

gsrubylib

```
require 'debuglog' unless $gs_nodebuglog
require 'pry' unless $gs_nopry
require 'contracts'
include Contracts
```

if object.in? collection	
if object.not_nil?	non_nil?
str = object.pp_s	
o.define_method(:add) do x,y x + y end	
squares = (1..10).build_hash { n [n, n*n] }	graph
squares.values.mapf(&:to_s)	collectf
h = squares.apply_keys { k k.to_s }	
h = squares.apply_values { k k.to_s }	
“foo”.indent(4)	
“bar”.tabto(4)	
USAGE = %{\n usage: prog [-o dir] -h file...\n where\n -o dir outputs to DIR\n -h prints this message\n}.trim("\n")	
StringIO.string { o o.puts “Hi...” }	
class Person\n attr_predicate :young\n attr_predicate_rw :successful\nend	

Labels

```
Result = GS::Label.create(:win, :lose,\n  :draw)\nresult = Result.lose\nresult.to_s / to_sym / symbol / inspect\nresult == Result[:lose]
```

Labels are safer than symbols because they guard against misspellings. They also “inspect” nicely.

values

```
Person =\n  GS::Value.new(name: String, age: Nat, married: Bool)\n  .default(married: false)\n  .create
```

```
p = Person.new(name: 'John', age: 25)
```

```
p.name; p.age; p.married; p.married?\np[:name] # etc.
```

```
p.with(age: 26, married: true)
```

```
p.attributes\np.values\np.values(:name, :married)
```

```
e = p.upgrade(Employee, title: 'Nurse', salary: 58400)\np = e.downgrade(Person)
```

Values are read-only structs with Contracts built-in, default values, predicate methods, copy-constructors (with).

