

# События. Часть 3

**№ урока:** 8 **Курс:** JavaScript Базовый

**Средства обучения:** Visual Studio Code  
Web Browser

## Обзор, цель и назначение урока

Изучить фазы распространения событий – погружение, на цели, всплытие. Научиться использовать фазы событий для эффективной обработки событий элементов.

## Изучив материал данного занятия, учащийся сможет:

- Использовать перехват и всплытие событий.
- Использовать одну функцию для обработки события нескольких элементов.
- Останавливать всплытие события.
- Останавливать действие по умолчанию, связанное с событием.

## Содержание урока

1. Фазы прохода события
2. Регистрация обработчика события на разные фазы
3. Отмена всплытия
4. Отмена действия по умолчанию
5. Примеры использования

## Резюме

- Фазы прохода событий:
  - Capturing phase – фаза погружения или фаза перехвата. Событие идет сверху вниз, от корневого элемента до того, который инициировал событие.
  - Target phase – фаза цели. Событие достигло элемента, из-за которого событие возникло.
  - Bubbling phase – фаза всплытия. Событие движется от целевого элемента к корневому.
- Фаза перехвата используется достаточно редко. В основном используется фаза всплытия, так как она позволяет сделать один обработчик события на контейнере, в котором находится несколько элементов, которые событие инициируют. Этот подход называется делегирование событий – дочерние элементы делегируют обработку события родительскому элементу. При этом, один обработчик на контейнере будет обрабатывать события от всех вложенных элементов. Такой подход позволяет сэкономить оперативную память, так как нужен всего один обработчик, вместо нескольких. Также это упрощает расширение приложения, так как с добавлением новых элементов, нет необходимости создавать для них обработчики.
- Объект события предоставляет свойство `event.target`, которое содержит самый глубокий элемент, который вызвал событие (целевой элемент).

- `Event.target` – целевой элемент, в процессе прохода события его значение не меняется. В свою очередь, значение `this` всегда содержит текущий элемент, до которого дошло событие. На этапе перехвата и на этапе всплытия значения `this` и `event.target` будут отличаться. В фазе цели значения `this` и `event.target` будут равны.
- По умолчанию все обработчики события регистрируются на фазе всплытия. Для регистрации на фазе перехвата, необходимо добавить в `addEventListener` третий параметр – `true`.
- Для того чтобы прекратить переход события к следующему элементу, необходимо вызвать метод `event.stopPropagation()`.
- Для того чтобы отменить действие по умолчанию, которое связано с событием, необходимо вызвать метод `event.preventDefault()`. Например, для события `submit` формы, происходит отрывка сетевого запроса. Если в обработчике события будет вызван метод `preventDefault()` то отправка запроса будет отменена.
- Некоторые события не всплывают, например, событие `focus` не поддерживает эту фазу.

### Закрепление материала

- Назовите фазы переходов события.
- Чем отличается свойство `event.target` от ключевого слова `this` в функции обработчике события?
- Как установить обработчик, который сработает на фазе перехвата?
- Как можно остановить переход события по узлам DOM дерева?
- Как можно отменить действие по умолчанию, связанное с событием?
- Приведите примеры использования перехвата событий для решения прикладной задачи.

### Самостоятельная деятельность учащегося

Выполните задания в директории `Exercises\Tasks\008 Events. Part 3`. Текст задач, находится в комментариях в тегах `script`

### Рекомендуемые ресурсы

Всплытие и перехват событий

[https://developer.mozilla.org/ru/docs/Learn/JavaScript/Building\\_blocks/Events#всплытие\\_и\\_перехват\\_событий](https://developer.mozilla.org/ru/docs/Learn/JavaScript/Building_blocks/Events#всплытие_и_перехват_событий)

`event.stopPropagation()`

<https://developer.mozilla.org/ru/docs/Web/API/Event/stopPropagation>

Делегирование событий

<https://davidwalsh.name/event-delegate>