

Нативные всплывающие окна с Popover API

Разберёмся, когда для всплывающих окон достаточно только HTML,
а когда будем жить по старинке



Егор Левченко

Разработчик добра,
Дока

Кто я?



Егор Левченко

Разработчик добра
Редакция Доки

- Помогаю новичкам в Доке и вебе с open-source
- Помогаю редакции со стримами, чатами и пространством на GitHub
- Верстаю эти ваши интерфейсы

Содержание

- 01 Всплывающие окна.
Что это и чего мы от них ждём?
- 02 Как мы жили до Popover API?
- 03 Как начать работать
без или почти без JS?
- 04 Диалог, поповер, тултип —
а есть ли разница?
- 05 Проблемы безопасности
Popover API
- 06 Как использовать на практике?
- 07 У моих клиентов старые браузеры
- 08 Что же выбрать?
- 09 Что мы узнали?

01

Всплывающие окна

Что это и чего мы от них ждём?

Всплывающие окна – часть повседневности



Найдётся всё



-3°

USD 99,93

EUR 103,29

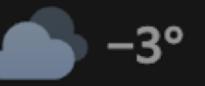
YDEX 4056,50

Орёл

Я

1

Найдётся всё



-3°

USD 99,93

EUR 103,29

YDEX 4056,50

найти сервис



Маркет



Игры



Карты



Кинопоиск



Переводчик



Авто.ру



Путешествия



Видео



Картинки



Музыка



Телепрограмма



Недвижимость



Яндекс
Афиша



Еадил



Практикум



Задания

Показать все

2

Бизнесу

Для бизнеса

Приложения

Все сервисы

Я

1

Найдётся всё

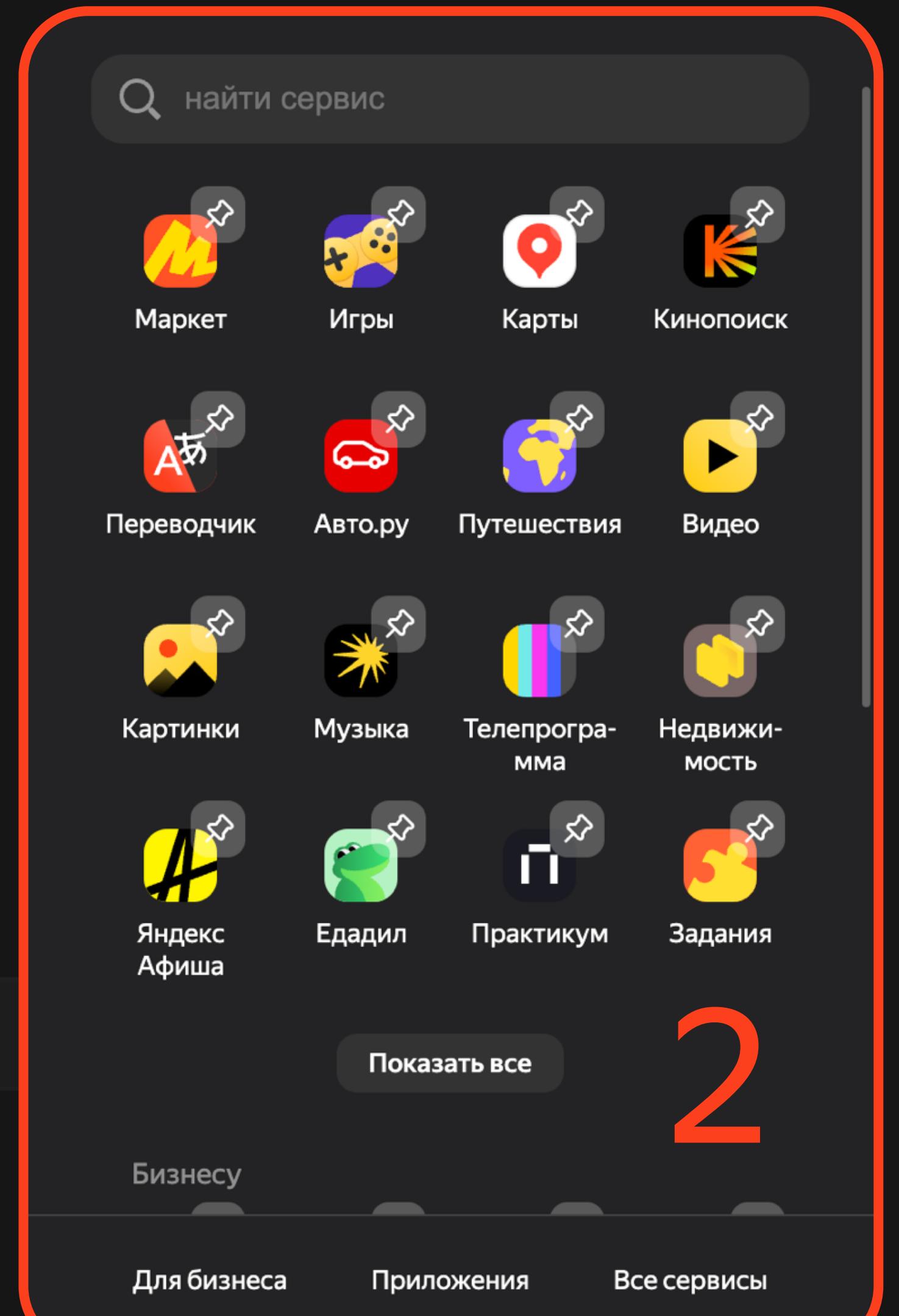


-3°

USD 99,93

EUR 103,29

YDEX 4056,50



2

Оверлеи

Попапы

Поповеры

Диалоги

Тултипы

Что должны уметь всплывающие окна

И попапы, и диалоги, и поповеры

01

Всплывать на верхнем слое

Что должны уметь всплывающие окна

И попапы, и диалоги, и поповеры

01

Всплывать на верхнем слое

02

Перехватывать фокус (не всегда)

Что должны уметь всплывающие окна

И попапы, и диалоги, и поповеры

01 Всплывать на верхнем слое

02 Перехватывать фокус (не всегда)

03 Обрабатывать события
клавиатуры

Что должны уметь всплывающие окна

И попапы, и диалоги, и поповеры

- 01 Всплывать на верхнем слое
- 02 Перехватывать фокус (не всегда)
- 03 Обрабатывать события клавиатуры
- 04 Открываться по клику на интерактивном элементе на странице

Главное отличие модального окна от немодального

Что должны уметь всплывающие окна

И попапы, и диалоги, и поповеры

- 01 Всплывать на верхнем слое
- 02 Перехватывать фокус (не всегда)
- 03 Обрабатывать события клавиатуры
- 04 Открываться по клику на интерактивном элементе на странице
- 05 Закрываться по клику вне контента*

02

Как мы жили до Ropover API?

