to your predicted output.

6. Open a Python Shell on your computer. Type each Python expression at the prompt, one line at a time, and write the corresponding Python output in the third column above. If an error occurs, write what type of error it was (i.e., the first word of the last line of the error message).

7. What does the ** operator do?

8. Based on the Python code in Model 1, identify four examples of:

a) mathematical operator

b) mathematical function

9. For addition and multiplication to produce an output with a decimal value, what type of number must be part of the input? Provide justification for your team's answer.

10. Does division follow the same rule as in #9? Provide justification for your team's answer.

11. The output of Model 1 displayed three different errors. Explain the reason for each:

a) TypeError

b) 1st NameError

c) 2nd NameError

12. Identify two differences between using a Python built-in function (e.g., `abs`) and a function from the `math` module.

Model 2 Order of Operations

Python follows a specific order for math and other operations. For example, multiplication and division take ***precedence*** over addition and subtraction. The following table lists several Python operators from highest precedence to lowest precedence.

Operator	Description
`**`	Exponentiation
`+ -`	Positive, Negative (***unary*** operators)
`* /`	Multiplication, Division
`+ -`	Addition, Subtraction (***binary*** operators)
`=`	Assignment

Questions (12 min) Start time:

13. Determine the order of operations in the statement: `y = 9 / 2`

a) First operator to be evaluated:

b) Second operator:

c) Value of `y`:

14. Determine the order of operations in the statement: `x = 5 * -3`

a) First operator to be evaluated:

b) Second operator:

c) Third operator:

d) Value of `x`:

15. Determine the order of operations in the statement: `z = 2 * 4 ** (3 + 1)`

a) First operator to be evaluated:

b) Second operator:

c) Third operator:

d) Fourth operator:

e) Value of `z`:

16. The `+` and `-` operators show up twice in the table of operator precedence. For the Python statement `x = 5 * -3`, explain how you know whether the `-` operator is being used as a unary or binary operator.

17. What do the words "unary" and "binary" mean in this context?

18. What operator has the lowest precedence? Why do you think it's designed that way?

19. What operators have the highest precedence? Why do you think it's designed that way?

20. Enter the expressions below into a Python Shell. Why are the results different? Explain your answer in terms of operator precedence.

- `-3 ** 2` Result:
- `(-3) ** 2` Result:

Model 3 Dividing Numbers

Table A

`9 / 4`	*evaluates to*	2.25
`10 / 4`	*evaluates to*	2.5
`11 / 4`	*evaluates to*	2.75
`12 / 4`	*evaluates to*	3.0
`13 / 4`	*evaluates to*	3.25
`14 / 4`	*evaluates to*	3.5
`15 / 4`	*evaluates to*	3.75
`16 / 4`	*evaluates to*	4.0

Table B

`9 // 4`	*evaluates to*	2
`10 // 4`	*evaluates to*	2
`11 // 4`	*evaluates to*	2
`12 // 4`	*evaluates to*	3
`13 // 4`	*evaluates to*	3
`14 // 4`	*evaluates to*	3
`15 // 4`	*evaluates to*	3
`16 // 4`	*evaluates to*	4

Table C

`9 % 4`	*evaluates to*	1
`10 % 4`	*evaluates to*	2
`11 % 4`	*evaluates to*	3
`12 % 4`	*evaluates to*	0
`13 % 4`	*evaluates to*	1
`14 % 4`	*evaluates to*	2
`15 % 4`	*evaluates to*	3
`16 % 4`	*evaluates to*	0

Questions (15 min) Start time:

21. For each operator in Model 3, identify the symbol and describe the type of numerical result.

22. If the result of the / operator were rounded to the nearest integer, would this be the same as the result of the // operator? Explain how the results in Table A compare to Table B.

23. If the table included more rows, list all numbers // 4 would evaluate to 2 and all the numbers // 4 would evaluate to 4.

24. Based on the results of Table C, propose another number % 4 evaluates to 0, and explain what all these numbers have in common.

25. Consider the expressions in Table C that evaluate to 1. How do the left ***operands*** in these expressions (i.e., 9, 13) differ from those that evaluate to 0?

26. Describe the reason for the repeated sequence of numbers (0, 1, 2, 3) for the result of % 4.

27. Recall how you learned to do long division in elementary school. Finish solving for $79 \div 5$ below. Which part of the answer is `79 // 5`, and which part is `79 % 5`?

$$\begin{array}{r} 1 \\ 5\overline{)\,79} \\ -\,5 \\ \hline 2 \end{array}$$

28. Imagine that you are given candy mints to divide evenly among your team members.

a) If your team receives 11 mints, how many mints would each student get, and how many are left over? Write a Python expression to compute each result.

b) If your team receives 2 mints, how many mints would each student get, and how many are left over? Write a Python expression to computes this result.

29. Python has three division operators: "floor division", "remainder", and "true division". Which operator (symbol) corresponds to each name?