JONAS SCHMEDTMANN

THE ULTIMATE REACT COURSE

@JONASSCHMEDTMAN

SECTION

A FIRST LOOK AT REACT

LECTURE

WHY DO FRONT-END FRAMEWORKS EXIST?

THE RISE OF **SINGLE-PAGE APPLICATIONS**

WordPress

NEXT.JS
Remix

The “old” way (but now getting modern again...)

SERVER-SIDE RENDERING

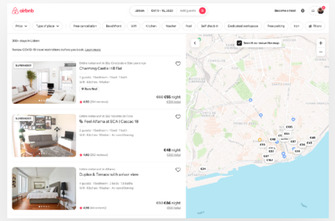
VIEW

DATA

RENDER WEBPAGE

HTML CSS JS

BROWSER



jQuery
write less, do more.

👉 *Developers started writing more and more JS code, which led to...*

SERVER / BACK-END

CLIENT / FRONT-END

The “modern” way

CLIENT-SIDE RENDERING

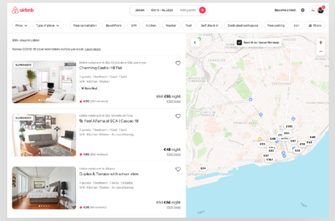
DATA

API

BROWSER

RENDER WEBPAGE

HTML CSS JS



VIEW

👉 **Single-Page Application (SPA)**

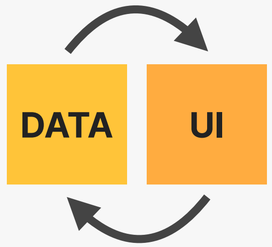
SINGLE-PAGE APPLICATIONS WITH **VANILLA JAVASCRIPT?**

👉 *Front-end web applications are all about...*

Handling data + **displaying** data in a user interface

↓

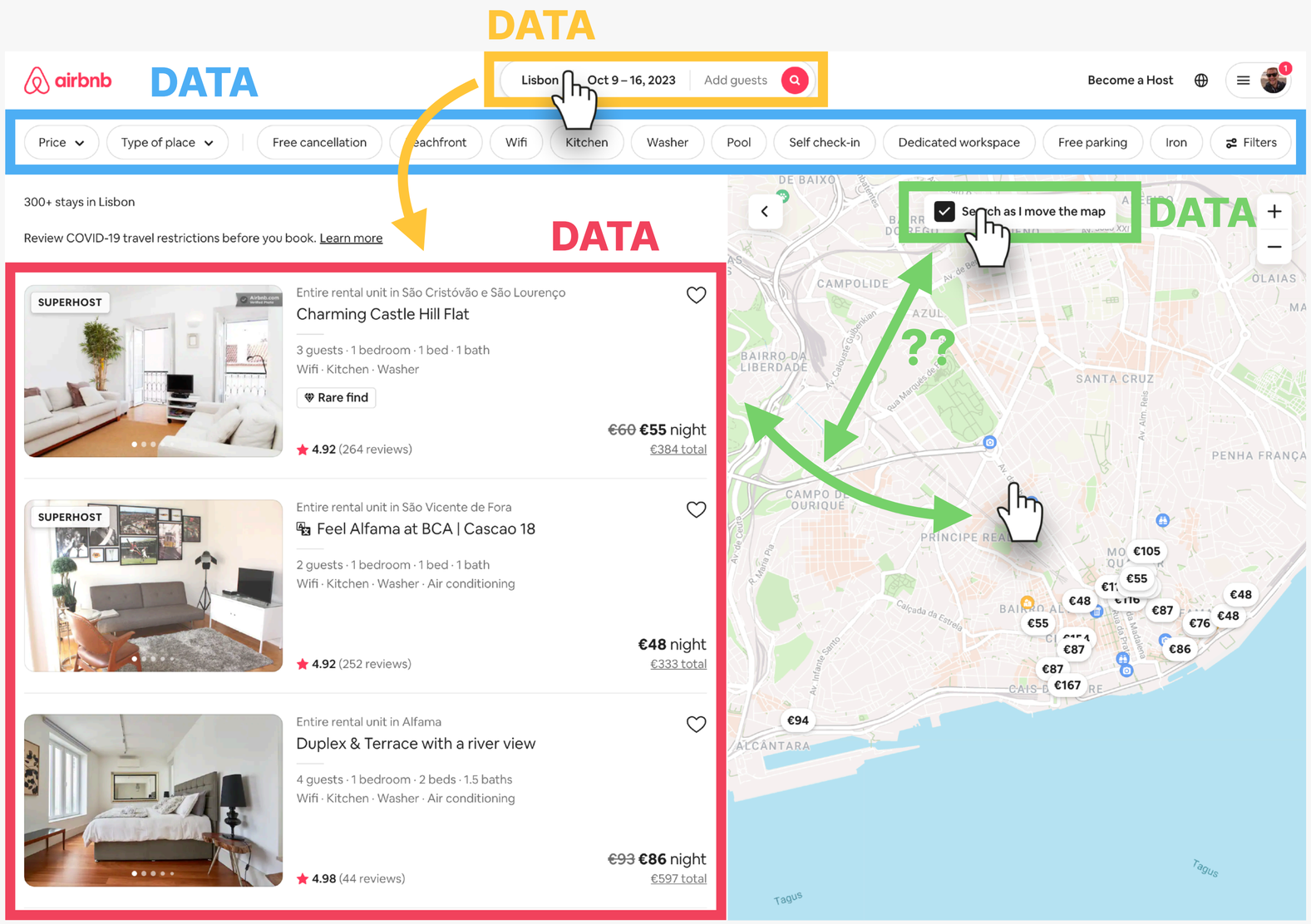
User interface needs to **stay in sync** with data

