

Jest, Snapshots and Beyond

ROGELIO GUZMAN
Frontend Engineer · Docker

 [@rogeliog](https://twitter.com/rogeliog)



@rogeliog



jest-core



I ❤️ testing

Test Runner
*Assertion
library*

Code coverage

Test Utilities

Watch mode

Test framework

Reporters

Mocking library

I ❤️ testing

Test Runner

Code coverage

Test framework

Watch mode

Reporters

*Assertion
library*

Test Utilities

Mocking library

I ❤️ the idea of testing

more effort => less tests

less effort => more tests

2016

Test Runner

*Assertion
library*

Code coverage

Jest

Test framework

Test Utilities

Watch mode

Reporters

Mocking library



Jest

Zero config

 6x faster instant run snapshot tests

Let's setup Jest

JS sum.test.js JS sum.js x

```
1 | function sum(a, b) {
2 |   return a + b;
3 |
4 |
5 | module.exports = sum;
6 |
7 |
```



~/dev/conferences/react-conf/live-code/live-code-jest-intro talk*

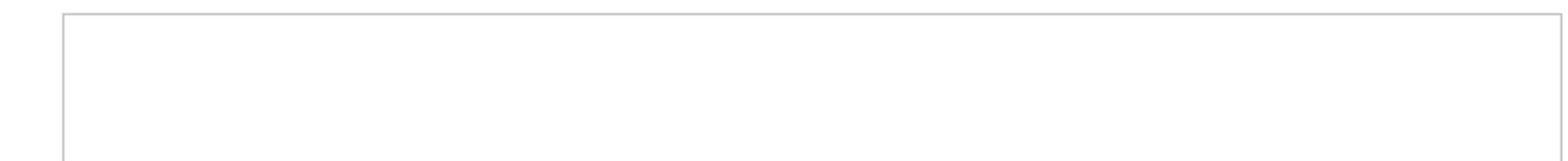
>

]



[rogeliog/jest-snapshot-talk](#)

Emoji Cinema



Start typing to see emojis!

