

**IT'S A
MAD,
MAD, MAD,
MAD,
SEARCH BAR
PT.1**







Dubai

29° ↓ show forecast



Hotels



Eat out



Malls



TASHEEL Moamalat



Groceries



Car service centers



Dental clinics



Beauty



Pharmacies



All categories

Interesting places in the city

Travel Guide



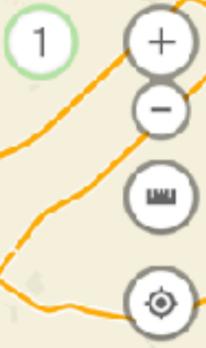
Sign in



Favorites



Download 2GIS



Базы данных

База данных и карта 2ГИС уникальны: точная информация о 3,2 миллионах организаций и 6 миллионах зданий, в более чем 340 городах 9 стран мира поможет вам в решении множества бизнес-задач.

[Скачать рубрикатор](#)

Справочные данные



Виды кухни

[+ Все атрибуты](#)



Дополнительные атрибуты
организации

[+ Все атрибуты](#)



Контакты

[+ Все атрибуты](#)



Оставить заявку

[Открыть карту →](#)

ПОМОЩНИК 2GIS

Какой у вас вопрос?

Популярные вопросы

[Как добавить компанию в 2ГИС](#)

[Как сообщить об ошибке](#)

[Как внести изменения в карточку компании](#)

[«Подарки 2ГИС»: ответы на главные вопросы о конкурсе](#)

[Как удалить компанию из справочника](#)

[Что такое «Премия 2ГИС»?](#)

[Как установить 2ГИС](#)

[Как установить 2ГИС на устройства с ранними версиями операционной системы](#)

[Зачем нужен профиль в 2ГИС](#)

[Как обновить карты городов](#)

[Все вопросы](#)

[Профиль для бизнеса](#)

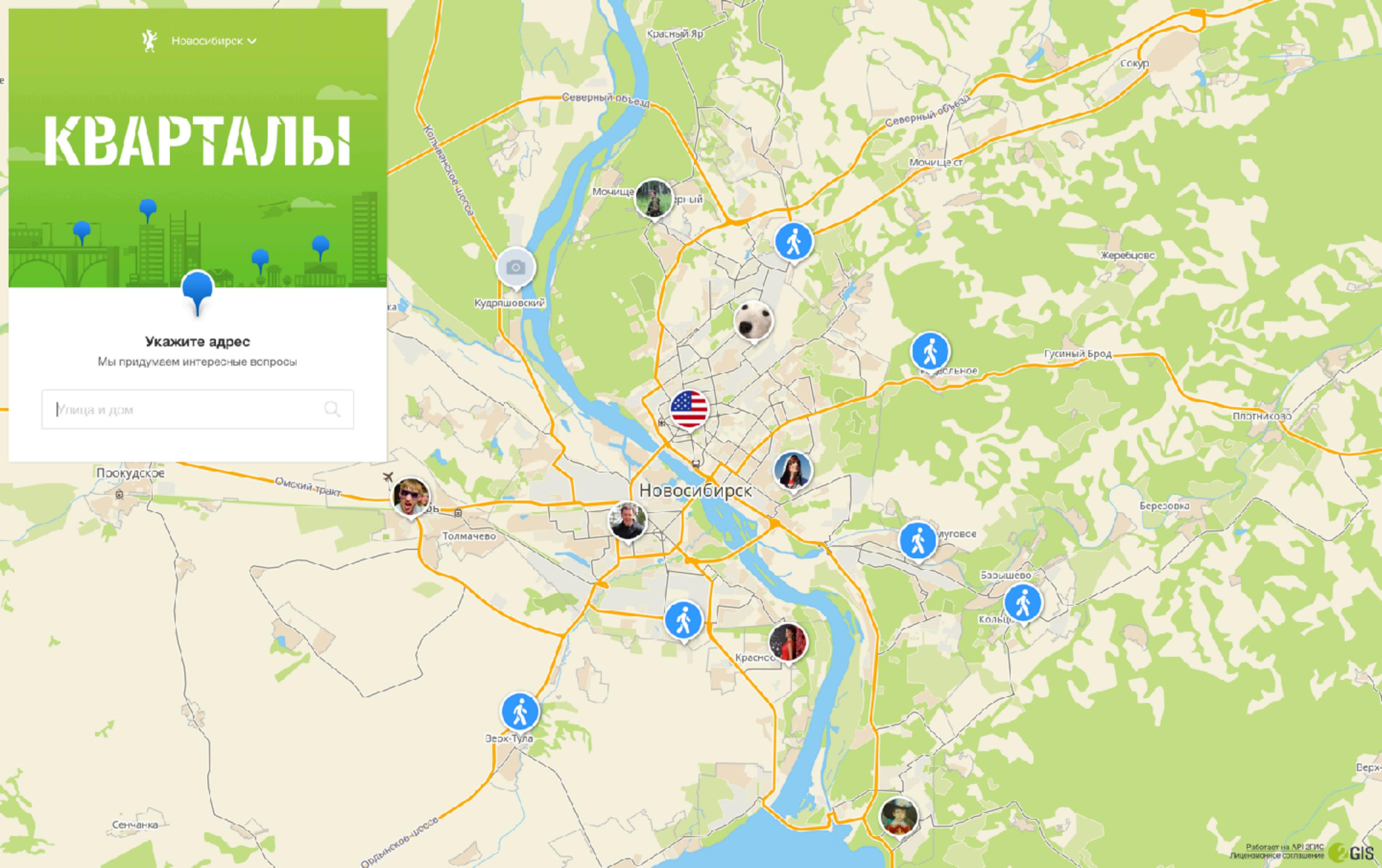
[Центр поддержки](#)

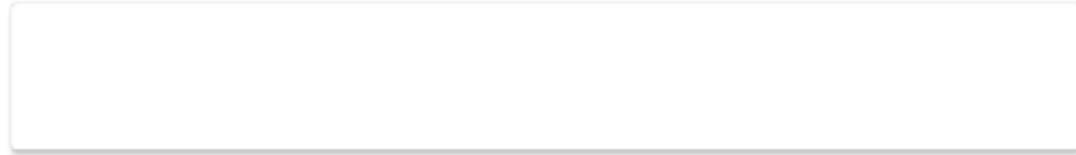
КВАРТАЛЫ

Укажите адрес

Мы придумаем интересные вопросы

Улица и дом







Содержание

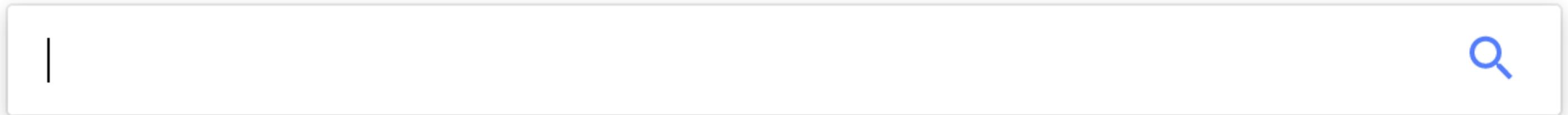
1. Как оно выглядит (звучит)
2. Ввести, очистить и найти

