

Краткое описание функций jQuery

[Перемещения](#) > [Обход](#)

[.add\(\)](#)

Добавляет заданные элементы в уже существующий набор элементов.

[Перемещения](#) > [Обход](#)

[.addBack\(\)](#)

Функция объединяет предыдущую выборку элементов с текущей. Это часто необходимо в тех случаях, когда Вам нужно осуществить над этими элементами общее действие.

[Атрибуты](#) | [Манипуляция DOM](#) > [Class атрибут](#) | [CSS](#)

[.addClass\(\)](#)

Добавляет класс(ы) выбранным элементам страницы. Если из этих элементов некоторые уже принадлежат каким-то классам, то новый (новые) класс добавится к существующим, а не заменит их.

[Манипуляция DOM](#) > [DOM Вставка вне](#)

[.after\(\)](#)

Функция вставляет заданное содержимое сразу после определенных элементов страницы.

[Аjax](#) > [Глобальные Ajax события](#)

[.ajaxComplete\(\)](#)

Регистрирует обработчик, который вызывается при завершении каждого Ajax запроса.

Является Ajax событием.

[Аjax](#) > [Глобальные Ajax события](#)

[.ajaxError\(\)](#)

Регистрирует обработчик, который вызывается при завершении Ajax запросов с ошибкой.

Является Ajax событием

[Аjax](#) > [Глобальные Ajax события](#)

[.ajaxSend\(\)](#)

Регистрирует обработчик, который вызывается при завершении Ajax запросов с ошибкой.

Является Ajax событием.

[Аjax](#) > [Глобальные Ajax события](#)

[.ajaxStart\(\)](#)

Регистрирует обработчик который вызывается при начале выполнения первого Ajax запроса. Является Ajax событием.

[Аjax](#) > [Глобальные Ajax события](#)

[.ajaxStop\(\)](#)

Регистрирует обработчик который вызывается когда все Ajax запросы будут выполнены.

Является Ajax событием.

[Аjax](#) > [Глобальные Ajax события](#)

[.ajaxSuccess\(\)](#)

Регистрирует обработчик который вызывается всякий раз когда Ajax запрос выполнится успешно. Является Ajax событием.

[Устаревшие методы](#) > [Версия 1.8](#) | [Перемещения](#) > [Обход](#) | [Удаленные](#)

[.andSelf\(\)](#)

Добавляет элементы из предыдущего набора, к текущему. Под предыдущим набором подразумевается набор элементов, который можно получить с помощью метода `.end()`.

[Эффекты](#) > [Пользовательские](#)

[.animate\(\)](#)

Выполняет произвольную анимацию набора CSS свойств.

[Манипуляция DOM](#) > [DOM Вставка внутрь](#)

[.append\(\)](#)

Вставляет содержимое, заданное параметром, в конец каждого элемента в наборе соответствующих элементов

[Манипуляция DOM](#) > [DOM Вставка внутрь](#)

[.appendTo\(\)](#)

Функция добавляет содержимое в конец элементов.

[Атрибуты](#) | [Манипуляция DOM](#) > [Общие атрибуты](#)

[.attr\(\)](#)

Возвращает или изменяет значение атрибутов у выбранных элементов страницы. Функция имеет четыре варианта использования.

[Манипуляция DOM](#) > [DOM Вставка вне](#)

[.before\(\)](#)

Функция помещает заданное содержимое перед определенными элементами страницы.

[Устаревшие методы](#) > [Версия 3.0](#) | [События](#) > [Прикрепление обработчиков](#)

[.bind\(\)](#)

Устанавливает обработчик события на выбранные элементы страницы.

[События](#) > [События формы](#) | [Формы](#)

[.blur\(\)](#)

Устанавливает обработчик потери фокуса, либо, запускает это событие.

[События](#) > [События формы](#) | [Формы](#)

[.change\(\)](#)

Устанавливает обработчик изменения заданного элемента формы, либо, запускает это событие.

[Перемещения](#) > [Обход элементов](#)

[.children\(\)](#)

Находит дочерние элементы.

[Эффекты](#) > [Пользовательские](#) | [Данные](#) | [Инструменты](#)

[.clearQueue\(\)](#)

Очищает очередь функций у выбранных элементов страницы.

[События](#) > [События мыши](#)

[.click\(\)](#)

Устанавливает обработчик "клика" мышью по элементу, либо, запускает это событие.

[Манипуляция DOM](#) > [Копирование](#)

[.clone\(\)](#)

Метод создает копии выбранных элементов страницы и возвращает их в виде объекта jQuery. Элементы копируются вместе со всеми содержащимися внутри них элементами.

[Перемещения](#) > [Обход элементов](#)

[.closest\(\)](#)

Для каждого элемента в наборе, получает первый элемент, который совпадает с селектором при движении вверх по DOM дереву элементов.

[Перемещения](#) > [Обход](#)

[.contents\(\)](#)

Возвращает все дочерние элементы выбранных элементов, а также текстовое содержимое.

[Устаревшие методы](#) > [Версия 1.10](#) | [Внутренние функции](#) | [Свойства](#) > [Экземпляр объекта jQuery](#) | [Удаленные](#)

[.context](#)

Данное свойство содержит текущий контекст.

[События](#) > [События мыши](#)

[.contextmenu\(\)](#)

Bind an event handler to the “contextmenu” JavaScript event, or trigger that event on an element.

[Манипуляция DOM](#) > [Свойства стилей](#) | [CSS](#)

[.css\(\)](#)

Возвращает или изменяет значения css-величин у выбранных элементов страницы.

[Данные](#) | [Разное](#) > [Хранилище данных](#)

[.data\(\)](#)

С помощью этого метода, появляется возможность привязывать собственные переменные к любым элементам страницы.

[События](#) > [События мыши](#)

[.dblclick\(\)](#)

Устанавливает обработчик двойного "клика" мышью по элементу, либо, запускает это событие.

[Эффекты](#) > [Пользовательские](#)

[.delay\(\)](#)

Устанавливает задержку следующей функции в очереди функций.

[Устаревшие методы](#) > [Версия 3.0](#) | [События](#) > [Прикрепление обработчиков](#)

[.delegate\(\)](#)

Устанавливает обработчик события на элементы, соответствующие заданному селектору.

[Эффекты](#) > [Пользовательские](#) | [Данные](#) | [Инструменты](#)

[.dequeue\(\)](#)

Начинает выполнение следующей функции в очереди.

[Манипуляция DOM](#) > [DOM Удаление](#)

[.detach\(\)](#)

Метод для удаления элементов страницы.

[Устаревшие методы](#) > [Версия 1.7](#) | [События](#) > [Прикрепление обработчиков](#) | [Удаленные](#)

[.die\(\)](#)

У выбранных элементов страницы, удаляет обработчики событий, установленные методом live().

[Разное](#) > [Манипуляция коллекциями](#) | [Перемещения](#)

[.each\(\)](#)

Итерация над объектом JQuery, выполняет функцию для каждого элемента

[Манипуляция DOM](#) > [DOM Удаление](#)

[.empty\(\)](#)

Удаляет все содержимое у выбранных элементов (без удаления самих элементов).

[Перемещения](#) > [Обход](#)

[.end\(\)](#)

Возвращает предыдущий набор элементов в текущей цепочке методов.

[Перемещения](#) > [Фильтрация](#)

[.eq\(\)](#)

Возвращает элемент, идущий под заданным номером в наборе выбранных элементов.

[События](#) > [События браузера](#) | [Устаревшие методы](#) > [Версия 1.8](#) | [Удаленные](#)

[.error\(\)](#)

Устанавливает обработчик ошибки при загрузке элементов (например отсутствие необходимой картинки на сервере).

[Эффекты](#) > [Затухание](#)

[.fadeIn\(\)](#)

С помощью этой функции можно показывать и скрывать выбранные элементы на странице, за счет плавного изменения прозрачности.

