УДК 004

Использование функциональности Drag-n-Drop на языке JavaScript

Халиманенков Андрей Сергеевич
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В данной статье рассматривается использование Drag-n-Drop (схвати и брось) в веб-приложениях на JavaScript. Использование этой функции помогает создавать интерфейсы таких приложений как Google Keep и т.п.
Ключевые слова: галерея, адаптивные изображения, разработка сайтов, JavaScript, HTML, CSS, UI.

Using Drag-n-Drop functionality in JavaScript

Khalimanenkov Andrey Sergeevich
Sholom-Aleichem Priamursky State University
Student

Abstract
This article discusses the use of Drag-n-Drop (grab and drop) in JavaScript web applications. Using this function helps to create interfaces for applications such as Google Keep, etc.
Keywords: slider, adaptive images, website development, JavaScript, HTML, CSS, UI.

Drag-n-Drop (в переводе с английского «хватай и бросай») часто используется в настольных операционных системах. Это тот принцип, по которому работает перемещение иконок с места на место на рабочем столе, перетаскивание файлов из одной папки в другую и открытие файлов в программах посредством перетаскивания первых на ярлыки последних. На сайтах же такой подход используется при загрузке аудио и видео файлов, а также изображений в социальные сети, облачные хранилища или площадки с развлекательным контентом. Но Drag-n-Drop можно использовать и в приложениях с отслеживанием задач.

Цель исследования – разобрать пример работы drag-n-drop в HTML5 и JavaScript на примере маленького списка задач.

Вопрос разработки интерфейсов при создании сайтов волнует некоторых исследователей и специалистов: Е. В. Пантелеева [1] отразила сущность разработки сайта с использованием языка разметки HTML, особенности таблицы стилей CSS и языка программирования JavaScript. Н. Д. Лушников и А. Д. Альтерман [2] рассмотрели основы (технические возможности) каскадных таблиц стилей CSS. Кроме того, освещены главные

преимущества и принцип работы каскадных таблиц. Н. О. Айдарбаев [3] раскрыл понятие адаптивного дизайна как одного из процессов веб разработки. Дал определения разновидностей фронтэнд фреймворков, используемых в веб разработке, и подробный анализ компонентов фреймворка Bootstrap. В. Е Селькин [4] в своей статье оценил эффективность модульного принципа на предмет временной задержки, которой обладают многосоставные приложения. Д.В. Ратов и др. [5] рассмотрели использование Drag-n-Drop при разработке интерфейса веб приложения.

Для реализации перетаскивания элементов страницы использовался чистый JavaScript [6] без сторонних библиотек и фреймворков, а также язык разметки HTML5 [7] и язык каскадных стилей CSS [8].

Для примера использования Drag-n-Drop создан небольшой список задач с тремя столбцами: «Начать», «В процессе» и «Готовы». В каждом столбце две строки. Изначально созданы четыре элемента задач (рис. 1):

1. Написать статью;
2. Написать курсовую;
3. Приготовить завтрак;
4. Постирать бельё.