Примеры из популярных
библиотек и фреймворков

Примеры реализации поповеров

React

Используем порталы

```
import { createPortal } from 'react-dom';

function MyComponent() {
  const [isVisible, setIsVisible] = useState(true);
  return (
    <>
      <PageComponent />
      {isVisible && createPortal(
        <MyModalComponent />,
        document.body
      )}
    </>
  );
}
```

Примеры реализации поповеров

Vue.js

Используем телепорты

```
<button @click="open = true">  
  Open Modal  
</button>  
  
<Teleport to="body">  
  <div v-if="open" class="modal">  
    <p>Hello from the modal!</p>  
    <button @click="open =  
false">Close</button>  
  </div>  
</Teleport>
```

03

Как начать работать без (или почти без) JavaScript

Обновление HTML Living Standard

В HTML Living Standard
добавлена поддержка
Popover API

Can I use

?  Settings

X Feature: HTMLElement API: popover



HTMLElement API: popover

Usage

% of all users



?

78.8% + 11.15% = 89.95%



Current aligned

Usage relative

Date relative

Filtered

All



Chrome Edge * Safari Firefox Opera IE ! * Chrome for Android

Safari on iOS * Samsung Internet

Opera Mini * Opera Mobile * UC Browser for Android

Android * Firefox for Android QQ Browser Baidu Browser KaiO Brows

Chrome	Edge	Safari	Firefox	Opera	IE !	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mini *	Opera Mobile *	UC Browser for Android	Android *	Firefox for Android	QQ Browser	Baidu Browser	KaiO Brows
4-113	12-113	3.1-16.6	2-124	10-99			3.2-16.7	4-22								
114-131	114-130	17.0-18.1	125-133	100-113	6-10		17.0-18.1	23-26		12-12.1		2.1-4.4.4				2.5
132	131	18.2	134	114	11	131	1 18.2	27	all	80	15.5	131	132	14.9	13.52	3.1
133-135		18.3-TP	135-137				1 18.3									

Notes

Test on a real browser

Feedback

See full reference on [MDN Web Docs](#).

¹ On iOS and iPadOS, popovers are not dismissed when the user taps outside of the popover area, see [bug 267688](#).

Support data for this feature provided by:



Пишем свой поповер

```
<button  
  popoverTarget="my-popover"  
  popoverTargetAction="toggle"  
>  
  Показать поповер  
</button>
```



Пишем свой popover

```
<button  
  popoverTarget="my-popover"  
  popoverTargetAction="toggle"  
>  
  Показать popover  
</button>  
  
<div id="my-popover" popover>  
  <p>  
    Я – простой, но крутой popover 😎  
  </p>  
</div>
```



Разберём атрибуты

popovertarget

Связывает управление, например кнопку, с нужным поповером по ID

```
<button  
    popovertarget="my-popover"  
    popovertargetaction="toggle"  
>  
    Показать поповер  
</button>
```

```
<div id="my-popover" popover>  
    <p>  
        Я – простой, но крутой поповер 😎  
    </p>  
</div>
```

Разберём атрибуты

popovertargetaction

Указывает, что делать с поповером:

- **toggle** — переключать
- **show** — показывать
- **hide** — скрывать

```
<button  
    popovertarget="my-popover"  
    popovertargetaction="toggle"  
>  
    Показать поповер  
</button>
```

```
<div id="my-popover" popover>  
    <p>  
        Я – простой, но крутой поповер 😎  
    </p>  
</div>
```

Разберём атрибуты

popover

Атрибут указывает, что блок ведёт себя как popover.

```
<button  
    popovertarget="my-popover"  
    popovertargetaction="toggle"  
>  
    Показать popover  
</button>
```

```
<div id="my-popover" popover>  
    <p>  
        Я – простой, но крутой popover 😎  
    </p>  
</div>
```

Модальное vs немодальное

Различия окон в разрезе атрибута popover

popover (popover="auto")

Значение атрибута по умолчанию

- Окно модальное — есть закрытие по оверлею
- На экране может быть только один поповер

Модальное vs немодальное

Различия окон в разрезе атрибута popover

popover (popover="auto")

Значение атрибута по умолчанию

- Окно модальное — есть закрытие по оверлею
- На экране может быть только один поповер

popover="manual"

Можно указать атрибут

1. Окно немодальное — закрытия по оверлею нет
2. Можно открыть несколько поповеров одновременно или вложить их друг в друга

04

Диалог, поповер, тултип — а есть ли разница?