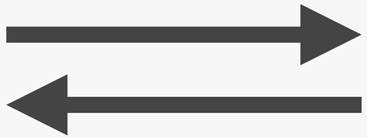


↓

Very hard problem to solve!

KEEPING UI **IN SYNC** WITH DATA





☝ Keeping UI and data in sync would be **virtually impossible with just vanilla JavaScript**

Piece of data
=
Piece of *state*

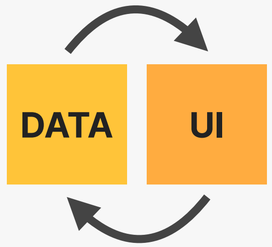
SINGLE-PAGE APPLICATIONS WITH **VANILLA JAVASCRIPT?**

👉 *Front-end web applications are all about...*

Handling data + **displaying** data in a user interface

↓

User interface needs to **stay in sync** with data



↓

Very hard problem to solve!

PROBLEMS WITH JS jQuery

write less, do more.

1

Requires lots of direct **DOM manipulation** and **traversing** (*imperative*) 👉 "Spaghetti code" 🍝

```
const guestsEl = document.querySelector('.guests');
const guestsPickerEl = document.querySelector('.picker');

guestsEl.addEventListener('click', function () {
  guestsEl.classList.toggle('inactive');
  guestsEl.classList.toggle('active');

  if (guestsPickerEl.style.display === 'block') {
    guestsPickerEl.style.display = 'none';
    guestsEl.firstElementChild.textContent = 'Add guests';
  } else {
    guestsPickerEl.style.display = 'block';
    guestsEl.firstElementChild.textContent = '';
  }
```

2

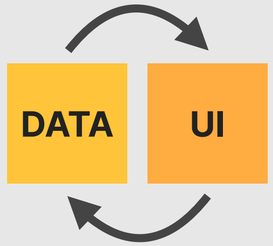
Data (state) is usually **stored in the DOM**, shared across entire app 👉 Hard to reason + bugs 🐛

WHY DO FRONT-END FRAMEWORKS **EXIST**?

1

JavaScript front-end frameworks exist because...

KEEPING A USER INTERFACE IN SYNC WITH DATA
IS **REALLY HARD** AND A **LOT OF WORK**



Front-end frameworks **solve this problem** and take hard work away from developers 🎉

Different approaches, same goal

2

They enforce a “**correct**” way of structuring and writing code (therefore contributing to solving the problem of “spaghetti code” 🍝)

3

They give developers and teams a **consistent** way of building front-end applications