

Лабораторная работа 15. Браузерный JavaScript. Изменение разметки в документе.

Цель работы. Познакомиться с особенностями работы JavaScript с элементами DOM при интерпретации разметки в браузере, а также при динамическом изменении разметки на странице пользователем при его взаимодействии и веб-интерфейсом.

Результатом работы должны быть выполненные задания (15.1–15.2). Задания выполняются в песочницах codepen, repl или в репозитории на гитхаб, сдается только второе задание в виде url-ссылки, доступной для просмотра.

Перед выполнением работы следует ознакомиться с теоретическим материалом, сопровождаемым примерами, в [конспекте](#).

Задание 15.1. Навигация по сайту библиотеки современных авторов

Техническое задание

О проекте

Библиотека современных авторов — это типовой сайт каталога. Есть алфавитный указатель авторов, группировка книг по автору. В области просмотра книг выбранного автора отображается подробная информация по каждой книге.

Входные данные

Предполагается, что данные для каталога приходят с сервера. Это массив объектов следующего вида:

```
[
  {
    key: 'A',
    items: [
      {
        author: 'Адамс Гай',
        books: [
          {
            genre: 'Детские книги',
            title: 'Очень добрая история',
            year: '2020',
            img: 'book-cover-345.jpg',
            price: '560 ₺',
            summary: 'Мудрая и добрая сказка, в которой просто и проникновенно говорится о самом важном: о дружбе, любви, о том, как важно помогать друг другу. Книга знакомит читателя с историей девочки Анны, которая приехала в маленький угрюмый городок. Своими поступками она запустила цепочку добрых дел и подружила всех жителей города, которые уже и забыли, как радоваться жизни.',
          },
          {
            genre: 'Детские книги',
            title: 'Чудище по имени Лень',
            year: '2019',
            img: 'book-cover-348.jpg',
            price: '480 ₺',
            summary: 'Сказка о необычном чудовище, которое иногда приходит ко всем. Герои сказки узнают, что это самая обычная лень, которая живёт в каждом из
```

```

нас. На борьбу с ленью они приглашают звёздного мага, который способен победить
чудовище.',
    },
    ...
  ],
},
{
  author: 'Агутин Леонид',
  books: [
    {
      genre: 'Детские книги',
      title: 'Лесной оркестр',
      year: '2019',
      img: 'book-cover-346.jpg',
      price: '1100 ₺',
      summary: 'Все жители леса объединились, чтобы устроить Мишке сюрприз
на День Рождения. Что же было подарком на День Рождения и как отметили этот день
лесные жители.',
    },
    {
      genre: 'Художественная литература',
      title: 'Хранитель снов',
      year: '2016',
      img: 'book-cover-353.jpg',
      price: '460 ₺',
      summary: 'Происходит ужасное – у всех людей в мире пропали хорошие
сны и мечты. На помощь приходит хранитель снов. Но ему не обойтись без помощи
своих юных друзей.',
    },
    ...
  ],
},
...
],
},
{
  key: 'Б',
  items: [
    {
      author: 'Белицкая Светлана',
      books: [
        {
          genre: 'Хобби и досуг',
          title: 'Беззаботная лама',
          year: '2017',
          img: 'book-cover-350.jpg',
          price: '460 ₺',
          summary: 'Легкая и юмористическая история о ламе, которую ничего
никогда не заботит. У нее все всегда хорошо. Ничто на свете не может огорчить
беззаботную ламу.',
        },
        ...
      ],
    },
    ...
  ],
},
{
  key: 'В',
  items: [],
}
...
]

```

key — это буква алфавитного указателя,
items — список авторов, имя которых начинается на букву *key*.

Отдельная запись об авторе состоит из полей:

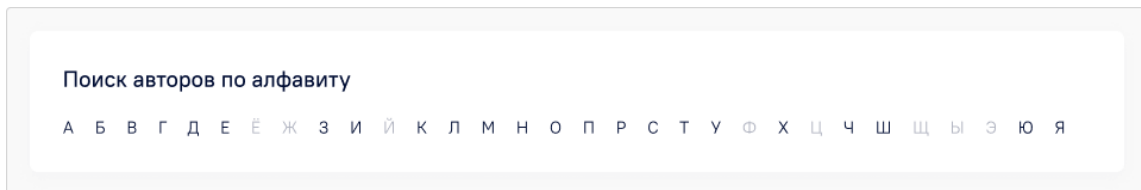
author — имя автора,
books — список книг автора.

Отдельная запись о книге содержит поля:

genre — жанр,
title — заголовок,
year — год издания,
img — изображение обложки книги,
price — цена,
summary — аннотация к книге.

Описание функциональности

Алфавитный указатель авторов содержит все буквы алфавита и позволяет быстро прокрутить сайт к списку авторов на выбранную букву. Буквы, которые содержат пустой список авторов (`items.length === 0`), визуально отличаются и становятся некликабельными.



Алфавитный указатель авторов

В списке авторов, помимо имени автора, отображается количество его книг в правильной числовой форме. Например, **11 книг**, **2 книги**, **31 книга**.

А		
Абрагимов Руслан 12 книг	Аверин Святослав 2 книги	Адлер Геворк 10 книг
Абрамова Валентина 6 книг	Аврелий Август 10 книг	Азарова Александра 8 книг

Список авторов

По умолчанию в области просмотра книг выводится текст с подсказкой **Нажмите на автора, чтобы посмотреть список его книг**.