Как оно выглядит (звучит)



Как оно ВЫГЛЯДИТ (ЗВУЧИТ)

Иконка увеличительного стекла



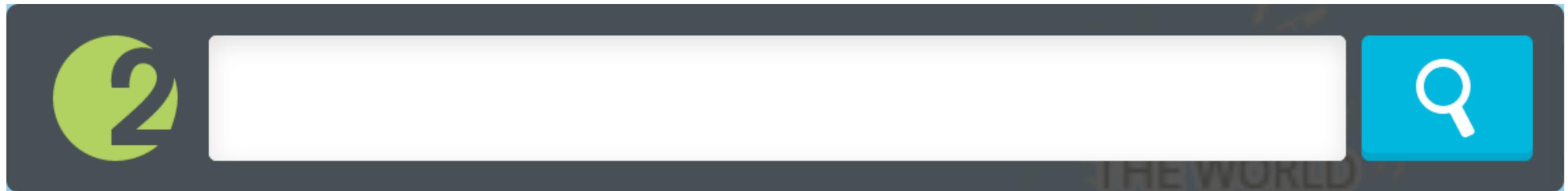
Как оно ВЫГЛЯДИТ (ЗВУЧИТ)

Иконка увеличительного стекла



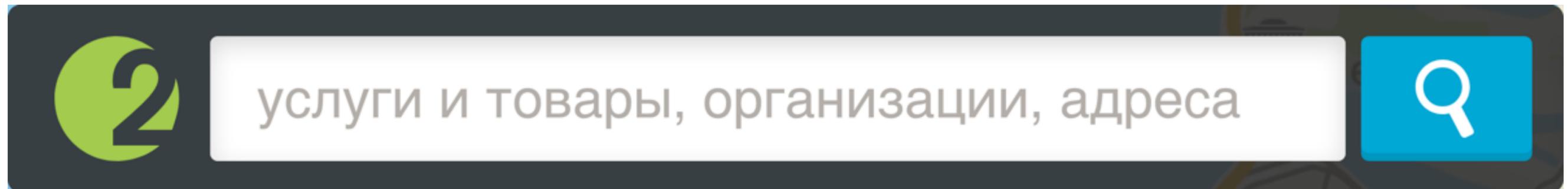
Как оно ВЫГЛЯДИТ (ЗВУЧИТ)

Иконка увеличительного стекла



Как оно ВЫГЛЯДИТ (ЗВУЧИТ)

Placeholder



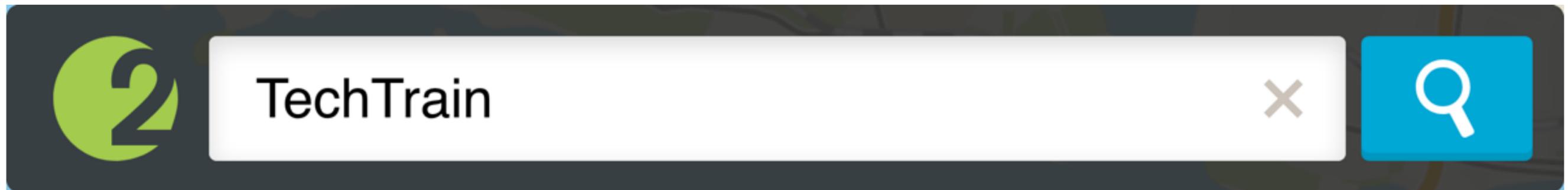
Как оно выглядит (звучит)

Placeholder

 Что вы хотите найти?

Как оно ВЫГЛЯДИТ (ЗВУЧИТ)

Крестик



Как оно выглядит (звучит)

Крестик

Как скачать карту Питера?



Как оно выглядит (звучит)

Доступность

- Что за инпут,
- Зачем он здесь,
- Что вводить,
- Что за форма,
- Какая у нее роль?

Как оно выглядит (звучит)

Что там в коде?

```
1 export Input extends React.PureComponent {  
2   render() {  
3     return (  
4       <input type="text" />  
5     );  
6   }  
7 }
```

Как оно ВЫГЛЯДИТ (ЗВУЧИТ)

Что там в коде?

```
1 export Input extends React.PureComponent {  
2   render() {  
3     return (  
4       <input type="text" value="Where is a big W" />  
5       <input type="search" value="Where is a big W" />  
6     );  
7   }  
8 }
```

Where is a big W?
Where is a big W?

Как оно ВЫГЛЯДИТ (ЗВУЧИТ)

Что там в коде?

```
1 input {  
2   height: 80px;  
3   font-size: 56px;  
4 }
```

Where is a big W?

Where is a big W?

Как оно ВЫГЛЯДИТ (ЗВУЧИТ)

Что там в коде?

```
1 input {  
2   height: 80px !important;  
3   font-size: 56px;  
4 }
```

Where is a big W?

Where is a big W?

FOX





```
<input type="text" value="Where is a big W?">  
<input type="search" value="Where is a big W?">
```

Elements Console Network Sources Performance Memory Application » 1 | ⋮ X

```
<!DOCTYPE >
<html class="gr__localhost">
  <head>...</head>
  <body data-gr-c-s-loaded="true" style> == $0
    <input type="text" value="Where is a big W?">
    <input type="search" value="Where is a big W?">
  </body>
</html>
```

Styles Computed >>

Filter :hov .cls +

```
element.style {
}

body { (index):7
  display: flex;
  justify-content: center;
  flex-flow: column wrap;
  width: 100vw;
  height: 100vh;
  align-items: center;
  box-sizing: border-box;
  margin: 0;
}

body { user agent stylesheet
  display: block;
  margin: 8px;
}
```

html.gr__localhost body input

margin -

```
▼ <input type="text" value="Where is a big W?">
  ▼ #shadow-root (user-agent)
    | <div>Where is a big W?</div>
  </input>
▼ <input type="search" value="Where is a big W?">
  ▼ #shadow-root (user-agent)
    ▼ <div id="text-field-container" pseudo="-webkit-textfield-decoration-container">
      ▼ <div id="editing-view-port">
        <div>Where is a big W?</div>
      </div>
      <div pseudo="-webkit-search-cancel-button" id="search-clear"></div>
    </div>
  </input>
```

```
▼ <input type="text" value="Where is a big W?">
  ▼ #shadow-root (user-agent)
    | <div>Where is a big W?</div>
  </input>
▼ <input type="search" value="Where is a big W?">
  ▼ #shadow-root (user-agent)
    ▼ <div id="text-field-container" pseudo="-webkit-textfield-decoration-container">
      ▼ <div id="editing-view-port">
        <div>Where is a big W?</div>
      </div>
      <div pseudo="-webkit-search-cancel-button" id="search-clear"></div>
    </div>
  </input>
```

Where is a big W?



