Functions

Example: write a function that computes the sum of the elements of a list of integers. (Hint: use recursion)

Cases:
[1,2,3,4,5] → 15
[6,12] → 18
[7] → 7
[ ] → 0

```
listsum [ ] = 0
listsum (x:rest) = x + listsum(rest)
```

recursion!

Functions

Example: write a function that computes the length of a list.

Cases:
[1,2,3] $\rightarrow$ 3
[1,2] $\rightarrow$ 2
[1] $\rightarrow$ 1
[ ] $\rightarrow$ 0

```
listlength [ ] = 0
listlength(x:rest) = 1 + listlength(rest)
```

$\nwarrow$ recursion!

Functions

Example: write a function that will reverse the order of the elements in a list.

Cases:

$[1,2,3] \rightarrow [3,2,1]$

$[1] \rightarrow [1]$

$[\,] \rightarrow [\,]$

```
reverselist [ ] = [ ]
reverselist (a:rest) = reverselist(rest)++[a]
```

Higher Order/Curried Functions

- Write a curried function that adds three values together.

```
add a = (\b->(\c->a+b+c))
```

Higher Order/Curried Functions

- Write a curried function that given two values it will return an ordered pair, e.g., 1 2 -> (1,2), 5 4 -> (4,5), 1001 1 -> (1,1001)

```
f a = (\b->if a<b then (a,b) else (b,a))
```

Higher Order/Curried Functions

- Using map, given a list of integers double the value of the integers on the list

```
map ((*) 2) [1,2,3]
```

Higher Order/Curried Functions

- Using map, write a function that given a list of integers doubles the value of the integers on the list

```
f  = map ((*) 2)
```

Functions

- Write a function that implements the bubble sort.

```
bubble [ ] = [ ]
bubble [a] = [a]
bubble (a:b:rest)
| a < b = a : bubble(b:rest)
| otherwise = bubble(b : bubble(a:rest))
```

Exercises

- Write a function *red3* of type (a,b,c)-> (a,c) that converts a tuple with three elements into one with two by eliminating the second element.
- Write a function *del3* of type [*a] ->* [a] whose output list is the same as the input list, but with the third element deleted (your function does not need to be defined on lists with length less than 3).
- Write a function *pow* of type (*Num a, Integral b) => (a,b) -> a* that raises a number to an integer power (your function does not need to be defined for integer values less than 0)
- Using map write a function that given a list of integers will compute a list of the squares of the input values.