

Раздел

[Интерфейсные события](#)

Навигация по уроку

Типы событий мыши

Получение информации о кнопке: `which`Модификаторы: `shift`, `alt`, `ctrl` и `meta`Координаты: `clientX/Y`, `pageX/Y`

Отключаем выделение

Итого

Задачи (1)

Комментарии

Поделиться

[Редактировать на GitHub](#)[🏠](#) → [Браузер: документ, события, интерфейсы](#)
→ [Интерфейсные события](#)

2-го октября 2020

Основы событий мыши

В этой главе мы более детально рассмотрим события мыши и их свойства.

Сразу заметим: эти события бывают не только из-за мыши, но и эмулируются на других устройствах, в частности, на мобильных, для совместимости.

Типы событий мыши

Мы можем разделить события мыши на две категории: «простые» и «комплексные».

Простые события

Самые часто используемые простые события:

`mousedown/mouseup`

Кнопка мыши нажата/отпущена над элементом.

`mouseover/mouseout`

Курсор мыши появляется над элементом и уходит с него.

`mousemove`

Каждое движение мыши над элементом генерирует это событие.

`contextmenu`

Вызывается при попытке открытия контекстного меню, как правило, нажатием правой кнопки мыши. Но, заметим, это не совсем событие мыши, оно может вызываться и специальной клавишей клавиатуры.

...Есть также несколько иных типов событий, которые мы рассмотрим позже.

Комплексные события

`click`

Вызывается при `mousedown`, а затем `mouseup` над одним и тем же элементом, если использовалась левая кнопка мыши.

`dblclick`

Вызывается двойным кликом на элементе.

Комплексные события состоят из простых, поэтому в теории мы могли бы без них обойтись. Но хорошо, что они существуют, потому что работать с ними очень удобно.

Порядок событий

Одно действие может вызвать несколько событий.

Например, клик мышью вначале вызывает `mousedown`, когда кнопка нажата, затем `mouseup` и `click`, когда она отпущена.

В случае, когда одно действие инициирует несколько событий, порядок их выполнения фиксирован. То есть обработчики событий вызываются в следующем порядке: `mousedown` → `mouseup` → `click`.

Кликните на кнопку ниже, и вы увидите события. Также попробуйте двойной клик.

Раздел

[Интерфейсные события](#)

Навигация по уроку

Типы событий мыши

Получение информации о кнопке: `which`

Модификаторы: `shift`, `alt`, `ctrl` и `meta`

Координаты: `clientX/Y`, `pageX/Y`

Отключаем выделение

Итого

Задачи (1)

Комментарии

Поделиться



[Редактировать на GitHub](#)

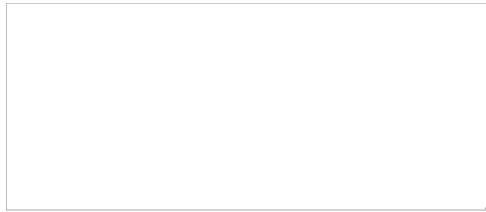


В окне теста ниже все события мыши записываются, и если задержка между ними более 1 секунды, то они разделяются горизонтальной чертой.

При этом мы также можем увидеть свойство `which`, которое позволяет определить, какая кнопка мыши была нажата.

Клики меня левой или правой кнопкой мыши

Очистить



Получение информации о кнопке: `which`

События, связанные с кликом, всегда имеют свойство `which`, которое позволяет определить нажатую кнопку мыши.

Это свойство не используется для событий `click` и `contextmenu`, поскольку первое происходит только при нажатии левой кнопкой мыши, а второе – правой.

Но если мы отслеживаем `mousedown` и `mouseup`, то оно нам нужно, потому что эти события срабатывают на любой кнопке, и `which` позволяет различать между собой «нажатие правой кнопки» и «нажатие левой кнопки».

Есть три возможных значения:

- `event.which == 1` – левая кнопка
- `event.which == 2` – средняя кнопка
- `event.which == 3` – правая кнопка

Средняя кнопка сейчас – скорее экзотика, и используется очень редко.