Как оно ВЫГЛЯДИТ (ЗВУЧИТ)

Что там в коде?

```
1 input::-webkit-search-cancel-button {  
2     -webkit-appearance: none;  
3     -moz-appearance: none;  
4     appearance: none;  
5 }
```

Styles

Event Listeners

DOM Breakpoints

Properties

Accessibility

Filter

:hov .cls +

```
input[type="search" i] {  
  -webkit-appearance: searchfield;  
  box-sizing: border-box;  
}
```

user agent stylesheet

```
input {  
  -webkit-appearance: textfield;
```

user agent stylesheet

6.1	Pointer interaction
6.1.1	Styling the Cursor: the <code>'cursor'</code> property
6.1.1.1	Cursor of the canvas
6.2	Insertion caret
6.2.1	Coloring the Insertion Caret: the <code>'caret-color'</code> property
6.2.2	Shape of the insertion caret: <code>'caret-shape'</code>
6.2.3	Insertion caret shorthand: <code>'caret'</code>
6.3	Keyboard control
6.3.1	Directional Focus Navigation: the <code>'nav-up'</code> , <code>'nav-right'</code> , <code>'nav-down'</code> , <code>'nav-left'</code> properties
6.3.1.1	Example: positioned buttons
6.3.1.2	Example: moving focus to inside a frame
6.3.2	Obsolete: the <code>ime-mode</code> property

7 User Interaction

7.1	Controlling content selection
-----	-------------------------------

8 Form Control Styling

8.1	Switching appearance
-----	----------------------

Appendix A. Acknowledgments

Appendix B. Changes

Appendix C. Considerations for Security and Privacy

Appendix D. Default style sheet additions for HTML

Conformance

Document conventions

Conformance classes

Requirements for Responsible Implementation of CSS

Partial Implementations

Implementations of Unstable and Proprietary Features

Implementations of CR-level Features

Index

Terms defined by this specification

Terms defined by reference

References

§ 8. Form Control Styling

§ 8.1. Switching appearance

ISSUE 9 This is not ready for shipping (see [Github issue 1250](#)).

This previously existed as a prefixed form in most browsers. Before standardization, in addition to none, the possible values were a very long list of all the ways an element could look; this list was different across browsers.

We concluded this was impossible to standardize, in part because many apply to pseudo-elements that other browsers don't have, as they construct form elements differently.

Instead, we'd just have an "auto" value that makes form controls look like whatever they need, and a "none" value that suppresses "native" look.

However, there is evidence that this may not be web compatible, due in part to uses such as:

```
input { appearance: none; }
input[type=checkbox] { appearance: checkbox; }
```

Such code is sometimes limited to the vendor-prefixed variant of this property, but not always, as authors often include the prefixless version for the sake of future proofing

Due to substantially different behavior, naïve aliasing of `'appearance'` and of `'-webkit-appearance'` is unlikely to work.

Possible ways forward include:

1. Keep the design as is, if the compatibility problem is not that serious
2. Keep the current design, and add a handful of values other than `'none'` and `'auto'`. The design of this property does not preclude having other values. As long as we have the `'auto'` value, there is no need to be exhaustive.
3. Keep the current design, but also accept other values (possibly white-listed, possibly accept anything) and let them behave the same as `'auto'`. As demonstrated in the example above, common uses of values other than `'none'` and `'auto'` are not to switch an element's appearance into something else, nor to give a native look to something that didn't have one, but merely to undo a previously applied `'none'`. Since `'auto'` did not exist in the non standard version, authors have been using specific values for this purpose, but the behavior of `'auto'` would be sufficient.
4. Combine the previous two options, adding a handful of explicit values with specific behaviors, and also accept any other value letting them behave the same as `'auto'`.
5. Abandon the current design, and specify instead the behavior of `'-webkit-appearance'`, without an `'auto'` value and with all the permutations instead. This probably requires specifying a few pseudo-elements as



Это просто div

Have a nice Day...



the finest coffee blends

see before

Как оно ВЫГЛЯДИТ (ЗВУЧИТ)

Что там в коде?

```
1 input[type="search"] {  
2     -webkit-appearance: textfield;  
3     -moz-appearance: textfield;  
4     appearance: textfield;  
5 }
```

Where is a big W?

Where is a big W?

Where is a big W?	X
Where is a big W?	

Как оно выглядит (звучит)

Что там в коде?

```
1  input::-ms-clear {  
2      display: none;  
3  }
```



192.168.0.11



e.g. streets, cities, organizati

Search

Clear



Done



I

The

Thanks

Q¹ W² E³ R⁴ T⁵ Y⁶ U⁷ I⁸ O⁹ P⁰ <

! A S D F G H J K L >

, ↑ Z X C V B N M ⊗ .

/ 123 🌐 😊 space Search 📄

Как оно ВЫГЛЯДИТ (ЗВУЧИТ)

Что там в коде?

```
1 input[type="search"] {  
2     -webkit-appearance: none;  
3     -moz-appearance: none;  
4     appearance: none;  
5 }
```

Как оно выглядит (звучит)

А как это звучит?

```
1 export Input extends React.PureComponent {  
2   render() {  
3     return (  
4       <input type="text" />  
5       <input type="search" />  
6     );  
7   }  
8 }
```

В чём разница для скринридеров?

