Lightning Talks

Schedule - Thursday

- Farrel Lifson
- Chris Stefano
- Brendan Mclean
- You?

Schedule - Friday

- Kevin McKelvin
- Lance Gleason
- ~~Salman Ansari~~ Thanks NY Weather!
- You?

Cape Town Ruby Brigade

- Since November 2007
- 2nd Wednesday of the Month
- Bandwidth Barn @ Woodstock Exchange
- http://capetownrb.org
- @capetown_rb

Survey!

Who lives outside Cape Town and has attended a CTRB talk?

Who lives in Cape Town and has NEVER attended a CTRB talk?

First Cape Town Ruby Brigade of 2014 is...

Bitcoin & BitX

12th March 2014

18:00 @ BWB

On to the lightning talks!

===

The neglected step child of comparison

Case Equality

“Threequal”

```
case number
when 1
 puts “one”
when 2
 puts “two”
end
```

```
if number === 1
 puts “one”
elsif number === 2
 puts “two”
end
```

Default behaviour

=== is ==

=== overloaded in four classes

=== Overloaded

Range	range.include?(arg)
Class	arg.kind_of?(klass)
Regexp	regexp =~ arg
Proc	proc.call(arg)

Unlike == multiple values will return true

=== really means
“is satisfied by”

```
(1..10) === 1
(1..10) === 10

/?at/ === “hat”
/?at/ === “cat”
```

Why?

```
case age
when 0..18
 “Too young”
when 18..65
 “Eligible”
when 66..120
 “Too old”
end
```

=== used in one other place
in Standard Library

Enumerable#grep

```
numbers.grep(Integer).grep(30..50)
```

instead of

```
numbers.
  select{ |n| n.kind_of?(Integer)}.
  select{ |n| (3..50).include?(n)}
```

=== is hardly ever overloaded

When to overload ===

Overload === when a class has a relationship to another class

Overload === when that relationship will be tested often

```ruby
class Polygon
  def pt_in_poly?(point)
    # Geometry goes here
  end

  def ===(point)
    pt_in_polygon?(point)
  end
end
```

```
case point
when triangle
  puts “In triangle”
when square
  puts “In Square”
end

intrsxn = points.
            grep(square).
            grep(triangle)
```

Thanks!

See you all(?) at CTRB on 12th March!