Поиск авторов по алфавиту

А Б В Г Д Е Ё Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Э Ю Я

А

Абрагимов Руслан

12 книг

Абрамова Валентина

6 книг

Агутин Леонид

2 книги

Адамс Гай

10 книг

Книги

Нажмите на автора, чтобы посмотреть список его книг

Область просмотра книг. Состояние по умолчанию

При клике на автора область заполняется списком книг этого автора. В списке содержится подробная информация по каждой книге: название, обложка, год издания, цена и аннотация.

Поиск авторов по алфавиту

А Б В Г Д Е Ё Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Э Ю Я

А

Абрагимов Руслан

12 книг

Абрамова Валентина

6 книг

Абрамс Рей

5 книг

Авесян Маринэ

7 книг

Авдеев Петр

7 книг

Аверин Святослав

2 книги

Аврелий Август

10 книг

Авченко Семен

8 книг

Агутин Леонид

2 книги

Адамс Гай

10 книг

Адлер Геворк

10 книг

Азарова Александра

8 книг

Айзенберг Евгений

8 книг

Аксенова Кристина

7 книг

Аллилуев Федор

3 книги

Адамс Гай

Очень добрая история

2020

560 Р

Купить

Лесной оркестр

2019

1100 Р

Купить

Чудище по имени Лень

2019

480 Р

Купить

Загадка таинственного озера

2018

630 Р

Купить

Мудрая и добрая сказка, в которой просто и проникновенно говорится о самом важном: о дружбе, любви...

Читать дальше

Все жители леса объединились, чтобы устроить Мишке сюрприз на День Рождения. Что же было...

Читать дальше

Сказка о необычном чудовище, которое иногда приходит ко всем. Герои сказки узнают, что это...

Читать дальше

Тайну этого озера не могут разгадать уже много лет. Но оно совсем не страшное, а наоборот...

Читать дальше

Область просмотра книг выбранного автора

Аннотация к книге отображается в двух видах: сокращенном и полном. В сокращенном варианте область просмотра ограничена тремя строками и появляется ссылка [Читать дальше](#). При полном варианте отображении текст аннотации показан полностью, а ссылка меняется на [Скрыть описание](#). Переключение между сокращенным и полным вариантом происходит кликом по этой ссылке.

Поиск авторов по алфавиту

А Б В Г Д Е Ё Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Э Ю Я

А

Абрагимов Руслан

12 книг

Абрамова Валентина

6 книг

Абрамс Рей

5 книг

Авесян Маринэ

7 книг

Авдеев Петр

7 книг

Аверин Святослав

2 книги

Аврелий Август

10 книг

Авченко Семен

8 книг

Агутин Леонид

2 книги

Адамс Гай

10 книг

Адлер Геворк

10 книг

Азарова Александра

8 книг

Айзенберг Евгений

8 книг

Аксенова Кристина

7 книг

Аллилуев Федор

3 книги

Адамс Гай

Очень добрая история

2020

560 Р

Купить

Мудрая и добрая сказка, в которой просто и проникновенно говорится о самом важном: о дружбе, любви...

Читать дальше

Лесной оркестр

2019

1100 Р

Купить

Все жители леса объединились, чтобы устроить Мишке сюрприз на День Рождения. Что же было...

Читать дальше

Чудище по имени Лень

2019

480 Р

Купить

Сказка о необычном чудовище, которое иногда приходит ко всем. Герои сказки узнают, что это...

Читать дальше

Загадка таинственного озера

2018

630 Р

Купить

Тайну этого озера не могут разгадать уже много лет. Но оно совсем не страшное, а наоборот...

Читать дальше

Сокращённый вариант отображения аннотации к книге

Поиск авторов по алфавиту

А Б В Г Д Е Ё Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Э Ю Я

А

Абрагимов Руслан

12 книг

Абрамова Валентина

6 книг

Абрамс Рей

5 книг

Авесян Маринэ

7 книг

Авдеев Петр

7 книг

Аверин Святослав

2 книги

Аврелий Август

10 книг

Авченко Семен

8 книг

Агутин Леонид

2 книги

Адамс Гай

10 книг

Адлер Геворк

10 книг

Азарова Александра

8 книг

Айзенберг Евгений

8 книг

Аксенова Кристина

7 книг

Аллилуев Федор

3 книги

Адамс Гай

Очень добрая история

2020

560 Р

Купить

Мудрая и добрая сказка, в которой просто и проникновенно говорится о самом важном: о дружбе, любви, о том, как важно помогать друг другу. Книга знакомит читателя с историей девочки Анны, которая приехала в маленький угрюмый городок. Своими поступками она запустила цепочку добрых дел и подружила всех жителей города, которые уже и забыли, как радоваться жизни.

Скрыть описание

Лесной оркестр

2019

1100 Р

Купить

Все жители леса объединились, чтобы устроить Мишке сюрприз на День Рождения. Что же было...

Читать дальше

Чудище по имени Лень

2019

480 Р

Купить

Сказка о необычном чудовище, которое иногда приходит ко всем. Герои сказки узнают, что это...

Читать дальше

Загадка таинственного озера

2018

630 Р

Купить

Тайну этого озера не могут разгадать уже много лет. Но оно совсем не страшное, а наоборот...

Читать дальше

Вариант отображения с полным текстом аннотации к книге

Имя автора, чьи книги находятся в области просмотра, визуально

отличается от других.

В макете содержится вариант с пустым список книг и вариант с заполненным списком книг выбранного автора.

[Скачать макет кейса в формате Figma](#)

Исходные материалы

В кейс включена готовая вёрстка сайта. Структура папок и файлов у сайта следующая:

- в файле `index.html` находится разметка сайта (тут же можно найти разметку для *DOM*-элементов, которые будут добавлены на сайт);
- в папках `css`, `fonts` и `img` находятся необходимые для оформления статические ресурсы;
- в папке `js` есть заготовка для скриптов — файл `script.js` — работу нужно вести в нём.

