Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Брянский государственный инженерно-технологический университет»

Кафедра «Информационные технологии»

**Лабораторная работа №1**

по дисциплине « SPA и MPA фреймворки»

**«»**

Выполнил ст. гр. ИСТм-101

Алешин. К.В.

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г.

Руководитель

Н.А. Афанасьева

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г.

Брянск 2023

Оглавление

[1. Задача 5. Товар в корзину 3](#_Toc126008910)

[2. Задача 6. Картинная галерея 5](#_Toc126008911)

[3. Задача 7. Тайное послание 8](#_Toc126008912)

[4. Ответь на вопросы 14](#_Toc126008913)

# Задача 5. Товар в корзину

Добавь товар в корзину при нажатии кнопки + число увеличивается, а при нажатии – число уменьшается.

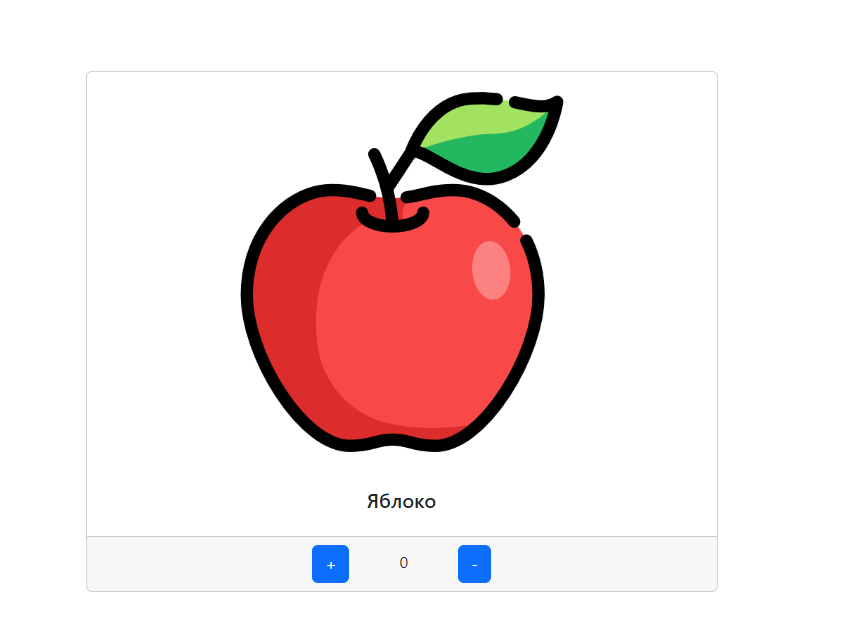


Рис. 1 Начально состояние

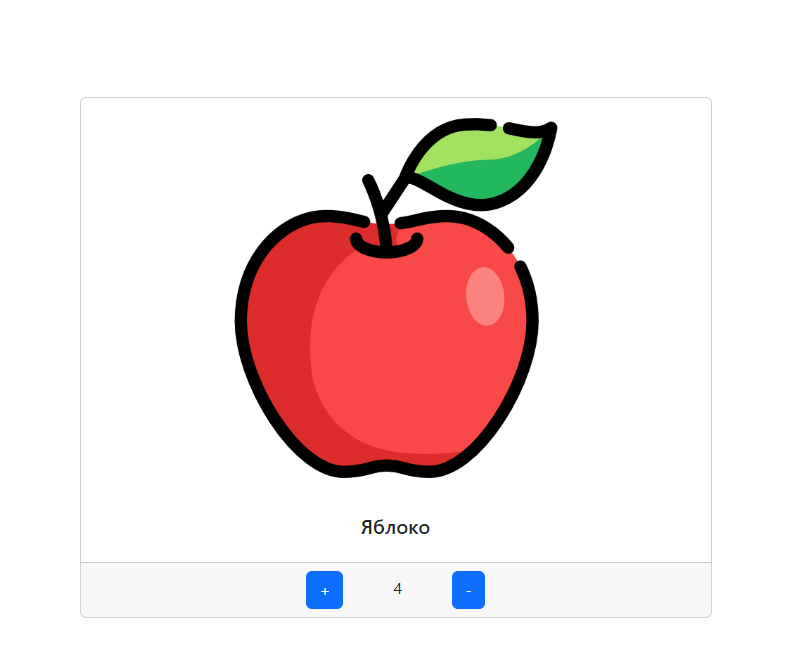


Рис. 2 После нажатия клавиш

Исходный js код

|  |
| --- |
| let buttonPlusNode=document.querySelector(`#plus`)  let buttonMinusNode=document.querySelector(`#minus`)  let amountCountNode=document.querySelector(`#amount`) |

Листинг 1Блок инициализации переменных

В данном блоке мы находим по id элементы, с которыми необходимо работать, это кнопка плюс, минус, и поле в которое мы выводи результат.

|  |
| --- |
| let count=0  amountCountNode.innerHTML= count |

Листинг 2 Присвоение полю первичного значения

Тут мы выводим в поле первично значение которое равно 0.

|  |
| --- |
| buttonPlusNode.addEventListener(`click`, function(){  count+=1  amountCountNode.innerHTML= count  })  buttonMinusNode.addEventListener(`click`, function(){  if(count-1>=0)  {  count-=1  amountCountNode.innerHTML= count  }  }) |

Листинг 3 Обработчики событий

И последним действием добавим 2 обработчика событий, по 1 на каждую кнопку, для увеличения числа или уменьшения количества товара в корзине.