[Эффекты](#) > [Затухание](#)

[.fadeOut\(\)](#)

Скрывает соответствующие элементы путем затухания их до прозрачного состояния.

[Эффекты](#) > [Затухание](#)

[.fadeTo\(\)](#)

Изменяет уровень прозрачности у выбранных элементов на странице. Позволяет менять прозрачность плавно.

[Эффекты](#) > [Затухание](#)

[.fadeToggle\(\)](#)

Вызов этого метода приводит к плавному исчезновению (если элемент не скрыт) или появлению (если элемент скрыт) выбранных элементов на странице, за счет изменения прозрачности. Отметим, что после скрытия элемента, его css-свойство display становится равным none, а перед появлением, оно получает свое прежнее значение обратно.

[Перемещения](#) > [Фильтрация](#)

[.filter\(\)](#)

Фильтрует набор выбранных элементов.

[Перемещения](#) > [Обход элементов](#)

[.find\(\)](#)

Поиск потомков внутри каждого элемента в текущем наборе соответствующих элементов с фильтрацией по селектору, объекту jQuery или элементу.

[Эффекты](#) > [Пользовательские](#)

[.finish\(\)](#)

Прекращает выполнение всех текущих анимаций, очищает всю очередь анимации и вызывает событие complete для всех заданных элементов.

[Перемещения](#) > [Фильтрация](#)

[.first\(\)](#)

Возвращает первый элемент из всех выбранных. Эквивалентно .eq(0).

[События](#) > [События формы](#) | [Формы](#)

[.focus\(\)](#)

Устанавливает обработчик получения фокуса, либо, запускает это событие.

[События](#) > [События формы](#) | [Формы](#)

[.focusin\(\)](#)

Устанавливает обработчик получения фокуса самим элементом или одним из его дочерних.

[События](#) > [События формы](#) | [Формы](#)

[.focusout\(\)](#)

Устанавливает обработчик потери фокуса элементом или одним из его дочерних.

[Разное](#) > [DOM методы элементов](#)

[.get\(\)](#)

Осуществляет запрос к серверу методом GET, без перезагрузки страницы.

[Перемещения](#) > [Фильтрация](#)

[.has\(\)](#)

Фильтрует набор выбранных элементов, оставляя только те, которые имеют определенных потомков.

[Атрибуты](#) | [Манипуляция DOM](#) > [Class атрибут](#) | [CSS](#)

[.hasClass\(\)](#)

Проверяет наличие класса у элементов страницы.

[CSS](#) | [Размеры элементов](#) | [Манипуляция DOM](#) > [Свойства стилей](#)

[.height\(\)](#)

Функция возвращает высоту элемента. Кроме этого, с помощью height(), можно установить новое значение высоты.

[Эффекты](#) > [Базовые](#)

[.hide\(\)](#)

С помощью этой функции можно плавно показывать и скрывать выбранные элементы на странице, за счет изменения размера и прозрачности.

[События](#) > [События мыши](#)

[.hover\(\)](#)

Устанавливает обработчик(и) двух событий: mouseenter и mouseleave.

[Атрибуты](#) | [Манипуляция DOM](#) > [DOM Вставка внутрь](#)

[.html\(\)](#)

Возвращает или изменяет html-содержимое выбранных элементов страницы.

[Разное](#) > [DOM методы элементов](#)

[.index\(\)](#)

Возвращает индекс заданного элемента в наборе (2 и 3 вариант использования) или относительно соседних элементов (1 вариант использования).

[CSS](#) | [Размеры элементов](#) | [Манипуляция DOM](#) > [Свойства стилей](#)

[.innerHeight\(\)](#)

Функция возвращает высоту элемента. Кроме этого, с помощью height(), можно установить новое значение высоты.

[CSS](#) | [Размеры элементов](#) | [Манипуляция DOM](#) > [Свойства стилей](#)

[.innerWidth\(\)](#)

Функция возвращает ширину элемента. Кроме этого, с помощью width(), можно установить новое значение ширины.

[Манипуляция DOM](#) > [DOM Вставка вне](#)

[.insertAfter\(\)](#)

Функция вставляет заданное содержимое сразу после определенных элементов страницы.

[Манипуляция DOM](#) > [DOM Вставка вне](#)

[.insertBefore\(\)](#)

Функция помещает заданное содержимое перед определенными элементами страницы.

[Перемещения](#) > [Фильтрация](#)

[.is\(\)](#)

Проверяет, соответствует ли хотя бы один из выбранных элементов определенному условию (оно зависит от заданного параметра: если задан селектор, то условием будет соответствие селектору; если задан объект jQuery, то условие — наличие в нем выбранных элементов; при передаче в .is() элемента DOM, будет проверяться его наличие среди выбранных элементов; и наконец если указать в качестве параметра функцию, то проверка условия будет возложена на нее).

[Внутренние функции](#) | [Свойства](#) > [Экземпляр объекта jQuery](#)

[.jquery](#)

Строка содержащая текущую версию jQuery.

[События](#) > [События клавиатуры](#)

[.keydown\(\)](#)

Устанавливает обработчик перехода клавиши клавиатуры в нажатое состояние, либо, запускает это событие.

[События](#) > [События клавиатуры](#)

[.keypress\(\)](#)

Устанавливает обработчик ввода символа с клавиатуры, либо, запускает это событие.

[События](#) > [События клавиатуры](#)

[.keyup\(\)](#)

Устанавливает обработчик возвращения клавиши клавиатуры в ненажатое состояние, либо, запускает это событие.

[Перемещения](#) > [Фильтрация](#)

[.last\(\)](#)

Возвращает последний элемент из всех выбранных. Эквивалентно .eq(-1).

[Свойства](#) > [Экземпляр объекта jQuery](#)

[.length](#)

Содержит количество выбранных элементов.

[Устаревшие методы](#) > [Версия 1.7](#) | [События](#) > [Прикрепление обработчиков](#) | [Удаленные](#)

[.live\(\)](#)

Устанавливает обработчик событий на выбранные элементы страницы. Метод имеет одну важную особенность, отличающую его от bind(): если на страницу будут вставлены новые элементы, которые соответствуют текущему селектору, то они также будут реагировать на заданные события.

[Устаревшие методы](#) > [Версия 1.8](#) | [События](#) > [Загрузка документа](#) | [Удаленные](#)

[.load\(\)](#)

Устанавливает обработчик полной загрузки выбранных элементов.

[Ajax](#) > [Сокращенные методы](#)

[.load\(\)](#)

Загрузка данных с сервера и размещение полученного HTML в указанный элемент.

[Перемещения](#) > [Фильтрация](#)

[.map\(\)](#)

Выполняет заданную функцию для каждого из выбранных элементов в отдельности. Значения, полученные в результате выполнения этой функции образуют новый набор в виде объекта jQuery, именно его и возвращает метод map.

[События](#) > [События мыши](#)

[.mousedown\(\)](#)

Устанавливает обработчик нажатия кнопки мыши на элементе, либо, запускает это событие.

[События](#) > [События мыши](#)

[.mouseenter\(\)](#)

Устанавливает обработчик появления курсора над элементом, либо, запускает это событие.

[События](#) > [События мыши](#)

[.mouseleave\(\)](#)

Устанавливает обработчик выхода курсора из области элемента, либо, запускает это событие.

[События](#) > [События мыши](#)

[.mousemove\(\)](#)

Устанавливает обработчик движения курсора мыши внутри элемента, либо, запускает это событие.

[События](#) > [События мыши](#)

[.mouseout\(\)](#)

Устанавливает обработчик выхода курсора из области элемента, либо, запускает это событие.

[События](#) > [События мыши](#)

[.mouseover\(\)](#)

Bind an event handler to the “mouseover” JavaScript event, or trigger that event on an element.

[События](#) > [События мыши](#)

[.mouseup\(\)](#)

Устанавливает обработчик возвращения кнопки мыши в ненажатое состояние, либо, запускает это событие.