В файл `script.js` добавлены тестовые (*mock*) входные данные, которые можно использовать для проверки работоспособности кода.

По [ссылке](#) можно скачать стартовый шаблон с версткой страницы.

Теперь разберем, как добавить *DOM*-элементы на веб-страницу и изменить их свойства в *JavaScript*-коде.

1. Первый шагом добавим вёрстку, которую скачаем со стартовым шаблоном.
2. Откроем файл `script.js`. Вместе со стартовым шаблоном пришёл массив данных **authors**. Эти данные используем, чтобы проверить наш код не отправляя запрос на сервер.

/ Тестовые данные */*

```
const authors = [
  {
    key: 'A',
    items: [
      {
        author: 'Адамс Гай',
        books: [
          {
            genre: 'Детские книги',
            title: 'Очень добрая история',
            year: '2020',
            img: 'book-cover-345.jpg',
            price: '560 ₺',
            summary: 'Мудрая и добрая сказка, в которой просто и проникновенно говорится о самом важном: о дружбе, любви, о том, как важно помогать друг другу. Книга знакомит читателя с историей девочки Анны, которая приехала в маленький угрюмый городок. Своими поступками она запустила цепочку добрый дел и подружила всех жителей города, которые уже и забыли, как радоваться жизни.',
          },
          {
            genre: 'Детские книги',
            title: 'Чудище по имени Лень',
            year: '2019',
            img: 'book-cover-348.jpg',
            price: '480 ₺',
            summary: 'Сказка о необычном чудовище, которое иногда приходит ко всем.
```

Герои сказки узнают, что это самая обычная лень, которая живёт в каждом из нас. На борьбу с ленью они приглашают звёздного мага, который способен победить чудовище.',

```

    },
    {
      genre: 'Детские книги',
      title: 'Загадка таинственного озера',
      year: '2018',
      img: 'book-cover-347.jpg',
      price: '630 ₺',
      summary: 'Тайну этого озера не могут разгадать уже много лет. Но оно совсем
не страшное, а наоборот, с волшебной и чистой водой, способной дать силу и уставшему
путнику и раненому зверю.',
    },
  ],
},
{
  author: 'Агутин Леонид',
  books: [
    {
      genre: 'Детские книги',
      title: 'Лесной оркестр',
      year: '2019',
      img: 'book-cover-346.jpg',
      price: '1100 ₺',
      summary: 'Все жители леса объединились, чтобы устроить Мишке сюрприз на
День Рождения. Что же было подарком на День Рождения и как отметили этот день лесные
жители.',
    },
    {
      genre: 'Художественная литература',
      title: 'Хранитель снов',
      year: '2016',
      img: 'book-cover-353.jpg',
      price: '460 ₺',
      summary: 'Происходит ужасное – у всех людей в мире пропали хорошие сны и
мечты. На помощь приходит хранитель снов. Но ему не обойтись без помощи своих юных
друзей.',
    },
  ],
},
],
},
{
  key: 'Б',
  items: [
    {
      author: 'Белицкая Светлана',
      books: [
        {
          genre: 'Хобби и досуг',
          title: 'Беззаботная лама',
          year: '2017',
          img: 'book-cover-350.jpg',
          price: '460 ₺',
          summary: 'Легкая и юмористическая история о ламе, которую ничего никогда не
заботит. У нее все всегда хорошо. Ничто на свете не может огорчить беззаботную ламу.',
        },
      ],
    },
  ],
},
],
},
{
  key: 'В',
  items: [
    {
      author: 'Вагнер Марк',

```

```

books: [
  {
    genre: 'Художественная литература',
    title: 'Единорожка',
    year: '2015',
    img: 'book-cover-354.jpg',
    price: '760 ₺',
    summary: 'Приключения забавного и неуклюжего единорожки Лилу, который
отправился на поиски друзей. Хорошая и поучительная история, полная забавных
приключений.',
  },
],
},
],
{
  key: 'Г',
  items: [],
},
{
  key: 'Д',
  items: [],
},
{
  key: 'Е',
  items: [],
},
{
  key: 'Ё',
  items: [],
},
{
  key: 'Ж',
  items: [],
},
{
  key: 'И',
  items: [],
},
{
  key: 'Й',
  items: [],
},
{
  key: 'К',
  items: [],
},
{
  key: 'Л',
  items: [],
},
{
  key: 'М',
  items: [],
},
{
  key: 'Н',
  items: [],
},
{
  key: 'О',
  items: [],
},
{
  key: 'П',
  items: [],
},

```



```

    },
    {
      key: 'Р',
      items: [],
    },
    {
      key: 'С',
      items: [],
    },
    {
      key: 'Т',
      items: [],
    },
    {
      key: 'У',
      items: [],
    },
    {
      key: 'Ф',
      items: [],
    },
    {
      key: 'Х',
      items: [],
    },
    {
      key: 'Ц',
      items: [],
    },
    {
      key: 'Ч',
      items: [],
    },
    {
      key: 'Ш',
      items: [],
    },
    {
      key: 'Щ',
      items: [],
    },
    {
      key: 'Ы',
      items: [],
    },
    {
      key: 'Э',
      items: [],
    },
    {
      key: 'Ю',
      items: [],
    },
    {
      key: 'Я',
      items: [],
    },
  ],
];

```

3. Переходим к реализации функционала. В первой строке включим **строгий режим**. Так проще будет выявить ошибки. Добавим определение и вызов функции **init**. Эта функция запускается при загрузке страницы.

```
const init = () => {
```

```
};
```

```
init();
```

