Лабораторная работа 17

Обработка событий в JS

Ход работы

Контрольные вопросы

1. Событие — это действие пользователя на сайте или в мобильном приложении, которое можно отслеживать без привязки к просмотру страниц или экранов: скачивание файлов, клик по кнопке, отправка формы, скроллинг страницы, просмотр видеоролика, взаимодействие с баннером или чатом и другие.
2. Ошибка (Error) — событие регистрируется в случае возникновения серьезного события (такого как потеря данных или функциональных возможностей). Событие данного типа будет зарегистрировано, если невозможно загрузить какой-либо из сервисов в ходе запуска системы.

 Предупреждение (Warning) — событие не является серьезным, но может привести к возникновению проблем в будущем. Например, если недостаточно дискового пространства, то в журнал будет занесено предупреждение.

Сведения (Information) — значимое событие, которое свидетельствует об успешном завершении операции приложением, драйвером или сервисом. Такое событие может, например, зарегистрировать успешно загрузившийся сетевой драйвер.

Аудит успеха (Audit Success) — событие, соответствующее успешно завершенному действию, относящемуся к безопасности системы. Примером такого события является успешная попытка входа пользователя в систему.

Аудит отказа (Audit Failure) — событие, соответствующее неудачно завершенному действию, относящемуся к безопасности системы. Например, такое событие будет зарегистрировано, если попытка доступа пользователем к сетевому диску закончилась неудачей.

1. Событию можно назначить обработчик, то есть функцию, которая сработает, как только событие произошло.

Именно благодаря обработчикам JavaScript-код может реагировать на действия пользователя.

Есть несколько способов назначить событию обработчик. Сейчас мы их рассмотрим, начиная с самого простого.

<script>

function countRabbits() {

for(let i=1; i<=3; i++) {

alert("Кролик номер " + i);

}

}

</script>

<input type="button" onclick="countRabbits()" value="Считать кроликов!">

1. Для взаимодействия с пользователем в JavaScript определен механизм событий. Например, когда пользователь нажимает кнопку, то возникает событие нажатия кнопки.

unction handler(bool, event) {

console.log(bool, event);

}

let handleFalse = handler.bind(null, false);

let handleTrue = handler.bind(null, true);

handleFalse('Event 1');

handleTrue('Event 2');

5.Метод addEventListener позволяет добавлять несколько обработчиков на одно событие одного элемента

6.Распространение события - это процесс, в ходе которого браузер решает, в каких объектах следует вызвать обработчики событий. В случае событий, предназначенных для единственного объекта (таких как событие «load» объекта Window), надобность в их распространении отсутствует. Однако, когда некоторое событие возникает в элементе документа, оно распространяется, или «всплывает», вверх по дереву документа.

7. Объект Event описывает событие, произошедшее на странице. Одной из причин возникновения событий являются действия пользователя, такие как клики мышкой MouseEvent или ввод с клавиатуры KeyboardEvent. Существует множество различных событий с разным набор информации.

element.addEventListener('click', function (event) {

console.log(event)

})