Министерство образования Республики Беларусь  
Учреждение образования  
«Белорусский государственный университет информатики и радиоэлектроники» Филиал  
«Минский радиотехнический колледж»

Учебная дисциплина «Программные средства создания Internet-приложений»

**Отчет**

по выполнению лабораторной работы

«Использование объектов HTML и объектной модели документа»

Выполнила Гончаров П.В.

Проверила Терешко О.И.

Минск

2020

Лабораторная работа № 24

Номер учебной группы: 7K2492.

Фамилия, инициалы учащегося: Гончаров П.В.

Дата выполнения работы: 07.02.2021

Тема работы: «Использование разных моделей событий. Остановка всплытия и перехват событий»

Цель работы: Формирование умений назначения обработчиков событий в сценариях JavaScript, использования разных моделей событий.

Результат выполнения работы:

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Lab 24</title>

  <script src="script.js"></script>

</head>

<body>

  <input type="text" oninput="task1(this)">

  <p id="result"></p>

  <a href="https://yandex.by/" onclick="task2(event)" id="link">ссылка задания 2</a>

  <div id="elements">

    <p id="element1">элемент1</p>

    <p id="element2">элемент2</p>

    <p id="element3">элемент3</p>

    <p id="element4">элемент4</p>

    <p id="element5">элемент5</p>

  </div>

  <button id="add-event" onclick="task3()">Добавить событие задание 3</button>

</body>

</html>

window.onload = function() {

  //часть Задания 1

  resultElement = document.querySelector("#result");

  //часть Задания 2

  const link = document.querySelector("#link");

  link.addEventListener("click", () => {

    console.log("link click");

  });

  link.addEventListener("mouseover", () => {

    console.log("link mouseover");

  });

  link.addEventListener("mouseout", () => {

    console.log("link mouseout");

  });

  link.addEventListener("blur", () => {

    console.log("link blur");

  });

  link.addEventListener("focus", () => {

    console.log("link focus");

  });

}

//Задание 1

let resultElement = null;

function task1(element) {

  resultElement.innerHTML = element.value;

}

//Задание 2

function task2(event) {

  event.preventDefault();

}

//Задание 3

let eventNames = [];

function task3() {

  eventNames = prompt("Введите имя события").split(",").map(eventName => {

    return eventName.trim();

  })

  alert("Выберите элемент");

  const elements = document.querySelector("#elements");

  elements.addEventListener("mouseover", addElementBorder);

  elements.addEventListener("mouseout", removeElementBorder, { capture: true });

  const removeElementsEventListeners = (event) => {

    const { target } = event;

    if (checkIsElementsChild(target)) {

      alert("Элемент выбран");

      removeElementBorder(event);

      eventNames.forEach(eventName => {

        target.addEventListener(eventName, () => {

          console.log(eventName);

        })

      });

      elements.removeEventListener("mouseover", addElementBorder);

      elements.removeEventListener("mouseout", removeElementBorder);

    }

  }

  elements.addEventListener("click", removeElementsEventListeners, { once: true });

}

function checkIsElementsChild(node) {

  return node?.parentNode.id === "elements";

}

function addElementBorder(event) {

  const { target } = event;

  event.stopPropagation();

  if (checkIsElementsChild(target)) {

    target.style.border = "2px solid black";

    target.style.backgroundColor = "skyblue";

    target.style.cursor = "pointer";

    target.style.width = "100px";

  }

}

function removeElementBorder(event) {

  const { target } = event;

  if (checkIsElementsChild(target)) {

    target.style.border = "";

