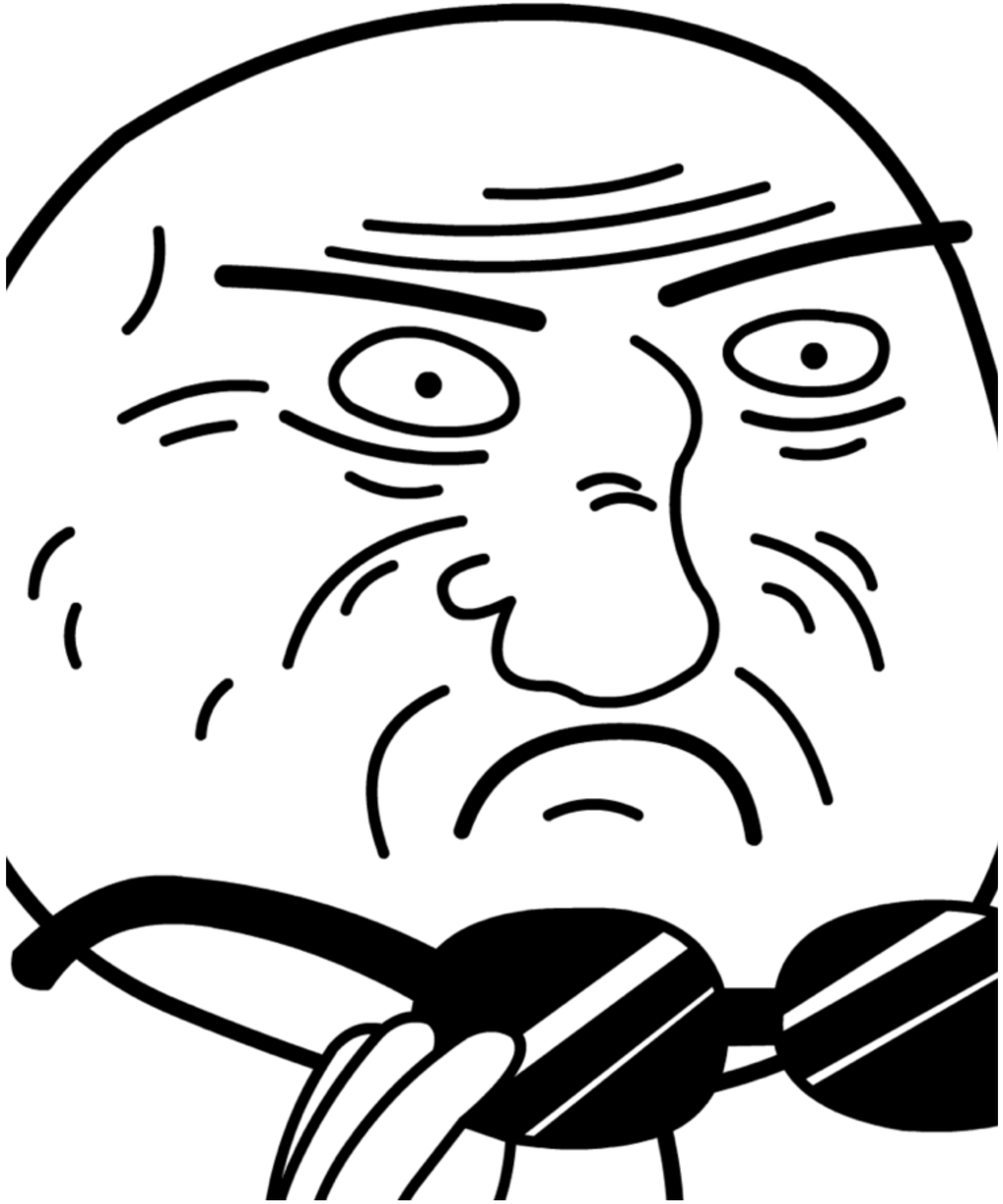
# Functional Programming

Monads

Endofunctors

# Abstract Data Types



There is no need

To be upset

# Pure Functions

# What makes a function pure?

Only relies on arguments for input

Doesn’t change their inputs

Always returns the same output for a given input

Doesn’t cause side effects (ie IO, update global state, throw exceptions)

# Not Pure

```
var counter = 0;

const increment => counter++;
```

# Pure

```
(num) => num + 1;
```

# Not Pure

```
(user, firstName, lastName) => {
    user.firstName = firstName;
    user.lastName = lastName;
};
```

# Pure

```
(user, firstName, lastName) =>

    Object.assign(user, {

        "firstName": firstName,

        "lastName": lastName

    });
```