Emoji Cinema

Frozen|



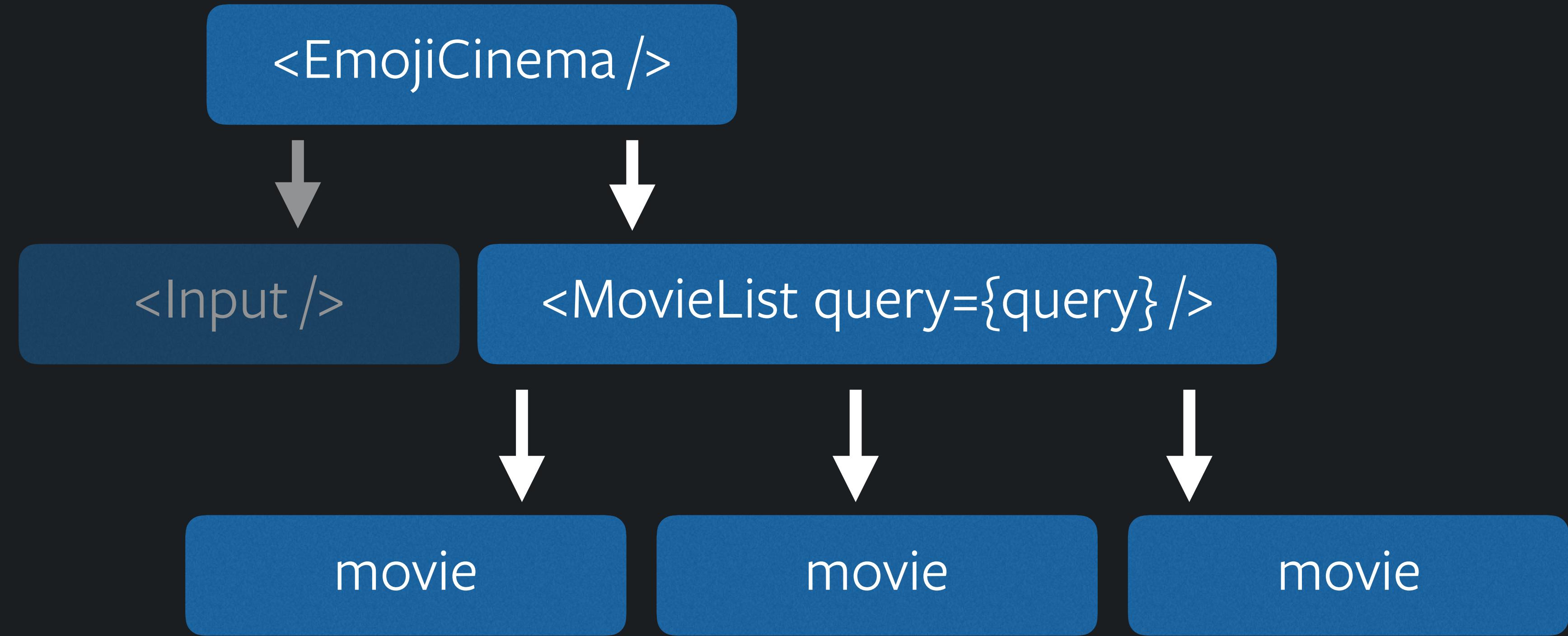
Frozen

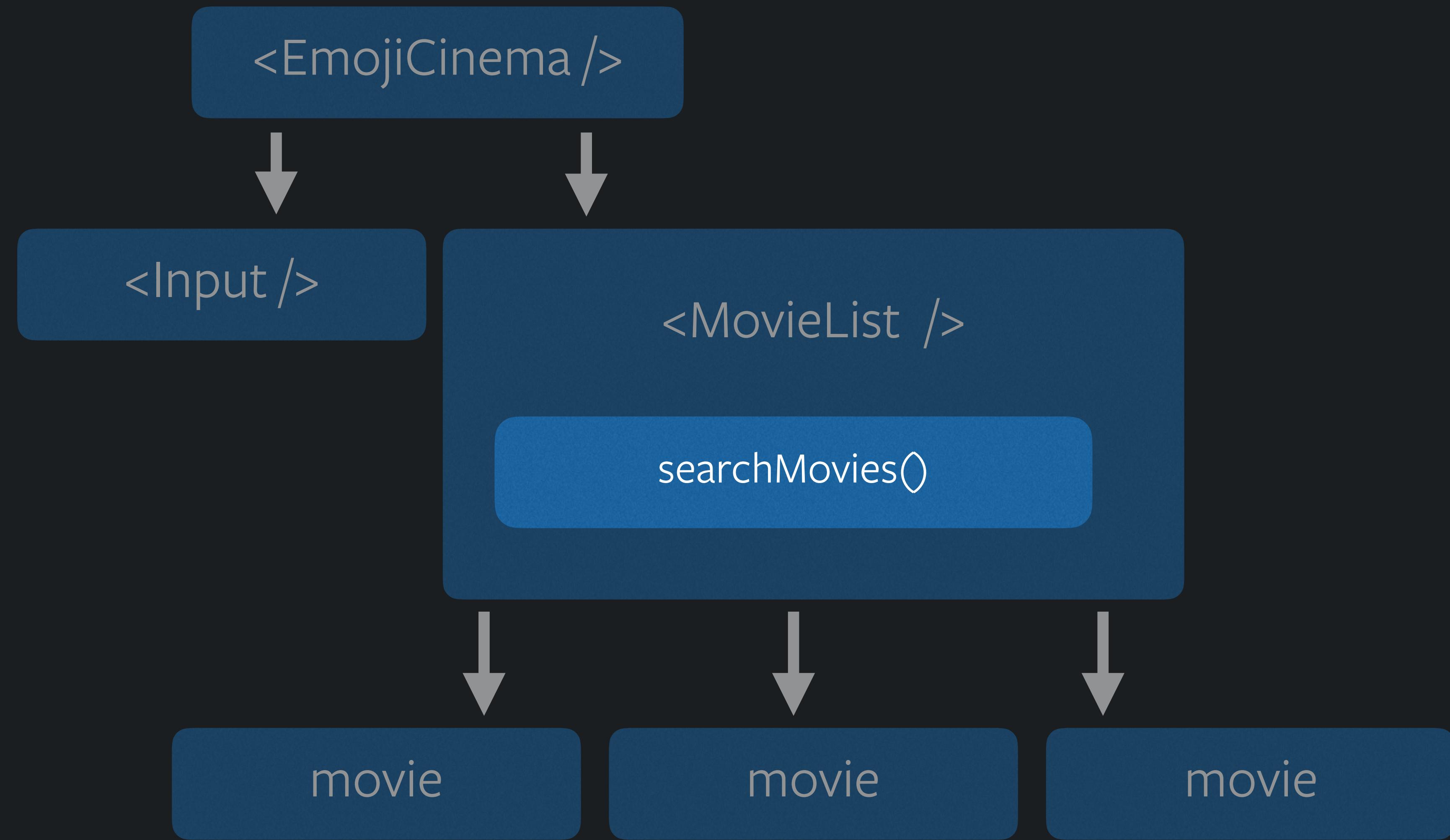
Emoji Cinema

Finding Nemo

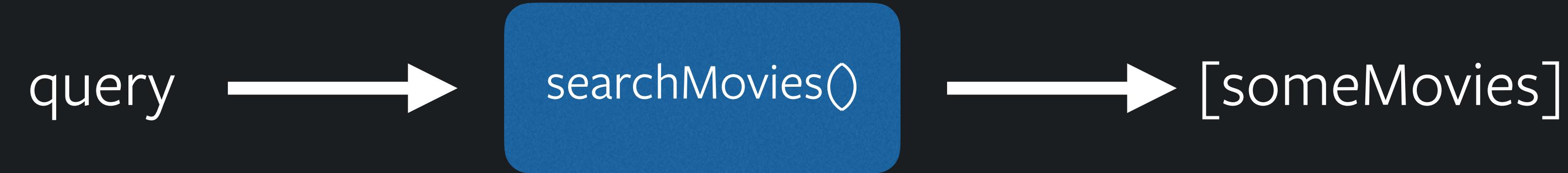


Finding Nemo





searchMovies



searchMovies

“F”



searchMovies()



[movies with an F]

```
function searchMovies(query) {  
  return emojiMap.filter((movie) => {  
    return movie.title.includes(query)  
  }) ;  
}  
}
```

```
const emojiMap = [
  { title: 'Frozen', emoji: '❄️👗' },
  { title: 'Finding Nemo', emoji: '🔍🐠' },
  { title: 'Kung Fu Panda', emoji: '👊💥🐼' },
  { title: 'Harry Potter', emoji: '🧙‍♂️👓⚡️' },
  { title: 'Hunger Games', emoji: '🍗🎮' },
  { title: 'The Lion King', emoji: '🦁👑' },
  { title: 'ET', emoji: '👽🚲🌙' },
];

function searchMovies(query) {
  return emojiMap.filter((movie) => {
    return movie.title.includes(query)
  });
}
```

How would we test this?

```
expect(searchMovies('F')).toEqual([someMovies]);
```

```
const searchMovies = require('./searchMovies');

it('Returns the movies that match the query', () => {
  expect(searchMovies('F')).toEqual([]);
})
```

```
const searchMovies = require('./searchMovies');

it('Returns the movies that match the query', () => {
  console.log(searchMovies('F'));
  // expect(searchMovies('F')).toEqual([]);
})
```

```
const searchMovies = require('./searchMovies');

it('Returns the movies that match the query', () => {
  console.log(searchMovies('F'));
  // expect(searchMovies('F')).toEqual([]);
})
```

Jest

```
console.log src/searchMovies.test.js:13
[ { title: 'Frozen', emoji: '❄️👗' },
  { title: 'Finding Nemo', emoji: '🔍🐠' },
  { title: 'Kung Fu Panda', emoji: '👊💥🐼' } ]
```

```
it('Returns the movies that match the query', () => {
  expect(searchMovies('F')).toEqual([
    { title: 'Frozen', emoji: '❄️👗' },
    { title: 'Finding Nemo', emoji: '🔍🐠' },
    { title: 'Kung Fu Panda', emoji: '👊💥🐼' },
  ]);
})
```

Jest

PASS

src/movieUtils.test.js

✓ Returns the movies that match the query (1ms)

FROZEN



Difference:

- Expected
- + Received

```
@@ -2,10 +2,14 @@
Object {
  "emoji": "\ud83d\udcda",
  "title": "Frozen",
},
Object {
+  "emoji": "\ud83d\udcda\udcbb",
+  "title": "Frozen 2",
+ },
+ Object {
    "emoji": "\ud83d\udcbb\udcbb",
    "title": "Finding Nemo",
},
```

```
it('Returns the movies that match the query', () => {
  expect(searchMovies('F')).toEqual([
    { title: 'Frozen', emoji: '❄️👗✌️' },
    { title: 'Finding Nemo', emoji: '🔍🐠' },
    { title: 'Kung Fu Panda', emoji: '👊💥🐼' },
  ]);
}) ;
```

Jest

```
"title": "Frozen",
},
Object {
+   "emoji": "❄️👗✌️",
+   "title": "Frozen 2",
+ },
+ Object {
"emoji": "🔍🐠",
```

```
it('Returns the movies that match the query', () => {
  expect(searchMovies('F')).toEqual([
    { title: 'Frozen', emoji: '❄️👗' },
    { title: 'Frozen 2', emoji: '❄️👗✌️' },
    { title: 'Finding Nemo', emoji: '🔍🐠' },
    { title: 'Kung Fu Panda', emoji: '👊💥🐼' },
  ]);
}) ;
```