[Перемещения](#) > [Обход элементов](#)

[.next\(\)](#)

Осуществляет поиск элементов, лежащих непосредственно после заданных элементов (по одному для каждого из заданных).

[Перемещения](#) > [Обход элементов](#)

[.nextAll\(\)](#)

Осуществляет поиск всех элементов, лежащих после заданных элементов.

[Перемещения](#) > [Обход элементов](#)

[.nextUntil\(\)](#)

Осуществляет поиск всех элементов, лежащих начиная от одних заданных элементов и заканчивая другими.

[Перемещения](#) > [Фильтрация](#) | [Перемещения](#) > [Обход](#)

[.not\(\)](#)

Фильтрует набор выбранных элементов, исключая из него определенные элементы.

[События](#) > [Прикрепление обработчиков](#)

[.off\(\)](#)

Удаляет с выбранных элементов страницы обработчики событий, установленные с помощью метода .on().

[CSS](#) | [Смещение](#) | [Манипуляция DOM](#) > [Свойства стилей](#)

[.offset\(\)](#)

С помощью этой функции, можно узнавать координаты элемента на странице. Кроме этого, с помощью `offset()`, можно изменить координаты элемента.

[Смещение](#) | [Перемещения](#) > [Обход элементов](#)

[.offsetParent\(\)](#)

Возвращает ближайшего предка (т.е. из числа: родитель, прародитель, и так далее, до начало дерева DOM), у которого задан тип позиционирования (`position` равен `relative` или `absolute` или `fixed`). Элементы с заданным позиционированием играют особую роль, поскольку именно относительно них происходит позиционирование всех элементов внутри.

[События](#) > [Прикрепление обработчиков](#)

[.on\(\)](#)

Устанавливает обработчики событий на выбранные элементы страницы.

[События](#) > [Прикрепление обработчиков](#)

[.one\(\)](#)

Устанавливает обработчик события выбранным элементам страницы. Особенностью метода является то, что обработчик будет вызван не более одного раза, на каждом из элементов.

[CSS](#) | [Размеры элементов](#) | [Манипуляция DOM](#) > [Свойства стилей](#)

[.outerHeight\(\)](#)

Метод `outerHeight()` возвращает высоту элемента (включая заполнение и границу).

[CSS](#) | [Размеры элементов](#) | [Манипуляция DOM](#) > [Свойства стилей](#)

[.outerWidth\(\)](#)

Ширина элемента с учетом внутренних отступов, `border-width` и (опционально) внешних отступов (`margin`).

[Перемещения](#) > [Обход элементов](#)

[.parent\(\)](#)

Осуществляет поиск родительских элементов всех заданных элементов.

[Перемещения](#) > [Обход элементов](#)

[.parents\(\)](#)

Осуществляет поиск всех предков выбранных элементов, то есть, не только прямых родителей, но и прародителей, прапрародителей и так далее, до начало дерева DOM.

[Перемещения](#) > [Обход элементов](#)

[.parentsUntil\(\)](#)

Осуществляет поиск всех предков у выбранных элементов (то есть, не только прямых родителей, но и прародителей, прапрародителей и так далее, до начало дерева DOM), вплоть до определенного элемента.

[CSS](#) | [Смещение](#) | [Манипуляция DOM](#) > [Свойства стилей](#)

[.position\(\)](#)

Получает значение позиции элемента `top` и `left` относительно отступов его родителя.

[Манипуляция DOM](#) > [DOM Вставка внутрь](#)

[.prepend\(\)](#)

Функция добавляет содержимое в начало определенных элементов.

[Манипуляция DOM](#) > [DOM Вставка внутрь](#)

[.prependTo\(\)](#)

Функция добавляет все выбранные элементы внутрь другого набора элементов перед существующим контентом.

[Перемещения](#) > [Обход элементов](#)

[.prev\(\)](#)

Для каждого из выбранных элементов находит предшествующий ему элемент (но только если он лежит на том же уровне иерархии DOM).

[Перемещения](#) > [Обход элементов](#)

[.prevAll\(\)](#)

Осуществляет поиск всех элементов, лежащих перед заданными элементами.

[Перемещения](#) > [Обход элементов](#)

[.prevUntil\(\)](#)

Осуществляет поиск всех элементов, лежащих начиная от одних заданных элементов и заканчивая другими.

[Объекты отложенного выполнения](#)

[.promise\(\)](#)

Создает заместителя deferred-объекта (объект с аналогичным набором методов, за исключением отсутствия методов изменения состояния объекта. Предоставление заместителя вместо исходного объекта, гарантирует, что его состояние нельзя будет изменить из вне).

[Атрибуты](#) | [Манипуляция DOM](#) > [Общие атрибуты](#)

[.prop\(\)](#)

Возвращает или изменяет значение свойств выбранных элементов страницы.

[Внутренние функции](#)

[.pushStack\(\)](#)

Добавляет коллекцию DOM элементов в jQuery стэк.

[Эффекты](#) > [Пользовательские](#) | [Данные](#) | [Инструменты](#)

[.queue\(\)](#)

Возвращает или изменяет (в зависимости от переданных параметров) очередь функций.

[События](#) > [Загрузка документа](#)

[.ready\(\)](#)

Устанавливает обработчик готовности дерева DOM.

[Манипуляция DOM](#) > [DOM Удаление](#)

[.remove\(\)](#)

Метод для удаления элементов страницы.

[Атрибуты](#) | [Манипуляция DOM](#) > [Общие атрибуты](#)

[.removeAttr\(\)](#)

Метод для удаления атрибутов (таких как id, class, title и.т.д) у элементов страницы.

[Атрибуты](#) | [Манипуляция DOM](#) > [Class атрибут](#) | [CSS](#)

[.removeClass\(\)](#)

Удаляет заданные классы у элементов на странице.

[Данные](#) | [Разное](#) > [Хранилище данных](#)

[.removeData\(\)](#)

Удаляет пользовательские переменные, установленные методом .data() у выбранных элементов страницы.

[Атрибуты](#) | [Манипуляция DOM](#) > [Общие атрибуты](#)

[.removeProp\(\)](#)

Удаляет свойство у каждого выбранного элемента страницы.

[Манипуляция DOM](#) > [DOM Замена](#)

[.replaceAll\(\)](#)

С помощью этой функции можно заменять элементы страницы новыми элементами или уже существующими.

[Манипуляция DOM](#) > [DOM Замена](#)

[.replaceWith\(\)](#)

С помощью этой функции можно заменять элементы страницы новыми элементами или уже существующими.

[События](#) > [События браузера](#)

[.resize\(\)](#)

Устанавливает обработчик изменения размеров окна браузера, либо, запускает это событие.

[События](#) > [События браузера](#)

[.scroll\(\)](#)

Устанавливает обработчик "прокрутки" элементов документа, либо, запускает это событие.

[CSS](#) | [Смещение](#) | [Манипуляция DOM](#) > [Свойства стилей](#)

[.scrollLeft\(\)](#)

С помощью этой функции можно получить или изменить величину горизонтального скроллинга элементов на странице.

[CSS](#) | [Смещение](#) | [Манипуляция DOM](#) > [Свойства стилей](#)

[.scrollTop\(\)](#)

С помощью этой функции можно получить или изменить величину вертикального скроллинга элементов на странице.

[События](#) > [События формы](#) | [Формы](#)

[.select\(\)](#)

Устанавливает обработчик выделения текста внутри элемента или , либо, запускает это событие.

[Устаревшие методы](#) > [Версия 1.7](#) | [Внутренние функции](#) | [Свойства](#) > [Глобальный объект jQuery](#) | [Удаленные](#)

[.selector](#)

Селектор передаваемый в метод jQuery (), если таковой использовался, при создании исходного набора элементов.

[Формы](#) | [Ajax](#) > [Вспомогательные методы](#)

[.serialize\(\)](#)

Возвращает строку, с именами и значениями выбранных элементов формы. Эта строка будет иметь формат параметров url-запроса.

