



Daily Coding Problem #5

Problem

This problem was asked by Jane Street.

`cons(a, b)` constructs a pair, and `car(pair)` and `cdr(pair)` returns the first and last element of that pair. For example, `car(cons(3, 4))` returns 3, and `cdr(cons(3, 4))` returns 4.

Given this implementation of `cons`:

```
def cons(a, b):  
    def pair(f):  
        return f(a, b)  
    return pair
```

Implement `car` and `cdr`.

Solution

This is a really cool example of using [closures](#) to store data. We must look at the signature type of `cons` to retrieve its first and last elements. `cons` takes in `a` and `b`, and returns a new anonymous function, which itself takes in `f`, and calls `f` with `a` and `b`. So the input to `car` and `cdr` is that anonymous function, which is `pair`. To get `a` and `b` back, we must feed it yet another function, one that takes in two parameters and returns the first (if `car`) or last (if `cdr`) one.

```
def car(pair):  
    return pair(lambda a, b: a)  
  
def cdr(pair):  
    return pair(lambda a, b: b)
```

Fun fact: `cdr` is pronounced "cudder"!

© Daily Coding Problem 2019

[Privacy Policy](#)

[Terms of Service](#)

[Press](#)