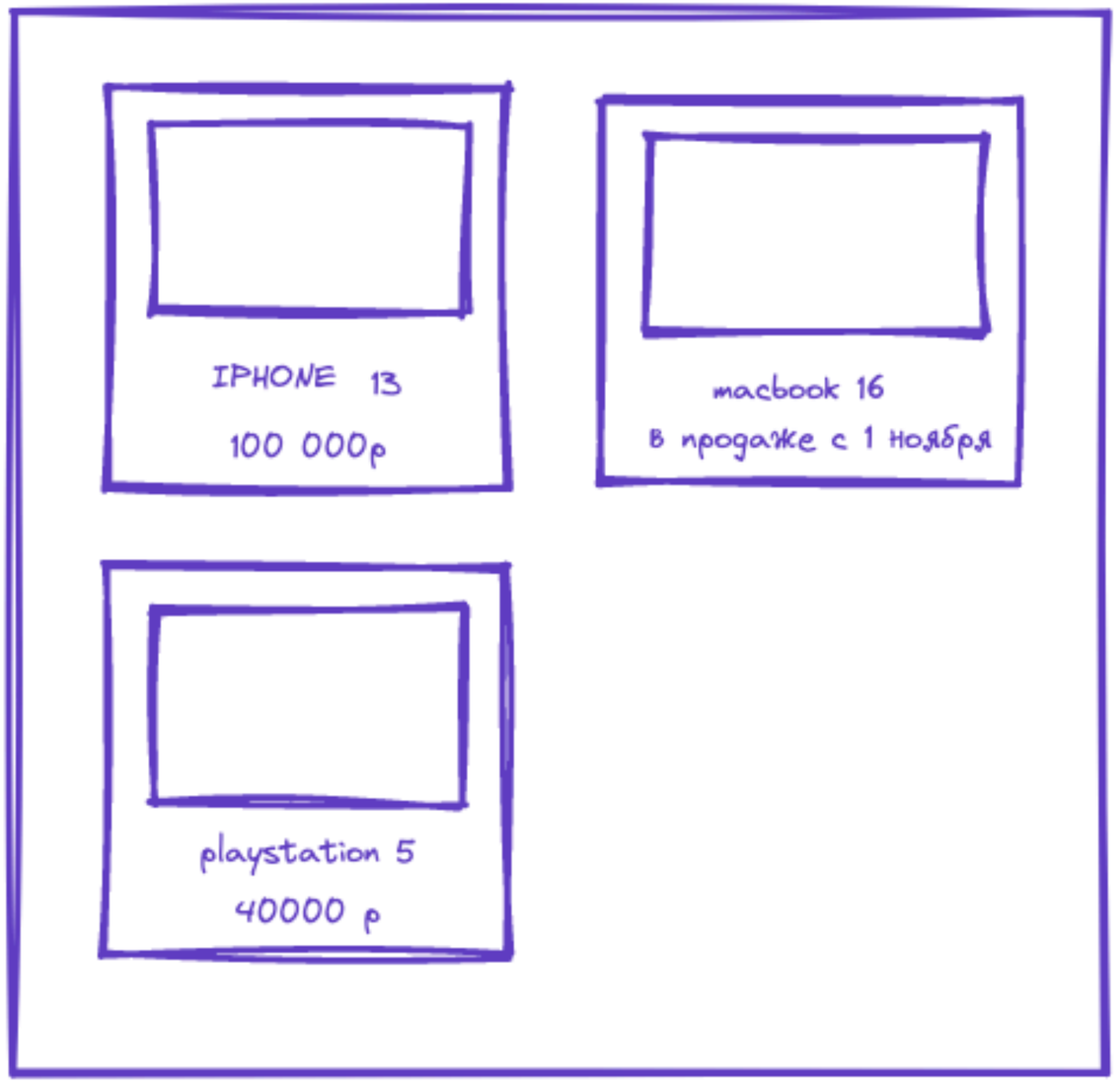
