Module Comparable

Relies on: <=>

The Comparable mixin is used by classes whose objects may be ordered. The class must define the <=> operator, which compares the receiver against another object, returning −1, 0, or +1 depending on whether the receiver is less than, equal to, or greater than the other object. Comparable uses <=> to implement the conventional comparison operators (<, <=, ==, >=, and >) and the method between?.

```
class CompareOnSize
  include Comparable
  attr :str
  def <=>(other)
    str.length <=> other.str.length
  end
  def initialize(str)
    @str = str
  end
end

s1 = CompareOnSize.new("Z")
s2 = CompareOnSize.new([1,2])
s3 = CompareOnSize.new("XXX")

s1 < s2                # =>   true
s2.between?(s1, s3)    # =>   true
s3.between?(s1, s2)    # =>   false
[ s3, s2, s1 ].sort    # =>   ["Z", [1, 2], "XXX"]
```

Instance methods

Comparisons

obj < *other_object* → true or false
obj <= *other_object* → true or false
obj == *other_object* → true or false
obj >= *other_object* → true or false
obj > *other_object* → true or false

Compares two objects based on the receiver's <=> method.

between?

obj.between?(*min*, *max*) → true or false

Returns false if *obj* <=> *min* is less than zero or if *obj* <=> *max* is greater than zero; returns true otherwise.

```
3.between?(1, 5)               # =>   true
6.between?(1, 5)               # =>   false
'cat'.between?('ant', 'dog')   # =>   true
'gnu'.between?('ant', 'dog')   # =>   false
```