# Бібліотека jQuery

<https://html5book.ru/vvedenie-v-jquery/>

# Введення в jQuery

**jQuery** - бібліотека JavaScript, що містить в собі готові функції мови JavaScript, всі операції jQuery виконуються з коду JavaScript.

Бібліотека jQuery проводить маніпуляції з html-елементами, керуючи їх поведінкою і використовуючи DOM для зміни структури веб-сторінки. При цьому вихідні файли HTML і CSS не змінюються, зміни вносяться лише в відображення сторінки для користувача.

Для вибору елементів використовуються селектори CSS. Вибір здійснюється за допомогою функції $(). При виклику функція $() повертає новий екземпляр об'єкта JQuery, який повертає нуль або більше елементів DOM і дозволяє взаємодіяти з ними різними способами.

Виконання різних сценаріїв можливе тільки після закінчення завантаження структури документа document, коли браузер перетворює html-код сторінки в дерево DOM. Управління процесом завантаження забезпечує конструкція

JavaScript:

jQuery(document).ready(function() { ... });

Спочатку проводиться вкладення примірника document в функцію jQuery(), після застосовується метод ready(), якому передається функція function(){...}, що виконується після завантаження документа.

На практиці зазвичай використовується скорочена форма такого запису jQuery(function(){...});, Або $(function(){...}); .

Для зберігання інформації при роботі з бібліотекою jQuery використовуються змінні JavaScript. У змінних можуть зберігатися елементи. Імена змінних, які призначені для зберігання повернутих jQuery-об'єктів, починаються зі знака $, наприклад:

JavaScript:

$h = $(".list").parent().parent().detach();

Для зберігання декількох елементів використовуються масиви JavaScript:

JavaScript:

$k[3] = 15;

Знак $ є дозволеним символом в імені змінної на мові Javascript, але його не прийнято ставити на початку імені змінних, які не містять jQuery-об'єкт, щоб чітко відрізняти змінні, які містять jQuery-об'єкт.

jQuery використовує знак $ як псевдонім (скорочення) для ідентіфіктора jQuery-об'єкту.

Наприклад:

$tab\_title\_input - просто ім'я, яке включає знак долара,

$ ("#tab\_title") - виклик функції jQuery.

// погано, але синтаксично все вірно:

var first = $('# first');

var second = $('# second');

var value = first. text ();

// добре - перед об'єктами, які управляються jQuery, ми ставимо символ $:

var $first = $('# first');

var $second = $('# second'),

var value = $first. text();

## Правила роботи з бібліотекою jQuery

### 1. Як додати jQuery на веб-сторінку

Додати бібліотеку jQuery на свою веб-сторінку можна двома способами:

Використовувати версію файлу jQuery, розміщену на ресурсах **Google,** **Microsoft** або **jQuery.com.**  
Даний метод використовує **«Мережа дистрибуції контенту» (CDN, content distribution network),** тобто файл jQuery розташований на іншому веб-сайті, який при запиті користувачем відправляє даний файл на його комп'ютер. Очевидні переваги даного способу - зниження навантаження на власний веб-сервер і прискорення завантаження файлу в силу розгалуженості мережі серверів дистриб'ютора.

Щоб скористатися таким способом, необхідно перейти по одній з посилань:  
[Google CDN](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&rurl=translate.google.com&sl=auto&sp=nmt4&tl=uk&u=https://html5book.ru/goto/https://developers.google.com/speed/libraries/devguide&usg=ALkJrhgLofeBogJSpLHRriPAeM1CYRL74g#jquery)  
[Microsoft CDN](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&rurl=translate.google.com&sl=auto&sp=nmt4&tl=uk&u=https://html5book.ru/goto/http://www.asp.net/ajaxlibrary/cdn.ashx&usg=ALkJrhgGjYzo7N9V-Xj9qo9ODLlTVUz6Yw#jQuery_Releases_on_the_CDN_0)  
[CDNJS CDN](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&rurl=translate.google.com&sl=auto&sp=nmt4&tl=uk&u=https://html5book.ru/goto/http://cdnjs.com/libraries/jquery/&usg=ALkJrhgj3g6co06xcrBv-Uk1cj6k2nxJsQ)  
Після переходу на сайт ресурсу вам буде потрібно всього лише скопіювати посилання на jQuery-файл і додати її на свою веб-сторінку між тегами <script> ... </ script>. В результаті у вас повинно вийти, наприклад,