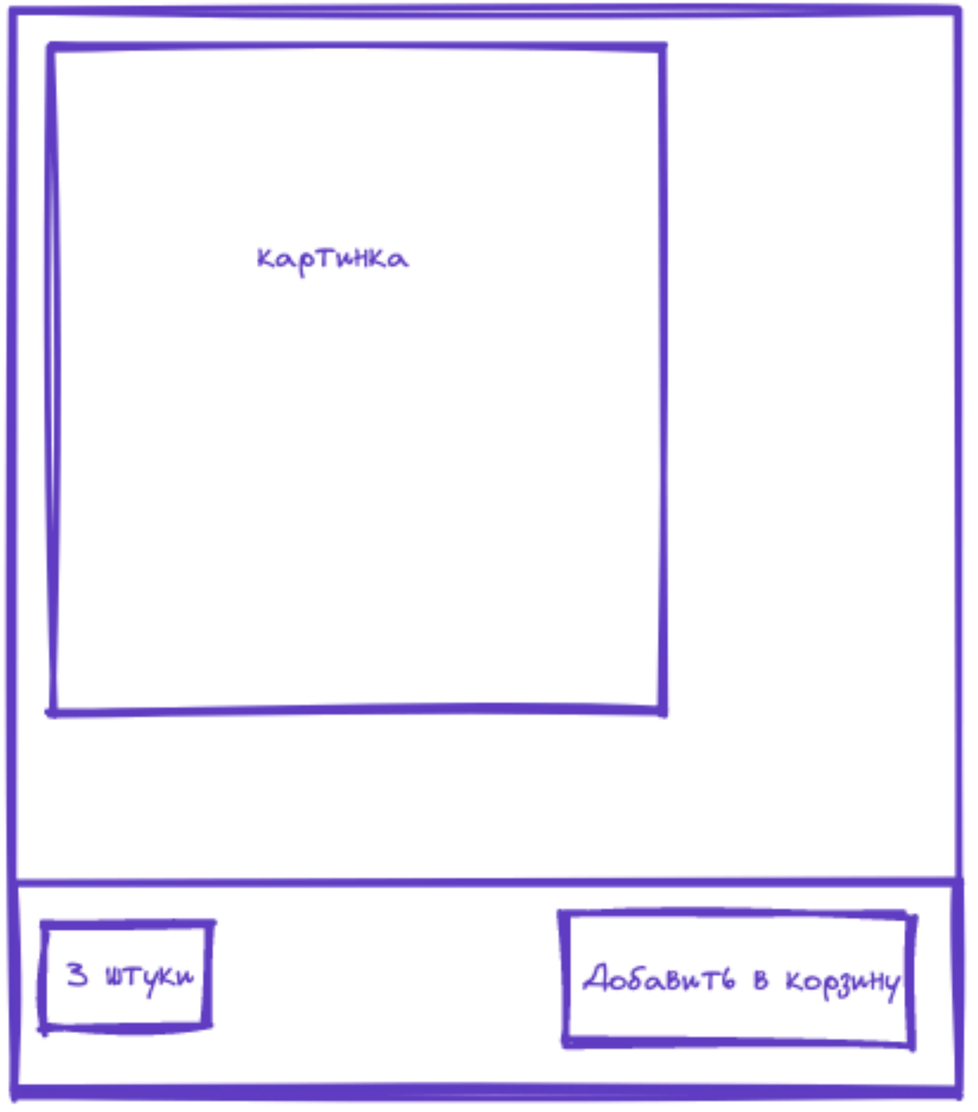
# UI библиотеки

