**Практическое занятие № 4**

**ТЕМА: События и функции**

|  |  |  |
| --- | --- | --- |
| **4.1. Теоретический материал** | | |
| **События JavaScript**  Практически все JavaScript-приложения выполняют те или иные действия, откликаясь на различные события.  Событие - это сигнал от браузера о том, что что-то произошло. События делятся на несколько категорий:   * события, связанные с документом; * события, связанные с элементами документа; * события, связанные с окнами.   Для того чтобы скрипт реагировал на событие - нужно назначить обработчик события.  Обычно обработчики называют "on+имя события", например: onclick. Существует несколько способов назначать обработчик на конкретное событие элемента. Один из этих способов – обработчик события записывается прямо в открывающем теге элемента. Например, для обработки события click на кнопке input, можно назначить обработчик onclick вот так:    **<input type="button" value="Нажми Меня " onclick="alert('Спасибо!');" />**    В этом случае JavaScript код пишется в кавычках в одну строку. Такой способ установки обработчиков очень удобен - он нагляден и прост, поэтому часто используется в решении простых задач.  Событие **click** – одинарный щелчок (нажата и отпущена кнопка мыши) возникает фактически для всех элементов страницы. Например, в *примере 4.1* после нажатия на кнопку вызывается метод alert().  *Событие Load и его обработчик onLoad*  Событие **Load** возникает для элементов body и frameset когда закончена загрузка документа. Например, в *примере 4.2* после загрузки страницы вызывается метод alert().  Как только обработчик начинает занимать больше одной строки – читабельность кода резко падает. В этом случае для обработки события нужно использовать функцию. При этом в обработчике события указывают только имя функции.  ***Описание функции***  Синтаксис:    Команда return, возвращающая значение функции, может быть не одна, может и вовсе отсутствовать. В последнем случае функция не возвращает никакого значения и ее вызов нельзя использовать в выражениях. Если в функцию или из нее не передаются параметры - то после имени функции ставятся круглые скобки без параметров.  ***Вызов функции***  Синтаксис:    Фактическим аргументом функции может быть константа, переменная, выражение и, в частности, вызов другой функции. Функция не может быть выполнена до тех пор, пока не будет явного обращения к ней  *Работа с функциями продемонстрирована в примерах 4.3 и 4.4* | | |
| **Пример 4.1 Демонстрация события onclick** | | |
|  | ***Задача:*** | |
|  | Продемонстрируйте вызов сообщения в диалоговом окне alert при нажатии на кнопку |
| ***Решение:*** | |
|  | Файл index.html |
| ***Ответ:*** | |
|  |  |
| **Пример 4.2. Демонстрация события onLoad** | | |
|  | ***Задача:*** | |
|  | Продемонстрируйте вызов сообщения в диалоговом окне alert при загрузке страницы |
| ***Решение:*** | |
|  | Файл index.html |
| ***Ответ:*** | |
|  |  |
| **Пример 4.3. Демонстрация функций** | | |
|  | ***Задача:*** | |
|  | Оформить ввод и вычисление суммы двух чисел как функцию, которая вызывается по щелчку кнопки |
| ***Решение:*** | |
|  | Файл index.html:    Файл script.js: |
| ***Ответ:*** | |
|  | Ввод первого числа после щелчка по кнопке:    Ввод второго числа:    Результат сложения: |
| **Пример 4.4. Демонстрация функций** | | |
|  | ***Задача:*** | |
|  | Оформить функцию, которая в качестве параметра получает два числа и находит их сумму. Числа вводятся вне функции. Функция вызывается по щелчку кнопки. |
| ***Решение:*** | |
|  | Файл index.html:    Файл script.js: |
| ***Ответ:*** | |
|  | Ввод первого числа:    Ввод второго числа:    Результат после нажатия на кнопку «Найдем сумму» |
| **Задание 4.1** | | |
|  | ***Задача:*** | |
|  | Написать функцию, которая проверяет, является ли число простым. Число вводится с помощью метода prompt (например, способом, который продемонстрирован в примере 4.4). Функция запускается по нажатии на кнопку. (Простыми называются числа, которые делятся только на 1 и на само себя) |
| ***Решение:*** | |
|  | Файл index.html:    Файл my.js: |
| ***Ответ:*** | |
|  |  |
| **Задание 4.2** | | |
|  | ***Задача:*** | |
|  | Написать функцию, которая печатает на странице (по нажатии на кнопку) единичную матрицу из n строк и n столбцов. Число n можно задать непосредственно в скрипте. (Единичная матрица – это матрица, у которой диагональные элементы равны 1, а остальные – нулевые). |
| ***Решение:*** | |
|  | Файл index.html:    Файл my.js: |
| ***Ответ:*** | |
|  |  |