4. Контейнер или целевой элемент для алфавитного указателя — это список `<ul class="site-nav">`. Найдём его с помощью метода `document.querySelector` и запишем в переменную `navContainer`. Добавим описание функции `render`, внутри которого очистим содержимое списка. Для очистки используем свойство `innerHTML`. Вызов функции `render` поместим внутрь `init`.

```
const navContainer = document.querySelector('.site-nav');
```

```
const render = () => {  
  navContainer.innerHTML = '';  
}
```

```
const init = () => {  
  render();  
};
```

```
init();
```

5. Создадим элементы списка. Для этого используем шаблонные строки и метод `insertAdjacentHTML`. С помощью HTML проще добавить новые элементы на страницу. В разметке можно сразу указать все теги для элементов, включая вложенные, добавить атрибуты для них. Определим функцию `createNavItemString`, в которой соберём строку с HTML для отдельного элемента навигации. Шаблон возьмём в разметке. В параметрах функции передадим ключ элемента и список авторов. У ссылки будем добавлять атрибут `href`, если в списке авторов есть хотя бы один элемент (автор на эту букву). На основе массива данных `authors` создадим новый массив с разметкой для элементов. Для этого используем метод `map`, в котором будем вызывать функцию `createNavItemString`. А затем соберём всю разметку в одну строку с помощью метода `join('')`. Результат запишем в переменную `navElementsString`.

```
const navContainer = document.querySelector('.site-nav');
```

```
const createNavItemString = ({key, items}) =>  
  `<li class="site-nav-item">  
    <a class="site-nav-link" ${items.length > 0 ? `href="#${key}"` :  
  ''} >${key}</a>  
  </li>  
  `;
```

```
const render = () => {  
  navContainer.innerHTML = '';
```

```
  const navElementsString = authors.map((item) =>  
    createNavItemString(item)).join('');  
}
```

```
const init = () => {
  render();
};
```

- Добавляем собранный HTML перед закрывающим тегом navContainer. Для этого в первом параметре метода insertAdjacentHTML укажем beforeend. После этого элементы списка появились в DOM.

```
const render = () => {
  navContainer.innerHTML = '';

  const navElementsString = authors.map((item) =>
    createNavItemString(item)).join('');

  navContainer.insertAdjacentHTML('beforeend', navElementsString);
}
```

- Добавим элементы, которые соответствуют разделам с авторами на указанную букву. Целевым элементом этого блока будет <section class="authors">. Найдём этот элемент и затем очистим его содержимое.

```
const navContainer = document.querySelector('.site-nav');
const authorsContainer = document.querySelector('.authors');

const createNavItemString = ({key, items}) =>
  `<li class="site-nav-item">
    <a class="site-nav-link" ${items.length > 0 ? `href="#${key}"` :
  ''} >${key}</a>
  </li>
  `;
```

- Добавим разделы. Для отдельного раздела снова используем HTML из вёрстки. Создаём функцию createSectionString. На вход подаем ключ раздела и список авторов. Ещё раз проходим по массиву данных authors, на этот раз с помощью createSectionString сформируем HTML для всех разделов. Сразу добавим новые элементы в DOM.

```
const createSectionString = ({key, items}) =>
  `<section class="authors-section">
    <h2 id="${key}">${key}</h2>
    <ul class="author-list">
      ${items.map(({author, books}) => `
    <li class="author-item">
      <a class="author-link" data-value="${author}" href="#">
        <h3>${author}</h3>
        <p>${books.length} книги</p>
      </a>
    </li>
  `).join('')}
    </ul>
  </section>
  `;
```

- Если вы посмотрите на список авторов, скажем, на букву Б, то заметите,

что слово книги используется в неверной форме. Должно быть: 1 книга, 2 книги, 5 книг. Для правильного склонения числительных используем внешнюю библиотеку get-declension. Мы выбрали эту библиотеку. Подключаем get-declension в файле index.html с помощью тега <script>.

```
</footer>
<script src="js/vendor/get-declension.min.js"></script>
<script src="script.js"></script>
</body>
```

10. На странице с описанием библиотеки в разделе Использование есть примеры вызова функции. Доработаем код, который выводит раздел с авторами, добавим склонение числительных.

```
const createSectionString = ({key, items}) =>
const createSectionString = ({key, items}) =>
`<section class="authors-section">
  <h2 id="${key}">${key}</h2>
  <ul class="author-list">
    ${items.map(({author, books}) => {
      const declension = window.getDeclension({count: books.length, one:
'книга', few: 'книги', many: 'книг'});
      return `
        <li class="author-item">
          <a class="author-link" data-value="${author}" href="#">
            <h3>${author}</h3>
            <p>${declension}</p>
          </a>
        </li>
      `;
    })}.join('')}
  </ul>
</section>
`;
```

11. Далее нам нужно заполнить область с книгами выбранного автора <section class="books">. Для начала найдем элемент, который соответствует разделу books. Очистим его содержимое.

```
const navContainer = document.querySelector('.site-nav');
const authorsContainer = document.querySelector('.authors');
const booksContainer = document.querySelector('.books');
---
const authorsElementsString = authors.map((item) =>
createSectionString(item)).join('');

authorsContainer.insertAdjacentHTML('beforeend',
authorsElementsString);

booksContainer.innerHTML = '';
}

const init = () => {
  render();
};
```

12. По умолчанию, когда автор еще не выбран, в этой области выводится

текст с подсказкой Нажмите на автора, чтобы посмотреть список его книг. Определим функцию `createEmptyBookString`. В ней создадим HTML с заголовком и текстом для раздела. И добавим эти элементы в DOM.

```
....
}).join('')}
  </ul>
  </section>
  `;

const createEmptyBookString = () =>
  `<h2>Книги</h2>
  <p>Нажмите на автора, чтобы посмотреть список его книг</p>
  `;

const render = () => {
  navContainer.innerHTML = '';
  const authorsElementsString = authors.map((item) =>
    createSectionString(item)).join('');

  authorsContainer.insertAdjacentHTML('beforeend',
    authorsElementsString);

  booksContainer.innerHTML = '';

  const emptyBookElementString = createEmptyBookString();

  booksContainer.insertAdjacentHTML('beforeend',
    emptyBookElementString);
}

const init = () => {
  render();
};
```

.....

