

CS344 - More Haskell

Note Title

3/19/2012

Announcements

- HW due Friday - bring to class!
- Next HW - on Haskell

Last time

- Types in Haskell

- Type classes
(not an O-O class!)

Ex: Num, Ord, Eq, Show, Read

Strong pattern Matching

Can use pattern matching in unexpected ways.

addVectors example

Another:

```
> let mylist = [(1,3), (4,3), (5,6)]  
> [a+b | (a,b) ← mylist]
```

Our own head function

head' :: [a] → a
head' [] = error 'Can't call head on empty list'
head' (x:_) = x

See firsttwo.hs

Exercise

Use pattern matching:

Write a function which computes the sum of the first 3 elements in a list.

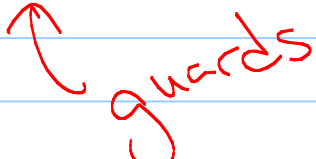
~~Write~~
don't call
it sum!

Now write a function which computes the sum of all elements in a list.
(Hint - use recursion!)

Guards

A way of testing if properties hold.
(a lot like if statements)

Example: where InAlpha :: String → String
where InAlpha word
| head word < 'h' = "beginning"
| head word < 'r' = "middle"
| otherwise = "end"

 guards

Where Bindings

We can set local names for expressions.

Ex:

initials :: String → String → String
initials firstname lastname =
[f] ++ ". " ++ [l] ++ ". "

where (f:_) = firstname
(l:_) = lastname