

События в деталях

JavaScript

Простые события

- Mousedown – кнопка мыши нажата над элементом
- Mouseup – кнопка мыши отпущена над элементом
- Mouseover – мышь появилась над элементом
- Mouseout – мышь ушла с элемента
- Mousemove – мышь движется

Комплексные события

- Click - вызывается при клике мышью (mousedown + mouseup)
- Contextmenu - вызывается при клике правой кнопкой мыши на элементе
- Dblclick - двойной клик по элементу (click + click)

Порядок срабатывания элементов

- Одно действие может вызывать несколько событий
- Клик вызывает сначала `mousedown` при нажатии, затем `mouseup` и `click` при отпускании кнопки
- Когда одно действие генерирует несколько событий, их порядок фиксирован, например `mousedown` → `mouseup` → `click`

Информация о кнопке

- Для получения кнопки мыши в объекте event есть свойство which
- Возможны следующие значения:
 - event.which == 1 – левая кнопка
 - event.which == 2 – средняя кнопка
 - event.which == 3 – правая кнопка

Правый клик

- `<div>`
 - Правый клик на этой кнопке выведет "Клик«
- `</div>`
- `<button oncontextmenu="alert('1');">`
 - Правый клик сюда
- `</button>`
- `<button oncontextmenu="alert('2');return false">`
 - Правый клик сюда
- `</button>`

Модификаторы

- Во всех событиях мыши присутствует информация о нажатых клавишах-модификаторах
- Соответствующие свойства:
 - `shiftKey`
 - `altKey`
 - `ctrlKey`
 - `metaKey` (для Mac)

Пример использования

- `<button>`
 - `Alt+Shift+Кликни меня!`
- `</button>`
- `<script>`
 - `document.body.children[0].onclick = function(e) {`
 - `if (!e.altKey || !e.shiftKey) return;`
 - `alert('Ура!');`
 - `}`
- `</script>`

Координаты в окне и документе

- Все мышинные события предоставляют текущие координаты курсора в двух видах:
 - `clientX/clientY` – относительно окна
 - `pageX/pageY` – относительно документа
- `<input onmousemove="this.value = event.clientX+':'+event.clientY">`
- `<input onmousemove="this.value = event.pageX+':'+event.pageY">`

