**Материалы для чайников**

## 1. **Событие**mousemove

### ****Что такое событие****mousemove****?****

Событие mousemove в JavaScript срабатывает каждый раз, когда пользователь перемещает курсор мыши внутри заданной области (например, внутри элемента HTML, всей страницы и т.д.).

### ****Как работает****mousemove****?****

Когда курсор перемещается, браузер генерирует событие mousemove, которое можно обработать в JavaScript. Для этого нужно добавить обработчик события, который будет выполнять действия каждый раз, когда мышь двигается.

* **Пример использования:** Вы можете использовать это событие для отслеживания движения мыши, например, чтобы отображать координаты мыши, менять стиль элемента или анимацию в зависимости от положения курсора.

#### **Свойства события**mousemove**:**

* **event.clientX** – возвращает горизонтальную координату мыши относительно области просмотра (в пикселях).

* **event.clientY** – возвращает вертикальную координату мыши относительно области просмотра (в пикселях).

* **event.pageX** и **event.pageY** – возвращают координаты мыши относительно всей страницы (с учетом прокрутки).

* **event.target** – возвращает элемент, на котором произошло событие.

### ****Пример:****

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

console.log('X:', event.clientX, 'Y:', event.clientY);

});

Этот код будет выводить в консоль координаты мыши в момент движения.

## 2. **Событие**mouseout

### ****Что такое событие****mouseout****?****

Событие mouseout срабатывает, когда курсор мыши покидает элемент, на котором оно было установлено. Это противоположность событию mouseenter, которое срабатывает, когда курсор входит в элемент.

### ****Как работает****mouseout****?****

Когда пользователь перемещает курсор мыши за пределы элемента, срабатывает событие mouseout. Это полезно для создания эффектов, которые должны исчезать, когда пользователь покидает область элемента.

#### **Пример:**

element.addEventListener('mouseout', function() {

console.log('Курсор покинул элемент');

});

Когда курсор покидает элемент, в консоль будет выведено сообщение.

## 3. **Событие**scroll

### ****Что такое событие****scroll****?****

Событие scroll срабатывает каждый раз, когда происходит прокрутка содержимого на странице или в элементе (например, при использовании колесика мыши, перетаскивании полосы прокрутки или на мобильных устройствах при прокрутке пальцем).

### ****Как работает****scroll****?****

Когда вы прокручиваете страницу или элемент с помощью мыши или клавиш, браузер генерирует событие scroll. Это событие часто используется для динамического изменения контента на странице в зависимости от того, насколько пользователь прокрутил страницу.

#### **Свойства события**scroll**:**

* **window.scrollY** – возвращает текущую вертикальную позицию прокрутки страницы (в пикселях).

* **window.scrollX** – возвращает текущую горизонтальную позицию прокрутки страницы (в пикселях).

* **element.scrollTop** – для прокрутки внутри элемента, возвращает текущее положение вертикальной прокрутки этого элемента.

### ****Пример:****

window.addEventListener('scroll', function() {

console.log('Текущая позиция прокрутки:', window.scrollY);

});

Этот код будет выводить текущую вертикальную позицию прокрутки страницы в консоль.

## 4. **Работа с координатами мыши**

### ****Что нужно учитывать при работе с координатами?****

Когда вы работаете с событием mousemove, часто возникает необходимость работы с координатами мыши. Важно понимать разницу между clientX/clientY и pageX/pageY.

* **event.clientX и event.clientY** – дают координаты мыши относительно видимой области браузера (без учета прокрутки).

* **event.pageX и event.pageY** – дают координаты мыши относительно всей страницы (с учетом прокрутки).

### ****Как использовать координаты для изменения позиции элементов?****

* Событие mousemove может быть использовано для отслеживания положения мыши и динамического изменения стилей элементов на странице, например, изменения размера элемента или его положения.

Пример:

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

element.style.left = `${event.pageX - element.offsetWidth / 2}px`;

element.style.top = `${event.pageY - element.offsetHeight / 2}px`;

});

## 5. **Прокрутка страницы**

### ****Что происходит при прокрутке?****

Когда происходит событие scroll, страница или элемент может прокручиваться вверх или вниз, и это меняет позицию видимой области страницы.

### ****Как отслеживать позицию прокрутки?****

Для отслеживания текущей позиции прокрутки страницы используется свойство window.scrollY, которое возвращает количество пикселей, на которое страница была прокручена по вертикали.