Ни в чём

Edit textfield blank



Как оно ВЫГЛЯДИТ (ЗВУЧИТ)

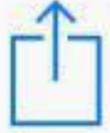
Type = search

```
1 export Input extends React.PureComponent {  
2   render() {  
3     return (  
4       <input type="search" />  
5     );  
6   }  
7 }
```

Done



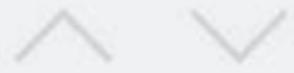
192.168.0.11



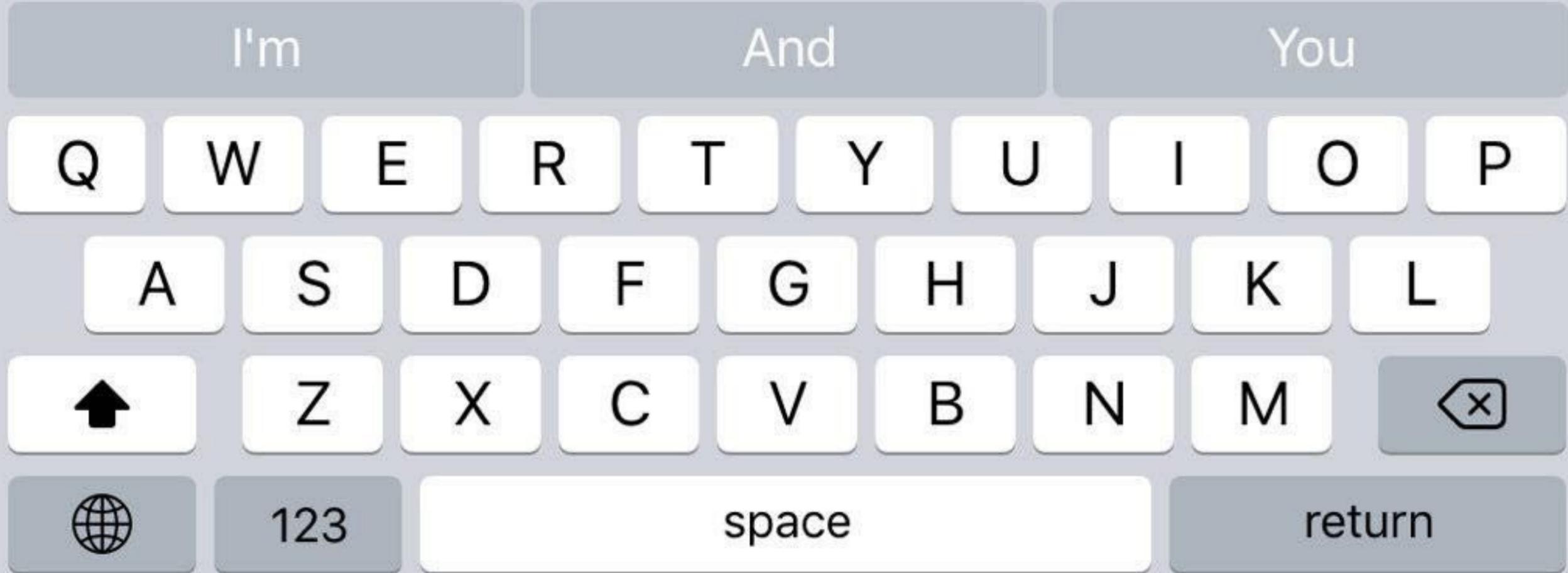
e.g. streets, cities, organizati

Search

Clear



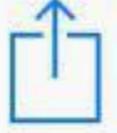
Done



Done



192.168.0.11



e.g. streets, cities, organizati

Search

Clear



Done

I'm

And

You

Q

W

E

R

T

Y

U

I

O

P

A

S

D

F

G

H

J

K

L



Z

X

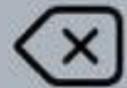
C

V

B

N

M



123

space

Search

Как оно выглядит (звучит)

Добавить контекста

```
1 export Input extends React.PureComponent {  
2   render() {  
3     return (  
4       <form action="">  
5         <input type="search" />  
6       </form>  
7     );  
8   }  
9 }
```

Как оно выглядит (звучит)

Добавить контекста

```
1 export Input extends React.PureComponent {  
2   render() {  
3     return (  
4       <form role="search">  
5         <input type="search" />  
6       </form>  
7     );  
8   }  
9 }
```

Edit textfield blank, search

Как оно выглядит (звучит)

А как это звучит?

```
1 export Input extends React.PureComponent {
2   render() {
3     return (
4       <form role="search">
5         <input type="search" aria-label="search on this website" />
6       </form>
7     );
8   }
9 }
```

search on this website, edit text, search

Как оно выглядит (звучит)

А как это звучит?

```
1 export Input extends React.PureComponent {
2   render() {
3     return (
4       <form role="search">
5         <input
6           type="search"
7           aria-label="search on this website"
8           placeholder="e.g. streets, cities, organizations"
9         />
10      </form>
11    );
12  }
13 }
```

search on this website, edit text, e.g. streets, cities, organizations, search

```
▼<form action role="search" aria-label="global search">
  ▼<input type="search" aria-label="Search 2GIS" placeholder="e.g. streets, cities, organizations">
    ▼#shadow-root (user-agent)
      <div pseudo="-webkit-input-placeholder" id="placeholder" style="display: block !important; text-overflow: clip;">e.g. streets, cities, organizations</div>
      ▼<div id="text-field-container" pseudo="-webkit-textfield-decoration-container">
        ▼<div id="editing-view-port">
          <div></div>
        </div>
        <div pseudo="-webkit-search-cancel-button" id="search-clear" style="opacity: 0; pointer-events: none;"></div>
      </div>
    </input>
  </form>
```

```
1 input::-webkit-input-placeholder {  
2     color: black;  
3 }
```

```
1 input::-webkit-input-placeholder {
2   color: black;
3 }
4 input:-ms-input-placeholder {
5   color: black;
6 }
7 input:-moz-placeholder { /* Firefox 18- */
8   color: black;
9 }
10 input::-moz-placeholder { /* Firefox 19+ */
11   color: black;
12 }
13 input::placeholder {
14   color: black;
15 }
```

```
1 @keyframes breathing {
2   from { outline-width: 1px; }
3   to { outline-width: 5px; }
4 }
5
6 input:placeholder-shown {
7   outline: 1px #a3cd4d solid;
8   animation: 1s linear 0s infinite alternate breathing;
9 }
```

e.g. streets, cities, organizations

Search

Clear

A region landmark is a perceivable section of the page that allows users to be able to navigate to the section.