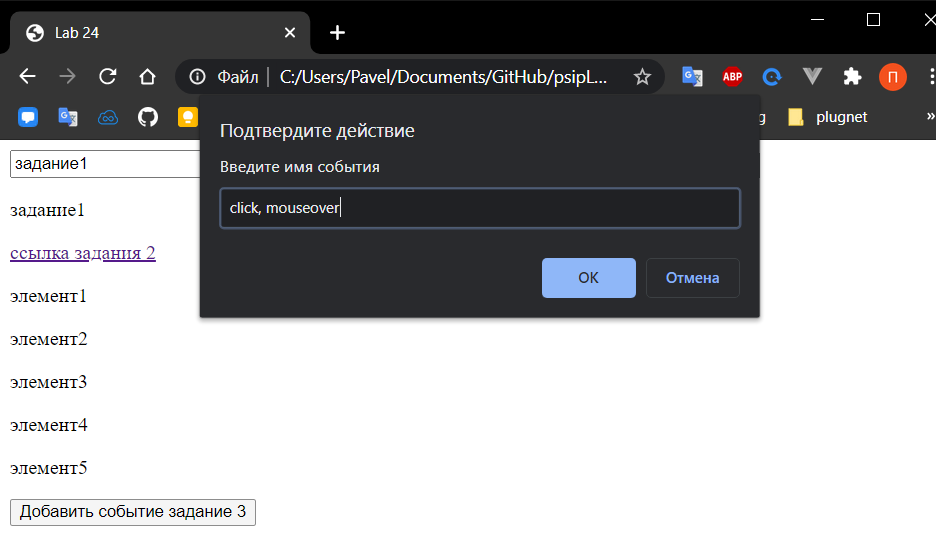
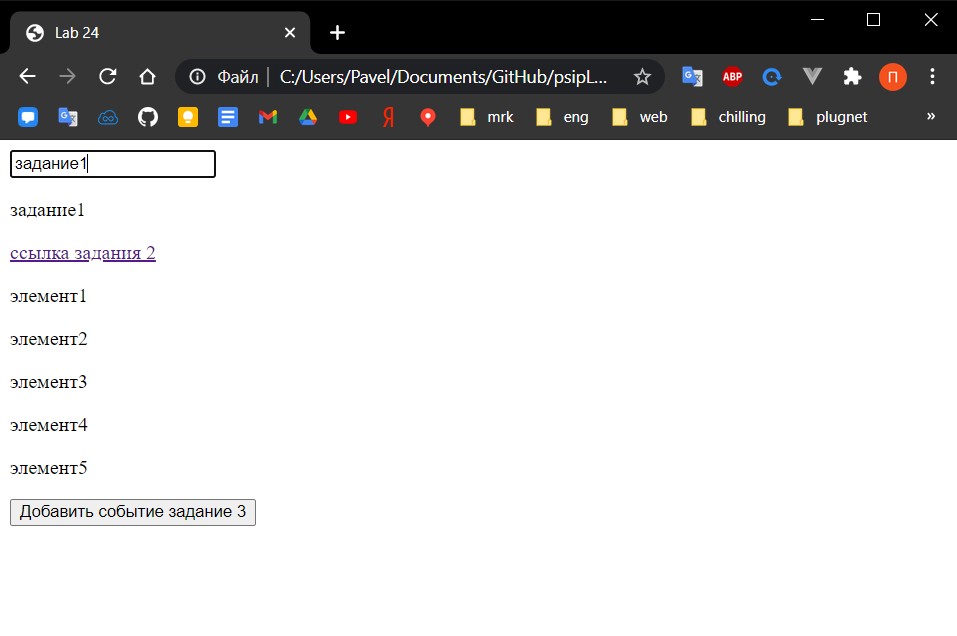
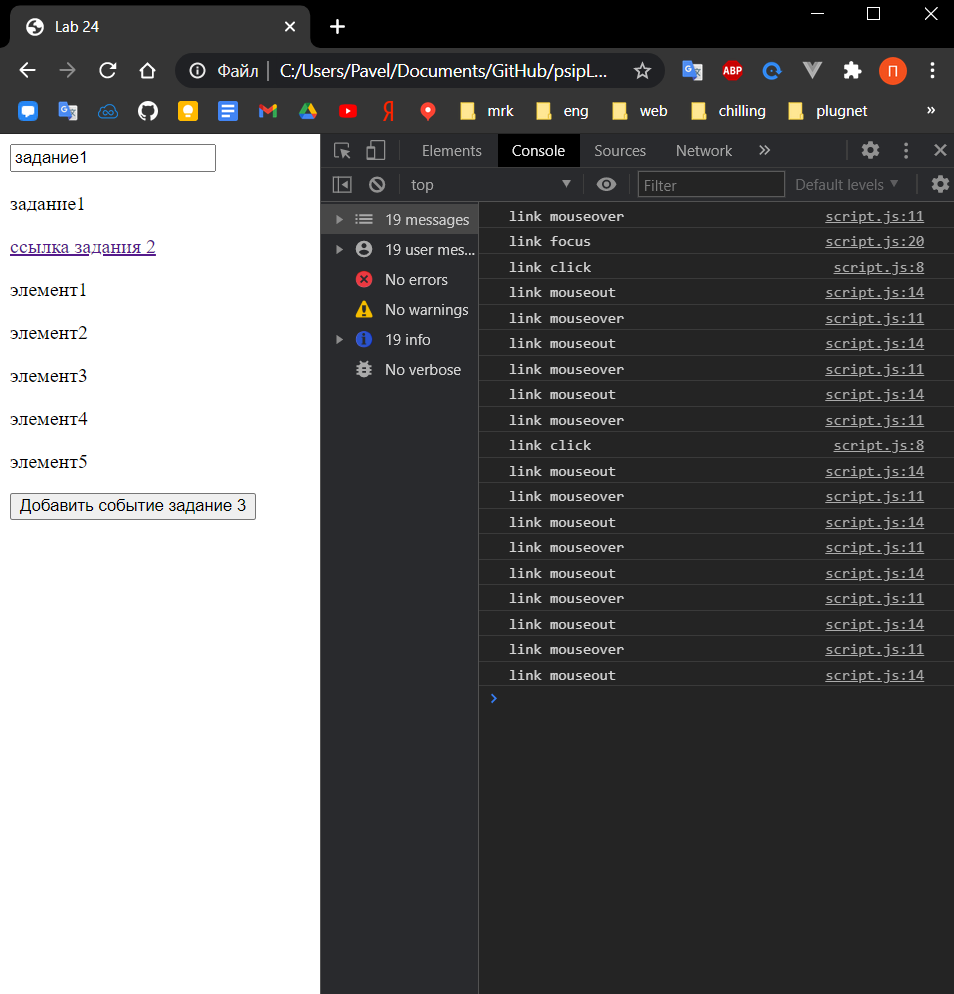
    target.style.backgroundColor = "";