### ****Как изменить поведение при прокрутке?****

Для создания динамического контента, например, подгрузки данных при прокрутке до конца страницы, используется событие scroll. Важно не забывать о производительности: слишком частые срабатывания события могут повлиять на скорость работы страницы, поэтому иногда используют оптимизации, например, с использованием requestAnimationFrame или debounce.

### ****Пример:****

window.addEventListener('scroll', function() {

console.log('Позиция прокрутки:', window.scrollY);

});

## 6. **Практическое применение этих событий**

### ****Задание 5:****mousemove****с изменением цвета фона****

* Используйте событие mousemove, чтобы отслеживать положение курсора внутри элемента.

* Генерируйте случайные значения для цветов и изменяйте стиль элемента с помощью JavaScript.

* Дополнительно можно использовать событие mouseout, чтобы вернуть элемент в исходное состояние.

### ****Задание 10:****scroll****с отображением позиции прокрутки****

* Используйте событие scroll, чтобы отслеживать прокрутку страницы или элемента.

* С помощью window.scrollY или element.scrollTop можно получить текущую позицию прокрутки и отображать ее на экране в реальном времени.

* Этот подход может быть полезен для создания динамических интерфейсов или пользовательских эффектов, например, для "подгрузки" контента при прокрутке.

**Задание 1: Событие click**

Создайте кнопку, при нажатии на которую в консоль выводится сообщение "Кнопка нажата".

**Требования:**

1. Создайте кнопку на странице с текстом "Нажми меня".
2. Добавьте обработчик события click, который выводит сообщение в консоль.

**Задание 2: Событие dblclick**

Создайте элемент, при двойном щелчке на котором его текст меняется на "Двойной клик!".

**Требования:**

1. Создайте элемент с текстом "Дважды щелкни меня".
2. Добавьте обработчик события dblclick, который меняет текст элемента.

**Задание 3: Событие mousedown**

Создайте элемент, который при удерживании левой кнопки мыши будет менять цвет на красный.

**Требования:**

1. Создайте элемент (например, div).
2. Добавьте обработчик события mousedown, который изменяет цвет элемента на красный, когда кнопка мыши нажата.

**Задание 4: Событие mouseup**

Создайте элемент, который меняет цвет обратно на первоначальный при отпускании кнопки мыши.

**Требования:**

1. Создайте элемент (например, div).
2. Добавьте обработчик события mouseup, который меняет цвет элемента обратно на изначальный.

### Задание 5: ****Событие**** mousemove ****с изменением цвета фона элемента****

**Требования:** Создайте элемент, который будет изменять цвет фона рандомно при движении мыши внутри него.

**Задание 6: Событие mouseover**

Создайте элемент, который при наведении мыши будет изменять свой фон на синий.

**Требования:**

1. Создайте элемент (например, div).
2. Добавьте обработчик события mouseover, который изменяет фон элемента на синий.

### Объяснение:

* Элемент меняет свой цвет фона при движении мыши внутри него, используя событие mousemove.
* Цвет генерируется случайным образом с помощью Math.random() и устанавливается как цвет фона.
* Когда курсор покидает элемент, его фон возвращается в исходное состояние (lightgray).

**Задание 7: Событие mouseout**

Создайте элемент, который при выходе мыши за его пределы возвращает свой фон в изначальный цвет.

**Требования:**

1. Создайте элемент (например, div).
2. Добавьте обработчик события mouseout, который изменяет фон обратно на исходный.

**Задание 8: Событие scroll**

Создайте страницу с длинным содержимым и добавьте обработчик события scroll, который будет выводить в консоль текущую позицию прокрутки страницы.

**Требования:**

1. Создайте длинную страницу, например, с несколькими параграфами текста.
2. Добавьте обработчик события scroll, который выводит текущую позицию прокрутки (window.scrollY).

**Задание 9: Событие click на нескольких элементах**

Создайте несколько кнопок, при нажатии на каждую из которых в консоль выводится сообщение с номером кнопки.

**Требования:**

1. Создайте 6 кнопok с текстом цвета "Красный", "Фиолетовый", "Розовый" и т д.
2. Добавьте обработчик события click, Который изменяет цвета <body>

### Задание 10: ****Событие**** scroll ****с отображением текущей позиции прокрутки****

**Требования:** Создайте страницу с длинным содержимым и добавьте обработчик события scroll, который будет отображать текущую позицию прокрутки в реальном времени.

### Объяснение:

* В этом задании добавляется длинное содержимое, чтобы появилась возможность прокрутки страницы.
* При прокрутке страницы отображается текущая вертикальная позиция прокрутки (window.scrollY).
* Позиция прокрутки обновляется в реальном времени в фиксированном блоке, который находится в верхнем правом углу страницы.