Jest

PASS

src/movieUtils.test.js

✓ Returns the movies that match the query (1ms)

```
it('Returns the movies that match the query', () => {
  expect(searchMovies('F')).toEqual([
    { title: 'Frozen', emoji: '❄️👗' },
    { title: 'Frozen 2', emoji: '❄️👗✌️' },
    { title: 'Finding Nemo', emoji: '🔍🐠' },
    { title: 'Kung Fu Panda', emoji: '👊💥🐼' },
  ]);
}) ;
```

Jest

```
- Expected
+ Received

@@ -1,18 +1,22 @@
Array [
Object {
  "emoji": "❄️👗",
  "title": "Frozen",
+   "year": 2013,
},
```

```
it('Returns the movies that match the query', () => {
  expect(searchMovies('F')).toEqual([
    { year: 2013, title: 'Frozen', emoji: '❄️👗' },
    { year: 'TBA', title: 'Frozen 2', emoji: '❄️👗✌️' },
    { year: 2003, title: 'Finding Nemo', emoji: '🔍🐠' },
    { year: 2008, title: 'Kung Fu Panda', emoji: '👊💥🐼' },
  ]);
}) ;
}) ;
```

Jest

PASS

src/movieUtils.test.js

✓ Returns the movies that match the query (1ms)

console.log

copy



run test

paste

failing test

manually updating

run test



Problems with this approach

- Manual process
- Does not adapt to changes
- Hard to maintain

Snapshot Testing



```
expect(sum(1, 2)).toBe(3)
```

```
expect(sum(1, 2)).toMatchSnapshot()
```

```
expect(sum(1, 2)).toMatchSnapshot()
```

↓ 3

get the output of the function

↓

has saved
snapshot?

→ y

save it into a .snap file

↓ 3

Pass

Fail

↑ n

same as
snapshot?

