Arithmetic Expressions

Now that you've written a few programs, let's take a step back and discuss how to do arithmetic. The behavior of Python operators (+, -, *, /) depends on what type of data you have.

Content Learning Objectives

After completing this activity, students should be able to:

- Execute mathematical expressions similar to a calculator.
- Describe the function of the three Python division operators.
- Explain differences between integer and floating-point data.

Process Skill Goals

During the activity, students should make progress toward:

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

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: ________

1. 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.

2. 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).

3. What does the ** operator do?

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

   a) mathematical operator

   b) mathematical function

5. 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.

6. Does division follow the same rule as in #5? Provide justification for your team's answer.

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

   a) TypeError

   b) 1st NameError

   c) 2nd NameError

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

Model 2 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:** ________

9. For each operator in Model 2, identify the symbol and describe the type of numerical result.

10. 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.

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

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

13. 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?

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

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

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

16. 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.

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

Model 3 Integers and Floats

Every value in Python has a ***data type*** which determines what can be done with the data. Enter the following code, one line at a time, into a Python Shell. Record the output for each line (if any) in the second column.

Python code	Shell output
`integer = 3`	
`type(integer)`	
`type("integer")`	
`pi = 3.1415`	
`type(pi)`	
`word = str(pi)`	
`word`	
`number = float(word)`	
`print(word * 2)`	
`print(number * 2)`	
`print(word + 2)`	
`print(number + 2)`	
`euler = 2.7182`	
`int(euler)`	
`round(euler)`	

Questions (15 min)

Start time: ________

18. What is the data type (`int`, `float`, or `str`) of the following values? (Note: if you're unsure, use the `type` function in a Python Shell.)

a) pi

c) word

b) integer

d) number

19. List the function calls that convert a value to a new data type.

20. How does the behavior of the operators (+ and *) depend on the data type?

21. What is the difference between the `int` function and the `round` function?

22. What is the value of `3 + 3 + 3`? What is the value of `.3 + .3 + .3`? If you enter these expressions into a Python Shell, what do you notice about the results?

23. In order to store a number with 100% accuracy, what data type is required? How might you precisely represent a bank account balance of $123.45?

24. Try calculating a very large integer in a Python Shell, for example, 123^{456}. Is there a limit to the integers that Python can handle?

25. Try calculating a very large floating-point number in a Python Shell, for example, 123.0^{465}. Is there a limit to the floating-point numbers that Python can handle?

26. Summarize the difference between the numeric data types (`int` and `float`). What are their pros and cons?