## Зачем они нужны

# IMPERATIVE
# VS
# DECLARATIVE


IPHONE 13
100 000р
macbook 16
в продаже с 1 ноября
playstation 5
40000 р

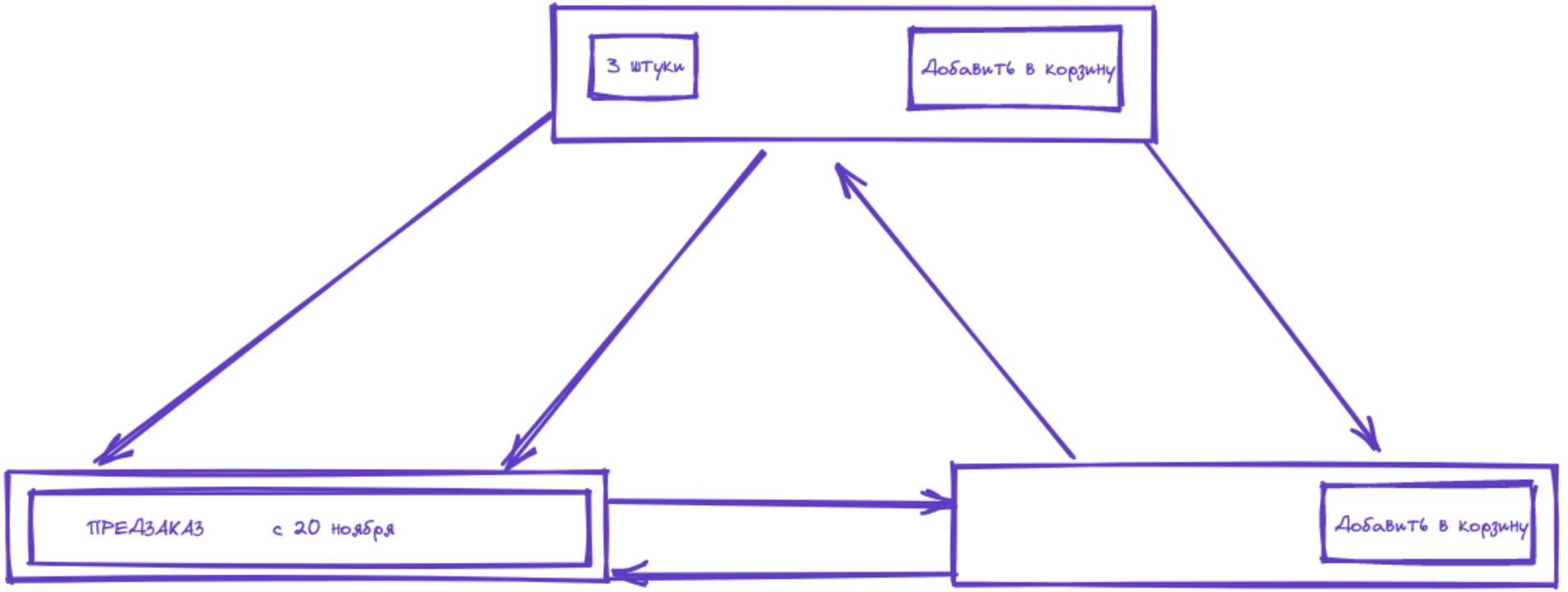

Картинка
3 штуки
Добавить в корзину

1 состояние

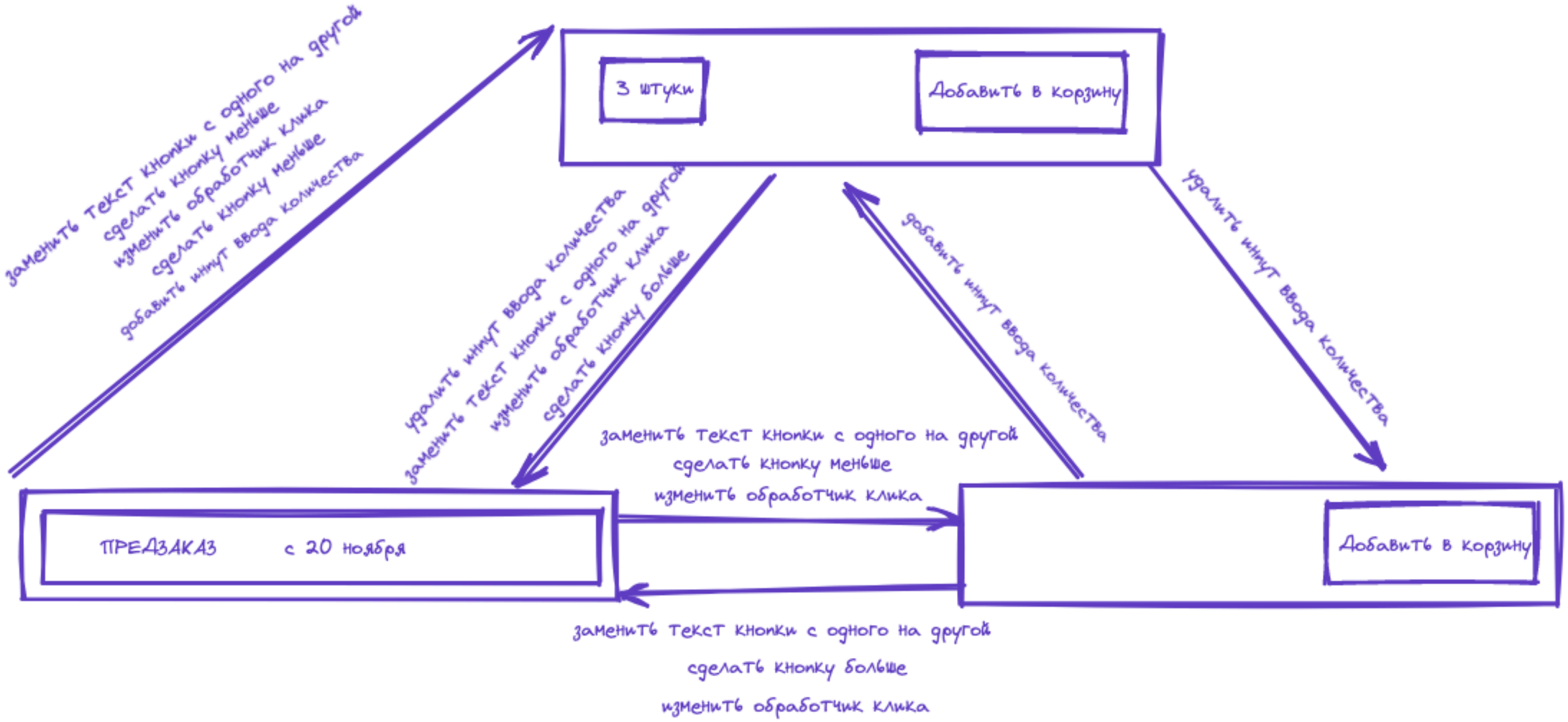

2 состояние


3 состояние

КАКИЕ БЫВАЮТ ПЕРЕХОДЫ МЕЖДУ СОСТОЯНИЯМИ?



# ОПИСАНИЕ ВСЕХ ПЕРЕХОДОВ



# N * N
# переходов

# Напишем код

сделать кнопку больше

```
button.classList.remove("small");
button.classList.add("large");
```

сделать кнопку меньше

```
button.classList.remove("large");
button.classList.add("small");
```

| | |
|---|---|
| добавить элемент | document.createElement() |
| удалить элемент | element.remove() |
| добавить обработчик события | element.addEventListener |
| удалить обработчик события | element.removeEventListener |


НЕ НАПИШЕМ КОД

# КОДА ОЧЕНЬ МНОГО

**ДАЖЕ ДЛЯ ПРИМЕРА ИЗ 3х состояний**

# Сколько состояний обычно в приложении

Товар Беспроводные наушники Xiaomi Redmi AirDots 3 добавлен к сравнению

Всего в списке 1 товар из категории Наушники и Bluetooth-гарнитуры

Сравнить

- JBL
- Panasonic
- Pioneer
- Samsung
- Sennheiser
- Skullcandy
- Sony
- Не определен