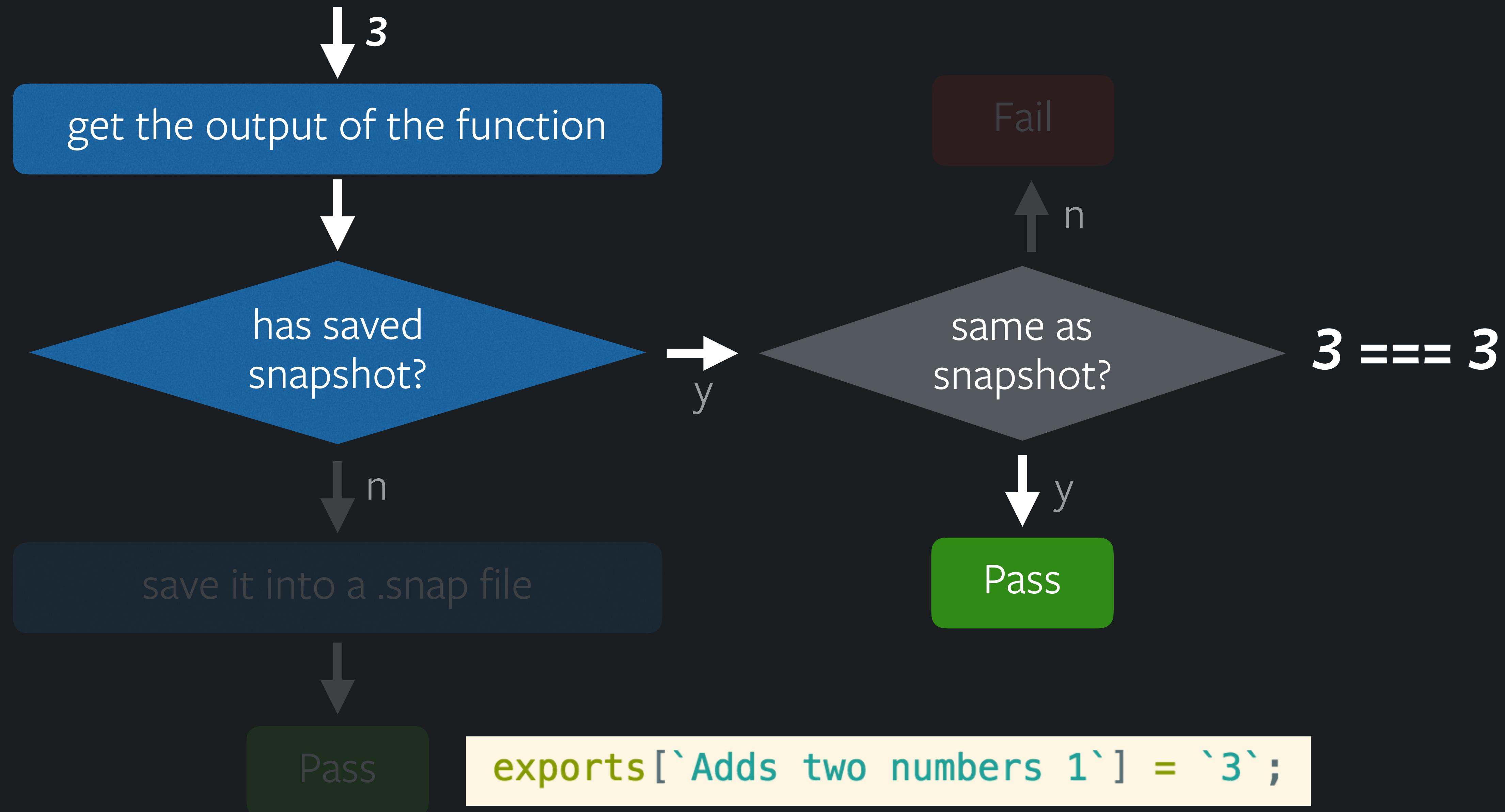
↓ y

Pass

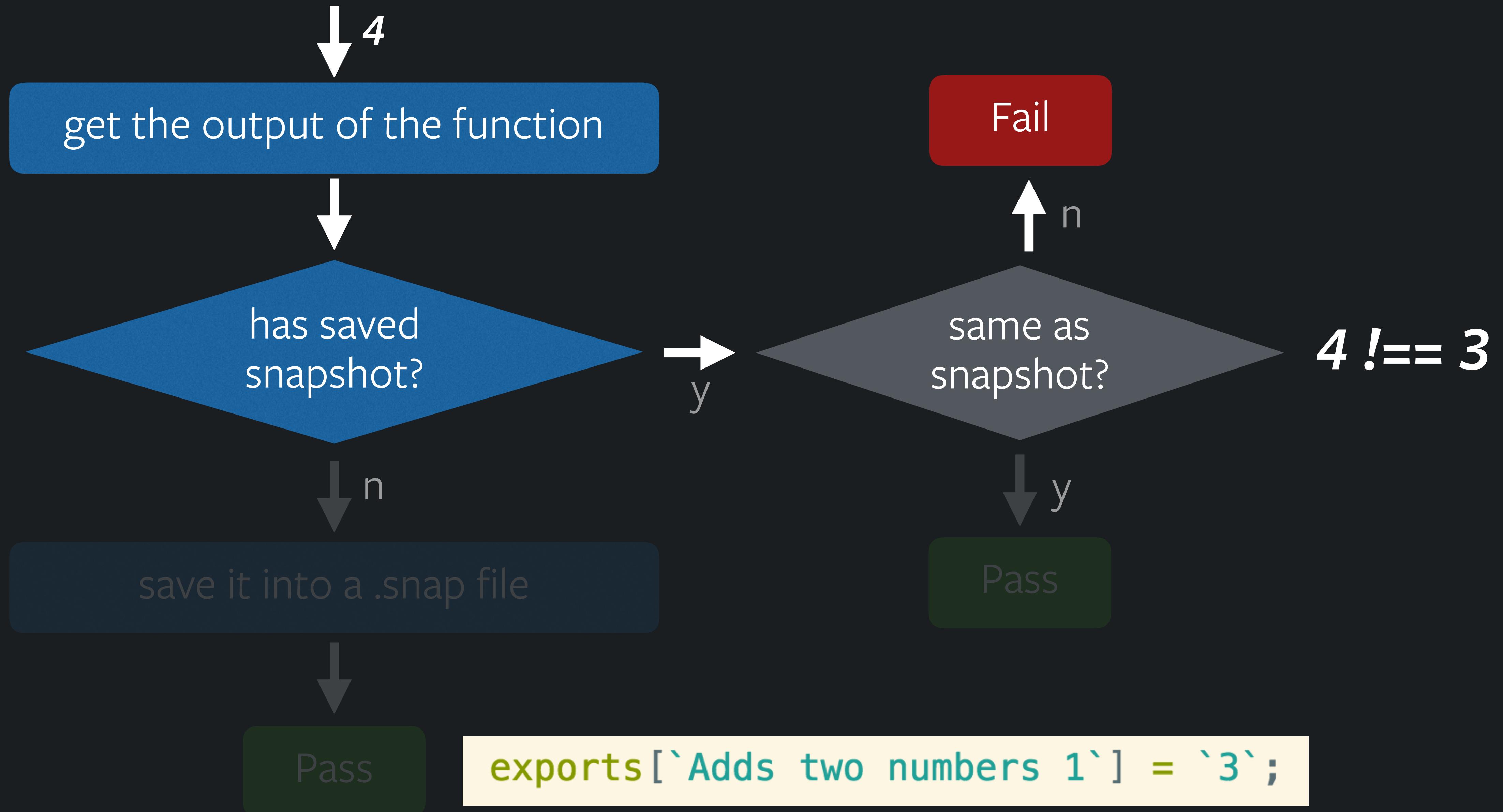
3 === 3

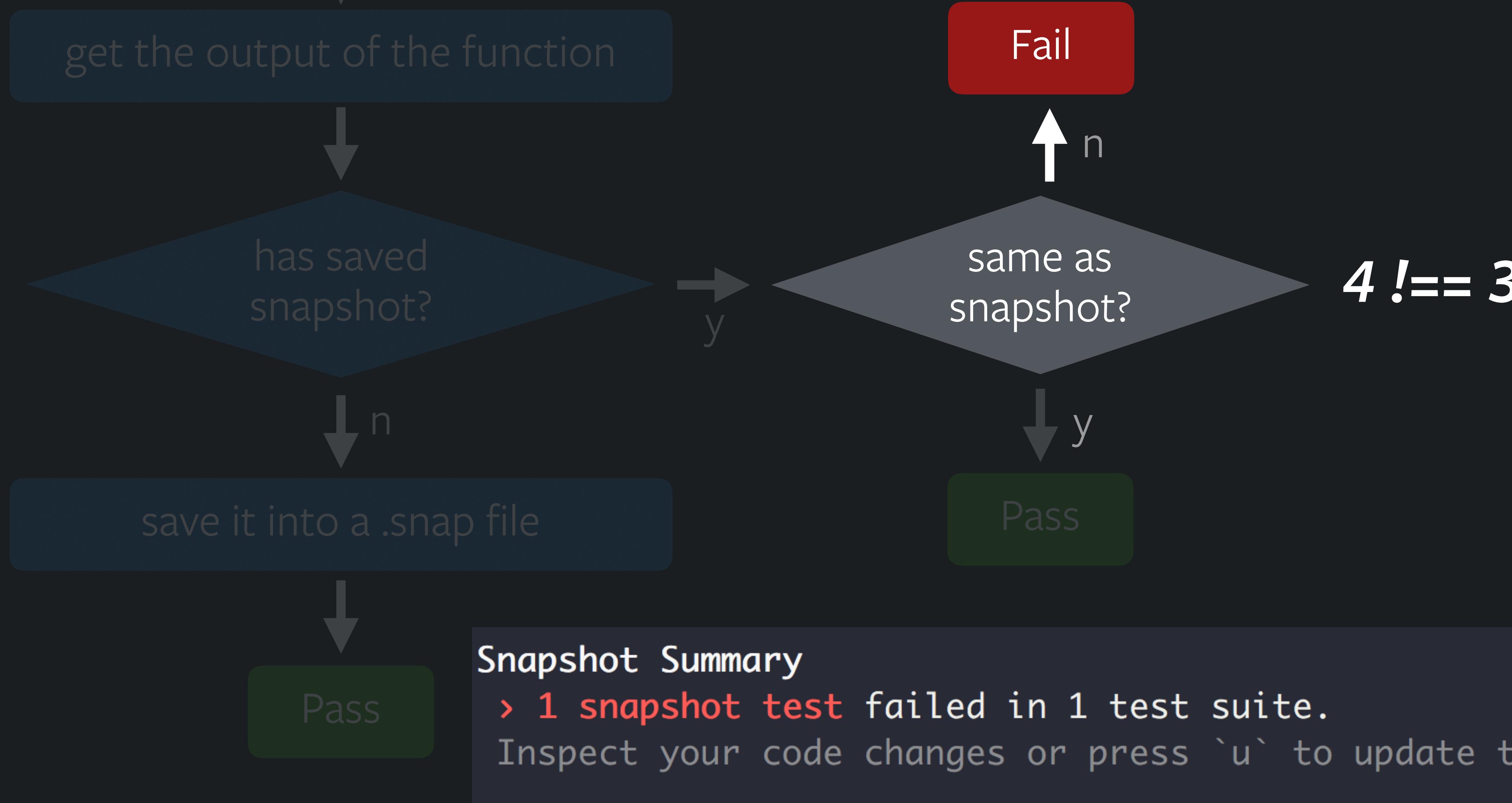
```
exports[`Adds two numbers 1`] = `3`;
```

```
expect(sum(1, 2)).toMatchSnapshot()
```



```
expect(sum(1, 2)).toMatchSnapshot()
```





Snapshot benefits

- 用工 Automated process, to manage snapshots
- 用工 Adapt to changes
- 用工 Easy to maintain

EXPLORER JS sum.test.js x JS sum.js

OPEN EDITORS

LIVE-CODE-JEST-INTRO

.vscode

coverage

node_modules

src

JS sum.js

JS sum.test.js

{ package.json

↳ yarn.lock

1 const sum = require('./sum');

2

3 it('Adds up two numbers', () => {

4 expect(sum(1, 2)).toBe(3);

5});

PASS src/sum.test.js

✓ Adds up two numbers (1ms)

Test Suites: 1 passed, 1 total

Tests: 1 passed, 1 total

Snapshots: 0 total

Time: 0.029s, estimated 1s

Ran all test suites related to changed files.

