Cheapest Item: Solution Review

Solution review.

Sorting

Getting the *cheapest* item implies you're sorting by price. After that grab the item's name.

A vanilla solution might look like this

index.js

cart.js

```
import cart from './cart';

const getCheapestItem = (items) => {
  const byPriceAsc = items
          .sort((a, b) => a.price - b.price);

  const cheapest = byPriceAsc[0];

  return cheapest.name;
};

const result = getCheapestItem(cart);

console.log({ result });
```

A $10 carrot is the cheapest item. What kind of grocery store is this?!

Anyways, we see the order of operations

1. Sort by price
2. Grab the first or last item (depending on how you sorted)
3. Return its name

We know `pipe` and `prop` from the last exercise and those seem like good candidates here. Ramda also carries a `sort` function.

index.js

cart.js

```
import { pipe, prop, sort } from 'ramda';
import cart from './cart';


const getCheapestItem = pipe(
  sort((a, b) => a.price - b.price),
  (list) => list[0],
  prop('name')
);

const result = getCheapestItem(cart);

console.log({ result });
```

This works, but a bit awkwardly. I'm not fond of how we're returning the *head* element

```
(list) => list[0]
```

Let's replace that with Ramda's `head` function. It returns a list's *head* element.

https://ramdajs.com/docs/#head

index.js

cart.js

```
import { head, pipe, prop, sort } from 'ramda';
import cart from './cart';

const getCheapestItem = pipe(
  sort((a, b) => a.price - b.price),
  head,
  prop('name')
);

const result = getCheapestItem(cart);

console.log({ result });
```

Much better. Pat yourself on the back if you got this.

But did you know about `sortBy`?

index.js

```
import { head, pipe, prop, sortBy } from 'ramda';
import cart from './cart';
```

cart.js

```
const getCheapestItem = pipe(
  sortBy(prop('price')),
  head,

  prop('name')
);

const result = getCheapestItem(cart);

console.log({ result });
```

It takes a function that describes how the data should be sorted. Useful if your sorts involve any complex logic.

https://ramdajs.com/docs/#sortBy