Показать всё

**В продаже**

**Со склада Яндекса**

**Конструкция**

- вкладыши
- внутриканальные
- накладные
- полноразмерные
- с костной проводимостью

Показать ещё 1

**Тип устройства**

- беспроводные TWS-наушники
- беспроводные наушники
- моно Bluetooth-гарнитура
- проводные наушники

**Система активного шумоподавления (ANC)**

- Да

express

## Беспроводные наушники Xiaomi Mi True Wireless Earbuds Basic 2, черный

4.5 2403 отзыва

конструкция: внутриканальные (закрытые)
подключение: Bluetooth 5.0
тип зарядки кейса: micro USB
тип излучателей: динамические

**1 390 ₽**

Доставка сегодня с 10:00

В корзину

express

## Беспроводные наушники Xiaomi Redmi AirDots 3, white

и ещё 2 варианта

4.4 64 отзыва

конструкция: внутриканальные (закрытые)
подключение: Bluetooth 5.2
тип зарядки кейса: USB Type-C
степень защиты: IPX4

**2 745 ₽**

Доставка сегодня с 11:30

В корзину

3 предложения от 2 745 ₽

Выбор покупателей

## Беспроводные наушники Apple AirPods 2 с беспроводным зарядным футляром MRXJ2, белый

4.7 539 отзывов

конструкция: вкладыши (открытые)
подключение: Bluetooth 5.0
тип зарядки кейса: беспроводная зарядка
количество микрофонов: 2