HTML:

<script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script>

Завантажити останню версію бібліотеки jQuery з офіційного сайту. Для цього вам необхідно перейти за адресою [jQuery.com](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&rurl=translate.google.com&sl=auto&sp=nmt4&tl=uk&u=http://jquery.com/download/&usg=ALkJrhg3TlfTcAKGYcEI2qCfTUdl7Aj79w) і вибрати цікаву для вас версію бібліотеки. Для завантаження пропонується дві версії jQuery-файлу - мінімізований і нестислий. Розмір нестислого файлу близько 300 Кб, він містить коментарі, тому його краще використовувати з метою розробки і налагодження коду.  
Мінімізована версія файлу важить близько 100 Кб, в ній видалені всі коментарі і непотрібні пропуски, що прискорює завантаження файлу браузером.

Для завантаження потрібно перейти за посиланням, і у вікні, натиснути правою кнопкою миші і вибрати «Зберегти як ...». Після цього помістити файл в потрібну папку (зазвичай для цього використовується папка «scripts») і додати його на вашу сторінку:

HTML:

<script src="scripts/jquery-2.0.0.min.js"></script>

### 2. Правила додавання jQuery на сторінку

* Розміщуйте посилання на jQuery-файл всередині тега <head>.
* Розміщуйте посилання на jQuery-файл після посилань на стилі CSS, так як часто бібліотека jQuery проводить маніпуляції зі стилями елементів веб-сторінки.
* Розміщуйте інші теги <script> ... </ script> тільки після посилання на файл jQuery, якщо даний скрипт використовує бібліотеку jQuery.

### 3. Як створити новий html-елемент

Створити порожній html-елемент, наприклад, блок, можна кількома способами:

1) за допомогою короткої форми запису $("<div>")  
2) за допомогою інструкції $( "<div> </div>")  
3) за допомогою інструкції $( "<div/>")

Всі три способи робочі, але, тим не менш, рекомендується включати відкривають та закривають теги, щоб показати, що даний елемент може містити інші елементи.

При створенні нового елемента методу $() можна передати другий параметр у вигляді об'єкта JavaScript, що визначає додаткові атрибути елемента:

$("<img/>",

{src: "images/flower.jpg",

title: "Rose\_in\_garden",

click: function() {...}

}).appendTo("body");

JavaScript

Таким чином створюється елемент <img> з заданими атрибутами і включається в дерево DOM.

Завантажити актуальну версію бібліотеки можна на [офіційному сайті](http://jquery.com/) <http://jquery.com/> абсолютно безкоштовно.

# [Селектори jQuery](https://basicweb.ru/jquery/jquery_selectors.php)

# [Події jQuery](https://basicweb.ru/jquery/jquery_events.php)

# [Методи jQuery](https://basicweb.ru/jquery/jquery_dom_methods.php)

# У документації до бібліотеки jQuery тема подій і їх обробки займає значне місце. Нище наведені «основні» (умовно звичайно) методи - ready, bind, one, trigger і triggerHandler, unbind, а у другій половині хелпери - такі собі помічниками, використовуючи які, Ви зможете як викликати певні події, так і пов'язувати з ними деякі дії ...

Page Load:

**ready (fn)**

повертає: jQuery   
Призначає функції, які будуть виконуватися щоразу, коли об'єктна модель документа (DOM) готова до використання. Це мабуть найважливіша функція, включена в розділ «Події», оскільки вона може поліпшити час реакції веб-додатків, і хороша альтернатива використанню події window.onload, оскільки при використанні цього методу, Ваші функції будуть викликані вже в той момент, коли об'єктна модель документа (DOM) готова до роботи. Тут не потрібно чекати, коли сторінка буде повністю завантажена.

Аргумент, що передається обробнику події, є посиланням на функцію jQuery. Можна використовувати jQuery або $ без ризику виникнення колізій у просторі імен. Однак необхідно переконатися в тому, що не має обробника події onload, в іншому випадку $ (document). Ready () може не виконатися.

На сторінці можна мати як завгодно багато $ (document). Ready подій. У цьому випадку функції, пов'язані з ними, будуть виконуватися в порядку їх слідування.

