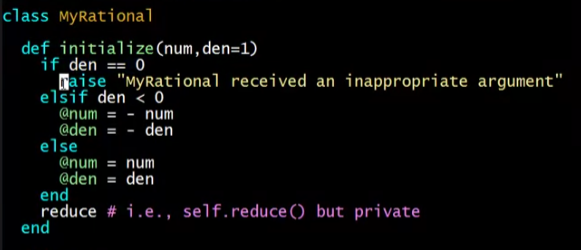
**Now**

* Use a small class for rational numbers
  + Called **MyRational**
    - Ruby has built-in function using a class *Rational*
* Will also use several new and useful expression forms
* Way our class work:
  + Keeps fractions in reduced form with a positive denominator
  + Like an ML-module example earlier in the course

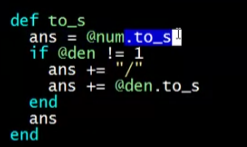
1st definition: **initialize**



* Note: ‘elsif’ – careful with the spelling

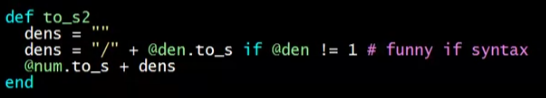
2nd definitions: toString definitions

**to\_s**



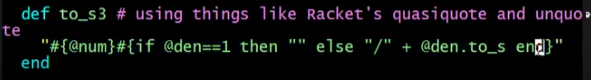
* .to\_s is a built-in function

**to\_s2**

****

* Funny if syntax
  + <action> if <condition>

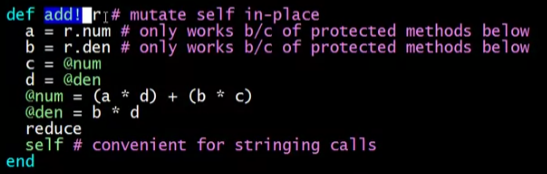
**to\_s3**

****

* This is like Javascript’s `${…}`
* Called expression interpolation

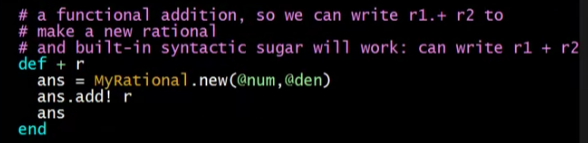
3rd definition: **add!** Function

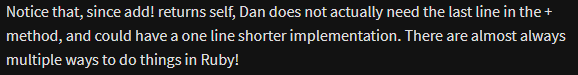
* Imperative addition



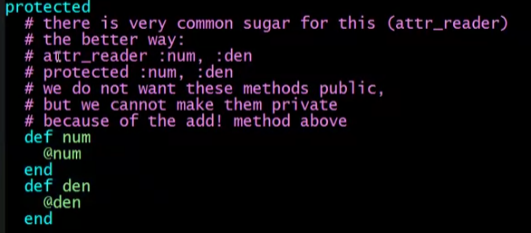
4th definition: **+** function

* Functional addition

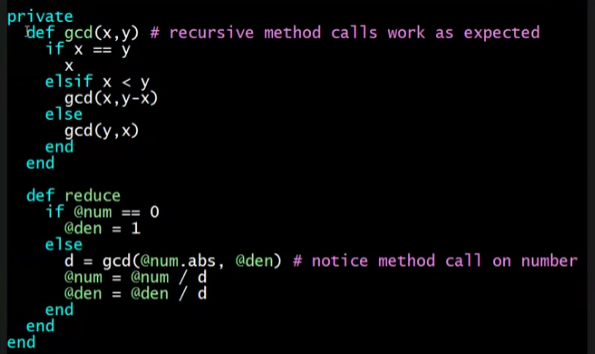




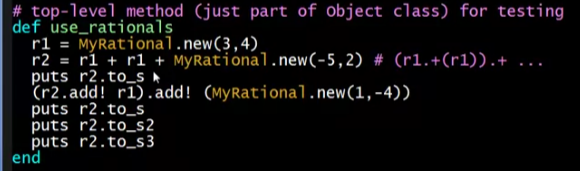
Protected definitions



Private definitions



**use\_rational** method



Loading the example