Watch Usage

- > Press a to run all tests.
- > Press p to filter by a filename regex pattern.
- > Press t to filter by a test name regex pattern.
- > Press q to quit watch mode.
- > Press Enter to trigger a test run.

Emoji Cinema

F



Frozen



Finding Nemo



Kung Fu Panda

```
it('Returns the movies that match the query', () => {
  expect(searchMovies('F')).toEqual([
    { title: 'Frozen', emoji: '❄️👗' },
    { title: 'Finding Nemo', emoji: '🔍🐠' },
    { title: 'Kung Fu Panda', emoji: '👊💥🐼' },
  ]);
}) ;
```

Jest

PASS

src/movieUtils.test.js

✓ Returns the movies that match the query (1ms)

```
it('Returns the movies that match the query', () => {
  expect(searchMovies('F')).toMatchSnapshot();
});
```

Jest

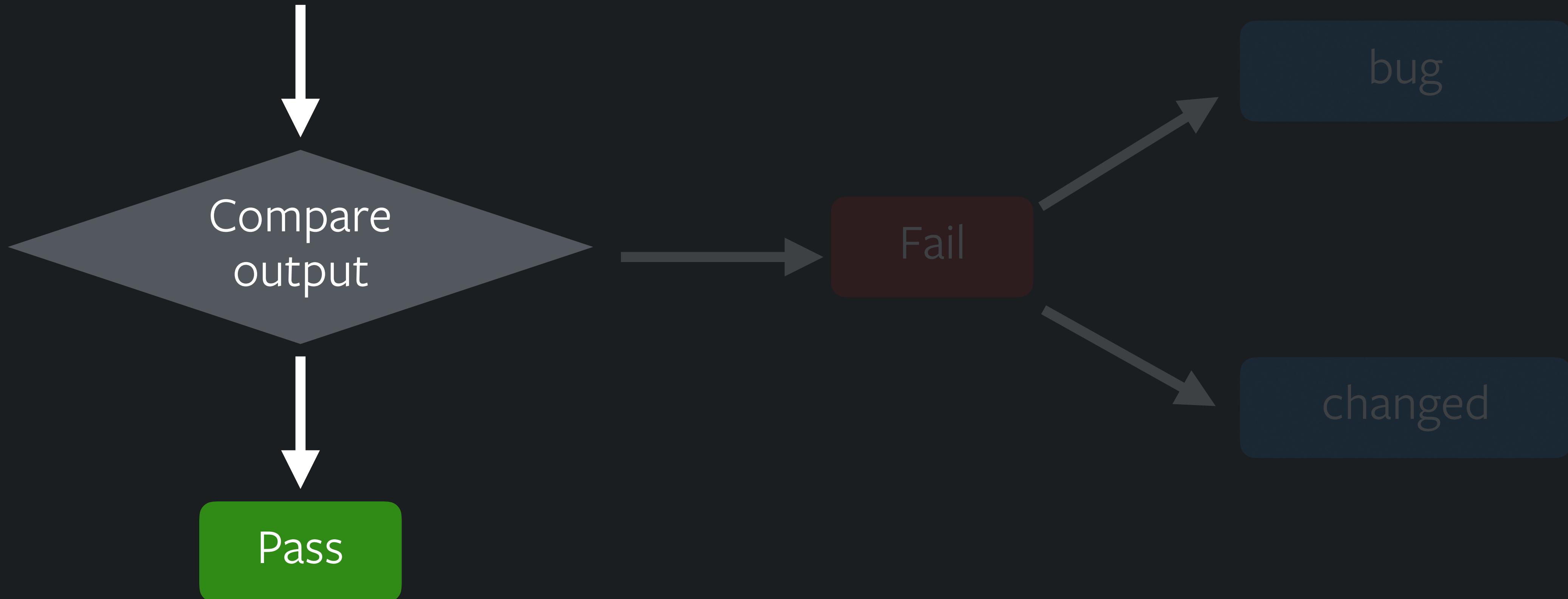
PASS src/searchMovies.test.js

✓ Returns the movies that match the query (2ms)

Snapshot Summary

→ > 1 snapshot written in 1 test suite.

searchMovie('F')



```
exports[`Returns the movies that match the query 1`] = `  
Array [  
  Object {  
    "emoji": "\ud83c\udc7f \ud83d\udcbb",  
    "title": "Frozen",  
  },  
  Object {  
    "emoji": "\ud83d\udcbb \ud83c\udc7f",  
    "title": "Finding Nemo",  
  },  
  Object {  
    "emoji": "\ud83d\udcbb \ud83c\udc7f \ud83d\udcbb",  
    "title": "Kung Fu Panda",  
  },  
];  
`;
```

FROZEN



```
expect(value).toMatchSnapshot()
```

Received value does not match stored snapshot 1.

- Snapshot
- + Received

```
@@ -2,10 +2,14 @@
Object {
    "emoji": "\ud83c\udc0d\ud83c\udc0e",
    "title": "Frozen",
},
Object {
+    "emoji": "\ud83c\udc0d\ud83c\udc0e\ud83c\udc1f",
+    "title": "Frozen 2",
},
Object {
    "emoji": "\ud83c\udc1e\ud83c\udc0d",
    "title": "Finding Nemo",
},
Object {
    "emoji": "\ud83c\udc1f\ud83c\udc0d\ud83c\udc1d",
}
```

searchMovie('F')



Compare
output



Fail



bug

changed

Pass

```
it('Returns the movies that match the query', () => {
  expect(searchMovies('F')).toMatchSnapshot();
}) ;
```

Jest

PASS src/searchMovies.test.js

✓ Returns the movies that match the query (2ms)

Snapshot Summary

→ > 1 snapshot written in 1 test suite.

```
exports[`test Returns the movies that match the query 1`] = `  
Array [  
  Object {  
    "emoji": "\ud83c\udc9e \ud83d\udcbb",  
    "title": "Frozen",  
  },  
  Object {  
    "emoji": "\ud83c\udc9e \ud83d\udcbb \ud83d\udcbb ",  
    "title": "Frozen 2",  
  },  
  Object {  
    "emoji": "\ud83d\udcbb \ud83c\udc9e",  
    "title": "Finding Nemo",  
  },  
  Object {  
    "emoji": "\ud83d\udcbb \ud83d\udcbb \ud83d\udcbb",  
    "title": "Kung Fu Panda",  
  },  
]
```

```
Array [  
  Object {  
    "emoji": "\ud83c\udc0d \ud83d\udcda",  
    "title": "Frozen",  
  },  
  Object {  
    "emoji": "\ud83c\udc0d \ud83d\udcda \ud83c\udc1f",  
    "title": "Frozen 2",  
  },  
  Object {  
    "emoji": "\ud83c\udc0d \ud83d\udcda \ud83d\udcbb",  
    "title": "Finding Nemo",  
  },  
  Object {  
    "emoji": "\ud83d\udcbb \ud83c\udc1f \ud83d\udcbb",  
    "title": "Kung Fu Panda",  
  },  
]
```



```
Array [  
  "\ud83c\udc0d \ud83d\udcda Frozen",  
  "\ud83c\udc0d \ud83d\udcda \ud83c\udc1f Frozen 2",  
  "\ud83c\udc0d \ud83d\udcda \ud83d\udcbb Finding Nemo",  
  "\ud83d\udcbb \ud83c\udc1f \ud83d\udcbb Kung Fu Panda",  
]
```