Поговорим про стилизацию, семантику,
доступность и всё остальное

Особенности стилизации попапов и поповеров в частности

Кастомизация элемента `<dialog>`

```
html,  
body {  
    scrollbar-gutter: stable;  
}
```

Если нужна «модалка»

1. Заблокируем скролл

Кастомизация элемента `<dialog>`

Если нужна «модалка»

1. Заблокируем скролл
2. JS для этого изменения

```
function openModalAndLockScroll() {  
    dialog.showModal()  
    document.body.classList.add("scroll-lock")  
}  
  
function returnScroll() {  
    document.body.classList.remove("scroll-lock")  
}  
  
dialogOpener.addEventListener(  
    "click", openModalAndLockScroll  
)  
dialog.addEventListener(  
    "close", returnScroll  
)
```

Кастомизация элемента `<dialog>`

Если нужна «модалка»

1. Заблокируем скролл
2. JS для этого изменения
3. Закрытие по оверлею
 - в HTML

```
<dialog class="dialog">
  <div class="dialog__wrapper">
    Содержимое диалога
  </div>
</dialog>
```

Кастомизация элемента `<dialog>`

Если нужна «модалка»

1. Заблокируем скролл
2. JS для этого изменения
3. Закрытие по оверлею
 - в HTML
 - в CSS

```
.dialog {  
    border: none;  
    padding: 0;  
}  
.dialog__wrapper {  
    padding: 1em;  
}  
dialog::backdrop {  
    background-color: rgb(0 0 0 / 70%);  
}  
dialog + .backdrop {  
    background-color: rgb(0 0 0 / 70%);  
}
```

Кастомизация элемента `<dialog>`

Если нужна «модалка»

1. Заблокируем скролл
2. JS для этого изменения
3. Закрытие по оверлею
 - в HTML
 - в CSS
 - в JS

```
function closeOnBackDropClick(  
  { currentTarget, target }  
) {  
  const dialogElement = currentTarget  
  const isClickedOnBackDrop = target ===  
    dialogElement  
  if (isClickedOnBackDrop) {  
    dialogElement.close()  
  }  
}  
  
dialogElement.addEventListener(  
  "click", closeOnBackDropClick  
)
```

Стилизация настоящего поповера

На примере по умолчанию

```
[popover] {  
    position: fixed;  
    width: fit-content;  
    height: fit-content;  
    inset: 0px;  
    margin: auto;  
    border: solid;  
    padding: 0.25em;  
}
```

Стилизация настоящего поповера

На примере по умолчанию

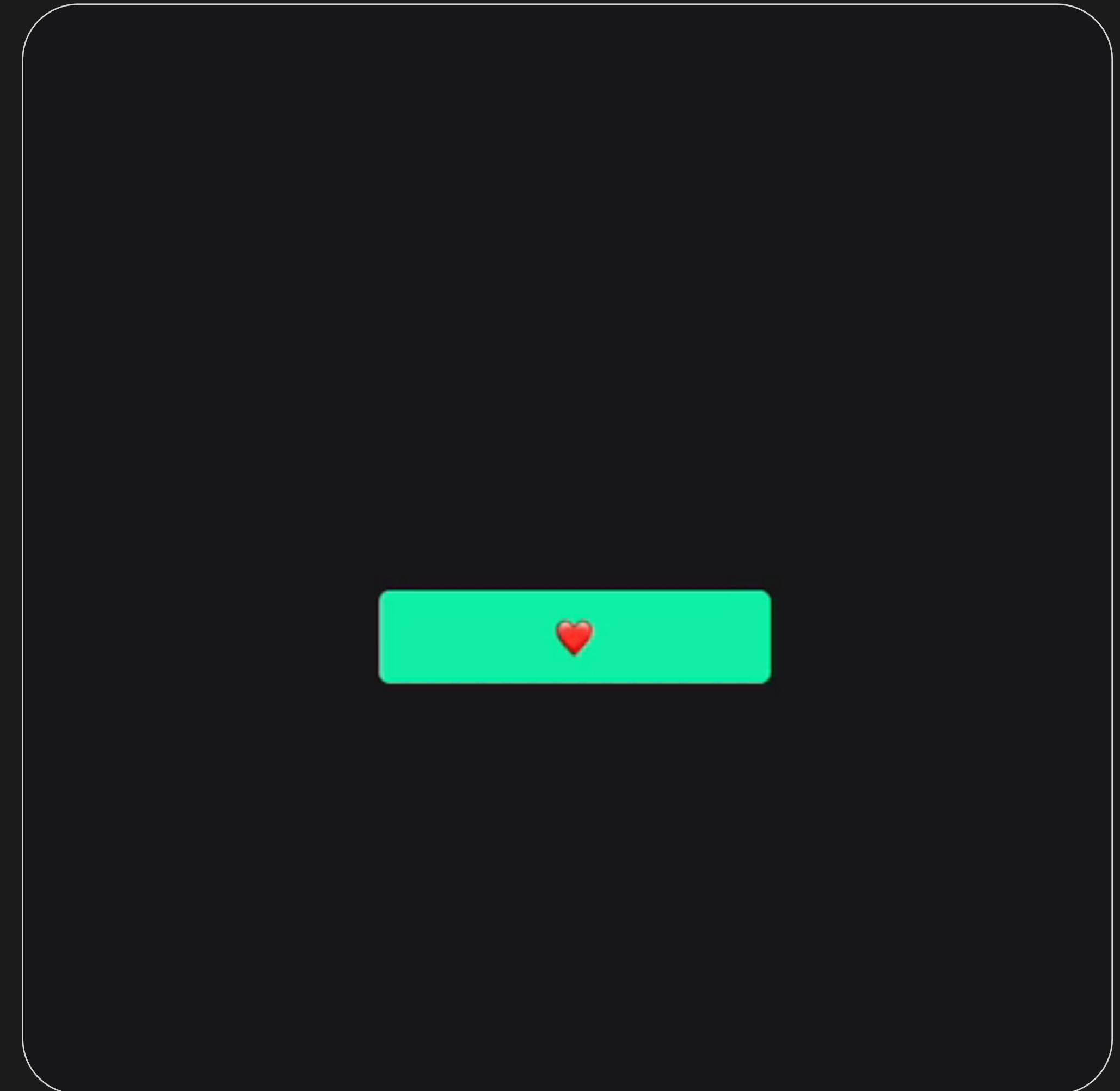
Обратите внимание
на стилизацию подложки

```
[popover] {  
    position: fixed;  
    width: fit-content;  
    height: fit-content;  
    inset: 0px;  
    margin: auto;  
    border: solid;  
    padding: 0.25em;  
}
```

```
[popover]::backdrop {  
    background-color: rgb(255 255 255 / 0.5);  
}
```