**12 980 ₽**

Доставка сегодня с 12:00

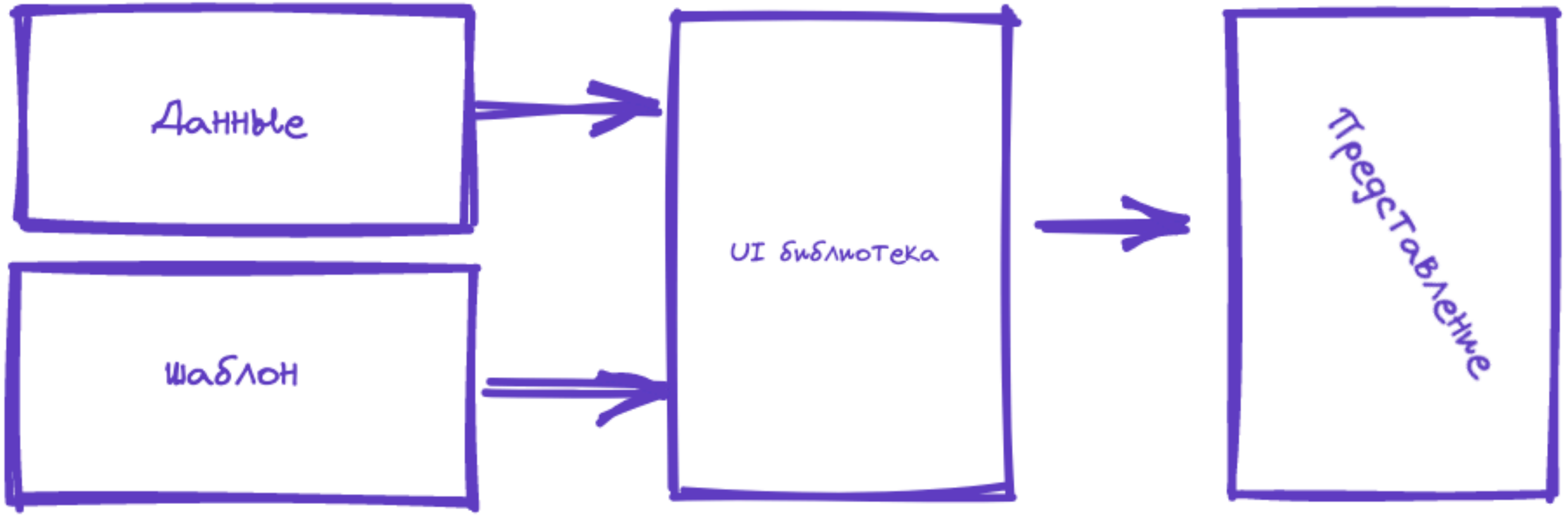
В корзину

# На помощь приходят шаблоны

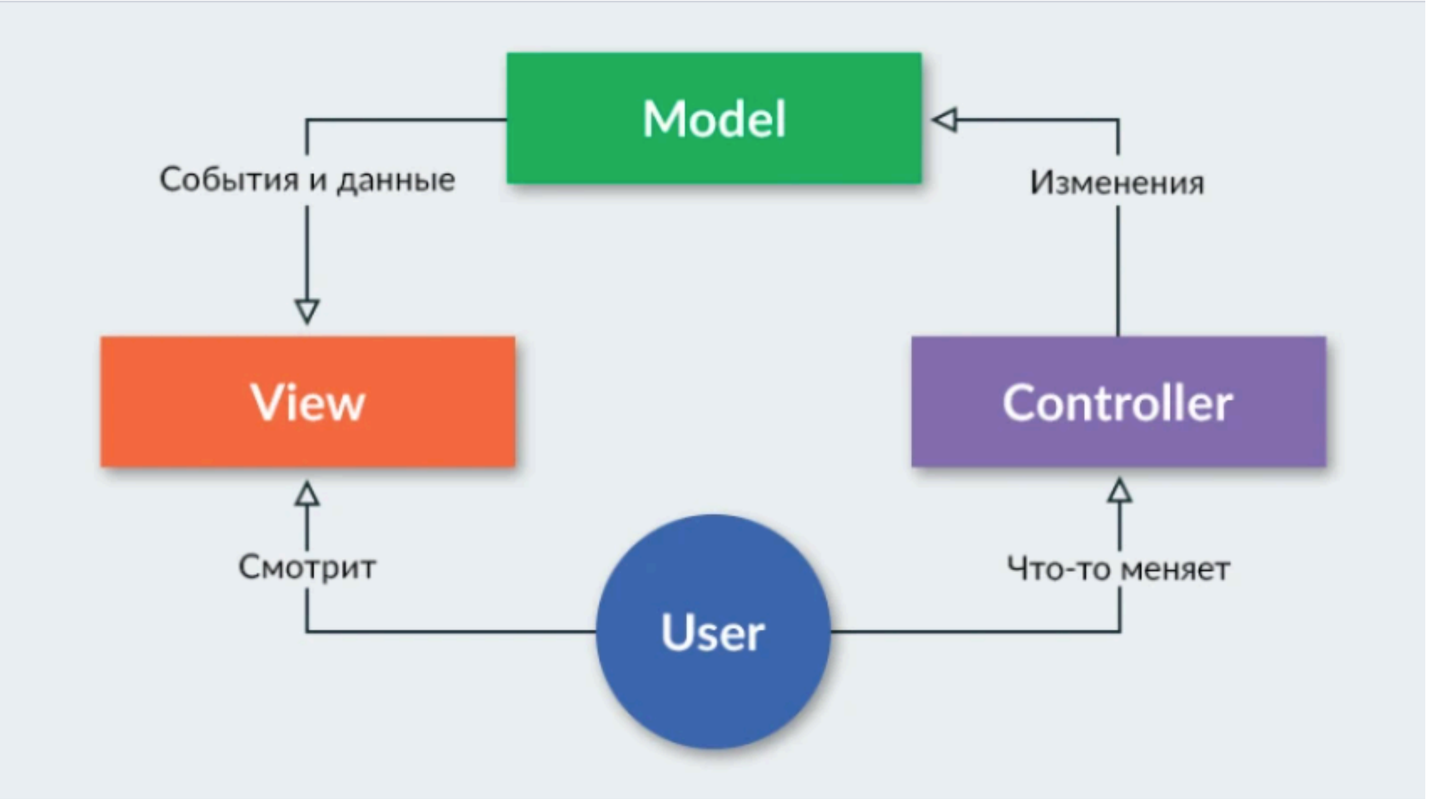
```
<div>
    <!-- показать если состояние 1 <input type="number" value="1"/> -->

    <!-- показать если состояние 1 или 2 <button class="big">Заказать</button> -->

    <!-- показать если состояние 3 <button class="big">Предзаказ с 20 ноября</button> -->
</div>
```

```
<div>
    <!-- показать если состояние 1 <input type="number" value="1"/> -->

    <!--
        <button
            class="состояние 1 или 2 ? small : big">
                состояние 1 или 2 ? Заказать: предзаказ с 20 ноября
        </button>
    -->
</div>
```

```
<div>
    <!-- показать если состояние 1 <input type="number" value="1"/> -->

    <!--
        <button
            class="состояние 3 ? big : small">
                состояние 3 ? предзаказ с 20 ноября: Заказать
        </button>
    -->
</div>
```


Данные
Шаблон
UI библиотека
Представление

# MVC


Model
События и данные
Изменения
View
Controller
Смотрит
Что-то меняет
User

# UI библиотеки

- Осуществляют парсинг шаблона + его исполнение
- Поддерживают компонентный подход
- Обновляют представление при изменении данных

# Примеры шаблонов

https://knockoutjs.com/

**Run it:**

Choose a ticket class: Choose... Clear

```
Choose a ticket class:
<select data-bind="options: tickets,
                   optionsCaption: 'Choose...',
                   optionsText: 'name',
                   value: chosenTicket"></select>

<button data-bind="enable: chosenTicket,
                   click: resetTicket">Clear</button>

<p data-bind="with: chosenTicket">
    You have chosen <b data-bind="text: name"></b>
    ($<span data-bind="text: price"></span>)
</p>

<script>
    function TicketsViewModel() {
        this.tickets = [
            { name: "Economy", price: 199.95 },
            { name: "Business", price: 449.22 },
            { name: "First Class", price: 1199.99 }
        ];
        this.chosenTicket = ko.observable();
        this.resetTicket = function() { this.chosenTicket(null) }
    }
    ko.applyBindings(new TicketsViewModel());
</script>
```

