jQuery Events

Alternate title: Mastering jQuery Event Handling.

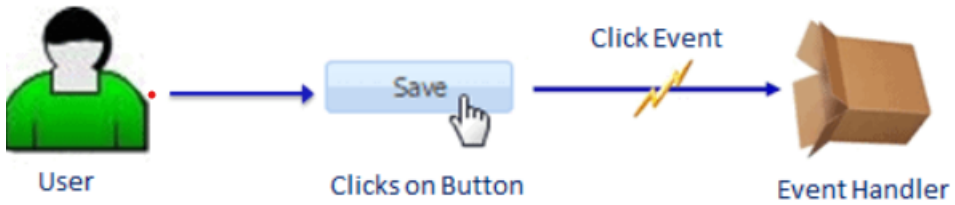
Meta Description: Learn how to implement events in queries and different types of events.

Introduction

An event is something that shows any actions or occurrences that happen on a web page. jQuery provides methods to handle and respond to these events. So, events are actions of some functionalities triggered in a particular scenario. When a user clicks something, it calls the event, which the event handler handles.

jQuery Event Handling and Methods

Event methods trigger the event using an event handler. This method is usually attached to a function or any click; this will, in turn, trigger the event.



Alt text: Process of event handling
Source: Event Handling

There are various types of events methods to handle different kinds of events.

Example of Events

- Whenever there is a mouse click.
- During some form submissions on the submit button.
- During the load of the webpage.
- Scrolling the webpage.

- Calling some methods.

Syntax:

```
$(“selector”). click(function) {
//implement the action here}
```

Types of Events

There are four major types of events in jQuery events. Below are the types:

- Mouse Event
- Keyboard Event
- Form Event
- Document/Windows Event

Mouse Event

This event is wholly triggered whenever the mouse performs any action. It can be a click or double click (or) on click (or) hover. A different type of event will be triggered based on each type.

Whenever any action happens, the method will bind that to event handlers.

Syntax for Click Event:

This will attach a function to the click event

```
$(“selector”). click(function) {
//implement the action here}
```

Syntax for Mouse Enter Event:

```
$(“selector”).mouseenter(function) {
//implement the action here}
```

Syntax for Mouse Leave Event:

```
$(“selector”).mouseleave(function) {
//implement the action here}
```

Event Name	Description
click ()	This is already a built-in method. It will be started whenever a click event occurs (or) and will attach a function to execute when a click event occurs.
dbclick()	This is also an inbuilt method that will be executed when the user double-clicks on the mouse.
mouseenter()	This inbuilt method will be executed when the user selects some elements.
mouseleave()	This inbuilt method will be executed when the user leaves the selected element.
mouseup()	This event will get triggered when the button is released from the element.
contextmenu()	This event will get triggered when the right Mouse button is clicked on the element.
mouseover()	This event will trigger when the Mouse pointer is moved to some element.
mouseout()	This event will occur when the Mouse pointer moves out of an element.

Example for Mouse Event:

```
$("document").ready(function()
        {
          $("#presskey").mouseup(up);
          $("#presskey").mousedown(down);
          function up()
           {
            $("#presskey").text("Mouse up event has occurred");
           }
          function down()
           {
            $("#presskey").text("Mouse down event has occurred");
              });
```

In the above example, when the respective key is pressed, the mouse up and mouse down events will be triggered. Each of these mouse up and mouse down events has separate functions defined. Based on the action, the function will be called.

Keyboard Event

This event is wholly triggered whenever there is any action from the keyboard. It can be a key press, key up, or key down. Based on each type, a different type of event will get triggered, and the respective method will bind the event handler.

Event Name	Description
keypress()	This method will be triggered whenever the browser registers keyboard input, which triggers this keyboard event.
keydown()	This will trigger the key down event whenever any key is pressed down.
keyup()	This event will specify whenever any key is released.

Example for Keyboard Event:

```
$(document).ready(function(){
    $("input").keypress(function(){
//perform one logic
    });
```

In the above example, when the input key is pressed, some action will be performed to execute the event.

Form Event

This event is wholly triggered when an action is taken from the form. Example: Form submit.

Event Name	Description
submit()	It gets executed when the user clicks the form submit button on a webpage.
change()	It gets executed whenever there is any change in an element. Example: Input/ Select.
focus()	This event will be fired whenever any focus is on a particular element. Example: textarea.
blur()	This will be fired when an element loses focus.

Example for Form Event:

```
$("select input").change(function () {
    //implement logic  });
```

In the above example, the user selects the input to be changed and calls the event change to implement the form.

Document/Windows Event

This event handles the interaction from windows or a document.

Event Name	Description
load()	This documented event will be triggered when the entire page and its resources get loaded.
resize()	This is a window-based event when the window is resized.
scroll()	This is an inbuilt method. This will be triggered when the user scrolls to a particular element.
unload()	This event will be triggered when the page gets unloaded.