Стилизация тултипа достаточно проста

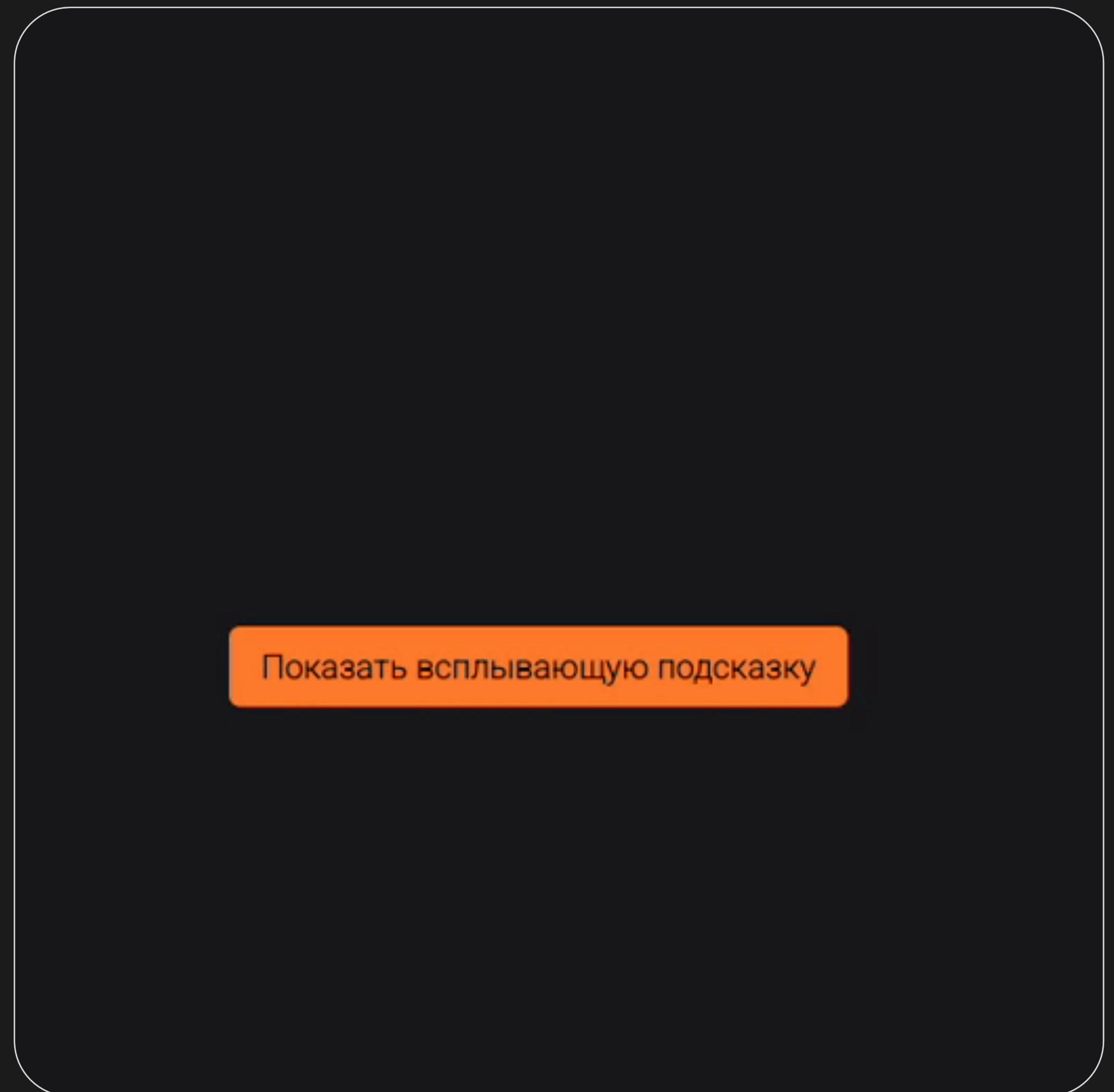
```
.tooltip {  
  position: absolute;  
  ...  
  transform: translateX(-50%);  
  visibility: hidden;  
  opacity: 0;  
  transition: opacity 0.2s, visibility  
  0.2s;  
}
```



Особенности анимации поповеров

Анимируем поповер: начало анимации

```
@starting-style {  
  [popover]:popover-open {  
    opacity: 0;  
    rotate: 1turn;  
  }  
}
```



Показать всплывающую подсказку

Анимируем поповер: открытый элемент

```
[popover]:popover-open {  
    opacity: 1;  
    rotate: 0turn;  
    transition:  
        rotate .5s,  
        opacity .5s,  
        display .5s,  
        overlay .5s;  
}
```



Анимируем поповер: элемент закрывается

```
[popover]:not(:popover-open) {  
    scale: 0;  
    transition:  
        scale .3s,  
        display .3s,  
        overlay .3s;  
}
```

Показать всплывающую подсказку

05

Проблема безопасности Popover API

Но всё в руках разработчиков
интерфейсов

Поповер не ограничен
размерами окна браузера

06

Как использовать Popover API на практике

Иногда без JavaScript
не обойтись

Методы Popover API

```
document  
  .getElementById('my-popover')  
  .showPopover()
```

Меняем состояния

1. Открываем

Методы Popover API

Меняем состояния

1. Открываем
2. Закрываем

```
document
  .getElementById('my-popover')
  .showPopover()
```

```
document
  .getElementById('my-popover')
  .hidePopover()
```

Методы Popover API

Меняем состояния

1. Открываем
2. Закрываем
3. Переключаем

```
document
  .getElementById('my-popover')
  .showPopover()
```

```
document
  .getElementById('my-popover')
  .hidePopover()
```

```
document
  .getElementById('my-popover')
  .togglePopover()
```

06

У моих клиентов старые браузеры

Или я не хочу обновлять свой React
и TypeScript прямо сейчас



Плюсы

Popover API – удобный инструмент

1. Можно собрать popover на HTML
2. Нет семантической нагрузки
3. Есть ловушка фокуса
4. Отрабатывается клавиатура
5. Отличный оверлей
6. Простая анимация
7. Простая стилизация