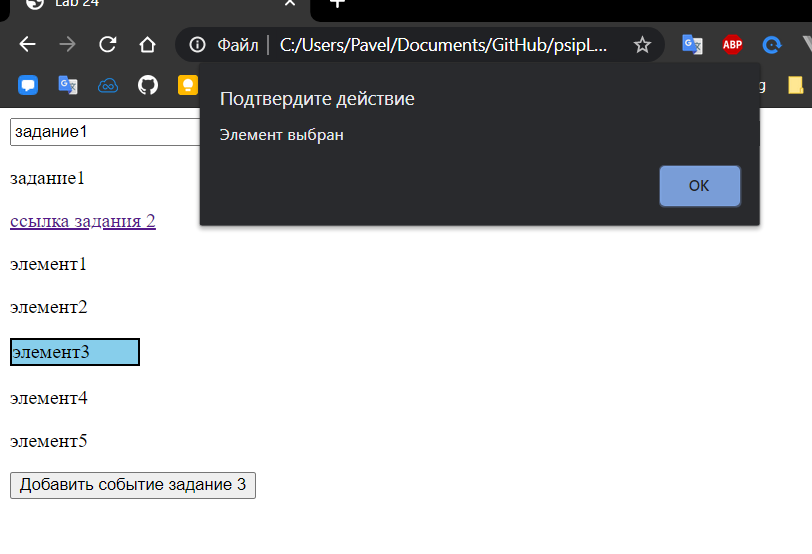
    target.style.cursor = "";