[Формы](#) | [Ajax](#) > [Вспомогательные методы](#)

[.serializeArray\(\)](#)

Упорядочивает все формы и элементы на форме (подобно .serialize()), но возвращает данные, которые можно использовать в структуре JSON

[Эффекты](#) > [Базовые](#)

[.show\(\)](#)

С помощью этой функции можно плавно показывать и скрывать выбранные элементы на странице, за счет изменения размера и прозрачности.

[Перемещения](#) > [Обход элементов](#)

[.siblings\(\)](#)

Осуществляет поиск элементов, являющихся соседними для выбранных элементов (под соседними понимаются элементы, которые имеют общего родителя). При этом, сами выбранные элементы в результат не включаются.

[.size\(\)](#)

Возвращает количество выбранных элементов.

[Перемещения](#) > [Фильтрация](#)

[.slice\(\)](#)

Фильтрует набор выбранных элементов, оставляя только те элементы, чьи индексы лежат в определенной области (например от 0 до 5).

[Эффекты](#) > [Скольжение](#)

[.slideDown\(\)](#)

С помощью этой функции можно показывать и скрывать выбранные элементы на странице, за счет плавного разворачивания и сворачивания.

[Эффекты](#) > [Скольжение](#)

[.slideToggle\(\)](#)

Вызов этого метода приводит к плавному сворачиванию (если элемент развернут) или разворачиванию (если элемент свернут) выбранных элементов на странице.

[Эффекты](#) > [Скольжение](#)

[.slideUp\(\)](#)

С помощью этой функции можно показывать и скрывать выбранные элементы на странице, за счет плавного разворачивания и сворачивания.

[Эффекты](#) > [Пользовательские](#)

[.stop\(\)](#)

Останавливает выполнение текущей анимации.

[События](#) > [События формы](#) | [Формы](#)

[.submit\(\)](#)

Устанавливает обработчик отправки формы на сервер, либо запускает это событие.

[Манипуляция DOM](#) > [DOM Вставка внутрь](#)

[.text\(\)](#)

Возвращает или изменяет текстовое содержимое выбранных элементов страницы.

[Разное](#) > [DOM методы элементов](#)

[.toArray\(\)](#)

Возвращает массив всех DOM-элементов, хранящихся в объекте jQuery.

[Устаревшие методы](#) > [Версия 1.8](#) | [События](#) > [События мыши](#) | [Удаленные](#)

[.toggle\(\)](#)

Поочередно выполняет одно из нескольких заданных действий.

[Эффекты](#) > [Базовые](#)

[.toggle\(\)](#)

Показывает или скрывает элементы в наборе объекта jQuery.

[Атрибуты](#) | [Манипуляция DOM](#) > [Class атрибут](#) | [CSS](#)

[.toggleClass\(\)](#)

Добавляет или удаляет заданный класс(ы) по принципу переключателя (добавляет, если элемент не содержит класса, и удаляет, если класс есть).

[События](#) > [Прикрепление обработчиков](#)

[.trigger\(\)](#)

Вызывает событие у выбранных элементов, что приводит к запуску обработчиков этого события.

[События](#) > [Прикрепление обработчиков](#)

[.triggerHandler\(\)](#)

Вызывает выполнение обработчиков заданного события у выбранных элементов. Само событие, при этом, не происходит.

[Устаревшие методы](#) > [Версия 3.0](#) | [События](#) > [Прикрепление обработчиков](#)

[.unbind\(\)](#)

Метод необходим для удаления обработчиков событий, установленных на выбранных элементах методами bind(), one() или методами с узким назначением (click(), focus() и.т.д.).

[Устаревшие методы](#) > [Версия 3.0](#) | [События](#) > [Прикрепление обработчиков](#)

[.undelegate\(\)](#)

У выбранных элементов страницы, удаляет обработчики событий, установленные методом delegate().

[Устаревшие методы](#) > [Версия 1.8](#) | [События](#) > [Загрузка документа](#) | [Удаленные](#)

[.unload\(\)](#)

Устанавливает обработчик выхода со страницы (при переходе по ссылке, закрытии браузера и.т.д.) для объекта window.

[Манипуляция DOM](#) > [DOM Вставка вокруг](#) | [Манипуляция DOM](#) > [DOM Удаление](#)

[.unwrap\(\)](#)

Удаляет родительские элементы у выбранных элементов, при этом, их содержимое останется на своих местах.

[Атрибуты](#) | [Формы](#) | [Манипуляция DOM](#) > [Общие атрибуты](#)

[.val\(\)](#)

Метод позволяет получать и изменять значения элементов форм. Для элементов input это значение атрибута value; для списков выбора (select) – значение value выбранного элемента (в случае множественного выбора – массив значений); в случае с textarea, метод .val() будет работать непосредственно с содержимым тега textarea.

[CSS](#) | [Размеры элементов](#) | [Манипуляция DOM](#) > [Свойства стилей](#)

[.width\(\)](#)

Функция возвращает ширину элемента. Кроме этого, с помощью width(), можно установить новое значение ширины.

[Манипуляция DOM](#) > [DOM Вставка вокруг](#)

[.wrap\(\)](#)

Функция помещает заданное содержимое вокруг выбранных элементов (как бы "обертывая" их).

[Манипуляция DOM](#) > [DOM Вставка вокруг](#)

[.wrapAll\(\)](#)

Функция помещает заданное содержимое вокруг выбранных элементов (как бы "обертывая" их).

[Манипуляция DOM](#) > [DOM Вставка вокруг](#)

[.wrapInner\(\)](#)

"Обертывает" содержимое выбранных элементов заданными html-элементами. Таким образом получается, что все выбранные объекты "обертываются изнутри".

[Объекты обратного вызова](#)

[callbacks.add\(\)](#)

Добавляет заданную функцию/функции в список функций callbacks-объекта.

[Объекты обратного вызова](#)

[callbacks.disable\(\)](#)

Блокирует текущий объект `callbacks`, отключая возможность выполнения его списка функций.

[Объекты обратного вызова](#)

[`callbacks.disabled\(\)`](#)

Определяет, был ли отключен список обратных вызовов.

[Объекты обратного вызова](#)

[`callbacks.empty\(\)`](#)

Удаляет все функции из текущего объекта `callbacks`.

[Объекты обратного вызова](#)

[`callbacks.fire\(\)`](#)

Выполняет все функции объекта `callbacks` (если тот не отключен).

[Объекты обратного вызова](#)

[`callbacks.fireWith\(\)`](#)

Выполняет все функции объекта `callbacks` с заданным контекстом.

[Объекты обратного вызова](#)

[`callbacks.fired\(\)`](#)

Проверяет, выполнялся ли текущий объект `callbacks` хотя бы один раз.

[Объекты обратного вызова](#)

[`callbacks.has\(\)`](#)

Проверяет наличие заданной функции в объекте `callbacks`.

[Объекты обратного вызова](#)

[`callbacks.lock\(\)`](#)

Блокирует список обратных вызовов в текущем состоянии.

[Объекты обратного вызова](#)

[`callbacks.locked\(\)`](#)

Определяет, был ли заблокирован список обратных вызовов.

[Объекты обратного вызова](#)

[`callbacks.remove\(\)`](#)

Удаляет заданную функцию из объекта `callbacks`.

[Объекты отложенного выполнения](#)

[`deferred.always\(\)`](#)

С помощью этого метода можно установить обработчики перехода объекта `deferred` в состояние удачного выполнения, ошибки выполнения (`resolved` или `rejected`).

[Объекты отложенного выполнения](#)

[`deferred.done\(\)`](#)

С помощью этого метода можно установить обработчики перехода объекта `deferred` в состояние удачного выполнения, ошибки выполнения (`resolved` или `rejected`).

