Compose

Pipe, but in the other direction. (2 min. read)

If `pipe` made sense, you already know `compose`. Instead of going **left-to-right**, we go **right-to-left**.

This returns 49 since we're using `pipe`.

```
import { pipe } from 'ramda';

const doMath = pipe(
  // start here
  double, // 2 * 2 = 4
  square, // 16
  triple, // 48
  increment // 49
);

const result = doMath(2);

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

But the same sequence with `compose` returns 162, since the order of operations have reversed.

```
import { compose } from 'ramda';

const doMath = compose(
  double, // 162
  square, // 81
  triple, // 9
  increment // 2 + 1 = 3
  // start here
);

const result = doMath(2);

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

To get 49, like `pipe`, flip the sequence.

```
import { compose } from 'ramda';

const doMath = compose(
  increment, // 49
  triple, // 48
  square, // 16
  double // 2 * 2 = 4
  // start here
);

const result = doMath(2);

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

`compose` mirrors the mathematical form of composition–the innermost function goes first.

```
increment(triple(square(double(x))));
```