- A **region** landmark must have a label.

```
1 export GlobalForm extends React.PureComponent {
2   render() {
3     return (
4       <form role="search" aria-label="global search">
5         <input
6           type="search"
7           aria-label="search on this website"
8           placeholder="e.g. streets, cities, organizations"
9         />
10      </form>
11    );
12  }
13 }
14
15 export BuildingForm extends React.PureComponent {
16   render() {
17     return (
18       <form role="search" aria-label="search in this building">
19         <input
20           type="search"
21           aria-label="search in this building"
22           placeholder="e.g. flowers, shoes"
23         />
24      </form>
25    );
26  }
27 }
```

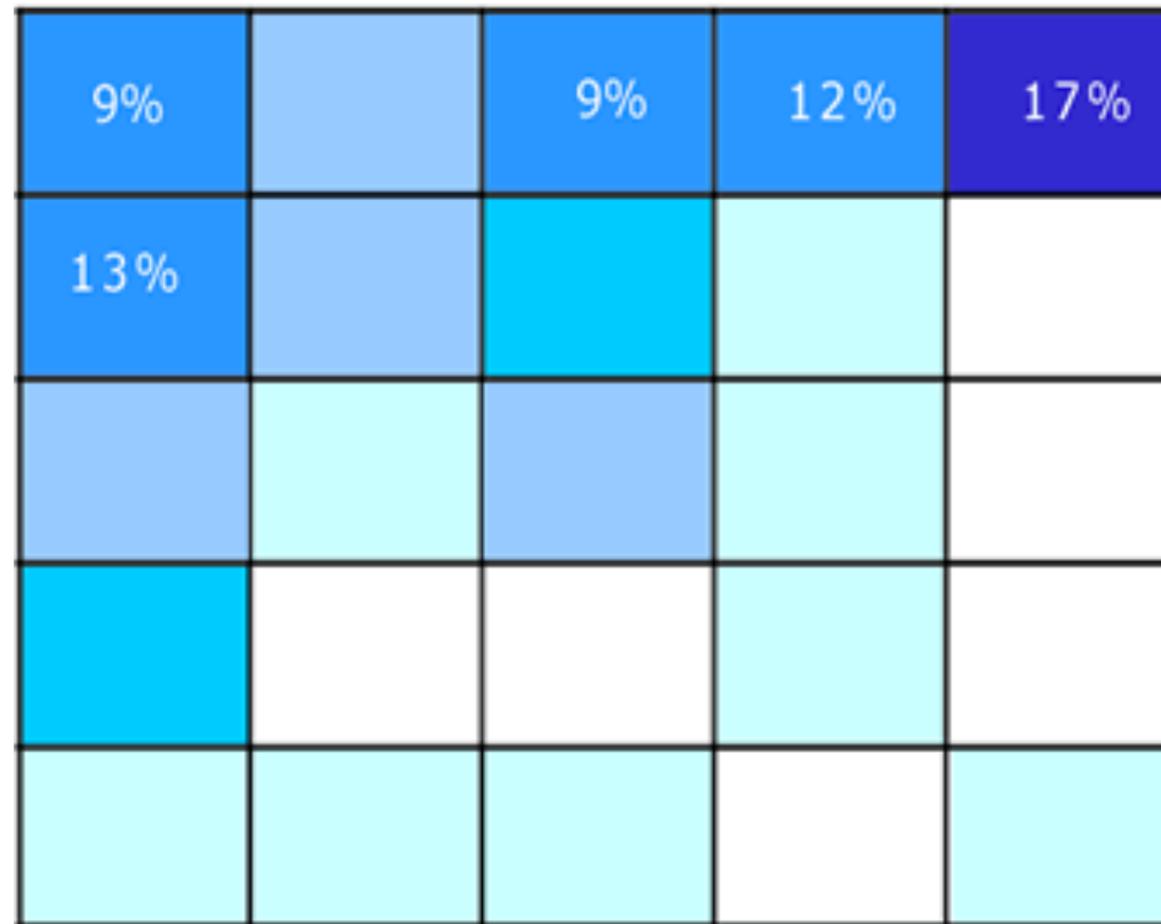
search on this website,
edit text,
e.g. streets, cities, organizations,
global search,
search

ДА ТЫ ШУТИШЬ?

The screenshot shows a mobile application interface. At the top, there is a dark navigation bar with a green circular logo on the left. A search bar contains the text "dubai mall" with a close button (X) on the right. To the right of the search bar is a blue square button with a white magnifying glass icon. Further right is a dark button with a white "A B" icon and the text "Route". Below the navigation bar, a light gray information card is displayed. The card has a title "The Dubai Mall" and a subtitle "3, Mohammed Bin Rashid Boulevard, Dubai". Below the subtitle is a search bar with the placeholder text "Search in the building" and a magnifying glass icon. At the bottom of the card, it says "1369 companies". On the left side of the screen, a vertical sidebar is partially visible, showing a list of items with a green dot and a clock icon, and a blue telephone icon at the bottom.

Как оно ВЫГЛЯДИТ (ЗВУЧИТ)

Где должен быть сёрч бар?



Как оно выглядит (звучит)

Где НЕ должен быть сёрч бар?

СКВЕР



Как оно выглядит (звучит)

Quick access

```
1 <form action="" role="search" aria-label="global search">
2   <input
3     type="search"
4     aria-label="search on 2GIS"
5     placeholder="e.g. streets, cities, organizations"
6     spellcheck="false"
7     accesskey="s"
8   />
9 </form>
```

Как оно ВЫГЛЯДИТ (ЗВУЧИТ)

Quick access

	Windows	Linux	Mac
Firefox	<code>Alt</code> + <code>Shift</code> + <code>key</code>		On Firefox 57 or newer, <code>Control</code> + <code>Option</code> + <code>key</code> -OR- <code>Control</code> + <code>Alt</code> + <code>key</code> On Firefox 14 or newer, <code>Control</code> + <code>Alt</code> + <code>key</code> On Firefox 13 or older, <code>Control</code> + <code>key</code>
Internet Explorer	<code>Alt</code> + <code>key</code>	N/A	
Google Chrome	<code>Alt</code> + <code>key</code>		<code>Control</code> + <code>Alt</code> + <code>key</code>
Safari	<code>Alt</code> + <code>key</code>	N/A	<code>Control</code> + <code>Alt</code> + <code>key</code>
Opera 15+	<code>Alt</code> + <code>key</code>		<code>Control</code> + <code>Alt</code> + <code>key</code>
Opera 12	<code>Shift</code> + <code>Esc</code> opens a contents list which are accessible by accesskey, then, can choose an item by pressing <code>key</code>		

Ввести, очистить и найти



Ввести, очистить и найти

Ожидания пользователя

- Поисковой запрос не исчезает,
- Esc очищает поле ввода,
- Enter начинает искать,
- Крестик,
- Кнопка «Найти»,
- Форма не «бесит».