[Объекты отложенного выполнения](#)

[`deferred.fail\(\)`](#)

С помощью этого метода можно установить обработчики перехода объекта `deferred` в состояние удачного выполнения, ошибки выполнения (`resolved` или `rejected`).

[Объекты отложенного выполнения](#) | [Устаревшие методы](#) > [Версия 1.7](#) | [Удаленные](#)

[`deferred.isRejected\(\)`](#)

Данный метод позволяет проверить, находится ли `deferred` объект в состоянии `rejected` ("ошибки выполнения").

[Объекты отложенного выполнения](#) | [Устаревшие методы](#) > [Версия 1.7](#) | [Удаленные](#)

[deferred.isResolved\(\)](#)

Данный метод позволяет проверить, находится ли deferred объект в состоянии resolved ("удачно выполненное").

Объекты отложенного выполнения

[deferred.notify\(\)](#)

Вызывает событие промежуточного выполнения deferred (прогресса выполнения).

Объекты отложенного выполнения

[deferred.notifyWith\(\)](#)

Вызывает событие частичного выполнения deferred (его прогресса выполнения), с указанием и контекста выполнения (значение переменной this) обработчика.

Объекты отложенного выполнения | Устаревшие методы > Версия 1.8

[deferred.pipe\(\)](#)

Позволяет проводить предварительную обработку параметров, которые будут передаваться в обработчики смены состояния объекта deferred.

Объекты отложенного выполнения

[deferred.progress\(\)](#)

Этот метод устанавливает обработчики прогресса выполнения объекта deferred.

Объекты отложенного выполнения

[deferred.promise\(\)](#)

Создает заместителя deferred-объекта (объект с аналогичным набором методов, за исключением отсутствия методов изменения состояния объекта).

Объекты отложенного выполнения

[deferred.reject\(\)](#)

С помощью этого метода можно изменить состояние объекта deferred на rejected ("ошибка выполнения").

Объекты отложенного выполнения

[deferred.rejectWith\(\)](#)

С помощью этого метода переводят объект deferred из состояния "ошибка выполнения", с указанием и контекста выполнения (значение переменной this) обработчиков.

Объекты отложенного выполнения

[deferred.resolve\(\)](#)

С помощью этого метода можно изменить состояние объекта deferred на resolved ("удачно выполненное").

Объекты отложенного выполнения

[deferred.resolveWith\(\)](#)

Resolve a Deferred object and call any doneCallbacks with the given context and args.

Объекты отложенного выполнения

[deferred.state\(\)](#)

Возвращает текущее состояние объекта deferred в виде строки.

Объекты отложенного выполнения

[deferred.then\(\)](#)

С помощью этого метода можно установить обработчики перехода объекта deferred в состояние удачного выполнения, ошибки выполнения (resolved или rejected). then() регистрирует 2 обработчика сразу.

События > Объект событие

[event.currentTarget](#)

Содержит DOM-элемент, событие которого обрабатывается. Внутри обработчика, `currentTarget` всегда совпадает с переменной `this`. Однако, этот элемент может не являться источником события, поскольку оно могло быть передано от дочернего элемента, в результате "всплытия" события, вверх по иерархии DOM. Первоначальный источник события содержится в `event.target`.

[События](#) > [Объект событие](#)

[event.data](#)

Дополнительные данные, которые передаются обработчику при его установке.

[События](#) > [Объект событие](#) | [События](#)

[event.delegateTarget](#)

Содержит DOM-элемент, на который действительно установлен сработавший обработчик. Это свойство имеет смысл, если обработчик установлен делегированно. Если обработчик установлен непосредственно, то значение `event.delegateTarget` будет совпадать с `event.currentTarget`.

[События](#) > [Объект событие](#)

[event.isDefaultPrevented\(\)](#)

Определяет, вызывался ли метод `event.preventDefault()` на данном объекте событий.

[События](#) > [Объект событие](#)

[event.isImmediatePropagationStopped\(\)](#)

Определяет, вызывался ли метод `event.stopImmediatePropagation()` на данном объекте событий.

[События](#) > [Объект событие](#)

[event.isPropagationStopped\(\)](#)

Определяет, вызывался ли метод `event.stopPropagation()` на данном объекте событий.

[События](#) > [Объект событие](#)

[event.metaKey](#)

Определяет была ли зажата клавиша META в момент действия события.

[События](#) > [Объект событие](#)

[event.namespace](#)

Содержит строковое значение пространства имен произошедшего события.

[События](#) > [Объект событие](#)

[event.pageX](#)

Координаты курсора мыши относительно левого верхнего угла документа.

[События](#) > [Объект событие](#)

[event.pageY](#)

Координаты курсора мыши относительно левого верхнего угла документа.

[События](#) > [Объект событие](#)

[event.preventDefault\(\)](#)

Отменяет выполнение события. Например, при "клике" по ссылке, переход по ней не произойдет, если внутри обработчика будет вызван этот метод.

[События](#) > [Объект событие](#)

[event.relatedTarget](#)

Не текущий DOM-элемент, участвующий в событии. Для события `mouseout` будет содержать элемент, куда переместился курсор, а для `mouseover` элемент, откуда курсор пришел.

[События](#) > [Объект событие](#)

[event.result](#)

Содержит значение, которое возвратил предыдущий обработчик этого события. Если предыдущего обработчика нет, возвратит undefined.

События > Объект событие

[event.stopImmediatePropagation\(\)](#)

Предотвращает выполнение следующих обработчиков текущего события, а также, дальнейшую передачу события вверх по иерархии дерева DOM. Если требуется лишь остановить передачу события вверх по иерархии, а другие обработчики на текущем элементе останавливать не требуется, воспользуйтесь методом event.stopPropagation().

События > Объект событие

[event.stopPropagation\(\)](#)

Предотвращает дальнейшую передачу текущего события, вверх по иерархии дерева DOM (некоторые события, такие как click, после выполнения на самом элементе, передают его родительским, что повторяется до самого основания дерева DOM).

События > Объект событие

[event.target](#)

DOM-элемент, являющийся источником события.

События > Объект событие

[event.timeStamp](#)

Содержит время, когда было произведено событие. Время представлено количеством секунд, прошедших с 1.01.1970.

События > Объект событие

[event.type](#)

Тип события.

События > Объект событие

[event.which](#)

При возникновении события, связанного с нажатием клавиш клавиатуры или кнопок на странице, это поле будет содержать информацию о нажатых клавишах или кнопках. В отличие от стандартных event.keyCode и event.charCode, содержимое в event.which кроссбраузерное.

Ядро

[jQuery\(\)](#)

Функция jQuery() является, пожалуй основной функцией в библиотеке. Она обычно, вызывается намного чаще других и поэтому для нее существует короткая форма записи — \$(). Функция jQuery() позволяет производить поиск элементов на странице, создавать новые элементы по заданному HTML тексту, а так же кое что еще. Возвращает объект jQuery.

Объекты обратного вызова

[jQuery.Callbacks\(\)](#)

Универсальный объект списка обратных вызовов, обеспечивающий эффективный способ управления списками обратных вызовов.

Объекты отложенного выполнения

[jQuery.Deferred\(\)](#)

Фабричная функция, которая возвращает служебный объект с методами регистрации нескольких обратных вызовов в очереди обратного вызова, вызова очереди колбэков и передаи состояние success или failure любой синхронной или асинхронной функции.

Ajax > Низкоуровневый интерфейс

[jQuery.ajax\(\)](#)

Выполняет асинхронный HTTP (Ajax) запрос.