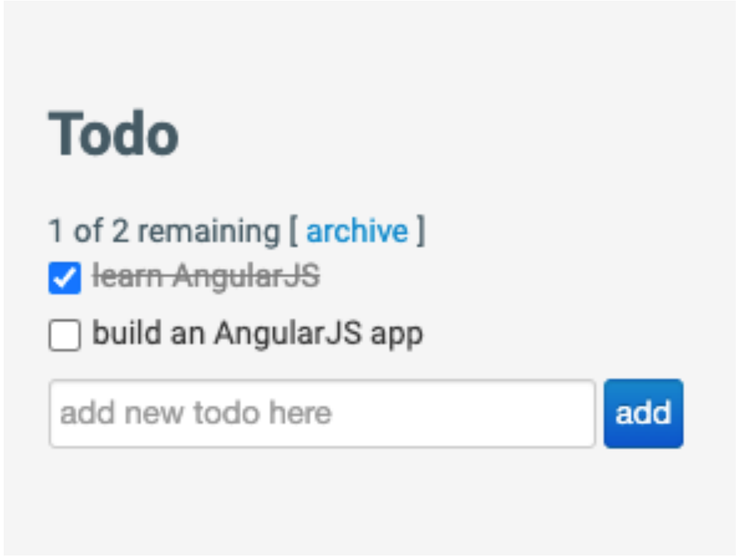
*Binding attributes* declaratively link DOM elements with model properties

Your *view model* holds the UI's underlying data and behaviors

Activates Knockout

# Примеры шаблонов

## https://angularjs.org/




```
<!doctype html>
<html ng-app="todoApp">
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
    <script src="todo.js"></script>
    <link rel="stylesheet" href="todo.css">
  </head>
  <body>
    <h2>Todo</h2>
    <div ng-controller="TodoListController as todoList">
      <span>{{todoList.remaining()}} of {{todoList.todos.length}} remaining</span>
      [ <a href="" ng-click="todoList.archive()">archive</a> ]
      <ul class="unstyled">
        <li ng-repeat="todo in todoList.todos">
          <label class="checkbox">
            <input type="checkbox" ng-model="todo.done">
            <span class="done-{{todo.done}}">{{todo.text}}</span>
          </label>
        </li>
      </ul>
      <form ng-submit="todoList.addTodo()">
        <input type="text" ng-model="todoList.todoText"  size="30"
               placeholder="add new todo here">
        <input class="btn-primary" type="submit" value="add">
      </form>
    </div>
  </body>
</html>
```

# Примеры шаблонов

## https://angular.io/

```
import { Component } from '@angular/core';

@Component({
  selector: 'hello-world',
  template: `
    <h2>Hello World</h2>
    <p>This is my first component!</p>
  `
})
export class HelloWorldComponent {
  // The code in this class drives the component's behavior.
}
```