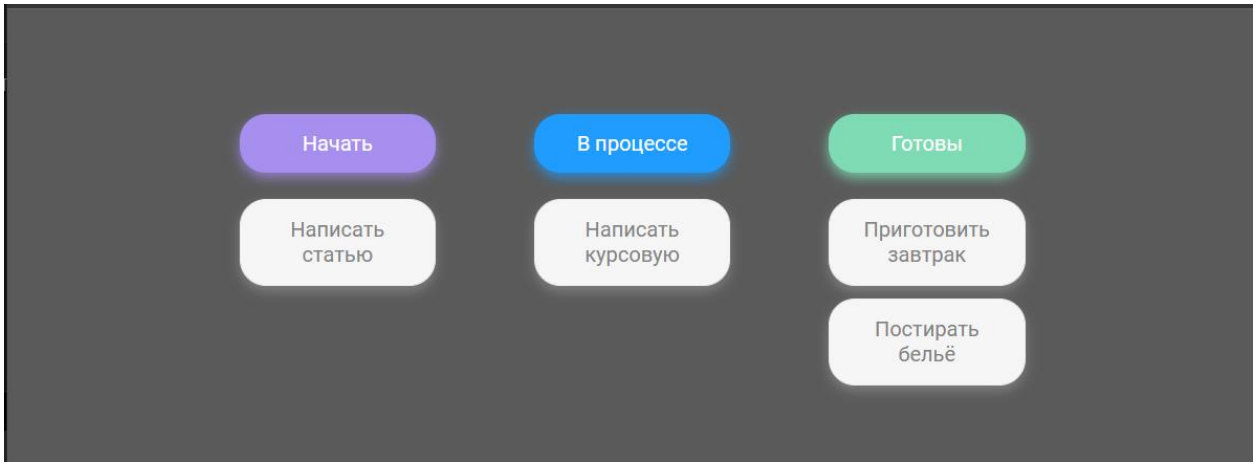

Рисунок 1 – Список задач

Каждый элемент задачи должен иметь атрибут draggable="true" в HTML разметке. Это даёт понять браузеру, что на этот элемент могут быть применены Drag-n-Drop события. Использоваться в этом веб-приложении будут пять из них:

1. Dragenter - событие вызывается, когда перетаскиваемый элемент или выделенный текст попадает в допустимую цель перетаскивания.
2. Dragleave - событие вызывается, когда перетаскиваемый элемент или выделенный текст покидает допустимую цель перетаскивания.
3. Drop - событие вызывается, когда перетаскиваемый элемент или выделенный текст сброшен в допустимую цель перетаскивания.
4. Dragstart - пользователь начал перетаскивать элемент.
5. Dragend - завершается перетаскивание. Например, отпускается кнопка мыши или нажимается Escape.

При начале перетаскивания элемент делается невидимым с помощью класса hide и его свойства display: none, чтобы было видно только его перетаскиваемую сущность (рис. 2).

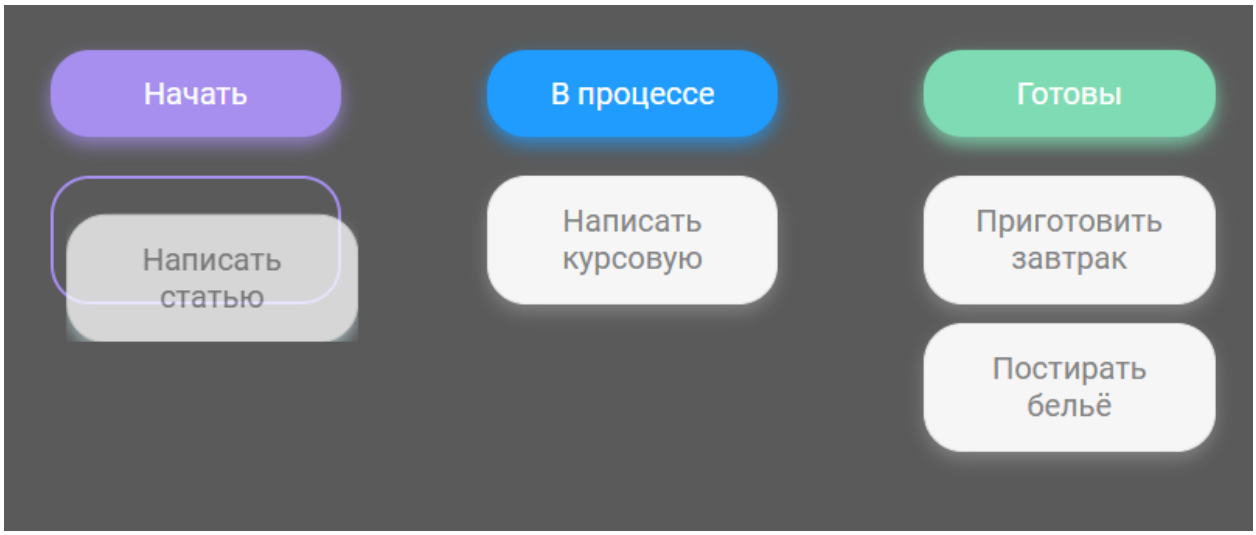

Рисунок 2 – Display: none у задачи в столбце «Начать»

В конце перетаскивания класс hide убирается и элемент снова видно. Когда перетаскиваемый объект входит в возможную область для вставки, то эта область выделяется цветной границей в зависимости от цвета столбца и получает класс hovered, который задаёт скругление границ (рис. 2). Во время покидания такой области класс hovered удаляется и очищаются стили элемента для возможной вставки задачи внутрь. Если мы на такой элемент перетаскиваем задачу, то она вставляется в тот столбец и строку, над которой находится курсор мыши пользователя (рис. 3 и 4).