[Ajax](#) > [Низкоуровневый интерфейс](#)

[jQuery.ajaxPrefilter\(\)](#)

Обрабатывает пользовательские Ajax параметры или изменяет существующие параметры прежде чем каждый запрос будет отправлен и прежде чем они будут обработаны методом \$.ajax()

[Ajax](#) > [Низкоуровневый интерфейс](#)

[jQuery.ajaxSetup\(\)](#)

Устанавливает значения по умолчанию для Ajax запросов

[Ajax](#) > [Низкоуровневый интерфейс](#)

[jQuery.ajaxTransport\(\)](#)

Создает объект, который обрабатывает фактическую передачу данных Ajax

[Устаревшие методы](#) > [Версия 1.3](#) | [Удаленные](#) | [Инструменты](#)

[jQuery.boxModel](#)

Указывает, отображается ли текущая страница в браузере пользователя с помощью W3C Box Model. Это свойство было удалено в jQuery 1.8. Попробуйте вместо этого использовать функцию обнаружения.

[Устаревшие методы](#) > [Версия 1.3](#) | [Свойства](#) > [Глобальный объект jQuery](#) | [Удаленные](#) | [Инструменты](#)

[jQuery.browser](#)

Содержит флаги для UserAgent, считанные из navigator.userAgent. Это свойство было удалено в jQuery 1.9 и доступно только через плагин jQuery.migrate. Попробуйте вместо этого использовать функцию обнаружения.

[Инструменты](#)

[jQuery.contains\(\)](#)

Проверяет, является ли элемент DOM потомком другого DOM элемента.

[CSS](#)

[jQuery.cssHooks](#)

Подключается непосредственно в jQuery, чтобы переопределить способы извлечения или установки определенных CSS свойств, нормализации присвоения CSS имен или создания настраиваемых свойств.

[CSS](#) | [Манипуляция DOM](#) > [Свойства стилей](#)

[jQuery.cssNumber](#)

An object containing all CSS properties that may be used without a unit. The .css() method uses this object to see if it may append px to unitless values.

[Данные](#) | [Инструменты](#)

[jQuery.data\(\)](#)

С помощью этого метода, появляется возможность привязывать собственные переменные к любым элементам страницы.

[Данные](#) | [Инструменты](#)

[jQuery.dequeue\(\)](#)

Execute the next function on the queue for the matched element.

[Инструменты](#)

[jQuery.each\(\)](#)

A generic iterator function, which can be used to seamlessly iterate over both objects and arrays. Arrays and array-like objects with a length property (such as a function's arguments object) are iterated by numeric index, from 0 to length-1. Other objects are iterated via their named properties.

Внутренние функции

[jQuery.error\(\)](#)

Takes a string and throws an exception containing it.

CSS

[jQuery.escapeSelector\(\)](#)

Escapes any character that has a special meaning in a CSS selector.

Инструменты

[jQuery.extend\(\)](#)

Объединяет содержимое двух или более заданных javascript-объектов. Результат объединения записывается в первый из этих объектов (он же будет возвращен функцией, после ее выполнения).

Инструменты

[jQuery.fn.extend\(\)](#)

Merge the contents of an object onto the jQuery prototype to provide new jQuery instance methods.

Эффекты > Пользовательские | Устаревшие методы > Версия 3.0 | Свойства > Глобальный объект jQuery

[jQuery.fx.interval](#)

The rate (in milliseconds) at which animations fire.

Эффекты > Пользовательские | Свойства > Глобальный объект jQuery

[jQuery.fx.off](#)

Globally disable all animations.

Аjax > Сокращенные методы

[jQuery.get\(\)](#)

Загружает данные с сервера при помощи HTTP GET запроса

Аjax > Сокращенные методы

[jQuerygetJSON\(\)](#)

Загружает JSON данные с сервера при помощи GET HTTP запроса

Аjax > Сокращенные методы

[jQuery.getScript\(\)](#)

Загружает JavaScript файл с сервера при помощи GET HTTP запроса и затем исполняет его.

Инструменты

[jQuery.globalEval\(\)](#)

Выполняет заданный скрипт в глобальной области видимости.

Инструменты

[jQuery.grep\(\)](#)

Ищет в заданном массиве элементы, удовлетворяющие условиям фильтрующей функции. Возвращает массив с найденными элементами (в исходный массив изменения не вносятся).

Данные

[jQuery.hasData\(\)](#)

Проверяет, закреплены ли данные за указанным элементом страницы.

Ядро

[jQuery.holdReady\(\)](#)

Holds or releases the execution of jQuery's ready event.

Манипуляция DOM

[jQuery.htmlPrefilter\(\)](#)

Modify and filter HTML strings passed through jQuery manipulation methods.

Инструменты

[jQuery.inArray\(\)](#)

Ищет заданный элемент в массиве. Возвращает индекс этого элемента или -1 в случае его отсутствия.

Инструменты

[jQuery.isArray\(\)](#)

Проверяет, является ли заданный элемент массивом.

Инструменты

[jQuery.isEmptyObject\(\)](#)

Проверяет, является ли заданный объект пустым.

Инструменты

[jQuery.isFunction\(\)](#)

Проверяет, является ли заданный элемент функцией.

Инструменты

[jQuery.isNumeric\(\)](#)

Проверяет, является ли заданный аргумент числом или может ли быть к нему сведен (например строка с числовым значением).

Инструменты

[jQuery.isPlainObject\(\)](#)

Проверяет, является ли заданный элемент пользовательским объектом, то есть он был задан пользователем именно в роли объекта (посредством "{}" или "new Object")(дело в том, что многие величины (например строки) в javascript являются объектами, однако они не задаются объектами явным образом).

Инструменты

[jQuery.isWindow\(\)](#)

Проверяет, является ли заданный элемент объектом типа window (объекты фреймов тоже принадлежат к этому типу).

Инструменты

[jQuery.isXMLDoc\(\)](#)

Проверяет, находится ли DOM-объект внутри XML-документа (или сам является XML-документом).

Инструменты

[jQuery.makeArray\(\)](#)

Конвертирует массивоподобные объекты, в массивы.

Инструменты

[jQuery.map\(\)](#)

Выполняет заданную функцию для каждого элемента массива (если задан массив) или каждого поля объекта (если задан объект) в отдельности. Значения, полученные в результате выполнения этой функции помещаются в новый массив и возвращаются как результат работы функции.

Инструменты

[jQuery.merge\(\)](#)

Объединяет содержимое двух массивов. Результат объединения записывается в первый из этих массивов (он же будет возвращен в качестве результата выполнения функции).

Ядро | Разное > Методы установки

[jQuery.noConflict\(\)](#)

Relinquish jQuery's control of the \$ variable.

[Инструменты](#)

[jQuery.noop\(\)](#)

Пустая функция (т. е. которая ничего не делает).

[Инструменты](#)

[jQuery.now\(\)](#)

Return a number representing the current time.

[Разное](#) > [Манипуляция коллекциями](#) | [Формы](#) | [Аjax](#) > [Вспомогательные методы](#)

[jQuery.param\(\)](#)

Преобразует объект, массив или массив объектов в строку, которая соответствует формату передачи данных через url.

[Инструменты](#)

[jQuery.parseHTML\(\)](#)

Parses a string into an array of DOM nodes.

[Инструменты](#)

[jQuery.parseJSON\(\)](#)

Конвертирует строку с json-данными в javascript-объект.

[Инструменты](#)

[jQuery.parseXML\(\)](#)

Конвертирует строку с xml-данными в xml-документ.

[Аjax](#) > [Сокращенные методы](#)

[jQuery.post\(\)](#)

Загружает данные с сервера при помощи HTTP POST запроса.

[События](#) > [Прикрепление обработчиков](#) | [Инструменты](#)

[jQuery.proxy\(\)](#)

По заданной функции, создает другую такую же, внутри которой переменная this будет равна заданному значению.

[Данные](#) | [Инструменты](#)

[jQuery.queue\(\)](#)

Show or manipulate the queue of functions to be executed on the matched element.

[Ядро](#)

[jQuery.readyException\(\)](#)

Handles errors thrown synchronously in functions wrapped in jQuery().

[Данные](#) | [Инструменты](#)

[jQuery.removeData\(\)](#)

Удаляет пользовательские переменные, установленные методом .data() у выбранных элементов страницы.