# Задача 6. Картинная галерея

Создай галерею из 3-х рисунков, которые можно переключать по нажатию на кнопки.

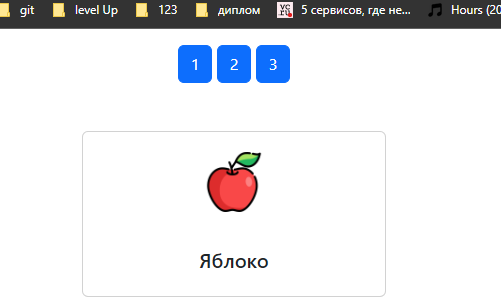


Рис. 3 Первое состояние

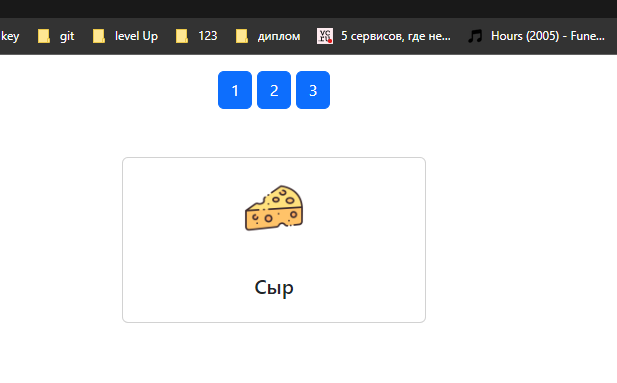


Рис. 4 Второе состояние

Во-первых определим html верстку для карточек с картинками которые мы будем менять, шаблоны.

|  |
| --- |
| let template1 = `<div class="col-3">  <div class="card text-center">  <img src="assets/apple.png" class="card-img-top" alt="...">  <div class="card-body">  <h5 class="card-title text-center">Яблоко</h5>  </div>  </div>  </div>  `;  let template2 = `<div class="col-3">  <div class="card text-center">  <img src="assets/cheese.png" class="card-img-top" alt="...">  <div class="card-body">  <h5 class="card-title text-center">Сыр</h5>  </div>  </div>  </div>  `;  let template3 = `<div class="col-3">  <div class="card text-center">  <img src="assets/eggs.png" class="card-img-top" alt="...">  <div class="card-body">  <h5 class="card-title text-center">Яйца</h5>  </div>  </div>  </div>  `; |

Листинг 4 Шаблоны

Теперь определим блок для поиска элементов по id и настроим стан драное поведение при загрузке страницы.

|  |
| --- |
| let rowNode = document.querySelector(`#row`);  let button1Node = document.querySelector(`#button1`);  let button2Node = document.querySelector(`#button2`);  let button3Node = document.querySelector(`#button3`);  rowNode.innerHTML = template1 |

Листинг 5 Определение переменных

Теперь при загрузки станицы мы будем видеть плашку с картинкой яблока.

Теперь добавив обработчики событий нажатия на кнопки.

|  |
| --- |
| button1Node.addEventListener(`click`, function()  {  rowNode.innerHTML = template1  })  button2Node.addEventListener(`click`, function()  {  rowNode.innerHTML = template2  })  button3Node.addEventListener(`click`, function()  {  rowNode.innerHTML = template3  }) |

Листинг 6 Обработчики событий

Теперь при нажатии кнопки будут меняться картики согласно установленному порядку, 1- яблоко, 2- сыр, 3- яйца.

# Задача 7. Тайное послание

Создайте свой телеграфный аппарат Морзе. Это простой интерфейс из четырех кнопок и элемента для вывода сообщения. Каждая кнопочка добавляет свой символ к сообщению, а последняя все удаляет. Используйте символы – и \*, а для паузы используй пробел.



Рис. 5 Интерфейс

Сделаем html верстку сайта согласно заданию и разместим на нем 5 функциональных кнопок, поля для отображения вводимых символов и поле с выходным результатам дешифровки.

|  |
| --- |
| <!DOCTYPE html>  <html lang="ru">  <head>  <meta charset="UTF-8" />  <meta name="viewport" content="width=device-width, initial-scale=1.0" />  <meta http-equiv="X-UA-Compatible" content="ie=edge" />  <title>Азбука Морзе</title>  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet"  integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65" crossorigin="anonymous" />  <script src="js/index.js" defer></script>  </head>  <body>  <div class="container vh-100 p-3">  <div class="row justify-content-center align-items-center mb-5">  <div class="col-20 text-center">  <button class="btn btn-primary" id="buttonMinus">-</button>  <button class="btn btn-primary" id="buttonStar">\*</button>  <button class="btn btn-primary" id="buttonSpace">Пауза</button>  <button class="btn btn-primary" id="buttonClean">Очистка</button>  <button class="btn btn-primary" id="buttonDecryptedMessage">Расшифровать</button>  </div>    <div class="row justify-content-center align-items-center">  <samp class="d-flex justify-content-center align-items-center input-group-text" id="encryptedMessage"></samp>  <span class="d-flex justify-content-center input-group-text" id="decryptedMessage"></span>  </div>  <div class="card text-center">  <img src="assets/morse.png" class="card-img-top" alt="...">  </div>  </div>  </div>  </body>  </html> |

Листинг 7 Исходный код html верстки

Теперь приступим к реализации js скрипта.