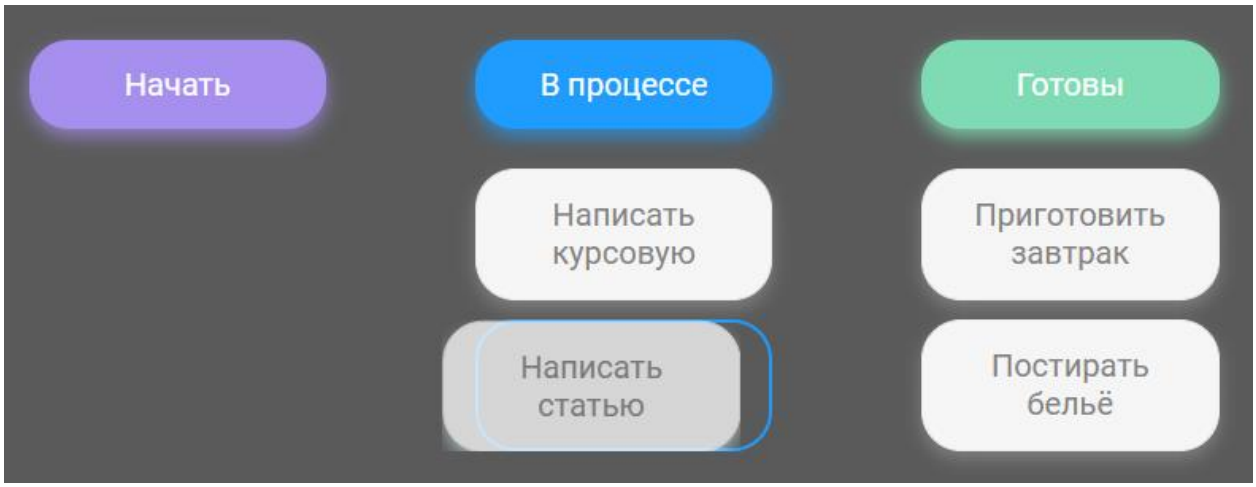

Рисунок 3 – Перемещение задачи над другим столбцом

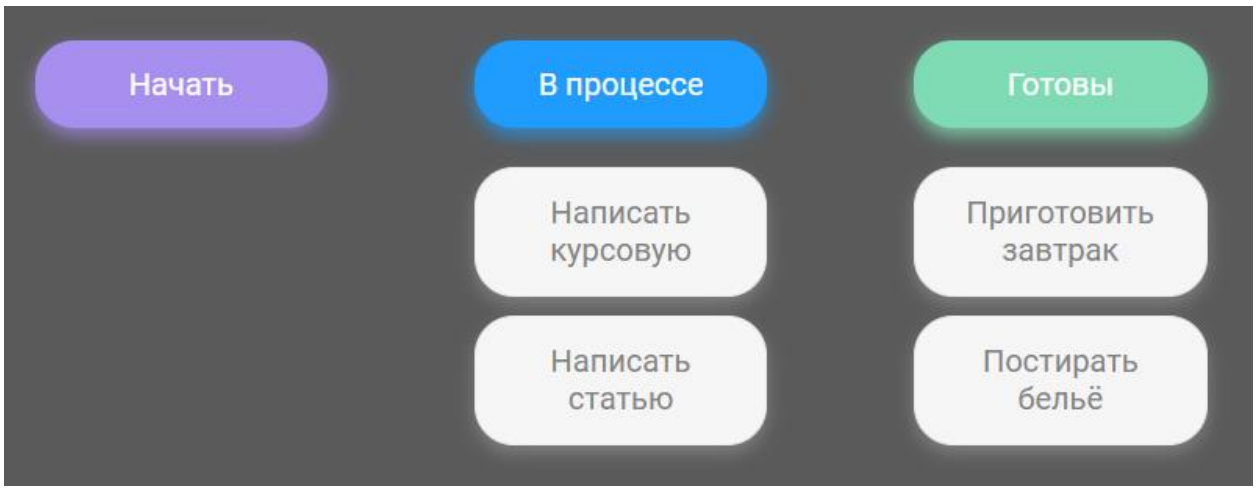

Рисунок 4 – Вставка задачи в новый столбец

HTML код тега body приложения:

```
      <body>
          <div class="container">

              <div class="column column-start">
                  <div class="col-header start">Начать</div>
                  <div class="column__items">
                      <div class="placeholder placeholder-start">
                                  <div  class="item"  draggable="true">Написать
статью</div>
                      </div>
                      <div class="placeholder placeholder-start"></div>
                  </div>
              </div>

              <div class="column column-progress">
                  <div class="col-header progress">В процессе</div>
                  <div class="placeholder placeholder-progress">
                      <div class="item" draggable="true">Написать курсовую</div>
                  </div>
                  <div class="placeholder placeholder-progress"></div>
              </div>

              <div class="column column-done">
                  <div class="col-header done">Готовы</div>
                  <div class="placeholder placeholder-done">
                    <div class="item" draggable="true">Приготовить завтрак</div>
                  </div>
                  <div class="placeholder placeholder-done">
                    <div class="item" draggable="true">Постирать бельё</div>
                  </div>
              </div>

          </div>
      </body>
```