У цьому прикладі ми покажемо повідомлення про те, що DOM завантажена і готова до роботи.

|  |  |
| --- | --- |
| 1 | $(document).ready(function(){ |

|  |  |
| --- | --- |
| 2 | $("p:first") |

|  |  |
| --- | --- |
| 3 | .text("DOM загружена и готова к работе!"); |

|  |  |
| --- | --- |
| 4 | }); |

|  |  |
| --- | --- |
| 5 | // еще раз тоже самое |

|  |  |
| --- | --- |
| 6 | jQuery(document).ready(function(){ |

|  |  |
| --- | --- |
| 7 | jQuery("p:last") |

|  |  |
| --- | --- |
| 8 | .text("Объектная модель документа готова!"); |

|  |  |
| --- | --- |
| 9 | }); |

Відкрийте [приклад](http://translate.googleusercontent.com/translate_c?depth=1&hl=ru&rurl=translate.google.com.ua&sl=ru&tl=uk&u=http://www.linkexchanger.su/jquery_events/ready_fn.html&usg=ALkJrhjkM9tNW9yl8PMB-Sl5BPz_ALChHQ) в окремому вікні і подивіться вихідний код.

Event Handling:

**bind (type, [data], fn)**

повертає: jQuery   
Пов'язує обробник з одним або більше подіями (наприклад click) для кожного елемента набору. Може також пов'язувати користувальницькі події. Можливі події: blur, focus, load, resize, scroll, unload, click, dblclick, mousedown, mouseup, mousemove, mouseover, mouseout, mouseenter, mouseleave, change, select, submit, keydown, keypress, keyup, error. Обробник події приймає об'єкт подія, що робить можливим запобігання поведінки за замовчуванням. Щоб запобігти і поведінку за замовчуванням і «спливання» події (тобто передачу події від одного елемента іншому), обробник повинен повертати false.

У більшості випадків можна визначити обробники події як анонімні функції, як показано в наступному прикладі. Всякий раз, коли здійснений клік на параграфі, в alert виводиться текст цього параграфа.

|  |  |
| --- | --- |
| 1 | $("p").bind("click", function(){ |

|  |  |
| --- | --- |
| 2 | alert( $(this).text() ); |

|  |  |
| --- | --- |
| 3 | }); |

У разі, якщо це неможливо, слід передавати додаткові дані в другому параметрі [data] (див. приклад).

|  |  |
| --- | --- |
| 1 | function handler(event) { |

|  |  |
| --- | --- |
| 2 | alert(event.data.foo); |

|  |  |
| --- | --- |
| 3 | } |

|  |  |
| --- | --- |
| 4 | $("p").bind("click", {foo: "bar"}, handler); |

Щоб скасувати дію за замовчуванням і запобігти «спливання», повертаємо false:

|  |  |
| --- | --- |
| 1 | $("form").bind("submit", function() { return false; } |

Щоб скасувати тільки дія за замовчуванням використовується метод preventDefault.

|  |  |
| --- | --- |
| 1 | $("form").bind("submit", function(event){ |

|  |  |
| --- | --- |
| 2 | event.preventDefault(); |

|  |  |
| --- | --- |
| 3 | }); |

Для запобігання «спливання» події використовується метод stopPropagation.

|  |  |
| --- | --- |
| 1 | $("form").bind("submit", function(event){ |

|  |  |
| --- | --- |
| 2 | event.stopPropagation(); |

|  |  |
| --- | --- |
| 3 | }); |

Також можна пов'язати деякі користувальницькі події.

Также можно связать некие пользовательские события.

|  |  |  |
| --- | --- | --- |
| 1 | $("p").bind("myCustomEvent", function(e,myName,myValue){ | |
| 2 | $(this).text(myName + ", hi there!"); |

|  |  |  |
| --- | --- | --- |
| 3 | $("span").stop().css("opacity", 1) | |
| 4 | .text("myName = " + myName) |

|  |  |  |
| --- | --- | --- |
| 5 | .fadeIn(30).fadeOut(1000); | |
| 6 | }); |

|  |  |
| --- | --- |
| 7 | $("button").click(function () { |
| 8 | $("p").trigger("myCustomEvent", [ "John" ]); | |

|  |  |
| --- | --- |
| 9 | }); |

Відкрийте [приклад](http://translate.googleusercontent.com/translate_c?depth=1&hl=ru&rurl=translate.google.com.ua&sl=ru&tl=uk&u=http://www.linkexchanger.su/jquery_events/bind_custom.html&usg=ALkJrhjvWQb666i1YFAuxbNts_M214KDhw) в окремому вікні і подивіться вихідний код.

