

ФГБОУ ВО “Чувашский государственный университет им. И. Н. Ульянова”

Факультет: ИВТ

Кафедра: Вычислительной техники

Предмет: Web-программирование

Лабораторная работа №5

Динамические WEB-страницы. Обработка событий.

Выполнил: студент группы ИВТ-41-20

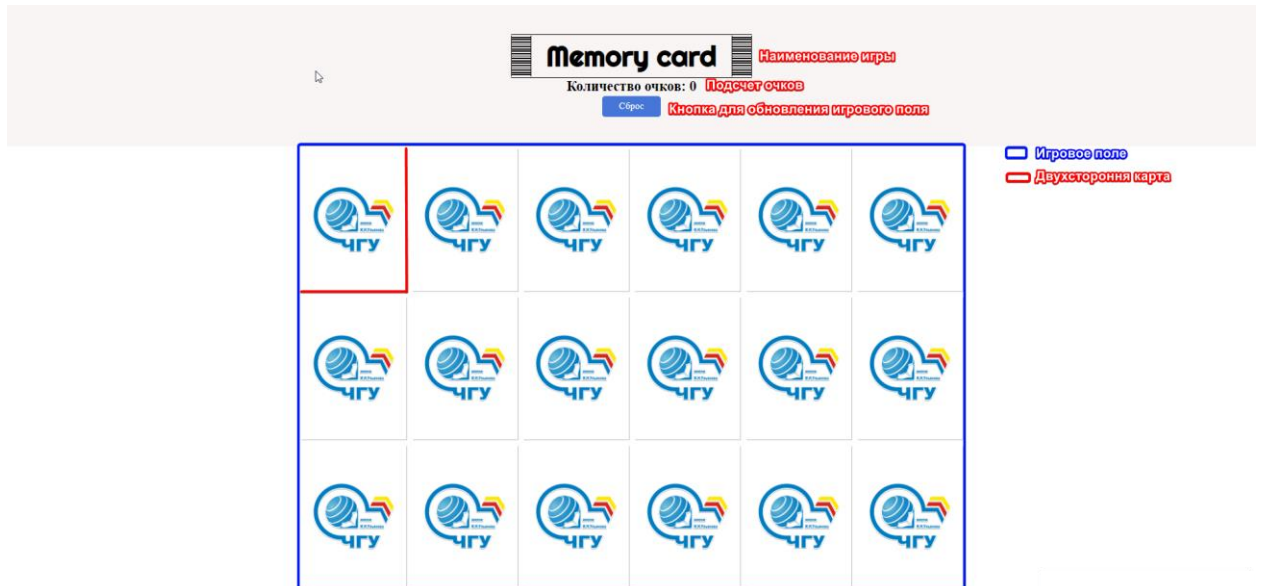
Галкин Дмитрий

Проверила: Перова Н.В

Цель работы

Изучить принципы разработки динамических web-страниц с использованием клиентских сценариев JavaScript и объектной модели документа. Научиться создавать кросс-браузерные html-страницы. Познакомиться с уровнями и особенностями API и обработкой событий DOM различных уровней.

Задание



Объявление данных

```
/*
 * Создаем список карточек всех уровней
 */
let list_cards = new Map();

list_cards.set("Christmas", "images/cards/christmas-gift.svg");
list_cards.set("Coffee", "images/cards/coffee.svg");
list_cards.set("Dog", "images/cards/dog.svg");
list_cards.set("Dumbbell", "images/cards/dumbbell.svg");
list_cards.set("Followers", "images/cards/followers.svg");
list_cards.set("Password", "images/cards/forget-password.svg");
list_cards.set("Healthy", "images/cards/healthy.svg");
list_cards.set("Ise", "images/cards/ice-cream.svg");
list_cards.set("Map", "images/cards/map-location.svg");
list_cards.set("Merry", "images/cards/merry-christmas.svg");
list_cards.set("Pocket", "images/cards/pocket-knife.svg");
list_cards.set("Sanitizer", "images/cards/sanitizer.svg");
list_cards.set("Santa", "images/cards/santa.svg");
list_cards.set("Sneezing", "images/cards/sneezing.svg");
list_cards.set("Tea", "images/cards/tea.svg");
list_cards.set("Wallet", "images/cards/wallet.svg");
list_cards.set("Wash", "images/cards/wash-hand.svg");
list_cards.set("Work1", "images/cards/work-from-home-1.svg");
list_cards.set("Work2", "images/cards/work-from-home-2.svg");
list_cards.set("Working", "images/cards/working.svg");

/*
 * Словарь наименования всех карточек
 */
const list_cards_names = {
  1: "Christmas",
  2: "Coffee",
  3: "Dog",
  4: "Dumbbell",
  5: "Followers",
  6: "Password",
  7: "Healthy",
  8: "Ise",
  9: "Map",
  10: "Merry",
  11: "Pocket",
  12: "Sanitizer",
  13: "Santa",
  14: "Sneezing",
  15: "Tea",
  16: "Wash",
  17: "Wallet",
  18: "Work1",
  19: "Work2",
  20: "Working"
}
```