Snapshot Serializers

What should the snapshot look like

```
expect.addSnapshotSerializer( {  
    test: (val) => val.title && val.emoji,  
    print: (val) => ` ${val.emoji} ${val.title}`  
} ) ;
```

```
expect.addSnapshotSerializer( {  
    test: (val) => val.title && val.emoji,  
    print: (val) => `${val.emoji} ${val.title}`  
} );
```

```
expect.addSnapshotSerializer( {  
    test: (val) => val.title && val.emoji,  
    print: (val) => ` ${val.emoji} ${val.title}`  
} ) ;
```

```
expect.addSnapshotSerializer( {  
    test: (val) => val.title && val.emoji,  
    print: (val) => ` ${val.emoji} ${val.title}`  
} ) ;
```

```
expect.addSnapshotSerializer({  
  test: (val) => val.title && val.emoji,  
  print: (val) => `${val.emoji} ${val.title}`  
}) ;
```

```
it('Returns the movies that match the query', () => {  
  expect(searchMovies('F')).toMatchSnapshot();  
}) ;
```

Jest

FAIL src/searchMovies.test.js
• Returns the movies that match the query

```
expect(value).toMatchSnapshot()
```

Received value does not match stored snapshot 1.

- Snapshot
- Received

```
exports[`test Returns the movies that match the query 1`] = `  
Array [  
  "❄️👗 Frozen",  
  "❄️👗✌️ Frozen 2",  
  "🔍🐠 Finding Nemo",  
  "👊💥🐼 Kung Fu Panda",  
]  
`;
```

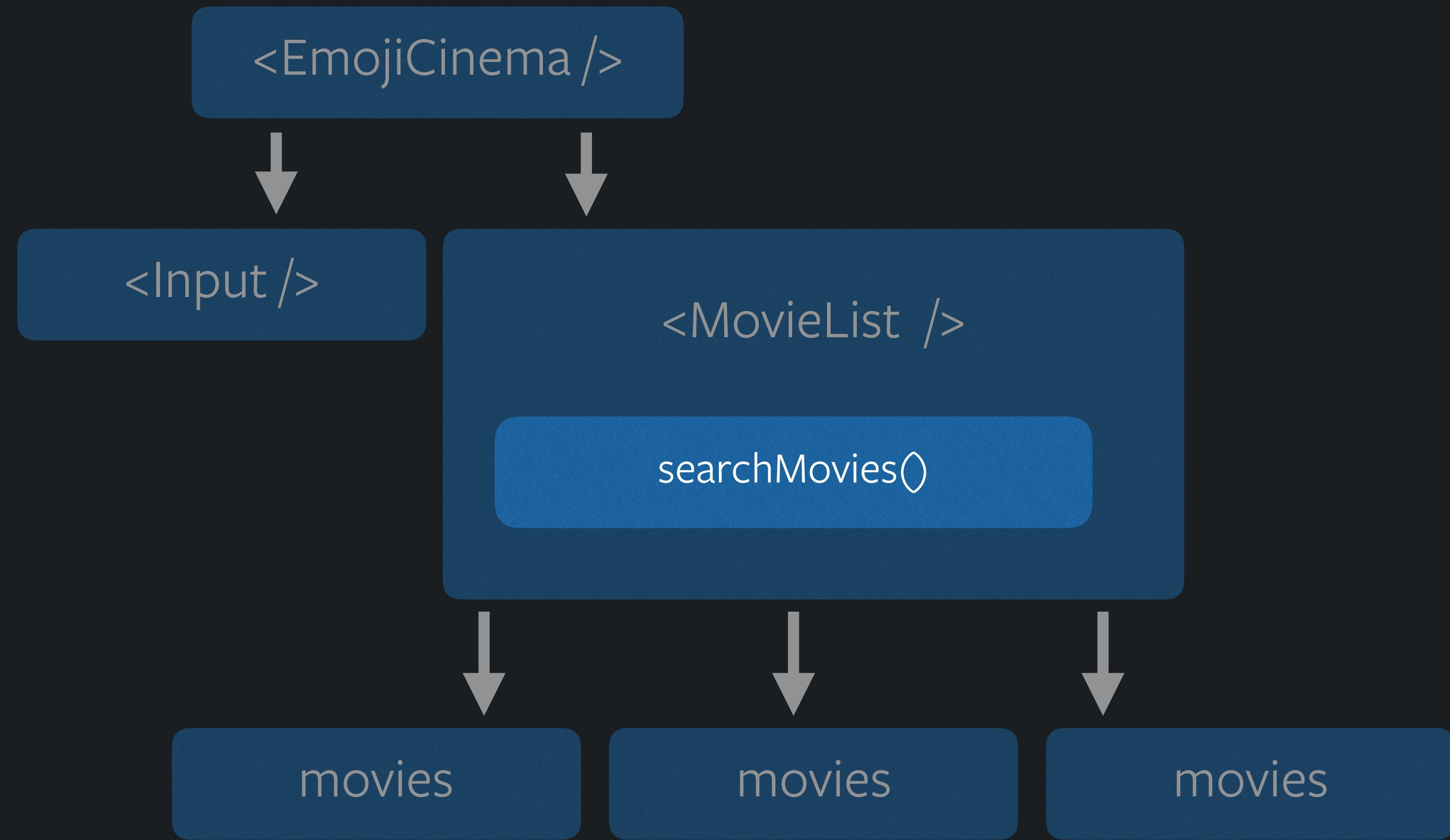
```
it('Returns the movies that match the query', () => {
  expect(searchMovies('F')).toMatchSnapshot();
}) ;
```