Example for Document Event:

```
$("selector").load(“sample.txt”) ;
```

In the above example, when the respective selector is called, it will load the text file sample.

Hover Event

Two events will be triggered when you roam the mouse pointer over the selected element.

1. Mouse Enter
2. Mouse Leave

The above two events are already explained in the section. Kindly refer to it.

For the mouse enter event, the Input parameter is mandatory. It gets executed when mouse enter occurs.

Event Name	Description
load()	This documented event will be triggered when the entire page and its resources get loaded.
resize()	This is a window-based event when the window is resized.

scroll()	This is an inbuilt method. This will be triggered when the user scrolls to a particular element.
unload()	This event will be triggered when the page gets unloaded.

Example for Hover Event:

```
$(window).resize(function() {
//implement resizing logic ) };
```

In the above example, the window will be resized according to the given value when the event is called. This resize is an in-built method that will resize the window when the browser size changes, but the user can also customize the value and change the size.

Handling the Events in jQuery

In jQuery event handling, we must first get the reference for the DOM elements. This can be done using the jQuery selector and invoking the appropriate jQuery event method.

Example:

Below is an example of using a DOM element.

```
$ ('#registerButton').click(function() {
specify some event action { }
<input type="button" id="registerButton" value="register">
```

In the above example, we used the DOM reference ID 'registerButton' and called it the click method.

Other than the above-listed events, there are still more events like jQuery event method chaining. This chaining method allows you to run multiple jQuery commands at a time.

To implement the chaining method, you need to add events one after another to get executed in the same order as they were implemented.

Example:

```
$("selector).slideup(1000).slidedown(2000);
```

In the above example, the slide-up event will run, after which the slide-down will be executed as a chaining event.

Event Object

Every event handler will have an event object associated with it. The object can include properties, methods, or related targets.

We can take a look at the example in the section above. Some function is passed inside the click event when the event register button is triggered.

This Keyword in Event Handler

This keyword in the event handler represents a DOM element of the raised event.

Example:

alert(this.value);

jQuery Event List

Event Name	Description
on()	It attaches one or more events to the selected element.
live()	It matches the event to the current selector for the given event type.
error()	When this method is called, the event handler will bind the respective error method. This is deprecated in the recent version. Use .on() instead.
event.isDefaultPrevented()	Returns whether event.preventDefault() was ever called on this object.
event.isPropagationStopped()	It checks whether the propagation on a particular event is stopped or not.
event.nameSpace()	It returns the custom namespace when the event is triggered.
event.pageX()	It returns the document X coordinate of the mouse pointer when the mouse event occurs.
event.pageY()	It returns the document Y coordinate of the mouse pointer when the mouse event occurs.
event.stopPropagation()	It stops the propagation.
load()	It is triggered when all the elements and its sub-element is loaded.

toggle()	It binds two or more event handlers to the set of matched elements.
bind()	It attaches one or more functions to the event. This is now deprecated. Hence, the alternative to use is **.on()**.
unbind()	It removes existing event handlers from elements.
unload()	It is triggered when the user navigates from the web page.
delegate()	This was one of the popular events for attaching the handler. However, this is now deprecated. Hence, the alternative to use for this event is -**on()**.

Other than the above events, many properties of events are also used for various purposes.

on() click event in jQuery is used to attach one or more events to the selected element. This is used as an alternative to many deprecated methods. Some examples of event properties are targets, current targets, getting time stamps, etc. Each of these should be used along with the keyword event.Propertyname.

Best Practices in Event Handling

- Use event delegation wherever possible, so instead of attaching them and having dependencies, you can implement them separately.
- In the current version, this event delegation is deprecated; hence, use the “on” function instead.
- Use proper selectors.
- Use unique Ids.
- Implement proper function methods for the respective events.
- Use the local variable wherever possible.

Wrapping Up

An event is a response to some actions that have been triggered. An event handler will execute these events. There are many inbuilt event methods and customized methods. We can implement events based on the user's requirements. There are four significant types of classification of events. However, there are many other events. Only a few are listed in this article.

FAQs

Q: What are the events in jQuery?
A: An action is a click or other action triggered by a webpage. An event represents something that has happened on the webpage.

Q: What is event data in jQuery?
A: Event data is something that is passed as a parameter to the event. It can be an input or output parameter. In some events, these event data are optional.

Q: What is an event method?
A: This method executes the event whenever an action has been triggered. This method will, in turn, call some event or return some alert. The event handler handles this method.

Q: What are Ajax events?
A: The events are attached or triggered whenever there is any Ajax request from the webpage. All the events that are related to Ajax requests will be handled by Ajax events.

Q: Why use jQuery for events?

A: We use jQuery for the events to manipulate the data and respective actions properly.