CSS код:

```
@import
url('https://fonts.googleapis.com/css?family=Roboto&display=swap');

* {
  box-sizing: border-box;
}

body {
  font-family: 'Roboto', sans-serif;
  background-color: #5a5a5a;
  display: flex;
  padding-top: 5rem;
  justify-content: center;
  overflow: hidden;
  margin: 0;
}

.container {
  display: flex;
  width: 600px;
  justify-content: space-between;
}

.col-header {
  width: 150px;
  border-radius: 20px;
  padding: 0.8rem 1rem;
  color: #fff;
  text-align: center;
  margin-bottom: 20px;
}

.item {
  width: 150px;
  height: 66px;
  border: 1px solid #eee;
  box-shadow: 0 4px 9px rgba(198, 198, 198, 0.36);
  border-radius: 20px;
  background: #f7f6f7;
  padding: 0.8rem 1rem;
  color: #828282;
  text-align: center;
  cursor: grab;
}

.item:active {
  cursor: grabbing;
}

.placeholder {
  width: 150px;
  height: 66px;
  margin-bottom: 10px;
}

.start {
```

```
    background-color: #a68fee;
    box-shadow: 0 4px 9px rgb(166,143,238, 0.7);
  }

  .progress {
    background-color: #209cff;
    box-shadow: 0 4px 9px rgba(32,156,255, 0.7);
  }

  .done {
    background-color: #7fdbb3;
    box-shadow: 0 4px 9px rgba(136,240,195, 0.7);
  }

  .hold {
    background: #fff;
    box-shadow: 0 4px 9px -2px rgb(214, 246, 255);
    transform: scale(1.1);
  }

  .hide {
    display: none;
  }

  .hovered {
    border-radius: 20px;
  }
```

JS код приложения:

```
const $items = document.querySelectorAll('.item');
const $placeHolders = document.querySelectorAll('.placeholder');
let dragItem = 0;

for (const item of $items) {
    item.addEventListener('dragstart', dragStart);
    item.addEventListener('dragend', dragEnd);
}

function dragStart(event) {
    dragItem = event.target;
    event.target.classList.add('hold');
    setTimeout(() => event.target.classList.add('hide'), 0);
}

function dragEnd(event) {
    dragItem = 0;
    event.target.className = 'item';
}

for (const placeHolder of $placeHolders) {
    placeHolder.addEventListener('dragover', dragOver);
    placeHolder.addEventListener('dragenter', dragEnter);
    placeHolder.addEventListener('dragleave', dragLeave);
    placeHolder.addEventListener('drop', dragDrop);
}

function dragOver(event) {
```

```
        event.preventDefault();
    }

    function dragEnter(event) {
      event.target.classList.add('hovered');
        let color='';

        switch (event.target.classList[1]) {
            case 'placeholder-start':
                                                                color =
window.getComputedStyle(document.querySelector('.start')).backgroundColor;
                event.target.style.cssText = `border: 2px solid ${color};`;
                break;
            case 'placeholder-progress':
                                                                color =
window.getComputedStyle(document.querySelector('.progress')).backgroundColor;
                event.target.style.cssText = `border: 2px solid ${color};`;
                break;
            case 'placeholder-done':
                                                                color =
window.getComputedStyle(document.querySelector('.done')).backgroundColor;
                event.target.style.cssText = `border: 2px solid ${color};`;
                break;
        }
    }

    function dragLeave(event) {
        event.target.style.cssText = '';
        event.target.classList.remove('hovered');
    }

    function dragDrop(event) {
        event.target.append(dragItem);
        event.target.style.cssText = '';
        event.target.classList.remove('hovered');
    }
```