Модификаторы: `shift`, `alt`, `ctrl` и `meta`

Все события мыши включают в себя информацию о нажатых клавишах-модификаторах.

Свойства объекта события:

- `shiftKey`: `Shift`
- `altKey`: `Alt` (или `Opt` для Mac)
- `ctrlKey`: `Ctrl`
- `metaKey`: `Cmd` для Mac

Они равны `true`, если во время события была нажата соответствующая клавиша.

Например, кнопка внизу работает только при комбинации

`Alt+Shift`+клик:

```
1 <button id="button">Нажми Alt+Shift+Click на мне!</butt
2
3 <script>
4   button.onclick = function(event) {
5     if (event.altKey && event.shiftKey) {
6       alert('Упа!');
7     }
8   };
9 </script>
```

Нажми Alt+Shift+Click на мне!

Раздел

Интерфейсные события

Навигация по уроку

Типы событий мыши

Получение информации о кнопке: which

Модификаторы: shift, alt, ctrl и meta

Координаты: clientX/Y, pageX/Y

Отключаем выделение

Итого

Задачи (1)

Комментарии

Поделиться



Редактировать на GitHub



⚠ Внимание: обычно на Мас используется клавиша `Cmd` вместо `Ctrl`

В Windows и Linux клавишами-модификаторами являются `Alt`, `Shift` и `Ctrl`. На Мас есть ещё одна: `Cmd`, которой соответствует свойство `metaKey`.

В большинстве приложений, когда в Windows/Linux используется `Ctrl`, на Мас используется `Cmd`.

То есть, когда пользователь Windows нажимает `Ctrl+Enter` и `Ctrl+A`, пользователь Мас нажимает `Cmd+Enter` или `Cmd+A`, и так далее.

Поэтому, если мы хотим поддерживать такие комбинации, как `Ctrl+клик`, то для Мас имеет смысл использовать `Cmd+клик`. Это удобней для пользователей Мас.

Даже если мы и хотели бы заставить людей на Мас использовать именно `Ctrl+клик`, это довольно сложно. Проблема в том, что левый клик в сочетании с `Ctrl` интерпретируется как *правый клик* на MacOS и генерирует событие `contextmenu`, а не `click` как на Windows/Linux.

Поэтому, если мы хотим, чтобы пользователям всех операционных систем было удобно, то вместе с `ctrlKey` нам нужно проверять `metaKey`.

Для JS-кода это означает, что мы должны проверить `if (event.ctrlKey || event.metaKey)`.

⚠ Не забывайте про мобильные устройства

Комбинации клавиш на клавиатуре – это хорошее дополнение к рабочему процессу. Если у пользователя есть клавиатура – они работают. Ну а если на его устройстве её нет – должен быть другой способ сделать то же самое.

Координаты: `clientX/Y`, `pageX/Y`

Все события мыши имеют координаты двух видов:

1. Относительно окна: `clientX` и `clientY`.
2. Относительно документа: `pageX` и `pageY`.

Например, если у нас есть окно размером 500x500, и курсор мыши находится в левом верхнем углу, то значения `clientX` и `clientY` равны 0. А если мышь находится в центре окна, то значения `clientX` и `clientY` равны 250 независимо от того, в каком месте документа она находится и до какого места документ прокручен. В этом они похожи на `position:fixed`.

Наведите курсор мыши на поле ввода, чтобы увидеть `clientX/clientY` (пример находится в `iframe`, поэтому координаты определяются относительно этого `iframe`):

```
1 <input onmousemove="this.value=event.clientX+' '+event.
```

Наведи на меня мышь

Координаты относительно документа `pageX`, `pageY` отсчитываются не от окна, а от левого верхнего угла документа. Подробнее о координатах вы можете узнать в главе [Координаты](#).

Отключаем выделение

Двойной клик мыши имеют побочный эффект, который может быть неудобен в некоторых интерфейсах: он выделяет текст.

Например, двойной клик на текст ниже выделяет его в дополнение к нашему обработчику:

```
1 <span onclick="alert('dblclick')">Сделайте двойной к.
```

Сделайте двойной клик на мне

Если нажать левую кнопку мыши и, не отпуская кнопку, провести мышью, то также будет выделение, которое в интерфейсах может быть «не кстати».