У наступному прикладі демонструється обробка подій click і double-click на параграфі.

|  |  |
| --- | --- |
| 01 | $("p").bind("click", function(e){ |

|  |  |
| --- | --- |
| 02 | var str = "( " + e.pageX + ", " + e.pageY + " )"; |

|  |  |
| --- | --- |
| 03 | $("span").text("Кликнуто! " + str); |

|  |  |
| --- | --- |
| 04 | }); |

|  |  |
| --- | --- |
| 05 | $("p").bind("dblclick", function(){ |

|  |  |
| --- | --- |
| 06 | $("span").text("Двойной клик сделан на "+this.tagName); |

|  |  |
| --- | --- |
| 07 | }); |

|  |  |
| --- | --- |
| 08 | $("p").bind("mouseenter mouseleave", function(e){ |

|  |  |
| --- | --- |
| 09 | $(this).toggleClass("over"); |

|  |  |
| --- | --- |
| 10 | }); |

Відкрийте [приклад](http://translate.googleusercontent.com/translate_c?depth=1&hl=ru&rurl=translate.google.com.ua&sl=ru&tl=uk&u=http://www.linkexchanger.su/jquery_events/bind.html&usg=ALkJrhhtqYXlJ9Lw9TM6zbKs3JoaR2XjvQ) в окремому вікні і подивіться вихідний код.

Візьміть до уваги, що приклад демонструється у фреймі, відповідно і координати обчислюються щодо кадру.

**one (type, [data], fn)**

повертає: jQuery   
Пов'язує обробник з одним або більше подіями, які будуть виконані тільки один раз, для кожного елемента в наборі.Оброблювач виповнюється тільки один раз для кожного елемента. В іншому діють ті ж правила, описані при застосуванні bind (). Обробник події приймає об'єкт подія, що робить можливим запобігання поведінки за замовчуванням.Щоб запобігти і поведінку за замовчуванням і «спливання» події (тобто передачу події від одного елемента іншому), обробник повинен повертати false.

**trigger (type, [data])**

повертає: jQuery   
Викликає подія (передане в type) для кожного елемента набору. Проте це також викличе виконання браузером дій за замовчуванням для цієї події. Наприклад, передаючи 'submit' у функцію trigger () Ви змусите браузер відправити форму. Дії браузера за замовчуванням можна запобігти, повертаючи false для однієї з функцій, пов'язаних з подією.

**triggerHandler (type, [data])**

повертає: jQuery   
Це особливий метод виклику всіх пов'язаних з елементом обробників подій БЕЗ виконання браузером дій за замовчуванням. Подія викликається тільки для першого елемента, включеного в набір. Якщо набір елементів порожній, цей метод поверне 'undefined'.

**unbind ([type], [data])**

повертає: jQuery   
Це метод - протилежність bind, він видаляє всі події пов'язані з елементом для кожного елемента набору. Можна також видалити користувальницькі події, зареєстровані в bind ().Якщо переданий аргумент type - видалені будуть всі події цього типу.

**Хелпери.**

 Використовуючи хелпери, можна як викликати певні події, так і пов'язувати з ними деякі дії ...

Interaction Helpers:

**hover (over, out)**

повертає: jQuery   
Симулює hovering - відстеження входу покажчика миші в межі об'єкта і виходу за його межі. Всякий раз, коли покажчик миші виявляється поверх об'єкту, викликається функція, передана в якості першого аргументу. Аналогічно, коли покажчик миші виходить за межі об'єкта - викликається функція, яка є другим аргументом.

**toggle (fn, fn1)(відсутній починаючи з версії 1.8.)**

повертає: jQuery   
Перемикається між двома функціями, які по черзі викликаються кожним наступним кліком. Кожен раз, коли клік здійснений на обраному елементі, спрацьовує перша функція. Якщо на тому ж елементі здійснений ще один клік - спрацьовує друга функція. Кожен наступний клік продовжує перемикання між цими двома функціями. Можна використовувати unbind («click»), щоб скасувати дію toggle ().

Event Helpers:

**blur ()**

повертає: jQuery   
Викликає подія blur для всіх елементів набору. У цьому випадку будуть викликані всі функції, пов'язані з цією подією blur і також викличе дію браузера за замовчуванням для цієї події.Поведінка браузера за замовчуванням можна попередити, повертаючи false з однієї або декількох функцій, пов'язаних з цією подією blur. Подія blur зазвичай викликається, коли елемент втрачає фокус через дію вказівного пристрою або клавіші Tab.

