

Structures

- Structures in lisp are rather like structs in C, defining a data type with named fields we can access individually
- We define a new type of structure with the defstruct function
- e.g. let's define a Fraction type, with fields numerator and denominator

```
(defstruct Fraction numerator denominator)
```

- This actually defines an entire suite of functions for us, designed to let us create and manipulate Fractions

Functions created by defstruct

- It defines a function to make a new Fraction (or whatever)
(make-Fraction) ; returns a new fraction
- It defines a function to test if something is a Fraction
(Fraction-p x) ; returns t iff x is a Fraction
- For each field, it defines a function to access that field
(Fraction-numerator f) ; return numerator of f
(setf (Fraction-numerator f) v) ; set numerator of f to v
- It defines a function to make a copy of a Fraction
(copy-Fraction f) ; returns a copy of f

Example

```
(defstruct Fraction num denom)
(defvar f (make-Fraction))
(setf (Fraction-num f) 4)
(setf (Fraction-denom f) 5)
(defvar g (copy-Fraction f))
(format t "numerator in g is ~A~%" (Fraction-num g))

; can also initialize values with make-Fraction, e.g.
(setf g (make-Fraction :num 11 :denom 27))
```