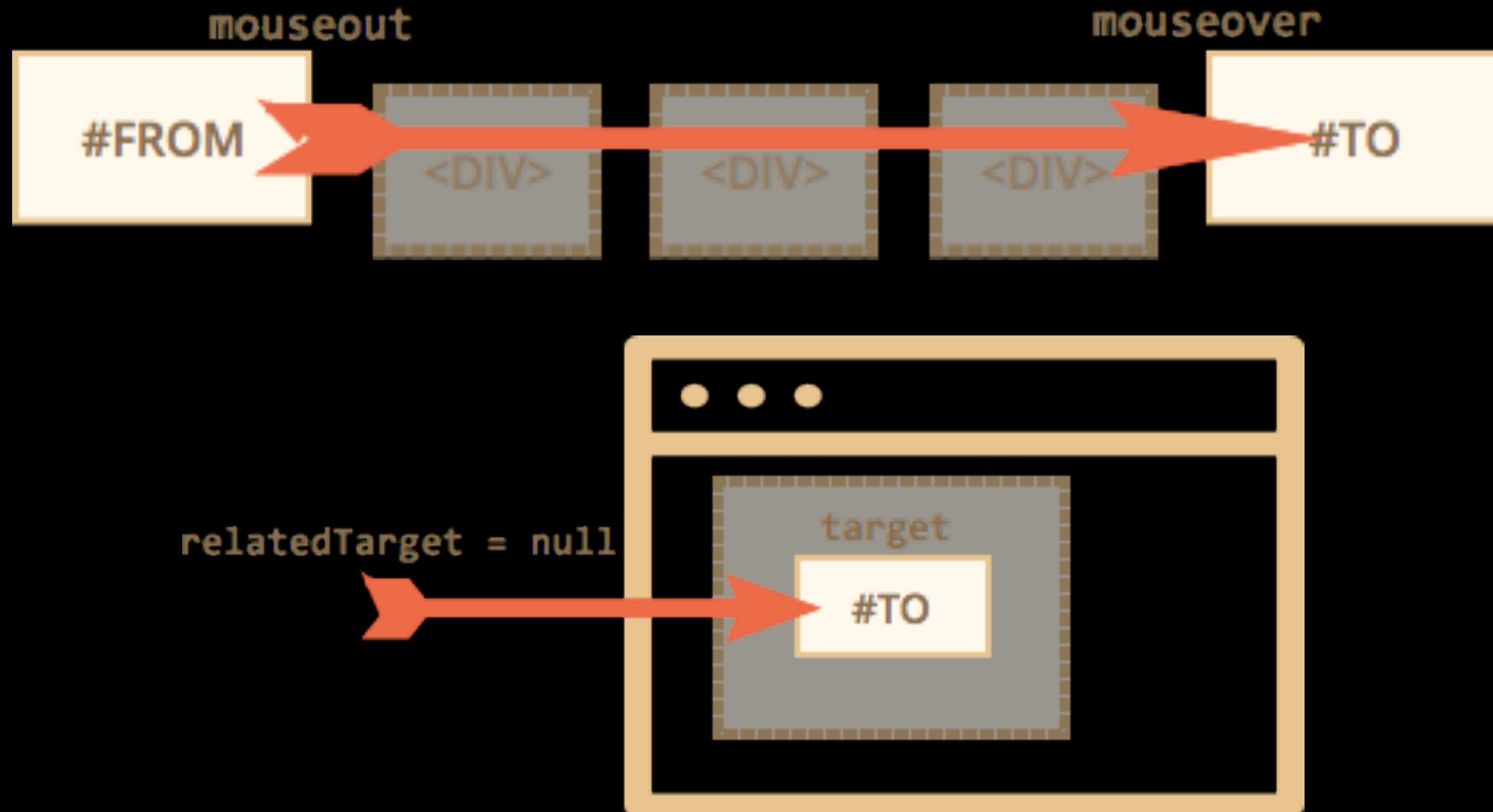
Отмена выделения

- ``
 - Текст
- ``
- `<div ondblclick="alert('Тест')" onselectstart="return false" onmousedown="return false">`
 - Текст
- `</div>`

Событие `mouseover/mouseout`

- `mouseover`:
 - `event.target` – элемент, на который пришла мышь
 - `event.relatedTarget` – элемент, с которого пришла мышь
- `Mouseout`
 - `event.target` – элемент, с которого ушла мышь
 - `event.relatedTarget` – элемент, на который перешла мышь

Частота событий



Уход на потомка



Mouseenter и mouseleave

- События `mouseenter/mouseleave` похожи на `mouseover/mouseout`
- Срабатывают, когда курсор заходит на элемент и уходит с него, но с двумя отличиями
 - Не учитываются переходы внутри элемента
 - События `mouseenter/mouseleave` не всплывают

Алгоритм Drag'n'Drop

- Алгоритм Drag'n'Drop выглядит так:
 - Отслеживаем нажатие кнопки мыши на переносимом (mousedown)
 - При нажатии — подготовить элемент к перемещению
 - Отслеживаем движение мыши через mousemove и передвигаем переносимый элемент на новые координаты путём смены left/top
 - При отпускании кнопки мыши (mouseup) остановить перенос элемента и произвести все действия, связанные с окончанием Drag'n'Drop

Scroll vs wheel

- При прокрутке срабатывает событие onscroll
- Оно произойдёт *при любой прокрутке*, в том числе через клавиатуру, но *только на прокручиваемых элементах*
- Событие wheel является чисто «мышиным»
- Оно генерируется *над* любым элементом при передвижении колеса мыши, при этом не важно, прокручиваемый он или нет
- Onscroll происходит *после* прокрутки, onwheel — *до* прокрутки