**blur (fn)**

повертає: jQuery   
Пов'язує функцію з подією blur для всіх елементів набору.

**change ()**

повертає: jQuery   
Викликає подія change для кожного елемента набору. У цьому випадку будуть викликані всі функції, пов'язані з цією подією change і також викличе дію браузера за замовчуванням для цієї події. Поведінка браузера за замовчуванням можна попередити, повертаючи false з однієї або декількох функцій, пов'язаних з цією подією change. Подія change зазвичай викликається, коли елемент управління втрачає фокус і його значення було змінено з моменту отримання ним фокусу.

**change (fn)**

повертає: jQuery   
Пов'язує функцію з подією change для всіх елементів набору.

**click ()**

повертає: jQuery   
Викликає подія click для кожного елемента набору. У цьому випадку будуть викликані всі функції, пов'язані з цією подією.

**click (fn)**

повертає: jQuery   
Пов'язує функцію з подією click для всіх елементів набору. Подія click настає, коли кнопка вказівного пристрою (наприклад миша) клікнути на елементі. Клік визначається як mousedown + mouseup в деякому місці екрана.

**dblclick ()**

повертає: jQuery   
Викликає подія dblclick для кожного елемента набору. У цьому випадку будуть викликані всі функції, пов'язані з цією подією dblclick і також викличе дію браузера за замовчуванням для цієї події. Поведінка браузера за замовчуванням можна попередити, повертаючи false з однієї або декількох функцій, пов'язаних з цією подією dblclick.

**dblclick (fn)**

повертає: jQuery   
Пов'язує функцію з подією dblclick для кожного елемента набору. Подія dblclick зазвичай викликається, коли кнопка вказівного пристрою двічі швидко клікнути на елементі.

**error ()**

повертає: jQuery   
Викликає подія error для кожного елемента набору. У цьому випадку будуть викликані всі функції, пов'язані з цією подією error і також викличе дію браузера за замовчуванням для цієї події. Поведінка браузера за замовчуванням можна попередити, повертаючи false з однієї або декількох функцій, пов'язаних з цією подією error. Подія error зазвичай викликається, коли елемент втрачає фокус через дію вказівного пристрою або клавіші Tab.

**error (fn)**

повертає: jQuery   
Пов'язує функцію з подією error для кожного елемента набору.  
Для події error немає загальних стандартів. У більшості браузерів, подія error об'єкта window викликається, коли на сторінці виявлена ​​помилка JavaScript. Для об'єкта image подія error викликається при некоректному вмісті атрибуту src - наприклад, неіснуючий файл.

Якщо подія відбулася в об'єкті window, обробник події передає три параметри:

* повідомлення, яке описує подію («varName is not defined», «missing operator in expression», і т.д.);
* повний URL документа, який містить помилку;
* номер рядка, в якій сталася помилка;

Якщо обробник події поверне true, це означає, що подія було оброблено і браузер не показує ніяких помилок.

**focus ()**

повертає: jQuery   
Викликає подія focus для кожного елемента набору. У цьому випадку будуть викликані всі функції, пов'язані з цією подією focus.

**focus (fn)**

повертає: jQuery   
Пов'язує функцію з подією focus для кожного елемента набору.Подія focus викликається, коли елемент отримує фокус через вказівник миші або клавішу Tab.

**keydown ()**

повертає: jQuery   
Викликає подія keydown для всіх елементів набору. У цьому випадку будуть викликані всі функції, пов'язані з цією подією keydown і також викличе дію браузера за замовчуванням для цієї події. Поведінка браузера за замовчуванням можна попередити, повертаючи false з однієї або декількох функцій, пов'язаних з цією подією keydown.

**keydown (fn)**

повертає: jQuery   
Пов'язує функцію з подією keydown для кожного елемента набору. Подія keydown зазвичай викликається, коли на клавіатурі натиснута яка-небудь клавіша.

**keypress ()**

повертає: jQuery   
Викликає подія keypress для всіх елементів набору. У цьому випадку будуть викликані всі функції, пов'язані з цією подією keypress і також викличе дію браузера за замовчуванням для цієї події. Поведінка браузера за замовчуванням можна попередити, повертаючи false з однієї або декількох функцій, пов'язаних з цією подією keypress. Подія keypress зазвичай викликається, коли на клавіатурі була натиснута яка-небудь клавіша.