13. Область с книгами должна перерисовываться при клике на автора. Нам понадобится обработчик события `click`. Чтобы не создавать множество обработчиков событий, используем делегирование и повесим обработчик `click` на весь элемент `authorsContainer`. Создадим заготовку функции `authorClickHandler`. За счет метода `closest` клик будем отлавливать только на ссылке. В обработчике события отменим поведение ссылки по умолчанию.

```
...
const authorClickHandler = (evt) => {
  const element = evt.target.closest('a');

  if (!element) {
    return;
  }
  evt.preventDefault();
};
```

```
const init = () => {
  render();
  authorsContainer.addEventListener('click', authorClickHandler);
};
```

```
init();
```

14. Продолжим реализацию. Нам потребуется функция для формирования HTML для списка книг автора `createAuthorBooksString`. На вход функция будет принимать объект, который содержит поля `author` и `books`. Разметку для элементов возьмем в вёрстке.

```
...
const createEmptyBookString = () =>
  `<h2>Книги</h2>
  <p>Нажмите на автора, чтобы посмотреть список его книг</p>
  `;
```

```
const createAuthorBooksString = ({author, books}) =>
  `<h2>${author}</h2>
  <ul class="book-list">
    ${books.map(({title, year, img, price, summary}) => `
      <li class="author-item">
        <article class="book">
          
          <div class="title-wrapper">
            <h3>${title}</h3>
            <p class="year">${year}</p>
          </div>
            <p class="price">${price} <a class="cart"
href="#">Купить</a></p>
            <p class="description short">${summary}</p>
            <a class="more" href="#">Читать дальше</a>
          </article>
        </li>
      `).join('')}
    </ul>
  `;
```

```
const render = () => {
  navContainer.innerHTML = '';
  ...
```

15. При формировании раздела с авторами в атрибуте `data-value` мы добавили имя автора. Используем эти данные для поиска. Сначала найдём группу авторов на выбранную букву: `authors.find(({key}) => value.toUpperCase().indexOf(key.toUpperCase()) !== 0)`. А затем в этой группе отыщем нашего автора: `authorsGroup.items.find((item) => item.author === value)`. Полученную запись будем передавать параметром в функцию `createAuthorBooksString`. Дополнительные проверки помогут избежать ошибок при некорректных данных. Сформированный *HTML* добавляем в `booksContainer`. Протестируем наш код, нажмём на

несколько ссылок с именами авторов. Область для просмотра книг меняется при смене автора.

```
...
evt.preventDefault();

const value = element.dataset.value;

const authorsGroup = authors.find(({key}) =>
value.toUpperCase().indexOf(key.toUpperCase()) === 0);

if (authorsGroup && authorsGroup.items) {
  const currentAuthor = authorsGroup.items.find((item) => item.author
=== value);

  if (currentAuthor) {
    const booksString = createAuthorBooksString(currentAuthor);

    booksContainer.innerHTML = '';
    booksContainer.insertAdjacentHTML('beforeend', booksString);
  }
}
};
```

```
const init = () => {
  render();
}
```

...

16. Автор, книги которого выводятся в области просмотра, внешне отличается от остальных, у ссылки появляется дополнительный класс `active`. Создадим функцию `SetActiveAuthor`, она будет снимать класс `active` у элемента, который был до этого активным и добавлять этот класс на элемент, по которому кликнули.

...

```
const setActiveAuthor = (target) => {
  const activeAuthor =
authorsContainer.querySelector('.author-link.active');
  if (activeAuthor) {
    activeAuthor.classList.remove('active');
  }

  target.classList.add('active');
};
```

```
const authorClickHandler = (evt) => {
  ...
  ...
  if (currentAuthor) {
    const booksString = createAuthorBooksString(currentAuthor);

    booksContainer.innerHTML = '';
    booksContainer.insertAdjacentHTML('beforeend', booksString);
    setActiveAuthor(element);
  }
}
```

```
    }  
  };
```

```
const init = () => {  
  render();
```

17. В техническом задании написано, что аннотация к книге по умолчанию отображается в сокращенном виде, а при клике на ссылку Читать дальше показывается полностью. Реализуем этот функционал. Определяем обработчик события click для области просмотра книг. Снова воспользуемся делегированием. На этот раз событие будем «ловить» на ссылках с классом more.

```
...  
const moreLinkClickHandler = (evt) => {  
  const element = evt.target.closest('a.more');
```

```
  if (!element) {  
    return;  
  }  
  evt.preventDefault();
```

```
};
```

```
const init = () => {  
  render();
```

```
  authorsContainer.addEventListener('click', authorClickHandler);  
  booksContainer.addEventListener('click', moreLinkClickHandler);  
};
```

```
init();
```

18. Нам понадобится элемент с классом description, для него будем переключать класс short. Находим элемент с помощью метода querySelector. Метод вызываем для bookContainer (ближайший родитель с классом book для ссылки, на которой произошло событие). Тут же будем менять текстовое содержимое самой ссылки. Если описание содержит класс short, то в ссылке показывается текст Читать дальше, в противном случае — Скрыть описание. Для замены текстового содержимого используем свойство textContent.

```
.....
```

```
const bookContainer = element.closest('.book');  
const descriptionElement =  
bookContainer.querySelector('.description');
```

```
if (descriptionElement) {  
  descriptionElement.classList.toggle('short');  
  element.textContent =  
descriptionElement.classList.contains('short') ? 'Читать дальше' :  
'Скрыть описание';  
}
```



```
};
```

19. Следующим шагом реализуем плавную прокрутку. Для начала добавим обработчик события `click` для контейнера с навигацией. Отлавливать событие будем на ссылках. Для ссылок отменим действие по умолчанию.

