

Equality In Ruby

Farrel Lifson

<http://www.aimred.com>

13th February 2008



Blatantly Stolen From

Navigating The Equality Maze

<http://www.texperts.com>

(don't sue!)



5 Types Of Equality

- Natural Equality: ==
- Object Identity: equal?
- Hash Equality: eql?
- Case Equality: ===
- Spaceship Operator: <=>

Natural Equality

- Compare two objects that are naturally comparable
- `1 == 1.0` is valid
- `1 == "1.0"` is not
- `1 == Rational(1,1)` is valid

Object Identity

- Compares if the actual objects are the same
- `1.equal?(1)` is true
- `1.equal?(1.0)` is false
- `a="1"`
`b="1"`
`a.equal?(b)` is false

Hash Equality

- True if both objects have the same value
- `1 == 1.0` is true
- `1.eql?(1.0)` is not true
- Default implementation in Object is not ideal
- `a.eql?(b)` implies `a.hash == b.hash`

Case Equality

- `===` - Used behind scenes of case statements
- Object default implementation of `===` is `==`
- Overwritten in `Class` (`kind_of?`), `Regex` (`match`)
- Useful for a one-to-many match

Spaceship Operator

- Define an ordering between objects
- $a <=> b$ returns:
 - -1 if $a < b$
 - 0 if $a == b$
 - 1 if $a > b$
- Hook for Comparable module - gives you `==`

What To Do?

- Do not override equal?
- Override eql? and hash
- Ensure $a.hash == b.hash$ implies $a.eql?(b)$
- Override $==$ if necessary