**keypress (fn)**

повертає: jQuery   
Пов'язує функцію з подією keypress для кожного елемента набору. Подія keypress зазвичай викликається, коли на клавіатурі була натиснута яка-небудь клавіша. Подія keypress визначається як keydown + keyup на який-небудь клавіші.

**keyup ()**

повертає: jQuery   
Викликає подія keyup для всіх елементів набору. У цьому випадку будуть викликані всі функції, пов'язані з цією подією keyup і також викличе дію браузера за замовчуванням для цієї події. Поведінка браузера за замовчуванням можна попередити, повертаючи false з однієї або декількох функцій, пов'язаних з цією подією keyup.

**keyup (fn)**

повертає: jQuery   
Пов'язує функцію з подією keyup для кожного елемента набору. Подія keyup зазвичай викликається, коли на клавіатурі була відпущена небудь клавіша.

**load (fn)**

повертає: jQuery   
Пов'язує функцію з подією load для кожного елемента набору.Будучи пов'язаним з елементом window, це подія викликається тільки тоді, коли агент користувача завершить завантаження всього вмісту документа (фрейми, об'єкти, графіком). Для елементів це викликається, коли цільової елемент і весь його вміст повністю завантажено.   
Необхідно пам'ятати, що load буде працювати тільки в тому випадку, якщо він визначений перед елементом, повного завантаження якого ми очікуємо. Якщо ж його визначити після такого елемента - нічого не відбудеться. 

**mousedown (fn)**

повертає: jQuery   
Пов'язує функцію з подією mousedown для кожного елемента набору. Подія mousedown зазвичай викликається, коли клавіша вказівного пристрою (миші) була натиснута на елементі.

**mousemove (fn)**

повертає: jQuery   
Пов'язує функцію з подією mousemove для кожного елемента набору. Подія mousemove зазвичай викликається, коли покажчик миші переміщається поверх елемента. Обробник події приймає один аргумент - об'єкт події, у властивостях clientX і clientY якого представлені координати покажчика миші.

**mouseout (fn)**

повертає: jQuery   
Пов'язує функцію з подією mouseout для кожного елемента набору. Подія mouseout зазвичай викликається, коли покажчик миші виходить за межі елемента.

**mouseover (fn)**

повертає: jQuery   
Пов'язує функцію з подією mouseover для кожного елемента набору. Подія mouseover зазвичай викликається, коли покажчик миші знаходиться в межах елемента. 

**mouseup (fn)**

повертає: jQuery   
Пов'язує функцію з подією mouseup для кожного елемента набору. Подія mouseup зазвичай викликається, коли клавіша вказівного пристрою (миші) була відпущена на елементі.

**resize (fn)**

повертає: jQuery   
Пов'язує функцію з подією resize для кожного елемента набору. Подія resize зазвичай викликається, коли змінюються розміри області перегляду документа.

**scroll (fn)**

повертає: jQuery   
Пов'язує функцію з подією scroll для кожного елемента набору.Подія scroll зазвичай викликається, коли прокручується область перегляду документа.

**select ()**

повертає: jQuery   
Викликає подія select для всіх елементів набору. У цьому випадку будуть викликані всі функції, пов'язані з цією подією select і також викличе дію браузера за замовчуванням для цієї події. Поведінка браузера за замовчуванням можна попередити, повертаючи false з однієї або декількох функцій, пов'язаних з цією подією select.   
Не варто плутати подія «select» з подією «change», яке викликається, коли HTML-елемент select використовується для вибору який (їх)-або опцій.

**select (fn)**

повертає: jQuery   
Пов'язує функцію з подією select для кожного елемента набору. Подія select зазвичай викликається, коли користувач виділяє небудь текст у текстовому полі, наприклад input і textarea.

**submit ()**

повертає: jQuery   
Викликає подія submit для всіх елементів набору. У цьому випадку будуть викликані всі функції, пов'язані з цією подією submit і також викличе дію браузера за замовчуванням для цієї події. Поведінка браузера за замовчуванням можна попередити, повертаючи false з однієї або декількох функцій, пов'язаних з цією подією submit.

**submit (fn)**

повертає: jQuery   
Пов'язує функцію з подією submit для кожного елемента набору. Подія submit зазвичай викликається, коли форма відправляється.

**unload (fn)**

повертає: jQuery   
Пов'язує функцію з подією unload для кожного елемента набору.