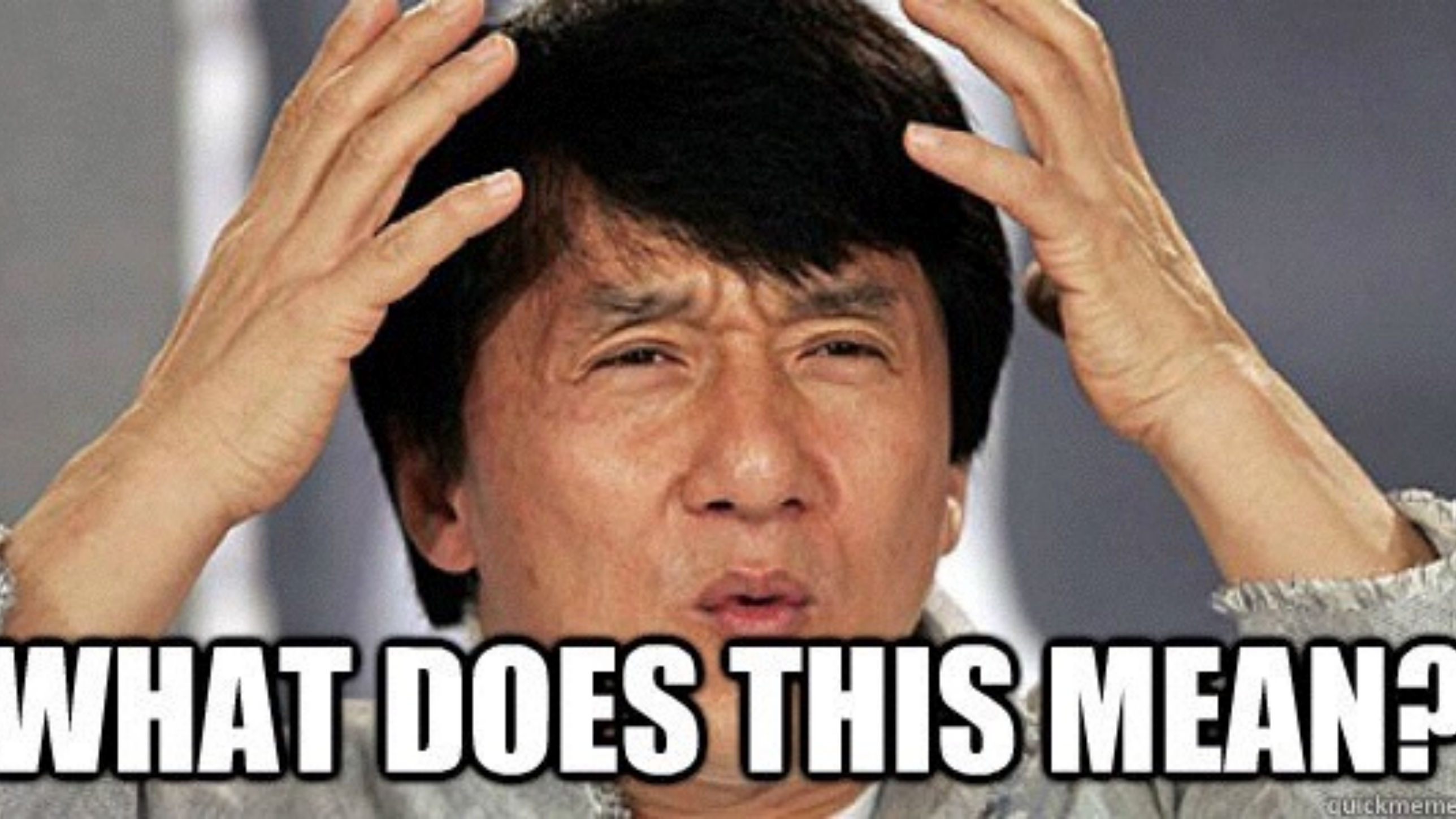
# Not Pure

```
(sql, callback) => getConnection().execute(sql, callback);
```

# (somewhat) Pure

```
// return a promise

(connection, sql) => connection.execute(sql);
```


WHAT DOES THIS MEAN?
quickmeme

# In short…

Lots of really cool mathematical guarantees for your code

Referential transparency

Can be reordered

Can be parallelized

Can be proven

Can be removed if not used

Can be memoized