.....

```
const navItemsClickHandler = (evt) => {  
  const element = evt.target.closest('a');  
  
  if (!element) {  
    return;  
  }  
  evt.preventDefault();  
};
```

20. Ссылки с буквами в алфавитном указателе содержат в значении атрибута `href` указатели на якоря соответствующих разделов. Используем значение атрибута `href`, чтобы найти нужный элемент и вызвать для него метод `scrollIntoView`. Реализацию плавной прокрутки мы разбираем в методике (см. выше по тексту).

...

```
evt.preventDefault();  
  
const blockId = element.getAttribute('href');  
  
if (blockId && blockId !== '#') {  
  const block = document.querySelector(blockId);  
  
  if (block) {  
    block.scrollIntoView({  
      behavior: 'smooth',  
      block: 'start',  
    });  
  }  
}  
};
```

21. Весь необходимый функционал реализован. Но наш код не идеален. Есть много повторяющихся фрагментов, например, у нас несколько раз повторяется последовательность команд: очистить содержимое родительского элемента, добавить HTML в родительский элемент. Выделим эти действия в отдельную функцию `addChildElementsFromString`. На вход будем подавать родительский элемент и HTML для дочерних элементы. Пройдемся по коду и заменим повторяющиеся команды на вызов новой функции.

....

```
const addChildElementsFromString = (parent, domString) => {  
  parent.innerHTML = '';  
  // добавляем разобранный текст как HTML перед закрывающим тегом  
  // родительского элемента (после последнего потомка)  
  parent.insertAdjacentHTML('beforeend', domString);  
};
```

....

22. Немного отрендерим синтаксис. Поскольку функции `createNavItemString` и `createSectionString` не используют данные из контекста (`this`), их можно передать как параметр в метод `map`.

23. Используем деструктурирующее присваивание, оператор `const value = element.dataset.value`; заменим на `const {value} = element.dataset`; Деструктурирующее присваивание и другие возможности синтаксиса *JavaScript* мы рассматриваем в разделе «Дополнительные возможности синтаксиса *JavaScript*». Обязательно после каждого «улучшения» проверяйте функционал. Сайт библиотеки современных авторов готов!

```
const value = element.dataset.value;  
const {value} = element.dataset;
```

Результат

Задание 15.2. Создайте скрипт для “Оживления” словаря психологических терминов

Техническое задание

О проекте

Словарь основных психологических терминов — это типовой сайт глоссария. На сайте есть алфавитный указатель.

Входные данные

Предполагается, что данные для глоссария приходят с сервера. Это массив объектов следующего вида:

```
[  
  {  
    key: 'A',  
    items: [  
      {  
        term: 'Абстиненция',  
        description: '<p><dfn>Абстинѐнция</dfn> (от <a href="#">нем.</a>  
<i>Abstinenz</i> и <a href="#">лат.</a> <i>Abstinentia</i>: воздержность,  
задержка, задержание, воздержание, воздержанность, а также пост, голодание,  
бескорыстие, честность) – в отличие от традиционного, <a  
href="#">наркологического</a> значения термина, в <a href="#">психоанализе</a>  
употребляется для описания состояния больных истерией страха и неврозом навязчивых  
состояний в процессе психоаналитической терапии. Абстиненция рассматривается не как  
патогенный фактор, а скорее как фактор, поддерживающий оптимальный уровень <a  
href="#">фрустрации</a>, необходимый для достижения терапевтического  
эффекта.</p>',  
        cite: 'Материал из Википедии – свободной энциклопедии',  
      },  
      {  
        term: 'Авторитетность',  
        description: '<p>Имеющий <a href="#">авторитет</a>, заслуживающий
```

признания, доверия и уважения.

Как видите, работа даже такого бесспорно авторитетного режиссера, как Станиславский, подвергалась некоторому сомнению со стороны критики. Блейхман отвечал без лишних слов: Совет для анархистов совершенно не авторитетен; если к его решению присоединятся большевики, то это ничего не значит, Совет в целом служит буржуазии и помещикам; никаких определенных намерений у анархистов на завтра нет; участвовать в манифестации они будут — со своими чёрными знамёнами, а насчет того, будут ли с оружием, то, может быть, пойдут без оружия, а может быть, и с оружием.

cite: 'Н. Н. Суханов, «Записки о революции / Книга 4», 1918–1921 г. (цитата из Национального корпуса русского языка, см. Список литературы)',

```
    },
    ...
  ],
},
{
  key: 'Б',
  items: [
    {
      term: 'Базальные эмоции',
      description: '<p><dfn>Базальные эмоции</dfn> – теоретический конструкт,
объединяющий эмоции минимального набора, на базе которых формируется все
многообразие эмоциональных процессов и состояний. К подобным эмоциям относят
эмоции радости, горя (печали), страха, гнева, удивления, отвращения.</p>',
      cite: 'Материал из Википедии – свободной энциклопедии',
    },
    {
      term: 'Бионика',
      description: '<p><dfn>Би́оника</dfn> (от <a href="#">др.-греч.</a> βίον
«живущее») – прикладная наука о применении в технических устройствах и системах
принципов организации, свойств, функций и структур живой природы, то есть формах
живого в природе и их промышленных аналогах.</p>',
      cite: 'Материал из Википедии – свободной энциклопедии',
    },
    ...
  ],
},
{
  key: 'В',
  items: [],
}
...
]
```

key — это буква алфавитного указателя,

items — список терминов, название которых начинается на букву *key*.

