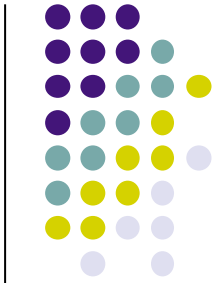
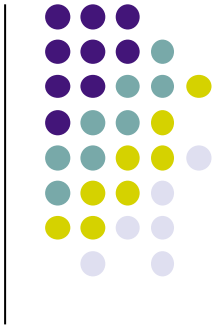
Atoms and records



- A record is a general compound data type
  - Records are used to build many other compound data types
  - To explain records, we first introduce the atom data type
- An atom is a symbolic value
  - A sequence of lowercase letters and digits that starts with a letter
  - Also, a sequence of any characters delimited by single quotes
  - Example of a list containing five atoms:

```
declare
L=[john paul george ringo ‘1337 5|*34|<‘]
{Browse L.1}
{Browse L.2.1}
{Browse {Length L}}
```

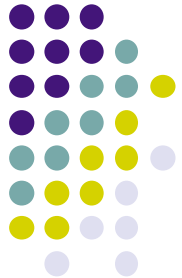
Records



- A record groups a set of values into a single compound value
  - A record has a fixed number of values that can be accessed directly
- Each record has a label and a set of pairs of field names and fields
  - The label is an atom, the field names are atoms or integers, and the fields can be any value
  - The field names and fields are separated by a colon ':'
  - The position of a field in the record is not important; the records point(x:10 y:20) and point(y:20 x:10) are identical
  - All field names must be different; the syntax box(in:deadcat in:livecat) is illegal while box(in:cat alive:X) is legal
- Example record with five fields:

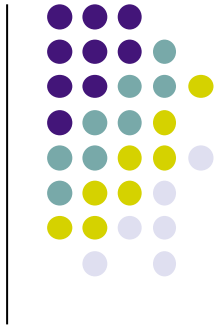
```
declare
R=rectangle(bottom:10 left:20 top:100 right:200 color:red)
```

Operations on records



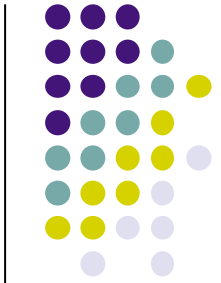
- We only give the basic operations; many other operations exist in the Record module
  - The following examples use this record:
    R=rectangle(bottom:10 left:20 top:100 right:200)
- Record fields are accessed through the dot operation
  - {Browse (R.top-R.bottom)*(R.right-R.left)}
- The label and fields can be extracted directly
  - {Label R} returns rectangle (the value of the label)
  - {Width R} returns 4 (the number of fields)
  - {Arity R} returns [bottom left right top] (list of field names alphabetically)
- Records can be used in comparisons and pattern matching
  - {Browse R==rectangle(top:100 bottom:10 left:20 right:200)} displays **true**
  - **case** R **of** rectangle(bottom:A top:B left:C right:D) matches with A=10, B=100, C=20, D=200

Records are the only compound type



- Records are the only compound type in the kernel language
  - An atom is a record whose width is 0
  - A tuple is a record whose field names are successive integers starting with 1
    - If the numbering condition is not satisfied, the data item is not a tuple but it is still a record
    - Fields without numbers are automatically numbered starting with 1: pair(H T) is syntactic sugar for pair(1:H 2:T)
  - A list is a recursive data type built with records nil and H|T
    - Syntactic sugar: H|T same as '|'(H T) same as '|'(1:H 2:T)
- This keeps the kernel language simple
  - A single compound data type suffices to understand execution
  - All other types (lists, trees, and so on) are encoded with records

Some examples



- Given the following records: are they tuples or lists?
  - A=a(1:a 2:b 3:c)
  - B=a(1:a 2:b 4:c)
  - C=a(0:a 1:b 2:c)
  - D=a(1:a 2:b 3:c d)
  - E=a(a 2:b 3:c 4:d)
  - F=a(2:b 3:c 4:d a)
  - G=a(1:a 2:b 3:c foo:d)
  - H='|'(1:a 2:'|'(1:b 2:nil))
  - I='|'(1:a 2:'|'(1:b 3:nil))