

Ruby



Facets of Ruby

- Object-Oriented to the core
 - Everything is an object
 - Tight encapsulation
 - Single Inheritance, but with **mixin** magic
- Dynamic
 - Duck typing
 - Advanced **metaprogramming**

Elegant

```
def quicksort(a)
  return a if a.length <= 1
  pivot = a.shift
  split = a.partition {|i| i < pivot}
  (quicksort(split[0]) << pivot << (quicksort(split[1]))).flatten
end
```

Concise

```
def fname_incr(fname)
  !File.exists?(fname) ?
    fname :
      /([0-9]+\.\.)/.match(fname) ?
        fname_incr(fname.sub(/([0-9]+)/) { |m| (m.to_i+1).to_s }) :
        fname_incr(fname.sub(/^[^\.]*/) { |m| m+'1' })
end
```

Powerful

```
def traverse_tree(addition = '', &proc)
  yield addition + "Q: #{self.value}"
  depth = $total_height - self.height
  # add 1 to depth because the root node has a depth of 0
  self.yes.traverse_tree("\n"+"t"*(depth+1), &proc)
  self.no.traverse_tree("\n"+"t"*(depth+1), &proc)
end
```