Минусы

Но пока только для энтузиастов

1. Не очень крутая поддержка браузерами
2. Требует обновления фреймворков и библиотек

А теперь обновим React и типы в TypeScript

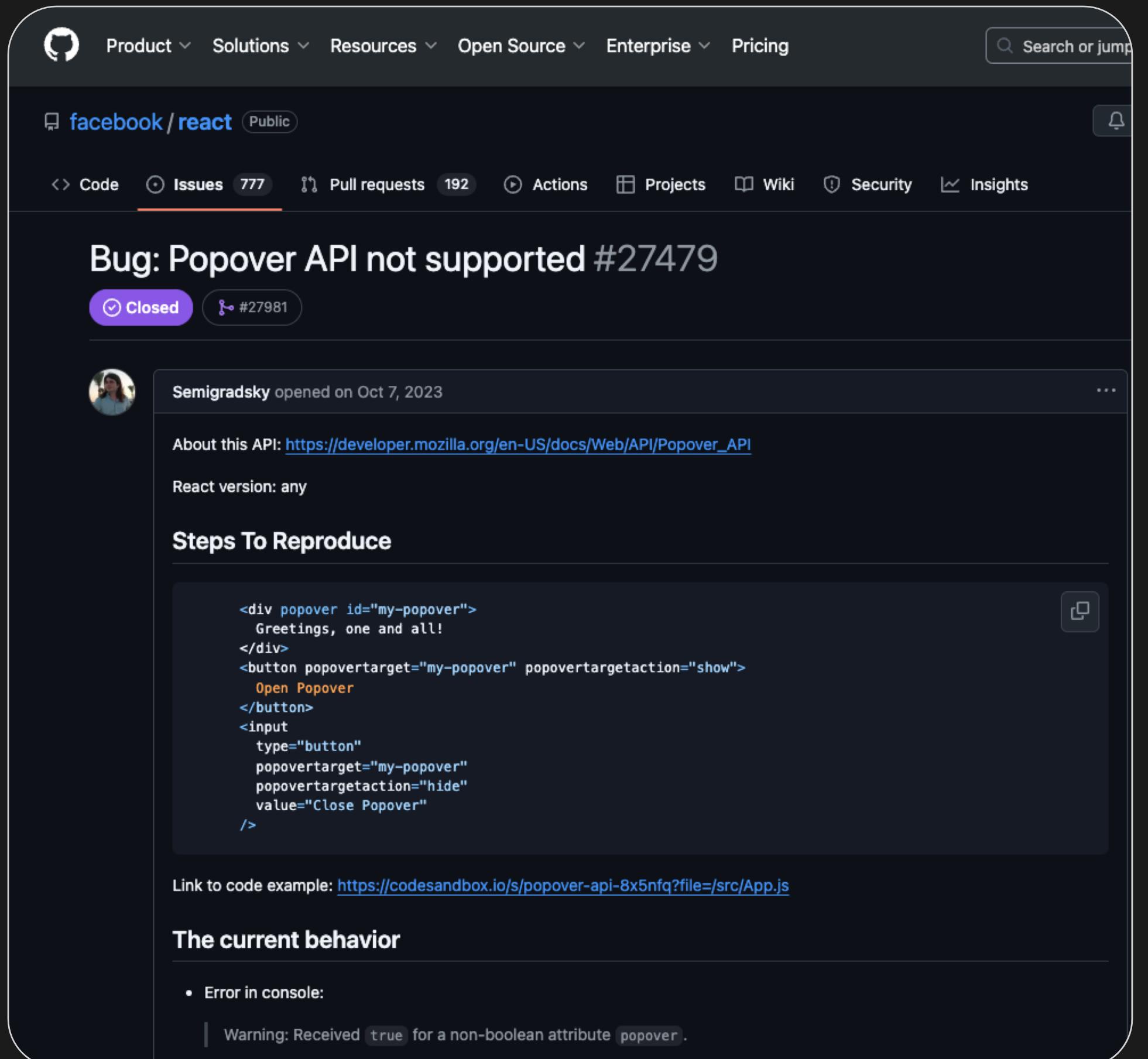
React 18 ещё не поддерживает Popover API

Переходите на свежую версию

- react@^19.0
- @types/react@^19.0.0
- @types/react-dom@^19.0.0.0

И используйте

- `popover="auto"`



App.tsx

test > popover > src > App.tsx > [App]

```
21  export const App: React.FC = () => {
44
45    const [currentPopover, setCurrentPopover] = React.useState<
46      IImage | undefined
47    >(undefined);
48
49    const showPopover = (image: IImage) => {
50      setCurrentPopover(image);
51    };
52
53    return (
54      <div className="relative">
55        {!!currentPopover && (
56          <Popover image={currentPopover} id={currentPopover.id} />
57        )}
58        <ul className="flex gap-10 overflow-scroll">
59          {images.map((item: IImage) => (
60            <li key={item.id}>
61              <button onClick={() => showPopover(item)} popoverTarget={item.id.toString()}>
62                <img src={item.url} alt="" className="max-w-4xl" />
63                <span className="sr-only">Open image #{item.id}</span>
64              </button>
65            </li>
66          ))}
67        </ul>
68      </div>
```

07

Что же мне выбрать?