Отдельный термин состоит из полей:

term — название термина,

description — описание термина, может содержать HTML,

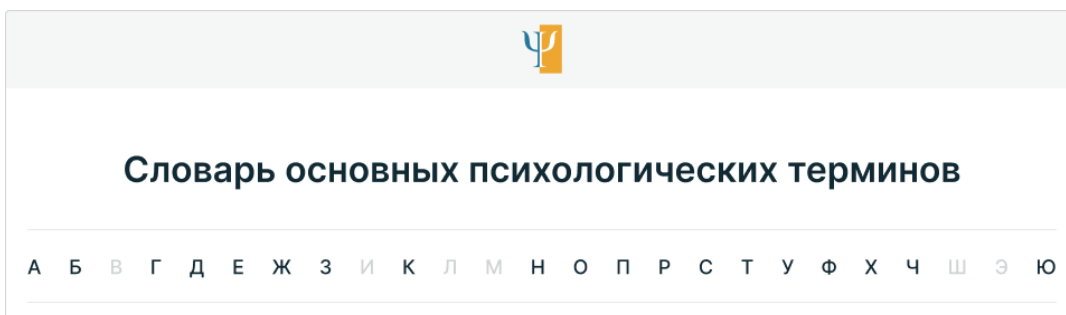
cite — источник.

Описание функциональности

Алфавитный указатель содержит все буквы алфавита и позволяет быстро прокрутить сайт к списку терминов на выбранную букву.

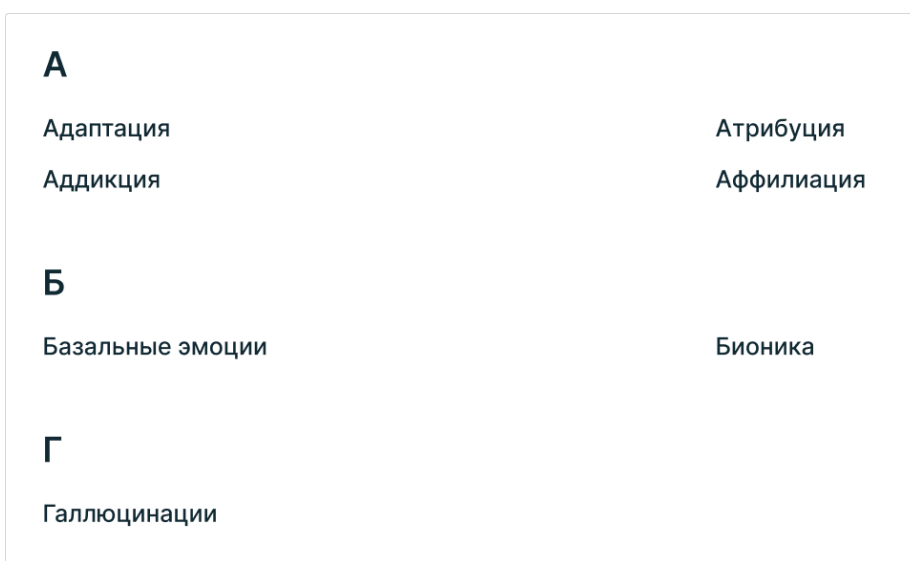
Буквы, которые содержат пустой список терминов (`items.length === 0`), становятся некликабельными (у ссылок отсутствует атрибут `href`,

например, `В`). Также они отличаются визуально.



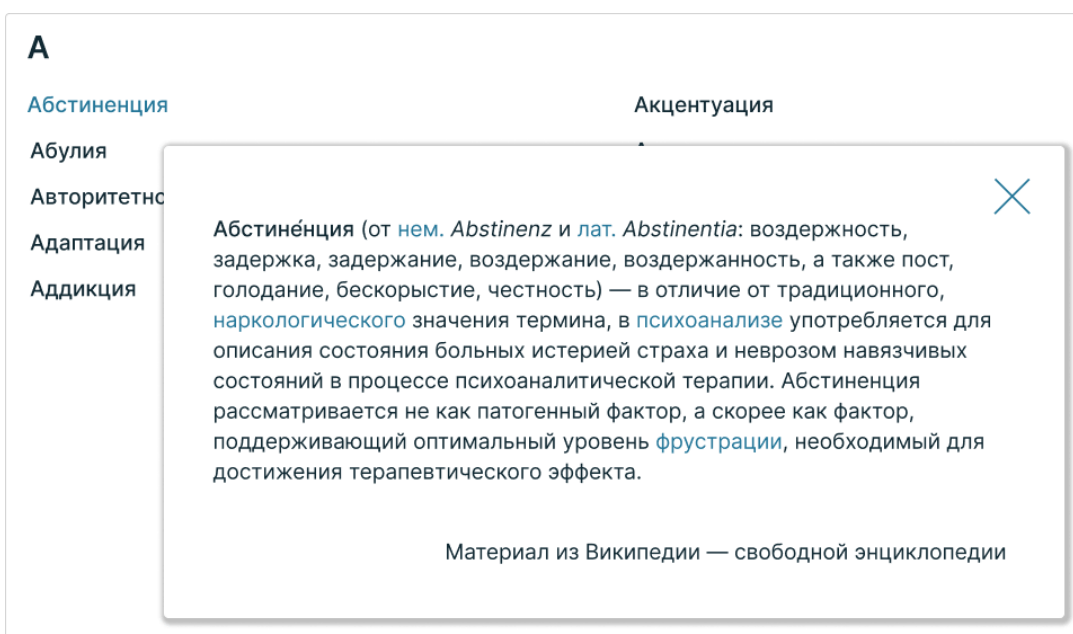
Алфавитный указатель психологических терминов

Для пустого списка терминов раздел на эту букву не отображается.



