



CHAPTER 6

EVENTS



WHAT IS AN EVENT?



Events are the browser's way of saying, "Hey, this just happened."



When an event **fires**, your script can then react by running code (e.g. a function).



By running code when an event fires, your website responds to the user's actions.

It becomes **interactive**.



DIFFERENT EVENT TYPES



USER INTERFACE EVENTS

load
unload
error
resize
scroll



KEYBOARD EVENTS

keydown

keyup

keypress



MOUSE EVENTS

click

dblclick

mousedown

mouseup

mouseover

mouseout



FOCUS EVENTS

focus / focusin
blur / focusout



FORM EVENTS

input
change
submit
reset
cut
copy
paste
select



HOW EVENTS TRIGGER JAVASCRIPT CODE



1

Select the
element
node(s) the
script should
respond to



1

Select the **element** node(s) the script should respond to

2

Indicate the **event** on the selected node(s) that will trigger a response



1

Select the **element** node(s) the script should respond to

2

Indicate the **event** on the selected node(s) that will trigger a response

3

State the code you want to run when the event occurs



BINDING AN EVENT TO AN ELEMENT



There are three ways to bind an event to an element:

- HTML event handler attributes
- Traditional DOM event handlers
- DOM Level 2 event listeners



The following examples show a **blur** event on an element stored in a variable called `e1` that triggers a function called `checkUsername()`.



HTML EVENT HANDLER ATTRIBUTES (DO NOT USE)

```
<input type="text" id="username"  
       onblur="checkUsername()">
```



HTML EVENT HANDLER ATTRIBUTES (DO NOT USE)

ELEMENT

```
<input type="text" id="username"  
      onblur="checkUsername()">
```



HTML EVENT HANDLER ATTRIBUTES (DO NOT USE)

```
<input type="text" id="username"  
      onblur="checkUsername()">
```

EVENT



HTML EVENT HANDLER ATTRIBUTES (DO NOT USE)

```
<input type="text" id="username"  
      onblur="checkUsername()">
```

FUNCTION




TRADITIONAL DOM EVENT HANDLERS

```
el.onblur = checkUsername();
```



TRADITIONAL DOM EVENT HANDLERS

```
el.onblur = checkUsername();
```




ELEMENT



TRADITIONAL DOM EVENT HANDLERS

```
el.onblur = checkUsername();
```



EVENT



TRADITIONAL DOM EVENT HANDLERS

```
el.onblur = checkUsername();
```

FUNCTION




EVENT LISTENERS

```
el.addEventListener('blur', checkUsername, false);
```



EVENT LISTENERS

```
el.addEventListener('blur', checkUsername);
```


ELEMENT



EVENT LISTENERS

```
el.addEventListener('blur', checkUsername);
```


EVENT



EVENT LISTENERS

```
el.addEventListener('blur', checkUsername);
```

FUNCTION



Because you cannot have parentheses after the function names in event handlers or listeners, passing arguments requires a workaround.



PARAMETERS WITH EVENT LISTENERS

```
el.addEventListener('blur', function() {  
    checkUsername(5);  
});
```

An anonymous function is used as the second argument.



PARAMETERS WITH EVENT LISTENERS

```
el.addEventListener('blur', function() {  
    checkUsername(5);  
});
```

Inside the anonymous function, a named function is called.



EVENT FLOW



HTML elements nest inside other elements. If you hover or click on a link, you will also be hovering or clicking on its parent elements.



EVENT BUBBLING (default behavior)



THE EVENT OBJECT



When an event occurs, the event object can tell you information about it and which element it happened upon.



ELEMENT AN EVENT OCCURRED ON

1: EVENT LISTENER CALLS FUNCTION

```
function checkUsername(e) {  
  var target = e.target;  
}  
  
var el = document.getElementById('username');  
el.addEventListener('blur', checkUsername, false);
```



ELEMENT AN EVENT OCCURRED ON

2: EVENT OBJECT PASSED TO FUNCTION

```
function checkUsername(e) {  
  var target = e.target;  
}  
  
var el = document.getElementById('username');  
el.addEventListener('blur', checkUsername, false);
```



ELEMENT AN EVENT OCCURRED ON

3: ELEMENT THAT EVENT HAPPENED ON

```
function checkUsername(e) {  
  var target = e.target;  
}  
  
var el = document.getElementById('username');  
el.addEventListener('blur', checkUsername, false);
```

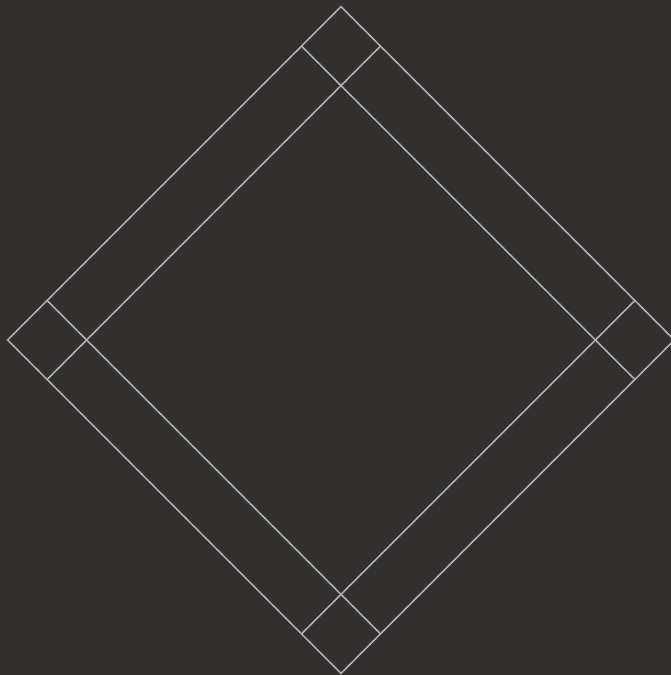


PROPERTIES

target
type
cancelable

METHODS

preventDefault()
stopPropagation()