Если я хочу сделать крутые попапы

Построил ДОМ,
вырасти дерево.
Что такое
Accessibility tree

Элементы Консоль Источники Сеть Производительность > ! 1 ⚙ : X

◀ div.services-more-popup__popup-footer ▶ a.home-link2.home-link2_color_black.home-link2_hover_tertiary

Стили Вычисленные Макет Прослушиватели событий Специальные возможности >

▼ Дерево доступности

Дерево доступности на всю страницу

▼ RootWebArea "Яндекс — быстрый поиск в интернете"

 ▼ dialog "Все сервисы"

 ▼ generic

 ▼ generic

 ▼ link "Приложения"

 > StaticText "Приложения"

▶ Атрибуты ARIA

▼ Вычисленные свойства

▼ Name: "Приложения"

 aria-labelledby: Не указано

 aria-label: Не указано

 Содержание: "Приложения"

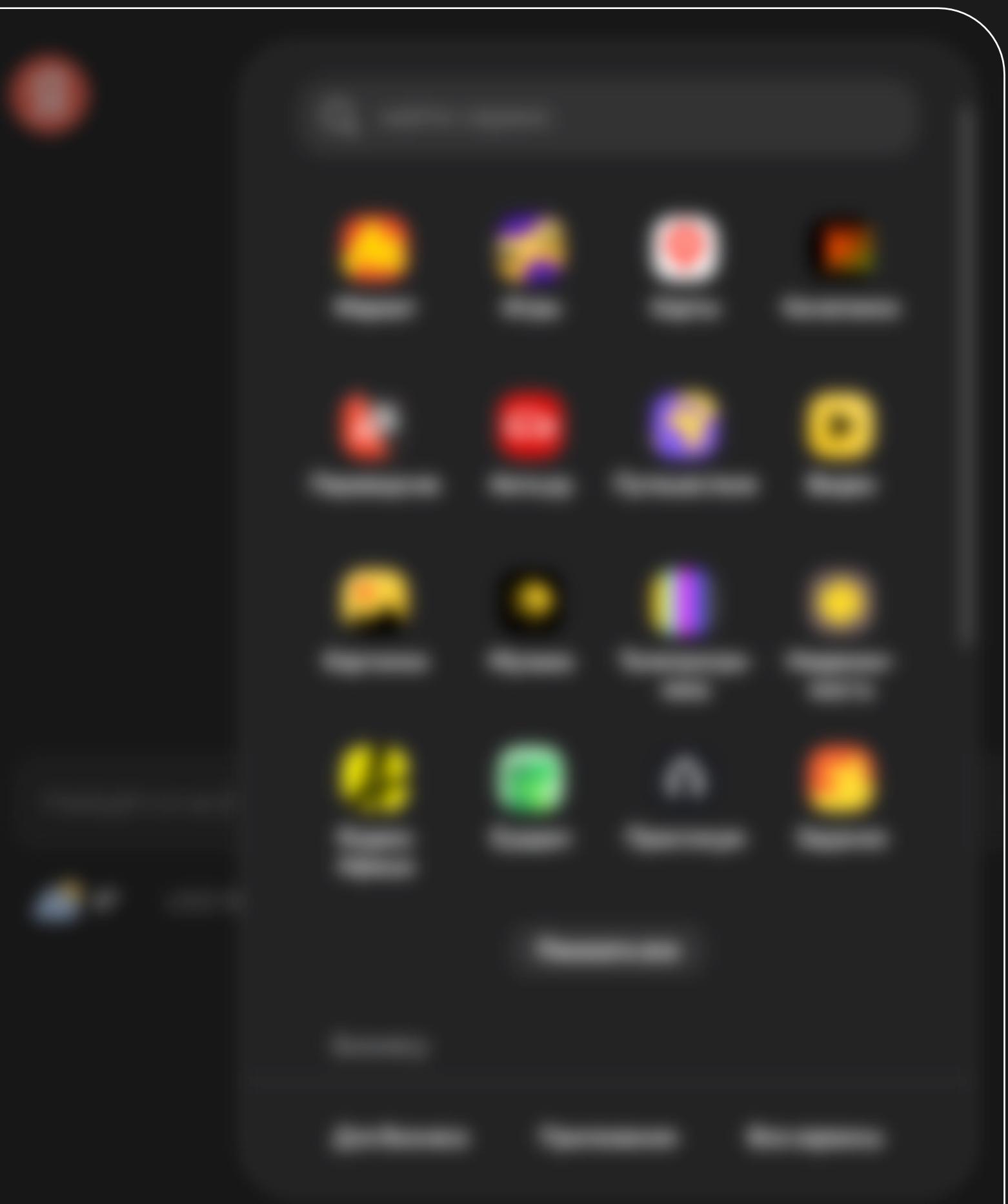
 title: Не указано

 Роль: link

 Фокусируемый: true

 url: "<https://yandex.ru/all?mode=apps>"

▶ Просмотр порядка источников



Особенности тултипа

01

Должен быть
связан с элементом
на странице при помощи
`aria-describedby`
или `aria-labelledby`

02

Имеет `role="tooltip"`,
который не обрабатывают
скринридеры, хотя он есть
в a11y tree

03

Это неинтерактивный
элемент, у него нет фокуса

04

Элемент зависит
от положения курсора
или клавиатуры



Особенности диалога



01

Имеет семантическую нагрузку

02

Имеет role="dialog", но надо добавлять имя через aria-label или aria-labelledby для улучшения доступности

03

Это интерактивный элемент с ловушкой фокуса и отработкой ESC

04

Это немодальное окно: фоновая страница доступна, по оверлею не закрывается*

Особенности поповера



01

Не имеет семантической
нагрузки и заимствует
её у родителя

02

Value tree popover
зависит от тегов, которые
вы используете в поповере

03

Это интерактивный
элемент с ловушкой
фокуса и отработкой ESC

04

Может быть как модальным,
так и немодальным окном,
но фоновый интерфейс
блокируется

07

Что же мы узнали?