[Эффекты](#) > [Пользовательские](#)

[jQuery.speed](#)

Creates an object containing a set of properties ready to be used in the definition of custom animations.

[Ядро](#) | [Устаревшие методы](#) > [Версия 1.7](#) | [Удаленные](#)

[jQuery.sub\(\)](#)

Creates a new copy of jQuery whose properties and methods can be modified without affecting the original jQuery object.

[Устаревшие методы](#) > [Версия 1.9](#) | [Свойства](#) > [Глобальный объект jQuery](#) | [Инструменты](#)

jQuery.support

A collection of properties that represent the presence of different browser features or bugs. Intended for jQuery's internal use; specific properties may be removed when they are no longer needed internally to improve page startup performance. For your own project's feature-detection needs, we strongly recommend the use of an external library such as Modernizr instead of dependency on properties in jQuery.support.

[Инструменты](#)

jQuery.trim()

Удаляет символы пробелов, табов и переносов строк из начала и конца строки.

[Инструменты](#)

jQuery.type()

Определяет класс заданного элемента (речь идет о внутренних классах javascript: string, boolean и т.д.).

[Инструменты](#)

jQuery.unique()

Сортирует массив с DOM-элементами, выстраивая их в порядке расположения в DOM, а также удаляя повторения.

[Инструменты](#)

jQuery.uniqueSort()

Sorts an array of DOM elements, in place, with the duplicates removed. Note that this only works on arrays of DOM elements, not strings or numbers.

[Ядро](#) | [Объекты отложенного выполнения](#)

jQuery.when()

На основе нескольких заданных объектов (обычно deferred), функция \$.when() создает новый deferred-объект, следящий за состоянием всех заданных. Стоит уточнить, что \$.when() возвращает promise-версию (заместителя) deferred-объекта.

[Селекторы](#) > [Базовые](#)

Групповые селекторы ("selector1, selector2, selectorN")

Соответствует элементам, удовлетворяющим хотя бы одному из селекторов (first, second, ...).

[Селекторы](#) > [Иерархия](#)

Дочерний селектор ("parent > child")

Соответствует элементам, удовлетворяющим селектору child, которые при этом являются дочерними (т.е. находятся на первом уровне вложенности) для элементов, удовлетворяющих селектору parent. (В отличие от него, селектор (E F) позволяет искать элементы на всех уровнях вложенности).

[Селекторы](#) > [Иерархия](#)

Нисходящий селектор ("ancestor descendant")

Соответствует элементам удовлетворяющим селектору inner, которые являются потомками (лежат внутри) элементов, удовлетворяющих селектору outer. В отличии от дочернего селектора (outer > inner), с помощью которого производится поиск элементов только на первом уровне вложенности в outer, нисходящий селектор задает поиск на всех уровнях вложенности.

[Селекторы](#) > [Базовые фильтры](#) | [Селекторы](#) > [jQuery расширение](#)

Селектор :animated

Соответствует элементам, которые в данный момент участвуют в анимации.

[Селекторы](#) > [Форма](#) | [Селекторы](#) > [jQuery расширение](#)

[Селектор :button](#)

Соответствует элементам, которые являются кнопками: input-элементы с типом button или button-элементы.

[Селекторы > Форма](#) | [Селекторы > jQuery расширение](#)

[Селектор :checkbox](#)

Соответствует элементам, которые являются флажками (input-элементы с типом checkbox).

[Селекторы > Форма](#)

[Селектор :checked](#)

Соответствует всем выбранным элементам формы. Это могут быть выбранные флажки (checkbox), переключатели (radio) или элементы списка (select).

[Селекторы > Фильтры содержимого](#)

[Селектор :contains\(\)](#)

Соответствует элементам, которые содержат строку text. Чувствителен к регистру.

[Селекторы > Форма](#)

[Селектор :disabled](#)

Соответствует всем заблокированным элементам формы (элементы с атрибутом disabled).

[Селекторы > Фильтры содержимого](#)

[Селектор :empty](#)

Соответствует элементам, которые не содержат текста и других элементов. Наличие одного только пробела уже считается текстом.

[Селекторы > Форма](#)

[Селектор :enabled](#)

Соответствует всем активным (незаблокированным от пользователя) элементам формы (элементы без атрибута disabled).

[Селекторы > Базовые фильтры](#) | [Селекторы > jQuery расширение](#)

[Селектор :eq\(\)](#)

Селектор соответствует элементу, занимающему позицию под номером index, среди уже выбранных с помощью селектора someSelector элементов. Нумерация элементов начинается с 0.

[Селекторы > Базовые фильтры](#) | [Селекторы > jQuery расширение](#)

[Селектор :even](#)

Соответствует элементам с четными номерами позиций, в наборе выбранных элементов. (вариант с нечетными — :odd). Поскольку счет элементов начинается с 0, :even будет соответствовать первому, третьему и т. д. элементам.

[Селекторы > Форма](#) | [Селекторы > jQuery расширение](#)

[Селектор :file](#)

Соответствует элементам, которые являются полями для загрузки файлов (input-элементы с типом file).

[Селекторы > Базовые фильтры](#) | [Селекторы > jQuery расширение](#)

[Селектор :first](#)

Соответствует первому элементу, из всех выбранных с помощью селектора someSelector. Эквивалентен селектору :eq(0).

[Селекторы > Фильтры потомков](#)

[Селектор :first-child](#)

Соответствует элементам, которые идут первыми в своих непосредственных предках.

[Селекторы > Фильтры потомков](#)

[Селектор :first-of-type](#)

Фильтрует набор выбранных элементов, оставляя только те элементы, которые первыми будут встречены в своих родителях.

[Селекторы > Базовые фильтры](#) | [Селекторы > Форма](#)

[Селектор :focus](#)

Соответствует всем элементам формы, находящимся в фокусе (в один момент времени, в фокусе может находиться только один или ни одного элемента на странице).

[Селекторы > Базовые фильтры](#) | [Селекторы > jQuery расширение](#)

[Селектор :gt\(\)](#)

Фильтрует набор выбранных элементов, оставляя только те, индекс которых превышает n.

[Селекторы > Фильтры содержимого](#) | [Селекторы > jQuery расширение](#)

[Селектор :has\(\)](#)

Соответствует элементам, которые обладают потомками, удовлетворяющими селектору selector.

[Селекторы > Базовые фильтры](#) | [Селекторы > jQuery расширение](#)

[Селектор :header](#)

Соответствует элементам, которые являются заголовками (h1, h2, ...).

[Селекторы > jQuery расширение](#) | [Селекторы > Фильтр видимости](#)

[Селектор :hidden](#)

Соответствует всем скрытым элементам страницы.

[Селекторы > Форма](#) | [Селекторы > jQuery расширение](#)

[Селектор :image](#)

Соответствует элементам с изображениями, при нажатии на которые происходит отправка данных формы на сервер (input-элементы с типом image).

[Селекторы > Форма](#) | [Селекторы > jQuery расширение](#)

[Селектор :input](#)

Соответствует всем элементам формы (элементы, представленные тегами input, textarea, select или button).

[Селекторы > Базовые фильтры](#)

[Селектор :lang\(\)](#)

Соответствует элементам, в которых указан подходящий язык содержимого. В параметре language указывается код языка.

[Селекторы > Базовые фильтры](#) | [Селекторы > jQuery расширение](#)

[Селектор :last](#)

Соответствует последнему элементу, из всех выбранных с помощью селектора someSelector.

[Селекторы > Фильтры потомков](#)

[Селектор :last-child](#)

Соответствует элементам, которые идут последними в своих непосредственных предках.

[Селекторы > Фильтры потомков](#)

[Селектор :last-of-type](#)

Фильтрует набор выбранных элементов, оставляя только те элементы, которые последними будут встречены в своих родителях.

