

Numbers

- Exercises
 - Exercise 0: Preparation
 - Exercise 1: Types of Numbers
 - Exercise 2: The Type of a Square Root
 - Exercise 3: More Types of Numbers
 - Exercise 4: A Large Number
 - Exercise 5: Inexact Roots
 - Exercise 6: Inexact Fractions
 - Exercise 7: Other Integer Procedures
 - Exercise 8: Exploring Rationals
 - Exercise 9: Rounding, Revisited
 - Exercise 10: Other Numeric Predicates
- Notes
 - Notes on Exercise 7

You may want to refer to the reading on numbers before or while you work on this lab.

Exercises

Exercise 0: Preparation

Start DrScheme.

Exercise 1: Types of Numbers

Have DrScheme confirm that $3/4$ is a rational number but not an integer and that the square root of -1 is a complex number but not a real number.

Exercise 2: The Type of a Square Root

Confirm that the value DrScheme computes for `(sqrt 2)` is an inexact real that is also rational.

Exercise 3: More Types of Numbers

As you've just seen, some kinds of numbers are subsets of other kinds of numbers. Determine the relationships between the various kinds of numbers.

Exercise 4: A Large Number

Write a Scheme numeral for 1.507 times ten to the fifteenth power, as an *exact* number. Have Scheme evaluate the numeral.

Exercise 5: Inexact Roots

- a. Have DrScheme find the square of the square root of 2 and subtract 2 from the result.
- b. Ideally, the difference should be 0; why isn't it?
- c. How big is the difference?
- d. Will you have the same problem if you start with 4? Why or why not?

Exercise 6: Inexact Fractions

Write a Scheme numeral for one-third, as an *inexact* number. Have Scheme evaluate the numeral.

Exercise 7: Other Integer Procedures

Scheme provides a number of numerical procedures that can produce integer results. We've already explored `expt`, `abs`, `+`, `-`, and `*`.

Here are some others. For each, try to figure (by experimentation, by discussing results with other students, and, eventually, by reading documentation) out how many parameters each procedure can take and what the procedure does. You may not be able to figure all of them out. Make sure to try a variety of values for each procedure, including positive and negative, integer and real.

- a. `quotient`
- b. `remainder`
- c. `modulo`
- d. `max`
- e. `min`
- f. `numerator`
- g. `denominator`
- h. `gcd`
- i. `lcm`

j. floor

k. ceiling

l. truncate

m. round

Exercise 8: Exploring Rationals

Since you've found that DrScheme seems to represent every real number as a rational, it might be worth finding a way to see what that rational number is. Determine the numerator and denominator of the rational representation of the square root of 2.

Exercise 9: Rounding, Revisited

For small numbers, the `exact->inexact` procedure produces about six digits after the decimal point. Figure out how to get just two digits after the decimal point. You may need to use multiplication, division, and some of the last procedures from the previous exercise.

You need not implement your algorithm; simply come up with one you think will work.

Exercise 10: Other Numeric Predicates

We've already seen a variety of predicates (procedures that return true or false) that can be applied to numbers. These predicates include `exact?`, `integer?`, and `real?`.

By reading the Scheme documentation, identify other predicates that can be applied to numbers.

Notes

Notes on Exercise 7

When you get stuck on this problem, it's probably worth skimming through DrScheme's *Help Desk*. The numerical operations are documented in section 6.2.5 of the Revised(5) Report on the Algorithmic Language Scheme.