CHAPTER 7

JQUERY



WHAT IS JQUERY?



jQuery is a script.

It is written in JavaScript.



Once included in your page, it
is used to:

Select elements

Do things with the elements you selected



1

Select elements
using CSS-style
selectors



```
jQuery( 'li.hot' );
```



FUNCTION



```
jQuery( 'li.hot' );
```

CSS-STYLE SELECTOR



SHORTHAND FOR JQUERY FUNCTION

```
$( 'li.hot' );
```



When you select an element
or set of elements, it creates a
jQuery object.



That object contains
references to the elements.



Like any object, the jQuery object has properties and methods. They allow you to work with those elements.



A jQuery object with selected elements can be called a:

jQuery selection
or
matched set



1

Select elements
using CSS-style
selectors

2

Do something
using methods of
the jQuery object



HIDE ALL MATCHING LIST ITEMS

```
$('li.hot').hide();
```



METHOD OF THE JQUERY OBJECT



With jQuery, you can:

- Select or find elements

- Update their content / size / visibility

- Simplify event handling



STORING SELECTIONS IN VARIABLES



When a variable holds a jQuery object, its name often begins with a \$ symbol.



```
var $titles = $('.title');
```

└──────────┘
VARIABLE NAME



CSS-STYLE SELECTORS

This variable holds a jQuery object containing the element whose `id` attribute has a value of `author`:

```
var $author = $('#author');
```



CSS-STYLE SELECTORS

This variable holds a jQuery object containing the first item from the unordered list whose `id` attribute has a value of `list`:

```
var $first = $('ul#list:first-child');
```



GETTING & SETTING INFORMATION



GETTING HTML CONTENT

The `.html()` method gets the content of the selection (including markup).

It only retrieves content from the *first element* in the matched set.



GETTING HTML CONTENT

This example retrieves the content of the first list item.

```
$ ( 'li' ) .html ( ) ;
```



GETTING TEXT CONTENT

The `.text()` method gets the *text content only* of the selection (excluding markup).

```
$ ( 'li' ) .text ( ) ;
```



SETTING CONTENT

New content is added inside the parentheses after the method name.

It updates *all* of the elements in the matched set (not just the first). This is known as **implicit iteration**.



SETTING HTML CONTENT

This example will replace the content of **each list item** with the word **Updated** in `` tags.

```
$('li').html('<b>Updated</b>');
```



SETTING TEXT CONTENT

This example will replace the text content of **each list item** with the word **Updated**.

```
$('li').text('Updated');
```



CHAINING

It is possible to call multiple methods on the same selection.

```
$('li').hide().fadeIn(500);
```

A horizontal line with a vertical tick mark pointing down to the word 'hide' in the code above.
FIRST METHOD



CHAINING

It is possible to call multiple methods on the same selection.

```
$('li').hide().fadeIn(500);
```

SECOND METHOD



CHECKING A PAGE IS
READY TO WORK WITH



jQuery's `.ready()` method checks that the page is ready for your code to work with.



Checks to see if the page has loaded before...

JQUERY METHOD

```
$(document).ready(function() {  
    // code goes here  
});
```



...running the code inside
the parentheses

CODE

```
$(document).ready(function() {  
    // code goes here  
});
```



This is a shortcut for writing
the `.ready()` method.



```
$(document).ready(function() {  
    // code goes here  
});
```

```
$(function() {  
    // code goes here  
});
```



MORE ABOUT UPDATING ELEMENTS



INSERT ELEMENTS

`.before()`

`.after()`

`item`

`.prepend()`

`.append()`



GET & SET ATTRIBUTES

```
.attr()           .removeAttr()  
.                .removeClass()  
addClass()
```

```
$( 'a#top' ).addClass( 'button' );
```



EVENTS



EVENTS

focus, blur, change, input,
keydown, keyup, keypress,
click, dblclick, mouseup,
mousedown, mouseover,
mousemove, mouseout, hover,
submit, select, ready, load,
unload, error, resize, scroll



EVENT METHODS

The `.on()` method handles events:

```
$('li').on('click', function() {  
    // code to happen on click  
});
```



EVENT METHODS

The type of event is the first argument:

```
$(`li`).on(`click`, function() {  
    // code to happen on click  
});
```



EVENT METHODS

When the event happens, an anonymous function is called:

```
$(`li`).on(`click`, function() {  
    // code to happen on click  
});
```



EVENT OBJECT

Passed into event handling function:

```
$( 'li' ).on( 'click', function(e) {  
    var eventType = e.type;  
});
```