MISSPELLED

ВВЕСТИ, ОЧИСТИТЬ И НАЙТИ

Spellchecking

```
1 return (  
2   <form role="search" aria-label="main search">  
3     <input  
4       type="search"  
5       aria-label="search on this website"  
6       placeholder="e.g. streets, cities, organizations"  
7       autoFocus={true}  
8       accessKey="s"  
9       spellCheck={false}  
10    />  
11    <button type="button">Clear</button>  
12    <button type="submit">Search</button>  
13  </form>  
14 );
```

Ввести, ОЧИСТИТЬ и НАЙТИ

Порядок кнопок

```
1 return (  
2   <form role="search" aria-label="main search">  
3     <input  
4       type="search"  
5       aria-label="search on this website"  
6       placeholder="e.g. streets, cities, organizations"  
7       autoFocus={true}  
8       accessKey="s"  
9       spellCheck={false}  
10    />  
11    <button type="button">Clear</button>  
12    <button type="submit">Search</button>  
13  </form>  
14 );
```

Ввести, ОЧИСТИТЬ и НАЙТИ

Порядок кнопок

```
1 return (  
2   <form role="search" aria-label="main search">  
3     <input  
4       type="search"  
5       aria-label="search on this website"  
6       placeholder="e.g. streets, cities, organizations"  
7       autoFocus={true}  
8       accessKey="s"  
9       spellCheck={false}  
10    />  
11    <button type="submit">Search</button>  
12    <button type="button">Clear</button>  
13  </form>  
14 );
```

Ввести, ОЧИСТИТЬ и НАЙТИ

Иконки как кнопки

```
1 return (  
2   <form role="search" aria-label="main search">  
3     <input  
4       type="search"  
5       aria-label="search on this website"  
6       placeholder="e.g. streets, cities, organizations"  
7       autoFocus={true}  
8       accessKey="s"  
9       spellCheck={false}  
10    />  
11    <Svg icon={searchIcon} onClick={this.submit} />  
12    <Svg icon={closeIcon} onClick={this.clear} />  
13  </form>  
14 );
```



Ввести, ОЧИСТИТЬ и НАЙТИ

Иконки как кнопки

```
1 return (  
2   <form role="search" aria-label="main search">  
3     <input  
4       type="search"  
5       aria-label="search on this website"  
6       placeholder="e.g. streets, cities, organizations"  
7       autoFocus={true}  
8       accessKey="s"  
9       spellCheck={false}  
10    />  
11    <Svg icon={searchIcon} onClick={this.submit} tabIndex={0} />  
12    <Svg icon={closeIcon} onClick={this.clear} tabIndex={0} />  
13  </form>  
14 );
```

Ввести, ОЧИСТИТЬ и НАЙТИ

Иконки как кнопки

```
1 return (  
2   <form role="search" aria-label="main search">  
3     <input  
4       type="search"  
5       aria-label="search on this website"  
6       placeholder="e.g. streets, cities, organizations"  
7       autoFocus={true}  
8       accessKey="s"  
9       spellCheck={false}  
10    />  
11    <Svg icon={searchIcon} onClick={this.submit} tabIndex={0} aria-label="search" />  
12    <Svg icon={closeIcon} onClick={this.clear} tabIndex={0} aria-label="clear" />  
13  </form>  
14 );
```

Ввести, очистить и найти

Иконки как кнопки

```
1 return (  
2   <form role="search" aria-label="main search">  
3     <input  
4       type="search"  
5       aria-label="search on this website"  
6       placeholder="e.g. streets, cities, organizations"  
7       autoFocus={true}  
8       accessKey="s"  
9       spellCheck={false}  
10    />  
11    <Svg icon={searchIcon} onClick={this.submit} tabIndex={0} aria-label="search" role="button" />  
12    <Svg icon={closeIcon} onClick={this.clear} tabIndex={0} aria-label="clear" role="button" />  
13  </form>  
14 );
```



Ввести, очистить и найти

Иконки как кнопки

```
1 return (  
2   <form role="search" aria-label="main search">  
3     <input  
4       type="search"  
5       aria-label="search on this website"  
6       placeholder="e.g. streets, cities, organizations"  
7       autoFocus={true}  
8       accessKey="s"  
9       spellCheck={false}  
10    />  
11  
12    <button type="submit">  
13      <Svg icon={searchIcon} />  
14    </button>  
15    <button type="button">  
16      <Svg icon={closeIcon} />  
17    </button>  
18  </form>  
19 );
```

Нам действительно нужны эти кнопки?



Google has no clear button |



All

Images

Videos

News

Maps

More

Settings

Tools

У Instagram нет кнопки «Найти»



🔍 pitercss



#pitercss
326 posts

У Instagram нет кнопки «Найти»
(но есть крестик)



🔍 pitercss



#pitercss
326 posts

Yandex

Yadnex has both at the same time|



Search



Log in

Web

Images

Video

News

Translate

Disk

Mail

All

[Открыть карту →](#)

ПОМОЩНИК 2GIS

Какой у вас вопрос?

Популярные вопросы

[Как добавить компанию в 2ГИС](#)

[Как сообщить об ошибке](#)

[Как внести изменения в карточку компании](#)

[«Подарки 2ГИС»: ответы на главные вопросы о конкурсе](#)

[Как удалить компанию из справочника](#)

[Что такое «Премия 2ГИС»?](#)

[Как установить 2ГИС](#)

[Как установить 2ГИС на устройства с ранними версиями операционной системы](#)

[Зачем нужен профиль в 2ГИС](#)

[Как обновить карты городов](#)

[Все вопросы](#)

[Профиль для владельца](#)

[Центр содержания вопросов](#)

Ввести, ОЧИСТИТЬ и НАЙТИ

Обойдёмся без кнопок

```
1 return (  
2   <form role="search" aria-label="main search">  
3     <input  
4       type="search"  
5       aria-label="search on this website"  
6       placeholder="e.g. streets, cities, organizations"  
7       autoFocus={true}  
8       accessKey="s"  
9       spellCheck={false}  
10    />  
11  
12    <Svg icon={searchIcon} />  
13    <Svg icon={closeIcon} onClick={this.clear} />  
14  </form>  
15 );
```

ВВЕСТИ, ОЧИСТИТЬ И НАЙТИ

Controlled vs uncontrolled

```
1 return (  
2   <form role="search" aria-label="main search">  
3     <input  
4       type="search"  
5       aria-label="search on this website"  
6       placeholder="e.g. streets, cities, organizations"  
7       autoFocus={true}  
8       accessKey="s"  
9       spellCheck={false}  
10      value={value}  
11      onChange={onChange}  
12    />  
13    <Svg icon={searchIcon} />  
14    <Svg icon={closeIcon} onClick={this.clear} />  
15  </form>  
16 );
```

ВВЕСТИ, ОЧИСТИТЬ И НАЙТИ

Clear button handler

```
1 clear = () => {  
2   this.props.onClearClick();  
3 }
```