Подведём итоги

Подведём итоги



HTML развивается

HTML забирает на себя
часть работы с JavaScript,
и поповеры для маленьких
проектов уже можно делать
быстрее и проще

Подведём итоги



HTML развивается

HTML забирает на себя часть работы с JavaScript, и поповеры для маленьких проектов уже можно делать быстрее и проще

Popover API

Разобрались, чем крут Popover API и чем различаются основные типы всплывающих окон

Подведём итоги



HTML развивается

HTML забирает на себя часть работы с JavaScript, и поповеры для маленьких проектов уже можно делать быстрее и проще



Popover API

Разобрались, чем крут Popover API и чем различаются основные типы всплывающих окон



Ещё не время

Несмотря на хорошую поддержку современных браузеров, popover требует поддержки со стороны фреймворков и библиотек

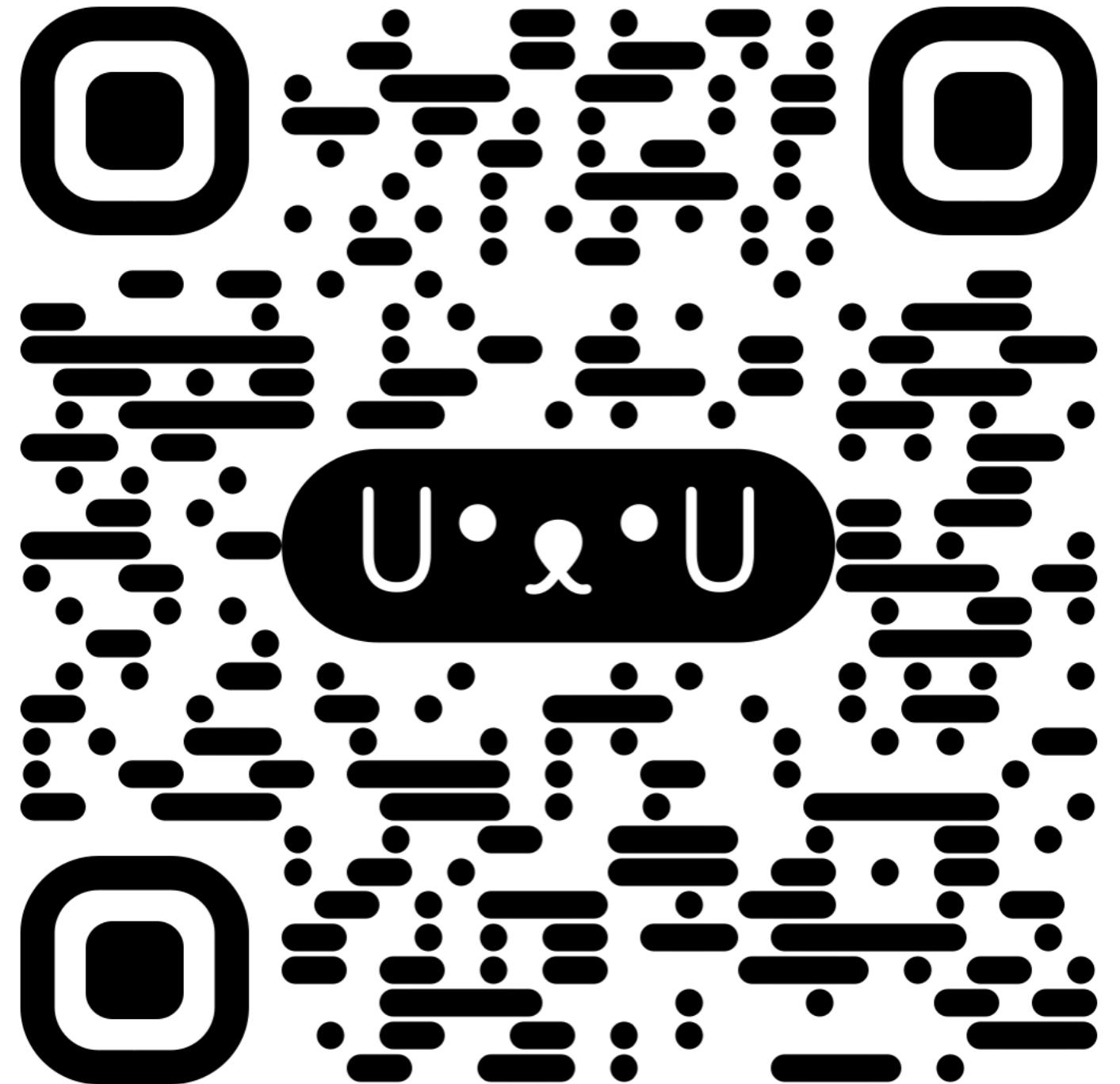


Тот, кто не смотрит
вперёд, оказывается
позади

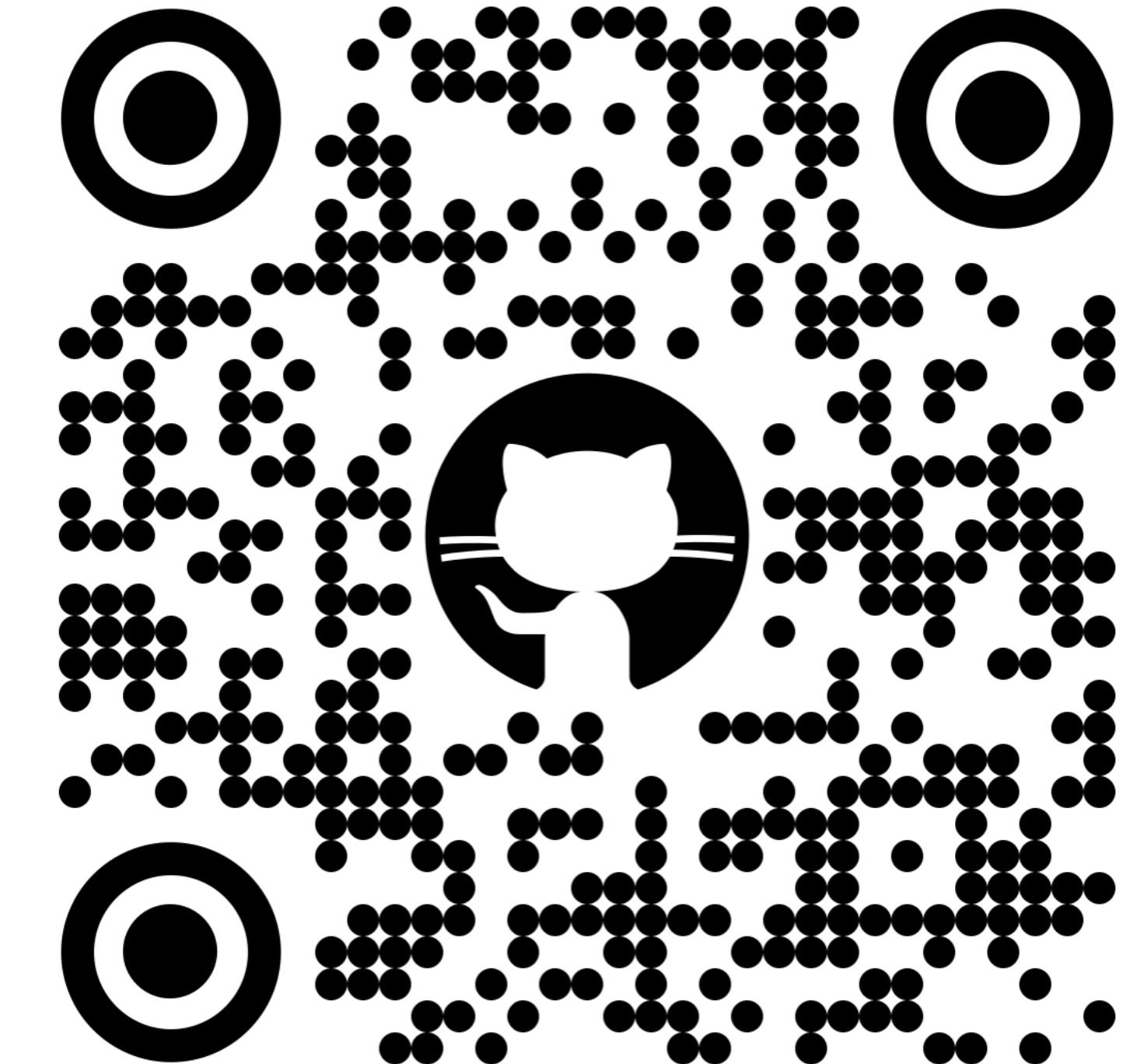
Герберт Уэллс

Мне помогла Дока, а вам?

Читайте Доку
или сделайте свой
вклад, наш проект
живёт благодаря
сообществу.



Дока



Наш GitHub

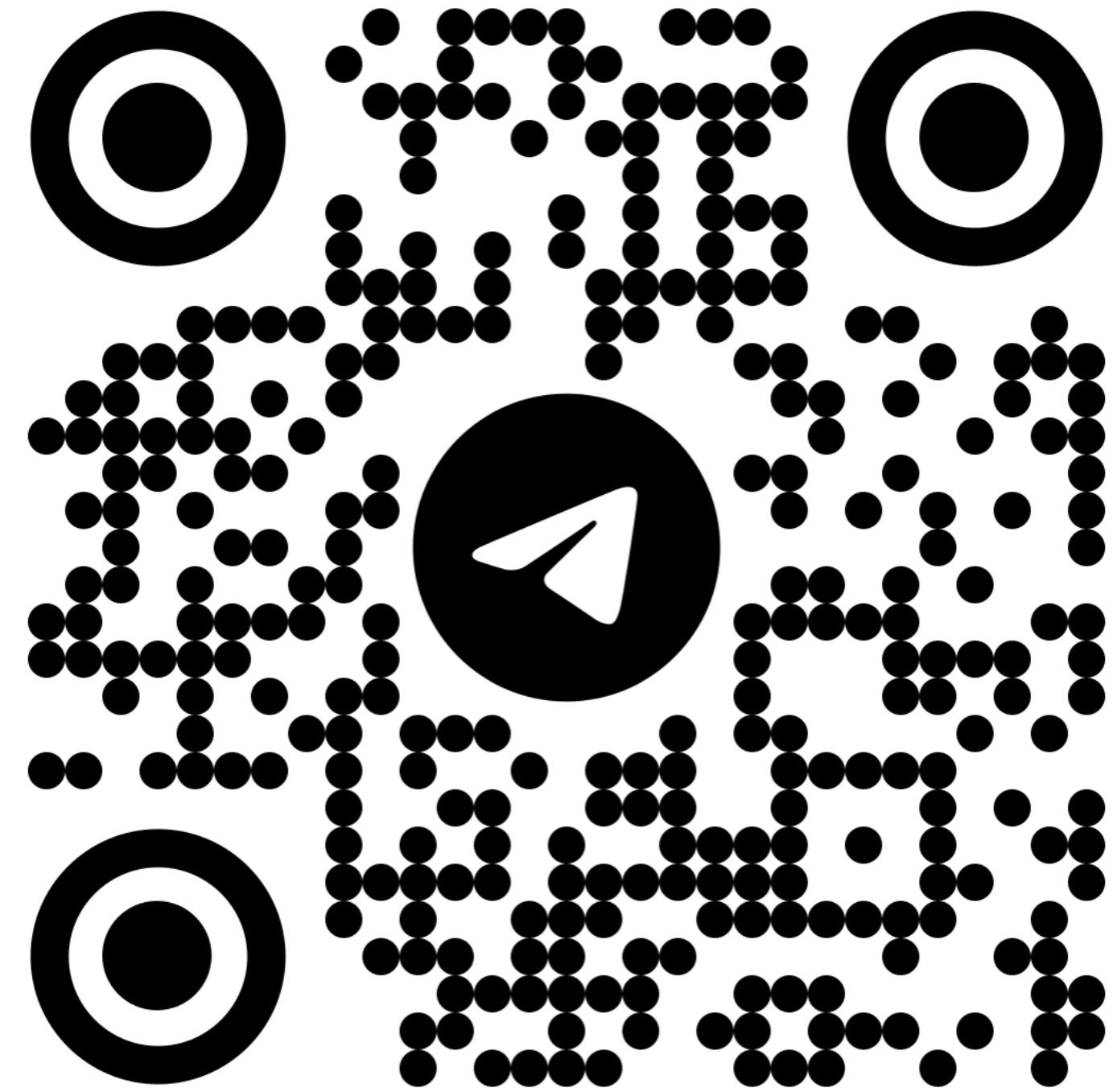
Яндекс

Спасибо!

Если есть желание связаться или почитать
длиннопосты, то вот телеграм-канал

Егор Левченко

Разработчик добра,
Дока



@frtvt