

Class exercises:

These are team exercises in the breakout rooms. That means, for each of the problems below a different person in your breakout room shares his/hers/theirs WeScheme screen, and everyone helps with creating the function. The someone else shares the screen for the next problem.

At the same time, y'all need to come up with at least 5 tests to make sure that the function works. Remember, these tests are **trying to get the function to give the incorrect answer**, so that y'all can fix the function to work correctly.

However, if the function has a limited domain (for instance, only positive integers), then do not test the function outside that domain (for instance on negative integers, or positive fractions, or elephants).

Here are a couple of new tools you may use:

- There's a built-in function called (RANDOM) which takes no arguments and returns a random decimal number between 0 and 1 (but never 1). Try it several times – you'll be using it. Examples:
(random) -> 0.5699463
(random) -> 0.0034668
- There's a built-in function called (FLOOR x), which, when given a number with a decimal value, will return the largest integer less than or equal to the number. We'll be using this function only on non-negative values. For instance:
(floor 4.6) -> 4
(floor 4.0) -> 4
- You may create any helper functions you need.

The Problems:

1. Create your own version of the *ROUND*(x) function -- *MY-ROUND*(x) -- which will take a non-negative number and round it to the nearest integer (4.5 -> 4 or 5, your choice). Of course, you may not use *ROUND*.

2. Create the function *ROUND-2* which does the same thing as *MY-ROUND* except for numbers half-way between integers. Those will be taken to the nearest even number. And no, you can't use the built-in functions *even?* or *odd?* (*Why not?* Because I said so. *Aw come on.* No.) Examples:
(*ROUND-2* 4.5) -> 4
(*ROUND-2* 5.5) -> 6
3. Create the function (*RANDOM-BETWEEN low high*), which will take two non-negative integers *low* and *high* with *low* < *high*, and will return an random integer between *low* and *high* (including *low* and *high*). All the integers in that range are equally probable. Examples:
(*RANDOM-BETWEEN* 0 1) -> 1
(*RANDOM-BETWEEN* 0 1) -> 0
(*RANDOM-BETWEEN* 47 904) -> 299
4. Create the function (*EQUAL-PROB-2 a b*), which will be given numbers *a* and *b* and will return either *a* or *b* with equal probability. Examples:
(*EQUAL-PROB2* 47 -89.3) -> -89.3
5. Create the function (*EQUAL-PROB-3 a b c*), which will return *a* or *b* or *c* with equal probability.