Москва ▾

РАЙОНЫ, КВАРТАЛЫ



Укажите адрес

Мы придумаем интересные вопросы

|e.g. streets, cities, organizations



Новосибирск

Раздольное

Гуси

Новолуговое

Барышево

Кольцово

Краснообск

Работает на API 2ГИС
Лицензионное соглашение



Ввести, ОЧИСТИТЬ и НАЙТИ

Clear button handler

```
1 clear = () => {  
2   this.props.onClearClick();  
3  
4   requestAnimationFrame(() => this.input.focus());  
5 }
```

Type in, clear and find

Controlled

```
1 return (  
2   <form role="search" aria-label="main search">  
3     <input  
4       type="search"  
5       aria-label="search on this website"  
6       placeholder="e.g. streets, cities, organizations"  
7       autoFocus={true}  
8       accessKey="s"  
9       spellCheck={false}  
10      value={value}  
11      onChange={onChange}  
12      ref={this.setSearchInputRef}  
13    />  
14    <Svg icon={searchIcon} />  
15    <Svg icon={closeIcon} onClick={this.clear} />  
16  </form>  
17 );
```

Ввести, ОЧИСТИТЬ и НАЙТИ

Controlled

```
1 setSearchInputRef = (searchInput) => { this.input = searchInput; };
```

ВВЕСТИ, ОЧИСТИТЬ И НАЙТИ

Controlled

```
1 return (  
2   <form role="search" aria-label="main search">  
3     <input  
4       type="search"  
5       aria-label="search on this website"  
6       placeholder="e.g. streets, cities, organizations"  
7       autoFocus={true}  
8       accessKey="s"  
9       spellCheck={false}  
10      value={value}  
11      onChange={onChange}  
12      ref={this.setSearchInputRef}  
13    />  
14    <Svg icon={searchIcon} />  
15    <Svg icon={closeIcon} onClick={this.clear} />  
16  </form>  
17 );
```

NOW

IT'S TIME

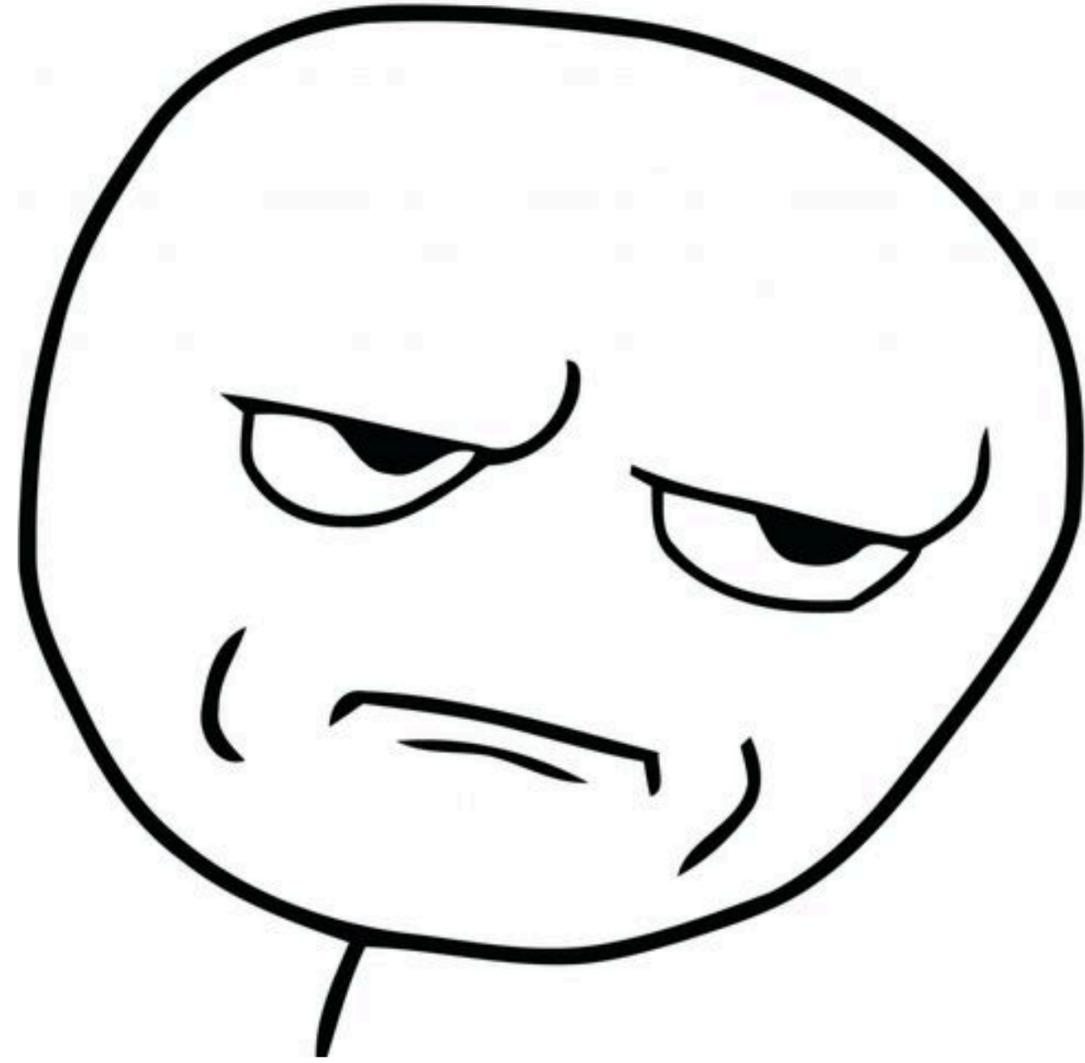
FOR

THE SUGGESTER

```

1 class Suggester extends React.PureComponent {
2   state = {
3     value: '',
4     isSearchTurnedOn: false
5   };
6
7   static getDerivedStateFromProps(nextProps, prevState) {
8     if (nextProps.isSearchTurnedOn !== prevState.isSearchTurnedOn) {
9       return {
10        value: '',
11        isSearchTurnedOn: nextProps.isSearchTurnedOn
12      };
13    }
14
15    return null;
16  }
17
18  onChange = (evt) => {
19    const val = evt.target.value;
20    this.setState({
21      value: val
22    });
23    const { isSearchTurnedOn, dispatch } = this.props;
24    const isEnoughLength = val.length >= 2;
25    const emptyVal = /^s*$/gi.exec(val);
26
27    if (!isSearchTurnedOn && !emptyVal && isEnoughLength) {
28      dispatch(searchingOn());
29    }
30
31    if (!emptyVal && isEnoughLength) {
32      dispatch(searchingRequest(val));
33    }
34
35    if (isSearchTurnedOn && emptyVal) {
36      dispatch(searchingOff());
37    }
38  };
39
40  onCrossClick = () => {
41    this.setState({
42      value: ''
43    });
44
45    this.props.dispatch(searchingOff());
46  };
47
48  render() {
49    return (
50      <input value={this.state.value} onChange={this.onChange} onClearClick={this.onCrossClick} />
51    );
52  }
53 }
54
55 const mapStateToProps = (state) => ({
56   isSearchTurnedOn: state.isSearchTurnedOn,
57 });
58
59 const connected = connect(
60   mapStateToProps,
61   dispatch => ({actions: bindActionCreators(actionCreators, dispatch)})
62 )(Suggester);

```



Open the map →

HELPER 2GIS

What is your question?