<pre> <div class="nav"> <h1>Memory card</h1> <h2 class="point_text"></h2> <div class="functional"> <button class="reset" onClick="window.location.reload();">Сброс</button> </div> </div> <div class="field"> <section class="chuvsu-game"> </section> </div> <script src="JS/script.js"></script> </pre>	<p>Nav – блок навигации (наименование игры, динамический подсчет очков, кнопка сброса)</p> <p>Field – секция динамического заполнения игрового поля картами.</p>
<pre> countPoint = 0; // Кол-во очков пользователя const sizeCards = 9; // Кол-во пар карт на поле const point = document.querySelector('.point_text'); // Вывод очков function textPoint() { point.textContent = "Количество очков: " + countPoint; } textPoint(); </pre>	<p>textPoint – Вывод очков</p>
<pre> const section = document.querySelector('.chuvsu-game'); /* * Создание карточек */ function creatingCard(card, index) { card.classList.add("chuvsu-card"); card.setAttribute("data-framework", list_cards_names[index].toLowerCase()); let image_1 = document.createElement('img'); image_1.classList.add("front-face"); image_1.setAttribute("src", list_cards.get(list_cards_names[index])); image_1.setAttribute("alt", list_cards_names[index]); card.appendChild(image_1); let image_2 = document.createElement('img'); image_2.classList.add("back-face"); image_2.setAttribute("src", "images/chuvsu.svg"); image_2.setAttribute("alt", "Memory card"); card.appendChild(image_2); return card; } </pre>	<p>В начале кода обращаемся к секции игрового поля.</p> <p>creatingCard – Создание карточек</p>
<pre> /* * Добавление карточек на поле */ function addCards () { for (let i = 1; i ≤ sizeCards; i++) { let card = creatingCard(document.createElement('div'), i); section.appendChild(card); section.appendChild(card.cloneNode(true)); } } addCards(); </pre>	<p>addCards – Добавление карточек на поле</p>

```
// Создаем массив карточек
const cards = document.querySelectorAll('.chuvsu-card');

let isFlippedCard = false; // Нажатие на 2 карточки
let lockBoard = false; // Предотвращает переворот других карточек
let firstCard, secondCard;

/*
 * Функция отвечает за переворот карточки
 */
function flipCard() {
    if (lockBoard) return;
    if (this === firstCard) return;

    this.classList.add('flip'); // При нажатии на карточку добавляет класс переворота

    // Если нажата карточка, контекст этой карточки передается в firstCard
    if (!isFlippedCard) {
        isFlippedCard = true;
        firstCard = this;
        return;
    }

    // Получаем контекст второй карточки, сразу же после первой
    secondCard = this;
    checkForMatch(); // запускаем проверку
}
```

Объявляем данные для работы с карточками игрового поля.

flipCard – Отвечает за подключение стиля переворота для нажатой карточки
Работа введется через неявный контекст **this**.

```
/*
 * Проверка на совпадение первой и второй карточки
 */
function checkForMatch() {
    let isMatch = firstCard.dataset.framework === secondCard.dataset.framework;
    isMatch ? disableCards() : enableCards();
}

/*
 * Карточки одинаковые, события сбросятся чтобы их больше нельзя было переворачивать
 */
function disableCards() {
    firstCard.removeEventListener('click', flipCard);
    secondCard.removeEventListener('click', flipCard);

    countPoint++;
    textPoint();

    if (countPoint === sizeCards) { win(); }

    resetBoard(); // сброс переменных
}
```

checkForMatch – Проверка на совпадение первой и второй карточки

disableCards – Сбрасывания событий нажатий, чтобы нельзя было переворачивать карточки, для **return true** из функции **checkForMatch**

<pre> /* * Карточки разные, сбрасываются классы переворота и переменные */ function enableCards() { lockBoard = true; setTimeout(() => { firstCard.classList.remove('flip'); secondCard.classList.remove('flip'); resetBoard(); }, 1500); } /* * Сброс всех значений */ function resetBoard() { [isFlippedCard, lockBoard] = [false, false]; [firstCard, secondCard] = [null, null]; } </pre>	<p>enableCards – Если карточки разные, то они переходят в исходное состояние с событием нажатия</p> <p>resetBoard – Сброс всех значений переменных</p>
<pre> /* * Рандомно расставляет карточки на поле */ (function randomBoardCards() { cards.forEach(card => { card.style.order = Math.floor(Math.random() * 12); }); })(); /* * Добавляем на каждую карточку ожидания нажатия на неё */ cards.forEach(card => card.addEventListener('click', flipCard)); </pre>	<p>randomBoardCards – Рандомно расставляет карточки на игровом поле</p> <p>card.addEventListener – каждой карточке добавляется событие нажатия</p>

Вывод

Изучил теоретические сведения о том, как работать с событиями и из лабораторной работы №4 попрактиковал работу с контекстом данных. Также научился создавать динамические web-страницы.