# Примеры шаблонов

## https://ru.reactjs.org/

Привет, Саша

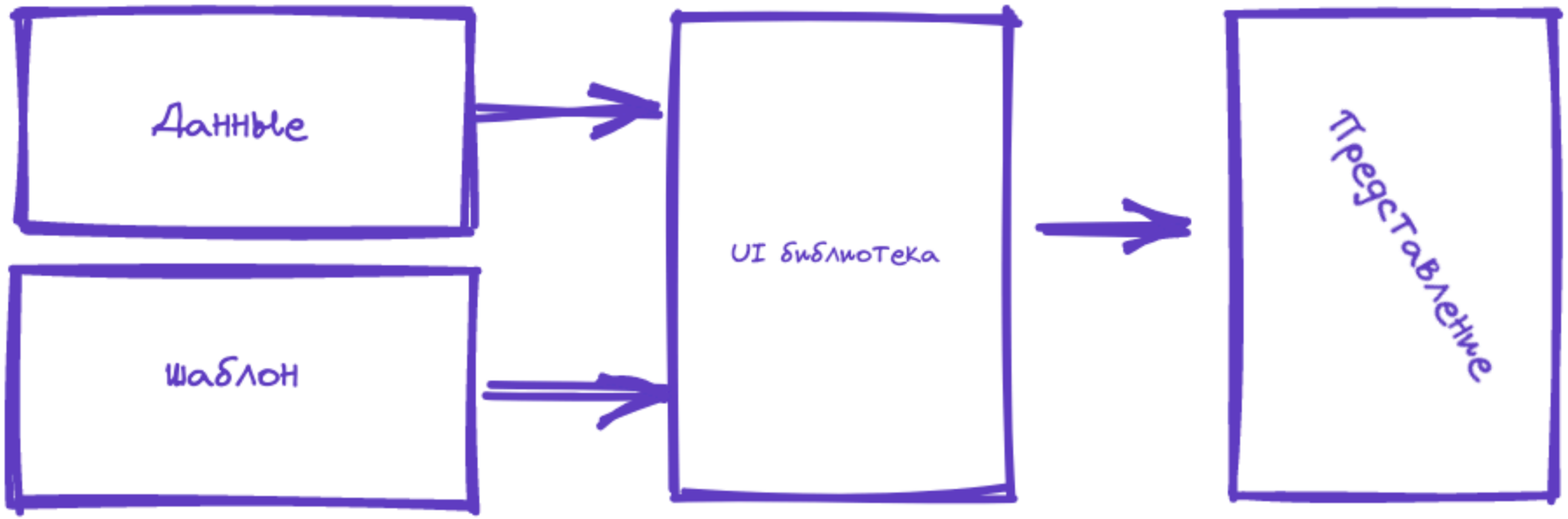
```
class HelloMessage extends React.Component {
  render() {
    return (
      <div>
        Привет, {this.props.name}
      </div>
    );
  }
}

ReactDOM.render(
  <HelloMessage name="Саша" />,
  document.getElementById('hello-example')
);
```

# Компонентный подход

https://alfa-laboratory.github.io/core-components/master/?path=/docs/%D0%BA%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D1%8B-button--button

- Меньше кода
- Инкапсуляция в одном месте => меньше правок
- Обеспечивает стилистическую консистентность
- Проще тестировать
- Требует проработки интерфейса взаимодействия компонента (Посмотреть на количество параметров)

# Автоматическое обновление представления при изменении данных



# Как библиотека узнает что данные меняются

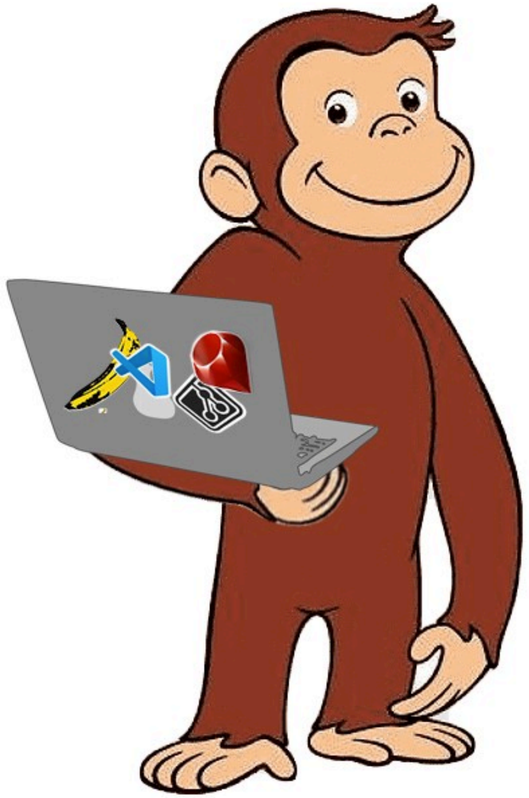
- Каждые 1000ms мы будем перерендеривать все
- Мы сами скажем что они изменились
- Мы скажем что они возможно изменились
- Библиотека сама как-то об этом узнает

# САМИ СКАЖЕМ ЧТО ДАННЫЕ ИЗМЕНИЛИСЬ

UIlibrary.dataForTemplateChanged(‘nameOfTemplate’, newData)

UIlibrary.dataForTemplateChanged(‘nameOfTemplate’, changeSet)

# MONKEY PATCHING



- XMLHTTPRequest
- setTimeout
- setTimeInterval
- Promise
- localstorage
- Etc

# ANGULARJS

## Super heroic framework

НАЗЫВАЕТ СЕБЯ НЕ БИБЛИОТЕКОЙ А ФРЕЙМВОРКОМ

Основной плюс - все из коробки

- $ajax
- $setTimeout
- $setTimeInterval
- $Promise
- $localstorage
- $Etc