Popular questions

[2gis fails in start, crushes, throws an errorrrr](#)

[How to find building, firm or place](#)

All questions

★ Frequently Asked Questions

⚠ Problems and errors

🔍 Search

Ввести, очистить и найти

Container component

1. Сделать инпут controlled

```
1 class Suggester extends React.PureComponent {
2   state = {
3     value: ''
4   };
5
6   onChange = (evt) => {
7     const val = evt.target.value;
8     this.setState({
9       value: val
10    });
11  };
12
13  onCrossClick = () => {
14    this.setState({
15      value: ''
16    });
17  };
18
19  render() {
20    return (
21      <Input value={this.state.value} onChange={this.onChange} onClearClick={this.onCrossClick} />
22    );
23  }
24 }
```

Ввести, очистить и найти

Container component

1. Сделать инпут controlled
2. Как минимум два символа для поиска

```
1 onChange = (evt) => {
2   const val = evt.target.value;
3   this.setState({
4     value: val
5   });
6   const { isSearchTurnedOn, dispatch } = this.props;
7   const isEnoughLength = val.length >= 2;
8
9   if (!isSearchTurnedOn && isEnoughLength) {
10    dispatch(searchingOn());
11  }
12
13  if (isEnoughLength) {
14    dispatch(searchingRequest(val));
15  }
16 };
```

Ввести, очистить и найти

Container component

1. Сделать инпут controlled
2. Как минимум два символа для поиска
3. Без пустых символов

```
1 onChange = (evt) => {
2   const val = evt.target.value;
3   this.setState({
4     value: val
5   });
6   const { isSearchTurnedOn, dispatch } = this.props;
7
8   const emptyVal = /^\\s*$\\/gi.exec(val);
9
10  if (isSearchTurnedOn && emptyVal) {
11    dispatch(searchingOff());
12    return;
13  }
14
15  const isEnoughLength = val.length >= 2;
16
17  if (!isSearchTurnedOn && isEnoughLength) {
18    dispatch(searchingOn());
19  }
20
21  if (isEnoughLength) {
22    dispatch(searchingRequest(val));
23  }
24 };
```

Ввести, очистить и найти

Container component

1. Сделать инпут controlled
2. Как минимум два символа для поиска
3. Без пустых символов
4. Очищать поиск по изменению location

```
1 state = {
2   value: '',
3   isSearchTurnedOn: false
4 };
5
6 static getDerivedStateFromProps(nextProps, prevState) {
7   if (nextProps.isSearchTurnedOn !== prevState.isSearchTurnedOn) {
8     return {
9       value: '',
10      isSearchTurnedOn: nextProps.isSearchTurnedOn
11    };
12  }
13
14  return null;
15 }
```

```

1 class Suggester extends React.PureComponent {
2   state = {
3     value: '',
4     isSearchTurnedOn: false
5   };
6
7   static getDerivedStateFromProps(nextProps, prevState) {
8     if (nextProps.isSearchTurnedOn !== prevState.isSearchTurnedOn) {
9       return {
10        value: '',
11        isSearchTurnedOn: nextProps.isSearchTurnedOn
12      };
13    }
14
15    return null;
16  }
17
18  onChange = (evt) => {
19    const val = evt.target.value;
20    this.setState({
21      value: val
22    });
23    const { isSearchTurnedOn, dispatch } = this.props;
24    const isEnoughLength = val.length >= 2;
25    const emptyVal = /^s*$/gi.exec(val);
26
27    if (!isSearchTurnedOn && !emptyVal && isEnoughLength) {
28      dispatch(searchingOn());
29    }
30
31    if (!emptyVal && isEnoughLength) {
32      dispatch(searchingRequest(val));
33    }
34
35    if (isSearchTurnedOn && emptyVal) {
36      dispatch(searchingOff());
37    }
38  };
39
40  onCrossClick = () => {
41    this.setState({
42      value: ''
43    });
44
45    this.props.dispatch(searchingOff());
46  };
47
48  render() {
49    return (
50      <input value={this.state.value} onChange={this.onChange} onClearClick={this.onCrossClick} />
51    );
52  }
53 }
54
55 const mapStateToProps = (state) => ({
56   isSearchTurnedOn: state.isSearchTurnedOn,
57 });
58
59 const connected = connect(
60   mapStateToProps,
61   dispatch => ({actions: bindActionCreators(actionCreators, dispatch)})
62 )(Suggester);

```

```

1 class Input extends React.PureComponent {
2   constructor(props) {
3     super(props);
4     this.input = React.createRef();
5   }
6
7   clear = () => {
8     this.props.onClearClick();
9     requestAnimationFrame(() => this.input.current.focus());
10  }
11
12  render() {
13    const {onChange, value} = this.props;
14    return (
15      <form role="search">
16        <input
17          type="search"
18          aria-label="search on this website"
19          placeholder="e.g. streets, cities, organizations"
20          accessKey="s"
21          spellCheck={false}
22          autoFocus={true}
23          value={value}
24          onChange={onChange}
25          ref={this.input}
26        />
27
28        {!value && <Svg icon={searchIcon} />}
29        {value && <Svg icon={closeIcon} onClick={this.clear} />}
30      </form>
31    );
32  }
33 }

```



**KEEP
CALM
AND
TRY
YOUR BEST**

МЫ ХОТИМ ЕЩЕ!

- <https://goo.gl/jNnZrQ> — ChromeVox
- <https://goo.gl/3yVMhD> — WAI-ARIA Authoring Practices 1.1
- <https://redux-saga.js.org> — Redux-Saga Tutorial
- <https://goo.gl/AgNQkv> — Why Did Wikipedia Move The Search Box?
- <https://goo.gl/AygJDi> — Slides

Спасибо

Вопросы?

@rayproud