Есть несколько способов запретить выделение, о которых вы можете прочитать в главе [Selection и Range](#).

В данном случае самым разумным будет отменить действие браузера по умолчанию при событии `mousedown`, это отменит оба этих выделения:

```
1 До...
2 <b onclick="alert('Клик!')" onmousedown="return false"
3   Сделайте двойной клик на мне
4 </b>
5 ...После
```

До... Сделайте двойной клик на мне ...После

Теперь выделенный жирным элемент не выделяется при двойном клике, а также на нём нельзя начать выделение, зажав кнопку мыши.

Заметим, что текст внутри него по-прежнему можно выделить, если начать выделение не на самом тексте, а до него или после. Обычно это нормально воспринимается пользователями.

Предотвращение копирования

Если мы хотим отключить выделение для защиты содержимого страницы от копирования, то мы можем использовать другое событие: `onscopy`.

```
1 <div oncopy="alert('Копирование запрещено!');return false">
2   Уважаемый пользователь,
3   Копирование информации запрещено для вас.
4   Если вы знаете JS или HTML, вы можете найти всю
5 </div>
```

Уважаемый пользователь, Копирование информации запрещено для вас. Если вы знаете JS или HTML, вы можете найти всю нужную вам информацию в исходном коде страницы.

Если вы попытаетесь скопировать текст в `<div>`, у вас это не получится, потому что срабатывание события `onscopy` по умолчанию запрещено.

Конечно, пользователь имеет доступ к HTML-коду страницы и может взять текст оттуда, но не все знают, как это сделать.

Итого

События мыши имеют следующие свойства:

- Кнопка: `which`.
- Клавиши-модификаторы (`true` если нажаты): `altKey`, `ctrlKey`, `shiftKey` и `metaKey` (Mac).
- Если вы планируете обработать `Ctrl`, то не забудьте, что пользователи Mac обычно используют `Cmd`, поэтому лучше проверить `if (e.metaKey || e.ctrlKey)`.
- Координаты относительно окна: `clientX/clientY`.

Раздел

Интерфейсные события

Навигация по уроку

Типы событий мыши

Получение информации о кнопке: `which`

Модификаторы: `shift`, `alt`, `ctrl` и `meta`

Координаты: `clientX/Y`, `pageX/Y`

Отключаем выделение

Итого

Задачи (1)

Комментарии

Поделиться



Редактировать на GitHub

Раздел

[Интерфейсные события](#)

Навигация по уроку

Типы событий мыши

Получение информации о кнопке: `which`

Модификаторы: `shift`, `alt`, `ctrl` и `meta`

Координаты: `clientX/Y`, `pageX/Y`

Отключаем выделение

Итого

Задачи (1)

Комментарии

Поделиться



[Редактировать на GitHub](#)



- Координаты относительно документа: `pageX/pageY`.

Действие по умолчанию события `mousedown` – начало выделения, если в интерфейсе оно скорее мешает, его можно отменить.

В следующей главе мы поговорим о событиях, которые возникают при передвижении мыши, и об отслеживании смены элементов под указателем.

✓ Задачи

Выделяемый список

важность: 5

Создайте список, в котором элементы могут быть выделены, как в файловых менеджерах.

- При клике на элемент списка выделяется только этот элемент (добавляется класс `.selected`), отменяется выделение остальных элементов.
- Если клик сделан вместе с `Ctrl` (`Cmd` для Mac), то выделение переключается на элементе, но остальные элементы при этом не изменяются.

Демо:

Клики на элемент списка, чтобы выделить его.

- Кристофер Робин
- Винни Пух
- Тигра
- Кенга
- Кролик. Просто Кролик.



P.S. В этом задании все элементы списка содержат только текст. Без вложенных тегов.



P.P.S. Предотвратите стандартное для браузера выделение текста при кликах.

[Открыть песочницу для задачи.](#)

решение

Проводим [курсы по JavaScript и фреймворкам](#).



💬 Комментарии

перед тем как писать...