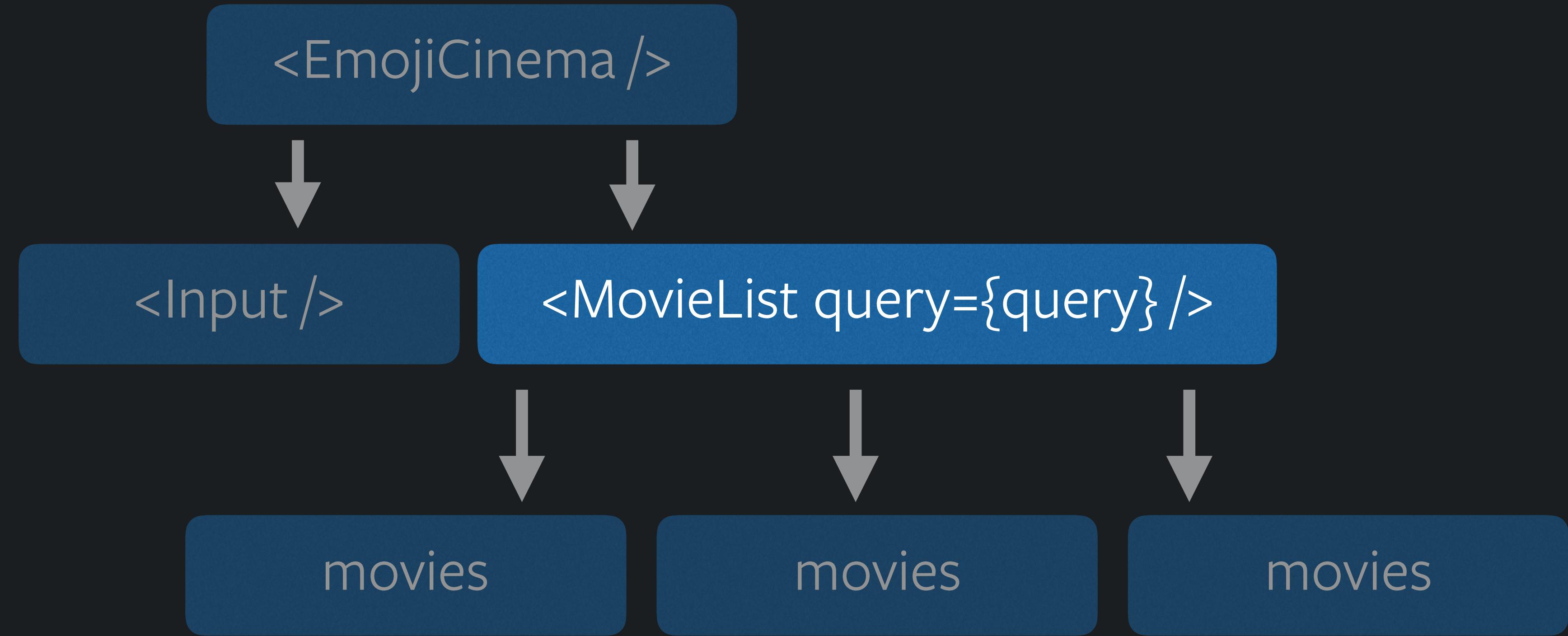
Jest

PASS

src/movieUtils.test.js

✓ Returns the movies that match the query (1ms)





How would we test this?

- 用工匠精神 Simply don't test it
- 用工匠精神 Selenium: flaky, slow, and high maintenance
- 用工匠精神 Snapshots

How would we test this?



```
import React from 'react';
import renderer from 'react-test-renderer'
import MovieList from './MovieList';

it('Renders movies', () => {
  expect(component).toMatchSnapshot();
}) ;
```

```
import React from 'react';
import renderer from 'react-test-renderer'
import MovieList from './MovieList';

it('Renders movies', () => {
  const component = renderer.create(
    <MovieList query="F" />
  );

  expect(component).toMatchSnapshot();
}) ;
```

```
it('Renders movies', () => {
  const component = renderer.create(
    <MovieList query="F" />
  ) ;

  expect(component).toMatchSnapshot();
}) ;
```

Jest

PASS src/MovieList.test.js

✓ Renders movies (5ms)

Snapshot Summary

→ > 1 snapshot written in 1 test suite.

```
exports[`test Renders movies 1`] = `<ul>
  <li
    className="Movie">
      <div>
        ❄️ 🎡 Frozen
      </div>
    </li>
  <li
    className="Movie">
      <div>
        🔎🐠 Finding Nemo
      </div>
    </li>
  <li
    className="Movie">
      <div>
        💥👊🐼 Kung Fu Panda
      </div>
    </li>
  </ul>
`;
```

FROZEN



```
expect(value).toMatchSnapshot()
```

```
Received value does not match stored snapshot 1.
```

- Snapshot
- + Received

```
@@ -6,10 +6,16 @@
      </div>
    </li>
    <li
      className="Movie">
      <div>
        +
        ❄️ 💃 🎉 Frozen 2
      +
      </div>
      +
      </li>
      +
      <li
        +
        className="Movie">
        +
        <div>
```

```
it('Renders movies', () => {
  const component = renderer.create(
    <MovieList query="F" />
  ) ;

  expect(component).toMatchSnapshot();
}) ;
```

Jest

PASS src/MovieList.test.js

✓ Renders movies (5ms)

Snapshot Summary

→ > 1 snapshot written in 1 test suite.

What about Enzyme?

Hint:  Serializers

```
it('Renders movies', () => {
  const component = renderer.create(
    <MovieList query="F" />
  );
  expect(component).toMatchSnapshot();
}) ;
```

```
import { mount } from 'enzyme';
import enzymeSerializer from 'enzyme-to-json/serializer';

expect.addSnapshotSerializer(enzymeSerializer);

it('Renders movies', () => {
  const component = mount(
    <MovieList query="F" />
  );
  expect(component).toMatchSnapshot();
}) ;
```

Jest

PASS src/MovieList.test.js
✓ Renders movies (3ms)

```
import { shallow } from 'enzyme';
import enzymeSerializer from 'enzyme-to-json/serializer';

expect.addSnapshotSerializer(enzymeSerializer);

it('Renders movies', () => {
  const component = shallow(
    <MovieList query="F" />
  );
  expect(component).toMatchSnapshot();
}) ;
```

Snapshots over time

- 用工匠精神 Our application is not static
- 用工匠精神 Capture the transformation of an object over time

```
it('Renders movies', () => {
  const component = renderer.create(
    <MovieList query="F" />
  ) ;

  expect(component).toMatchSnapshot();
}) ;
```

Jest

PASS src/MovieList.test.js
✓ Renders movies (3ms)

```
it('Renders movies', () => {
  ['F', 'Fro', 'Frozen 2'].forEach((query) => {
    const component = renderer.create(
      <MovieList query={query} />
    );
    expect(component).toMatchSnapshot();
  });
});
```

Jest

PASS src/MovieList.test.js
✓ Renders movies (2ms)

Snapshot Summary
→ > 2 snapshots written in 1 test suite.

```
<MovieList query="Fro"/>
```

```
exports[`test Renders movies 2`] = `<ul>
<li
  className="Movie">
    <div>
      ❄️ 👗 Frozen
    </div>
  </li>
<li
  className="Movie">
    <div>
      ❄️ 👗 ✌️ Frozen 2
    </div>
  </li>
</ul>
`;
```

```
<MovieList query="Frozen 2"/>
```

```
exports[`test Renders movies 3`] = `<ul>
<li
  className="Movie">
    <div>
      ❄️ 👗 ✌️ Frozen 2
    </div>
  </li>
</ul>
`;
```

jest-movie

- Movie, where each snapshot is a frame
- Tools for visualizing snapshots over time
- Is built on top of other Jest packages

File: MovieList.test.js.snap

Movie: test Renders movies

Frame: 1/3

- > Press left arrow to see previous frame.
- > Press right arrow to see next frame.

