IMPORTANT NOTE:

/run/media/lijunjie/资料/Learning Video/writing-running-fixing-code/02_compiling-and-running/05_review/01_assignment-06-rect_instructions.html

After launching the Practice Programming Environment, move into the **06_rect** folder to begin the assignment.

$ cd learn2prog/06_rect

In this folder you will find a README file that contains instructions for the assignment. Don't forget to add your file(s) to git, commit and push, and then run the *grade* command to pass this assignment.

One quick note about this assignment:

You write a function called **canonicalize** , which takes a rectangle and returns a rectangle. Many people have asked about this in the forums, so I am going to post a more detailed explanation here.

When we describe a rectangle with x,y, width, and height , there are many different ways to describe the same rectangle. For example if you want the rectangle from (0,0) to (2,3), you could say it has

x=0, y=0, width=2, height=3

x=0, y=3, width=2, height = **-** 3

x=2, y=0, width= **-** 2, height=3

x=2, y=3, width= **-** 2, height= **-** 3

All of these describe the same rectangle, but we really prefer the first one: it is the canonical description of that rectangle. The others name the same rectangle, but are "strange" ways to do it. We want our programs to be robust in how they handle strange inputs. However, if we write our code to deal with rectangles to handle any of these 4 ways to describe one rectangle, the complexity of the code explodes.

A great approach when faced with this situation is to take the input and put it into the canonical form before doing anything else with it. That is, take any "weird" rectangles (options 2, 3, or 4 above) and turn them into the "normal" one (option 1 above). This is exactly what canonicalize does: given a rectangle, turn it into the "normal" way of describing the same rectangle: turn negative width or heights into positive widths or heights.

If you did not pass the previous assignment (**05_squares**) then you won't see the **06_rect** folder with this assignment. Complete the previous assignment before attempting this assignment.

We strongly encourage you to do the assignments in order, as each assignment builds on previous assignments, and you need to pass all assignments to pass this course. If you didn't pass the previous assignment but really want to work on this assignment, you can release it using the command:

$ fast-forward 06_rect