DOM и приложение после перемены задач местами (рис. 5 и 6).

```
▼<div class="container"> flex
  ▼<div class="column column-start">
     <div class="col-header start">Начать</div>
    ▼<div class="column__items">
     ▼<div class="placeholder placeholder-start" style>
        <div class="item" draggable="true">Приготовить завтрак</div>
      </div>
     ▼<div class="placeholder placeholder-start" style>
        <div class="item" draggable="true" style>Постирать бельё</div>
      </div>
    </div>
  </div>
  ▼<div class="column column-progress">
     <div class="col-header progress">В процессе</div>
     <div class="placeholder placeholder-progress" style></div>
    ▼<div class="placeholder placeholder-progress" style>
      <div class="item" draggable="true">Написать курсовую</div>
    </div>
  </div>
  ▼<div class="column column-done">
     <div class="col-header done">Готовы</div>
     <div class="placeholder placeholder-done" style></div>
    ▼<div class="placeholder placeholder-done" style>
      <div class="item" draggable="true" style>Написать статью</div>
    </div>
  </div>
</div>
```

Рисунок 5 – Изменённое DOM дерево после переменны задач местами

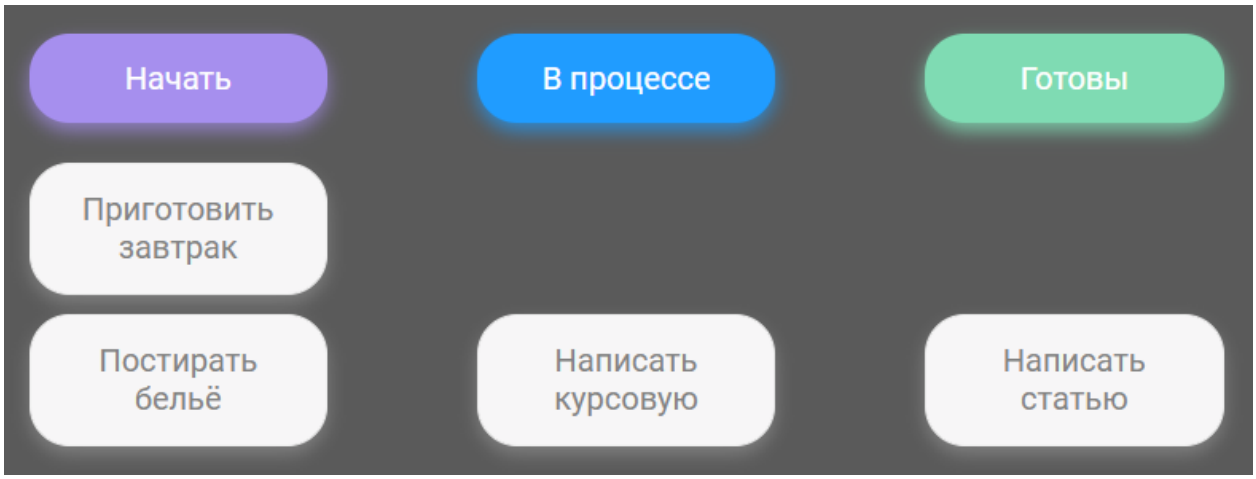

Рисунок 6 – Изменённое состояние списка задач

Таким образом была использована функциональность HTML5 и JavaScript для перетаскивания элементов DOM дерева с помощью мыши. Эта технология может быть использована при создании списка задач, перетаскивании файлов в облачное хранилище и т.д.

Библиографический список

1. Пантелеева Е. В. Разработка сайта с использованием языка разметки HTML, таблицы стилей CSS и языка программирования JavaScript. // Информационные системы и технологии в образовании, науке и бизнесе. 2020. С. 94-96
2. Лушников Н. Д., Альтерман А. Д. Основы каскадных таблиц стилей (CSS). // Наука и образование: новое время. 2019. №. 1. С. 69-72.
3. Айдарбаев Н. О. Адаптивный дизайн веб-сайта с использованием фронтэнд-фреймворка Bootstrap // Молодой ученый. 2018. №. 21. С. 115-119
4. Селькин В. Е. Временной анализ модульной сборки пользовательского интерфейса на JavaScript //Студенческая наука XXI века. 2016. №. 2-1. С. 279-281.
5. Ратов Д. В. и др. Object adaptation of Drag and Drop technology for web-system interface components //ВІСНИК СХІДНОУКРАЇНСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ імені Володимира Даля. 2021. №. 4 (268). С. 7-12.
6. JavaScript URL: https://ru.wikipedia.org/wiki/JavaScript (дата обращения: 03.01.2022).
7. HTML URL: https://ru.wikipedia.org/wiki/HTML (дата обращения: 03.01.2022).
8. CSS URL: https://ru.wikipedia.org/wiki/CSS (дата обращения: 03.01.2022).