Определим набор переменных, которые будут отвечать за элементы управления.

|  |
| --- |
| let buttonMinusNode=document.querySelector(`#buttonMinus`)  let buttonStarNode=document.querySelector(`#buttonStar`)  let buttonSpaceNode=document.querySelector(`#buttonSpace`)  let buttonCleanNode=document.querySelector(`#buttonClean`)    let encryptedMessageNode=document.querySelector(`#encryptedMessage`)  let decryptedMessageNode=document.querySelector(`#decryptedMessage`)  let buttonDecryptedMessageNode=document.querySelector(`#buttonDecryptedMessage`) |

Листинг 8 Определение переменных.

Теперь определим словарь азбуки Морзе.

|  |
| --- |
| // алфавит Морзе  const MORSE\_CODE = {  "\*-":"А",  "-\*\*\*":"Б",  "\*--":"В",  "--\*":"Г",  "\*":"Е",  "\*\*\*-":"Ж",  "--\*\*":"З",  "\*\*":"И",  "\*---":"Й",  "-\*-":"К",  "\*-\*\*":"Л",  "--":"М",  "---":"О",  "\*--\*":"П",  "\*-\*":"Р",  "\*\*\*":"С",  "-":"Т",  "\*\*-":"У",  "\*\*-\*":"Ф",  "\*\*\*\*":"Х",  "-\*-\*":"Ц",  "---\*":"Ч",  "----":"Ш",  "--\*-":"Щ",  "-\*\*-":"Ь,Ъ",  "-\*--":"Ы",  "\*\*-\*\*":"Э",  "\*\*--":"Ю",  "\*-\*-":"Я", //  "\*----":"1",  "\*\*---":"2",  "\*\*\*--":"3",  "\*\*\*\*-":"4",  "\*\*\*\*\*":"5",  "-\*\*\*\*":"6",  "--\*\*\*":"7",  "---\*\*":"8",  "----\*":"9",  "-----":"0",  "\*\*\*\*\*\*":".",  "\*-\*-\*-":",",  "-\*-\*-\*":";",  "---\*\*\*":":",  "\*\*--\*\*":"?",  "--\*\*--":"!",  "-\*\*\*\*-":"-",  "\*-\*\*-\*":"<<",  "-\*--\*-":"(",  "-\*\*-\*":"/"  }; |

Листинг 9 Определение алфавита Морзе

Добавим обработчики событий на элементы управления

|  |
| --- |
| // добавляем "\*" в поле с зашифрованным сообщением  buttonStarNode.addEventListener(`click`, function(){  encryptedMessageNode.innerHTML+=`\*`  })  // добавляем " "(пробел) в поле с зашифрованным сообщением для раздения символов  buttonSpaceNode.addEventListener('click', function()  {  encryptedMessageNode.innerHTML+=` `  })  // добавляем "-" в поле с зашифрованным сообщением  buttonMinusNode.addEventListener(`click`, function(){  encryptedMessageNode.innerHTML+=`-`  })  // очищаем поле ввода сообщениЯ  buttonCleanNode.addEventListener(`click`, function()  {  encryptedMessageNode.innerHTML=``  decryptedMessageNode.innerHTML=``  })  // декодируем закодированне сообщение  buttonDecryptedMessageNode.addEventListener(`click`, function()  {  decryptedMessageNode.innerHTML=decodeMorse(encryptedMessageNode.innerHTML)  }) |

Листинг 10 Обработчики событий

Реализуем функцию декодирования сообщения.

|  |
| --- |
| // функция декодирования закодированного сообщения  decodeMorse = function( morseCode ) {  return morseCode  .split(' ') // get word code, 3 spaces apart  .map(word => word  .split(' ') // получаем код символа  .map(symbol => MORSE\_CODE[symbol]) // декодируем символ  .join('') // объединение символов в слово  )  .join(' ') // добавляем пробел между словами  .trim()  } |

Листинг 11 Функция декодирования закодированного сообщения

На этом разболтка скрипта закончена.

# Ответь на вопросы

1. Какое ключевое слово(а) объявляет(ют) переменные? let, var(устарела)

2. Какое свойство управляет содержимым элемента в DOM? Ответ: document.

3. Ставиться ли ; в конце строк в JS? Ответ: да, но не обязательн.

4. Какое сообщение закодировано? \*--\* \*-\* \*\* \*-- \* - \*-\*-\*- -- \*\* \*-\* --\*-- Ответ: ПРИВЕТ,МИР