

ERLANG BASICS: TUPLES, LAMBDA, AND LIST COMPREHENSIONS

Curt Clifton

Rose-Hulman Institute of Technology

SVN Update your repository,
Open *ErlangInClass/basics2.erl*

A large, textured red rectangular area with a white border. The red area has a slightly grainy, brush-stroke-like texture. The text "BACK TO BASICS" is centered in white, bold, sans-serif capital letters.

BACK TO BASICS

HELP!

- Helpful (but hard to find) module reference:
 - http://erlang.org/doc/man_index.html
 - See *lists* and *math* links from there in particular

MORE ERLANG BASICS

- Tuples — curly braces, typically tagged with atoms
- Lambdas — more *fun* than you can shake a stick at
 - You'll miss Scheme's and Haskell's syntax after working with Erlang *fun*

New Pet

WHAT ARE YOU DOING?

MOUNTING YOUR EEE PC IN A HAMSTER BALL.

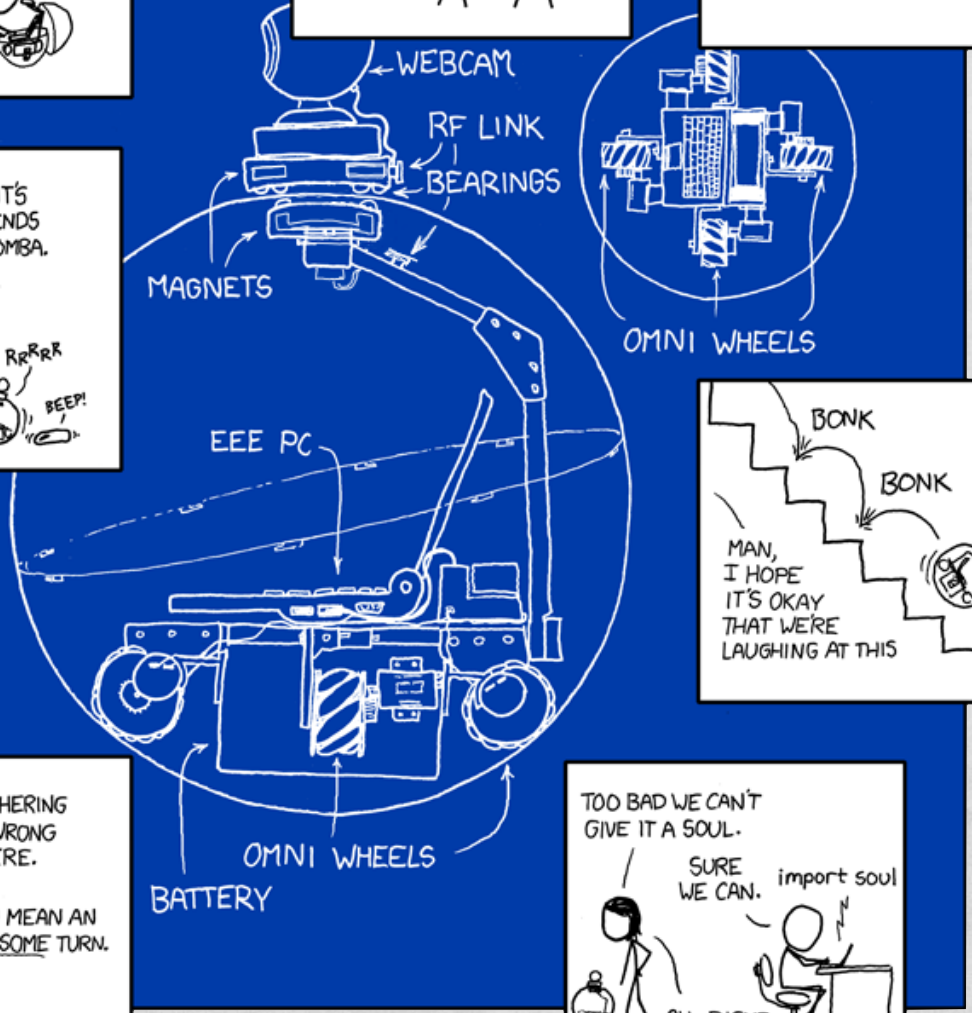
WELL, THE TCO OF A CAT IS LIKE \$1,000/YEAR, SO WE'RE SAVING MONEY.

MICROCONTROLLERS ARE ALL WIRED UP! HOW'S THE BRAIN COMING?

I'VE TAUGHT IT OBSTACLE AVOIDANCE AND BLOGGING.

AWW, LOOK, IT'S MAKING FRIENDS WITH THE ROOMBA.

RRRRR
BEEP!



BONK
BONK

MAN, I HOPE IT'S OKAY THAT WE'RE LAUGHING AT THIS

I THINK MY MOTHERING INSTINCT TOOK A WRONG TURN SOMEWHERE.

YOU MEAN AN AWESOME TURN.

TOO BAD WE CAN'T GIVE IT A SOUL.

SURE WE CAN. `import soul`

OH, RIGHT. PYTHON.

One laptop per hamster

LIST COMPREHENSIONS

- Syntax: `[expr | qualifier, ...]`
- Qualifiers give a list of possible variable bindings
- Expression is mapped over this list to get result
- Qualifiers can be:
 - `pattern <- listExpr`
 - `booleanExpr`

QUICKSORT

INEFFICIENT BUT ELEGANT

```
qsort([]) -> [];  
qsort([Pivot|T]) ->  
  qsort([X || X <-T, X < Pivot])  
  ++ [Pivot] ++  
  qsort([X || X <-T, X >= Pivot]).
```

Beautiful code, but it has two significant problems:

- Using first element as pivot is $O(n^2)$ for already sorted lists
- ++ operator stinks in Erlang
 - Why was it OK in Haskell?

DEALING WITH LISTS IN ERLANG

- Don't use ++
- Take items from head of list: $[H|T] = L$
- Add items to head of list: $L2 = [myNewItem|T]$
- Use `lists:reverse/1` when needed
- Use tail recursion whenever you can, it's *lightning fast*