**Действия браузера по умолчанию**

Многие события автоматически влекут за собой действие браузера.

Например:

* Клик по ссылке инициирует переход на новый URL.
* Нажатие на кнопку «отправить» в форме – отсылку ее на сервер.
* Двойной клик на тексте – инициирует его выделение.

Если мы обрабатываем событие в JavaScript, то зачастую такое действие браузера нам не нужно. К счастью, его можно отменить.

**[Отмена действия браузера](http://learn.javascript.ru/default-browser-action" \l "otmena-deystviya-brauzera)**

Есть два способа отменить действие браузера:

* **Основной способ – это воспользоваться объектом события. Для отмены действия браузера существует стандартный метод event.preventDefault().**
* Если же обработчик назначен через onсобытие (не через addEventListener), то можно просто вернуть falseиз обработчика.

В следующем примере при клике по ссылке переход не произойдет:

<a href="/" onclick="return false">Нажми здесь</a>

или

<a href="/" onclick="event.preventDefault()">здесь</a>

**Возвращать true не нужно**

Обычно значение, которое возвращает обработчик события, игнорируется.

Единственное исключение – это return false из обработчика, назначенного через onсобытие.

Иногда в коде начинающих разработчиков можно увидеть return других значений. Но они не нужны и никак не обрабатываются.

**[Пример: меню](http://learn.javascript.ru/default-browser-action" \l "primer-menyu)**

Рассмотрим задачу, когда нужно создать меню для сайта, например такое:

<ul id="menu" class="menu">

<li><a href="/php">PHP</a></li>

<li><a href="/html">HTML</a></li>

<li><a href="/javascript">JavaScript</a></li>

<li><a href="/flash">Flash</a></li>

</ul>

Данный пример при помощи CSS может выводиться так:

HTML-разметка сделана так, что все элементы меню являются не кнопками, а ссылками, то есть тегами <a>.

Это потому, что некоторые посетители очень любят сочетание «правый клик – открыть в новом окне». Да, мы можем использовать и <button> и <span>, но если правый клик не работает – это их огорчает. Кроме того, если на сайт зайдёт поисковик, то по ссылке из <a href="..."> он перейдёт, а выполнить сложный JavaScript и получить результат – вряд ли захочет.

Поэтому в разметке мы используем именно <a>, но обычно клик будет обрабатываться полностью в JavaScript, а стандартное действие браузера (переход по ссылке) – отменяться.

Например, вот так:

menu.onclick = function(event) {

if (event.target.nodeName != 'A') return;

var href = event.target.getAttribute('href');

alert( href ); // может быть подгрузка с сервера, генерация интерфейса и т.п.

return false; // отменить переход по url

};

В конце return false, иначе браузер перейдёт по адресу из href.

Так как мы применили делегирование, то меню может увеличиваться, можно добавить вложенные списки ul/li, стилизовать их при помощи CSS – обработчик не потребует изменений.

**[Другие действия браузера](http://learn.javascript.ru/default-browser-action" \l "drugie-deystviya-brauzera)**

Действий браузера по умолчанию достаточно много.

Вот некоторые примеры событий, которые вызывают действие браузера:

* mousedown – нажатие кнопкой мыши в то время как курсор находится на тексте начинает его выделение.
* click на <input type="checkbox"> – ставит или убирает галочку.
* submit – при нажатии на <input type="submit"> в форме данные отправляются на сервер.
* wheel – движение колёсика мыши инициирует прокрутку.
* keydown – при нажатии клавиши в поле ввода появляется символ.
* contextmenu – при правом клике показывается контекстное меню браузера.
* …

Все эти действия можно отменить, если мы хотим обработать событие исключительно при помощи JavaScript.

**События могут быть связаны между собой**

Некоторые события естественным образом вытекают друг из друга.

Например, нажатие мышкой mousedown на поле ввода <input> приводит к фокусировке внутри него. Если отменить действие mousedown, то и фокуса не будет.

Попробуйте нажать мышкой на первый <input> – произойдёт событие onfocus. Это обычная ситуация.

Но если нажать на второй, то фокусировки не произойдёт.

<input value="Фокус работает" onfocus="this.value=''">

<input onmousedown="return false" onfocus="this.value=''" value="Кликни меня">

Это потому, что отменено стандартное действие при onmousedown.

…С другой стороны, во второй <input> можно перейти с первого нажатием клавиши Tab, и тогда фокусировка сработает. То есть, дело здесь именно в onmousedown="return false".

**[Особенности IE8-](http://learn.javascript.ru/default-browser-action" \l "osobennosti-ie8)**

В IE8- для отмены действия по умолчанию нужно назначить свойство event.returnValue = false.

Кроссбраузерный код для отмены действия по умолчанию:

element.onclick = function(event) {

event = event || window.event;

if (event.preventDefault) { // если метод существует

event.preventDefault(); // то вызвать его

} else { // иначе вариант IE8-:

event.returnValue = false;

}

}

Можно записать в одну строку:

...

event.preventDefault ? event.preventDefault() : (event.returnValue=false);

...

**[Итого](http://learn.javascript.ru/default-browser-action" \l "itogo)**

* Браузер имеет встроенные действия при ряде событий – переход по ссылке, отправка формы и т.п. Как правило, их можно отменить.
* Есть два способа отменить действие по умолчанию: первый – использовать event.preventDefault() (IE8-: event.returnValue=false), второй – return false из обработчика. Второй способ работает только если обработчик назначен через onсобытие.