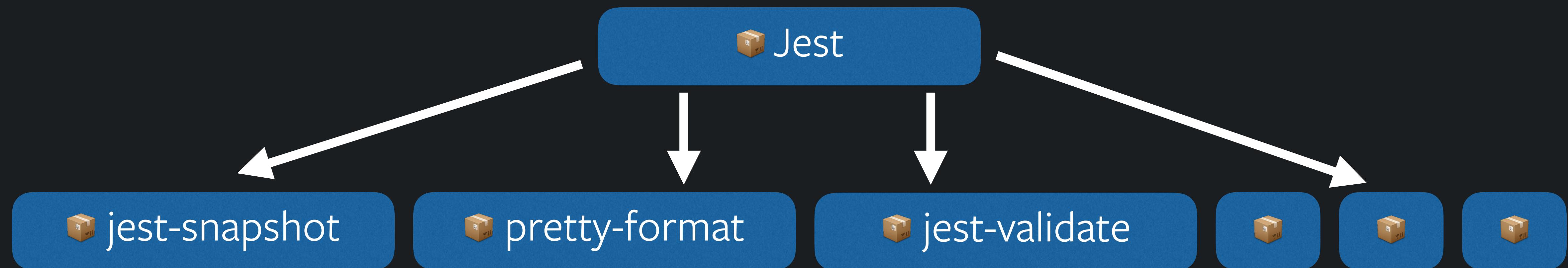
```
<ul>
  <li className="Movie">
    <div>❄️👗Frozen</div>
  </li>
  <li className="Movie">
    <div>❄️👗✌️Frozen 2</div>
  </li>
  <li className="Movie">
    <div>🔍🐠Finding Nemo</div>
  </li>
  <li className="Movie">
    <div>🥋💥⚽️Kung Fu Panda</div>
  </li>
</ul>
```

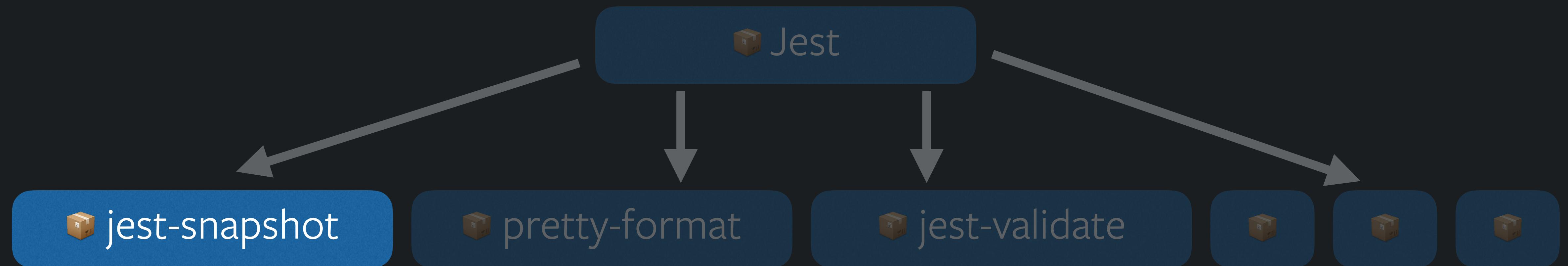
Jest is a testing platform

Jest is a testing platform



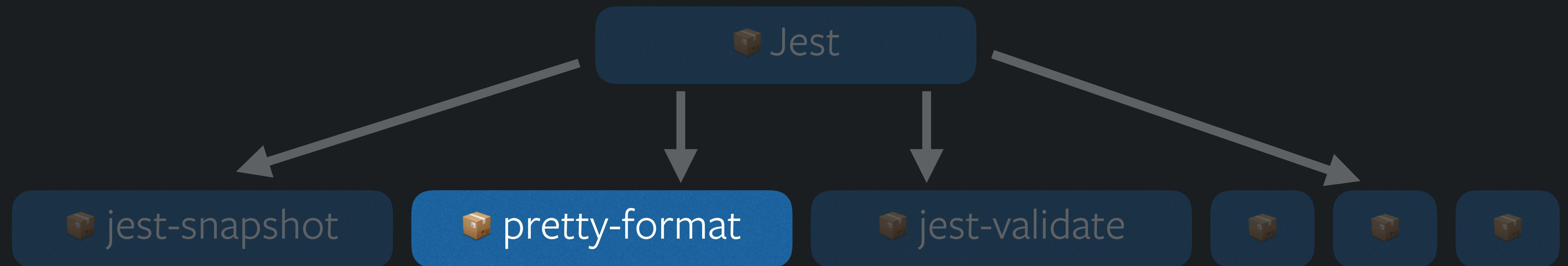
Jest





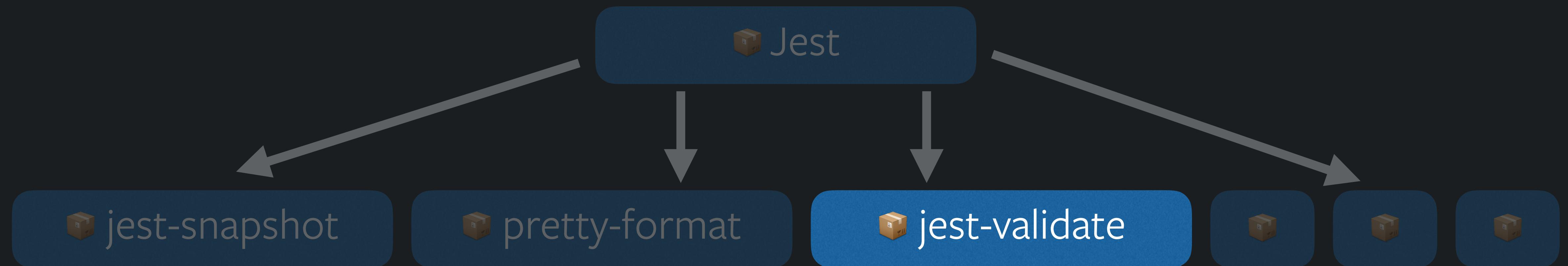
Standalone snapshot package

- Manage and update snapshots
- Manage serializers
- toMatchSnapshot()



Stringify any JavaScript value

- Supports all built-in JavaScript types
- Plugin system
- React and Immutable.js



Generic configuration validation tool

- ⌚ descriptive validation errors
- ⌚ deprecation warnings
- ⌚ configuration examples

What's next?

- 大型多人研究
- 邀请您贡献

Thanks! 😊

@rogeliog