[Селекторы > Базовые фильтры](#) | [Селекторы > jQuery расширение](#)

[Селектор :lt\(\)](#)

Фильтрует набор выбранных элементов, оставляя только те, индекс которых меньше n. Не забывайте, что индексирование начинается с 0.

[Селекторы](#) > [Базовые фильтры](#)

Селектор :not()

Исключает элементы удовлетворяющие селектору selector из найденных элементов.

[Селекторы](#) > [Фильтры потомков](#)

Селектор :nth-child()

Соответствует элементам, которые расположены в своих непосредственных предках по определенным условиям.

[Селекторы](#) > [Фильтры потомков](#)

Селектор :nth-last-child()

Соответствует элементам, которые расположены в своих непосредственных предках по определенным условиям. :nth-child-last() рассматривает позиции элементов в обратном порядке (т. е. с конца).

[Селекторы](#) > [Фильтры потомков](#)

Селектор :nth-last-of-type()

Рассматривает позиции элементов в обратном порядке (т. е. с конца).

[Селекторы](#) > [Фильтры потомков](#)

Селектор :nth-of-type()

Фильтрующий селектор (отфильтровывает элементы, выбранные с помощью предшествующей части селектора).

[Селекторы](#) > [Базовые фильтры](#) | [Селекторы](#) > [jQuery расширение](#)

Селектор :odd

Соответствует элементам с нечетными номерами позиций, в наборе выбранных элементов. (вариант с четными — :even). Поскольку счет элементов начинается с 0, :odd будет соответствовать второму, четвертому и т. д. элементам.

[Селекторы](#) > [Фильтры потомков](#)

Селектор :only-child

Соответствует элементам, которые являются единственными в своих непосредственных предках.

[Селекторы](#) > [Фильтры потомков](#)

Селектор :only-of-type

Элемент соответствует селектору :only-child, если среди своих соседей нет элементов с таким же именем тега (именно именем тега).

[Селекторы](#) > [Фильтры содержимого](#) | [Селекторы](#) > [jQuery расширение](#)

Селектор :parent

Соответствует всем непустым элементам. Наличие одного только пробела уже будет считаться наличием содержимого.

[Селекторы](#) > [Форма](#) | [Селекторы](#) > [jQuery расширение](#)

Селектор :password

Соответствует элементам, которые являются полями для ввода пароля (input-элементы с типом password).

[Селекторы](#) > [Форма](#) | [Селекторы](#) > [jQuery расширение](#)

Селектор :radio

Соответствует элементам, которые являются переключателями (input-элементы с типом radio).

[Селекторы](#) > [Форма](#) | [Селекторы](#) > [jQuery расширение](#)

[Селектор :reset](#)

Соответствует кнопкам, которые сбрасывают введенные в формы данные: input-элементы с типом reset.

[Селекторы](#) > [Базовые фильтры](#)

[Селектор :root](#)

Соответствует корневому элементу в документе. В HTML-документе, такой элемент всегда html.

[Селекторы](#) > [Форма](#) | [Селекторы](#) > [jQuery расширение](#)

[Селектор :selected](#)

Соответствует всем элементам со статусом selected. Это могут быть выбранные элементы типа

[Селекторы](#) > [Форма](#) | [Селекторы](#) > [jQuery расширение](#)

[Селектор :submit](#)

Соответствует кнопкам, которые отправляют данные формы на сервер: input-элементы с типом submit и button-элементы с типом submit.

[Селекторы](#) > [Базовые фильтры](#)

[Селектор :target](#)

Выбирает целевой элемент, указанный идентификатором фрагмента URI документа.

[Селекторы](#) > [Форма](#) | [Селекторы](#) > [jQuery расширение](#)

[Селектор :text](#)

Соответствует элементам, которые являются текстовыми полями (input-элементы с типом text).

[Селекторы](#) > [jQuery расширение](#) | [Селекторы](#) > [Фильтр видимости](#)

[Селектор :visible](#)

Соответствует всем видимым элементам страницы.

[Селекторы](#) > [Базовые](#)

[Селектор всех элементов \("*"\)](#)

Соответствует всем элементам.

[Селекторы](#) > [Атрибут](#)

[Селектор по значению атрибута \[name="value"\]](#)

Соответствует всем элементам с атрибутом attribute равным value. Если value состоит из нескольких слов, между которыми есть пробелы, то нужно заключать value в кавычки.

Если value не содержит пробелов — кавычки не обязательны.

[Селекторы](#) > [Базовые](#)

[Селектор по идентификатору \("#id"\)](#)

Соответствует элементу с идентификатором idName. Селектор по идентификатору является одним из наиболее востребованных средств jQuery.

[Селекторы](#) > [Базовые](#)

[Селектор по имени тэга \("element"\)](#)

Соответствует всем элементам с тегом tagname.

[Селекторы](#) > [Базовые](#)

[Селектор по классу \(".class"\)](#)

Соответствует всем элементам страницы с классом className. Селектор по классу является одним из наиболее востребованных средств jQuery.

[Селекторы](#) > [Атрибут](#)

[Селектор по концу атрибута \[name\\$="value"\]](#)

Соответствует всем элементам, у которых значение атрибута attribute заканчивается на value. Если value состоит из нескольких слов, между которыми есть пробелы, то нужно заключать value в кавычки. Если value не содержит пробелов — кавычки не обязательны.

[Селекторы > Атрибут](#)

Селектор по наличию атрибута [name]

Соответствует всем элементам, которые имеют атрибут attributeName. При этом, не важно, какие им заданы значения.

[Селекторы > Атрибут](#)

Селектор по началу атрибута [name^="value"]

Соответствует всем элементам, у которых значение атрибута attribute начинается с value. Если value состоит из нескольких слов, между которыми есть пробелы, то нужно заключать value в кавычки. Если value не содержит пробелов — кавычки не обязательны.

[Селекторы > Атрибут](#)

Селектор по нескольким атрибутам [name="value"][name2="value2"]

Соответствует элементам, удовлетворяющим всем заданным условиям на атрибуты (first, second, ...).

[Селекторы > Атрибут](#) | [Селекторы > jQuery расширение](#)

Селектор по несовпадению с атрибутом [name!="value"]

Соответствует всем элементам, у которых значение атрибута attribute не равно value. Если value состоит из нескольких слов, между которыми есть пробелы, то нужно заключать value в кавычки. Если value не содержит пробелов — кавычки не обязательны.

[Селекторы > Атрибут](#)

Селектор по подстроке в элементе [name*="value"]

Соответствует всем элементам, у которых значение атрибута attribute содержит value. Если value состоит из нескольких слов, между которыми есть пробелы, то нужно заключать value в кавычки. Если value не содержит пробелов — кавычки не обязательны.

[Селекторы > Атрибут](#)

Селектор по префиксу элемента [name|="value"]

Соответствует всем элементам, с атрибутом attribute содержащим префикс value, т.е. либо полностью совпадает с value, либо начинается со строки value- (наличие знака переноса существенно). Если value состоит из нескольких слов, между которыми есть пробелы, то нужно заключать value в кавычки. Если value не содержит пробелов — кавычки не обязательны.

[Селекторы > Атрибут](#)

Селектор по слову в атрибуте [name~="value"]

Соответствует всем элементам с атрибутом attribute, содержащим слово value (именно слово, а не просто подстроку. То есть вхождение value должно содержать с обеих сторон разделители: пробелы или начало/конец строки). Если value состоит из нескольких слов, между которыми есть пробелы, то нужно заключать value в кавычки. Если value не содержит пробелов — кавычки не обязательны.

[Селекторы > Иерархия](#)

Следующие селекторы ("prev ~ siblings")

Соответствует тем элементам, удовлетворяющим селектору next, которые расположены после элементов, удовлетворяющих селектору prev.

[Селекторы > Иерархия](#)

Смежные селекторы ("prev + next")

Соответствует элементам, удовлетворяющим селектору `next` и расположены непосредственно за элементами, удовлетворяющими селектору `prev`.