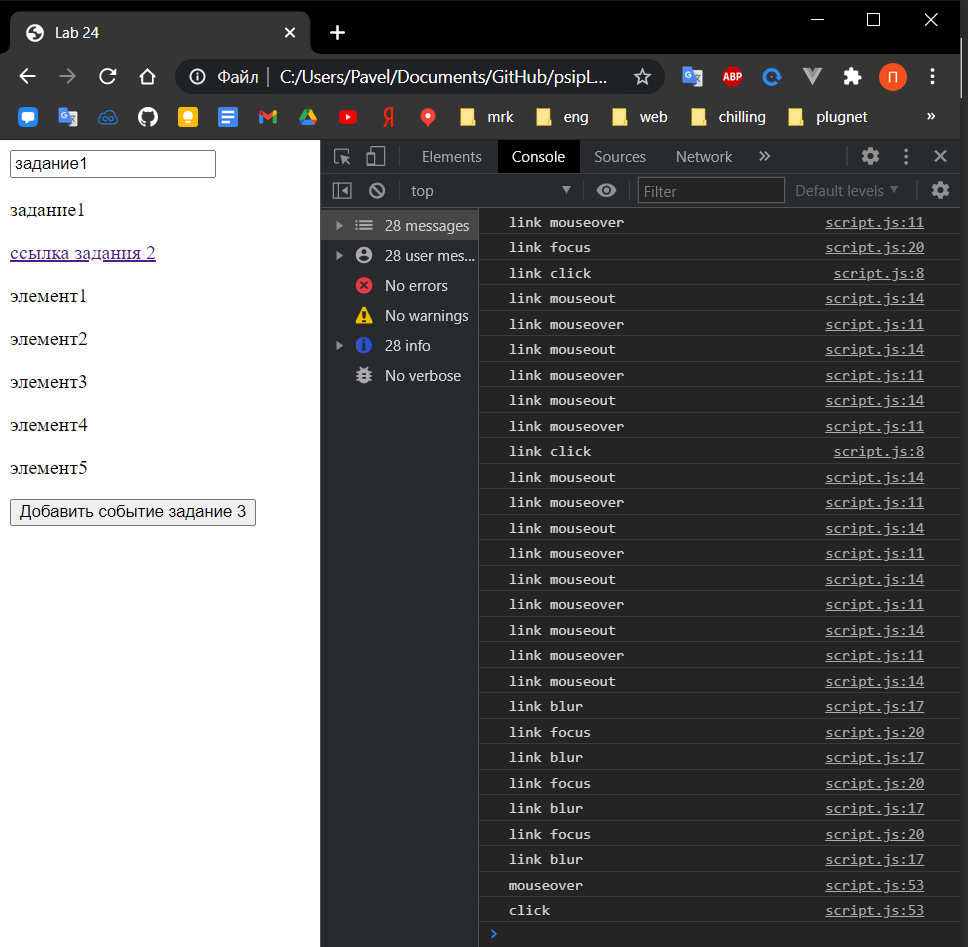
    target.style.width = "";

  }

}







Контрольные вопросы и задания

1. Что представляет собой событие?

Событие в JavaScript – это определённое действие, которое вызвано либо пользователем, либо браузером.

2. Для каких целей могут быть использованы события на веб-странице?

К ним относятся движение мыши, нажатия на кнопки мыши и клавиатуры,

наведение фокуса на элемент, изменение значения в каком-нибудь текстовом поле, изменение размеров окна браузера и так далее.

3. Назовите способы задания обработчиков событий. В чем отличие?

1. Самый простой способ задания обработчика – это использование атрибутов событий прямо в HTML-коде.  
2. Ещё один способ установки обработчика события – это прямое использование DOM свойств, то есть сразу записывать обработчик в свойство элемента.  
3. Для установки и удаления любого количества обработчиков на одно событие используются методы addEventListener() и removeEventListener()

4. Для чего предназначен объект Event, и как его получить?

Объект event предназначен для получения различной информации о событии в обработчике события. Т.е. позволяет получить элемент, который вызвал обработчик события; элемент, который сгенерировал событие; определить какая была нажата кнопка мыши (для событий связанных с мышью) и многое другое.

Объект event в соответствии со стандартом всегда передаётся обработчику события через первый параметр

5. Перечистите свойства объекта Event.

* type - строка, содержащее имя события.
* target - DOM-элемент, который сгенерировал событие.
* currentTarget - DOM-элемент, который вызвал обработчик события.
* eventPhase - число, показывающее на каком этапе произошло событие (1 - этапе погружения (перехвата), 2 - на цели, 3 - на этапе всплытия).
* timestamp - число (дата), когда произошло событие
* bubbles - возвращает логическое значение, указывающее может ли данное событие всплывать
* defaultPrevented - проверяет можно ли вызвать метод preventDefault() для данного события.
* view - возвращает ссылку на объект window, в котором произошло событие.

6. Опишите назначение методов объекта Event.

* preventDefault() - отменить стандартное действие браузера, если это конечно возможно.
* stopPropagation() - предотвратить всплытие события (пузырька)