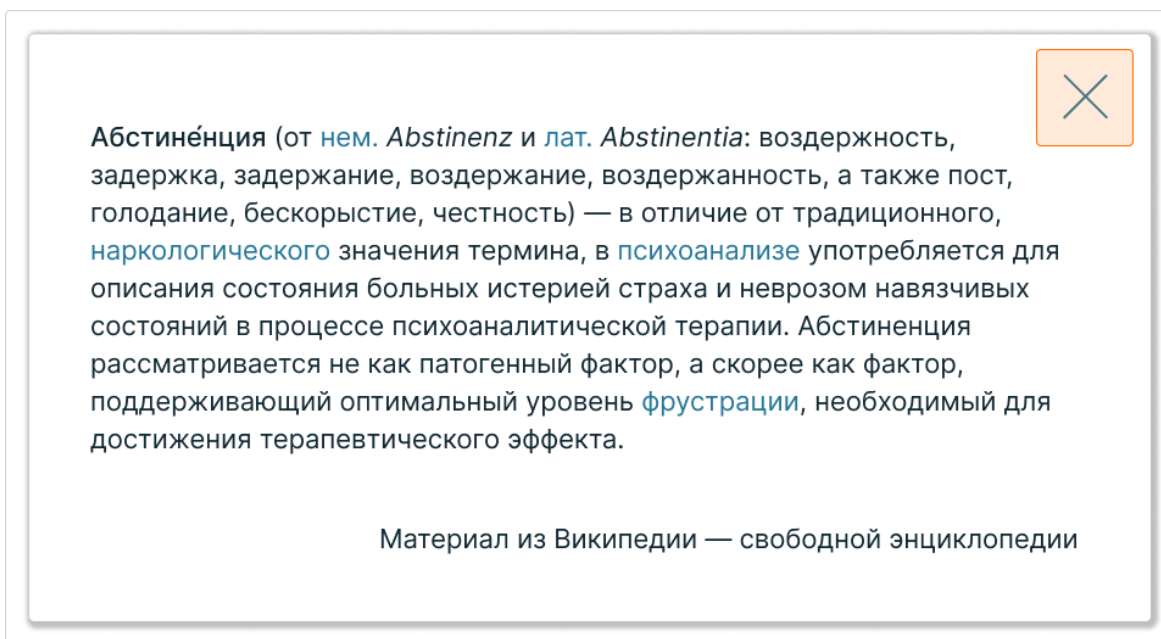
За разделом на букву Б идёт раздел на букву Г, раздел на букву В не выводится

При клике на термин открывается попап-окно с полным определением этого психологического термина и источником.



Попап-окно с полным определением психологического термина

Попап-окно закрывается по клику на кнопку **Заккрыть**.



Кнопка **Заккрыть**.

Определение термина может содержать в себе теги, их следует отображать как есть. В макете показаны оба состояния: попап-окно закрыто и попап-окно открыто.

[Скачать макет кейса в формате Figma](#)

Исходные материалы

В кейс включена готовая вёрстка сайта. Структура папок и файлов у сайта следующая:

- в файле **index.html** находится разметка сайта (тут же можно найти разметку для DOM-элементов, которые будут добавлены на сайт);

- в папках **css**, **fonts** и **img** находятся необходимые для оформления статические ресурсы;

- в папке **js** есть заготовка для скриптов — файл **script.js** — работу нужно вести в нём.

В файл **script.js** добавлены тестовые (*mock*) входные данные, которые можно использовать для проверки работоспособности кода.

По [ссылке](#) можно скачать стартовый шаблон с версткой страницы.

Дополнительная информация

Все необходимые CSS-классы и CSS-правила есть в исходных материалах. Для реализации функционала достаточно добавить элементам соответствующие классы.

Описание CSS-классов

CSS-класс `shown` — класс для вывода попап-а с определением термина. Для того, чтобы показать окно, нужно добавить этот класс для элемента `<div class="modal">` и разметка окна будет выглядеть так:

```
<div class="modal shown">
```

Скачайте [стартовый шаблон](#) навыка и напишите JS-код в файле `js/script.js`.

Инструкция

Чтобы проверить своё решение и довести его до идеала, действуйте так:

1. Выполните задачу кейса.
2. Загрузите файлы в `repl.it` или на `github` для проверки.
3. Сформируйте внешнюю ссылку на деплой проекта.

Каким должен быть проект с версткой?

- Сразу внутри папки с проектом должен находиться файл `index.html`.
- В папке `js` должен быть `script.js`, подключенный в `index.html`.
- Папки и файлы должны содержать только латинские символы.

Других ограничений нет.

Вспомогательные материалы

1. https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/JavaScript_basics
2. <https://developer.mozilla.org/ru/docs/Learn/JavaScript/Asynchronous>
3. https://www.w3schools.com/js/js_window.asp