

Event

Th.S Lê Quang Song
Founder: MyClass.vn - CyberSoft.edu.vn

CyberSoft Academy [CyberSoft.edu.vn]

UI EVENTS

Occur when a user interacts with the browser's user interface (UI) rather than the web page

EVENT	DESCRIPTION
load	Web page has finished loading
unload	Web page is unloading (usually because a new page was requested)
error	Browser encounters a JavaScript error or an asset doesn't exist
resize	Browser window has been resized
scroll	User has scrolled up or down the page

KEYBOARD EVENTS

Occur when a user interacts with the keyboard (see also `input` event)

EVENT	DESCRIPTION
keydown	User first presses a key (repeats while key is depressed)
keyup	User releases a key
keypress	Character is being inserted (repeats while key is depressed)

MOUSE EVENTS

Occur when a user interacts with a mouse, trackpad, or touchscreen

EVENT	DESCRIPTION
click	User presses and releases a button over the same element
dblclick	User presses and releases a button twice over the same element
mousedown	User presses a mouse button while over an element
mouseup	User releases a mouse button while over an element
mousemove	User moves the mouse (not on a touchscreen)
mouseover	User moves the mouse over an element (not on a touchscreen)
mouseout	User moves the mouse off an element (not on a touchscreen)

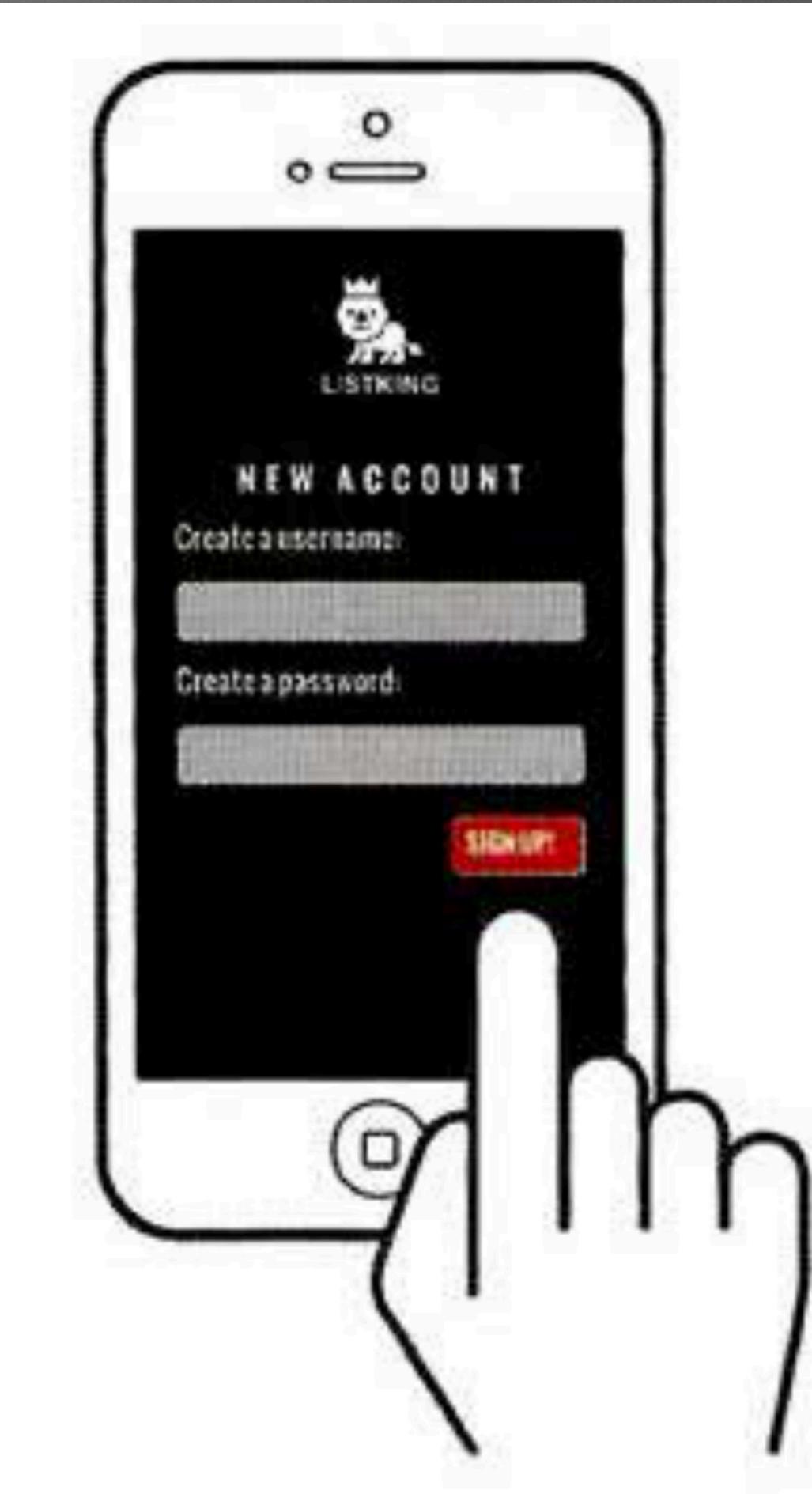
TERMINOLOGY

EVENTS FIRE OR ARE RAISED

When an event has occurred, it is often described as having **fired** or been **raised**. In the diagram on the right, if the user is tapping on a link, a **click** event would fire in the browser.

EVENTS TRIGGER SCRIPTS

Events are said to **trigger** a function or script. When the **click** event fires on the element in this diagram, it could trigger a script that enlarges the selected item.



FOCUS EVENTS

Occur when an element (e.g., a link or form field) gains or loses focus

EVENT

DESCRIPTION

focus / focusin

Element gains focus

blur / focusout

Element loses focus

FORM EVENTS

Occur when a user interacts with a form element

EVENT

DESCRIPTION

input

Value in any <input> or <textarea> element has changed (IE9+) or any element with the contenteditable attribute

change

Value in select box, checkbox, or radio button changes (IE9+)

submit

User submits a form (using a button or a key)

reset

User clicks on a form's reset button (rarely used these days)

cut

User cuts content from a form field

copy

User copies content from a form field

paste

User pastes content into a form field

select

User selects some text in a form field

Event Handling

1

Select the **element** node(s) you want the script to respond to.

For example, if you want to trigger a function when a user clicks on a specific link, you need to get the DOM node for that link element. You do this using a DOM query (see Chapter 5).

2

Indicate which **event** on the selected node(s) will trigger the response.

Programmers call this **binding** an event to a DOM node.

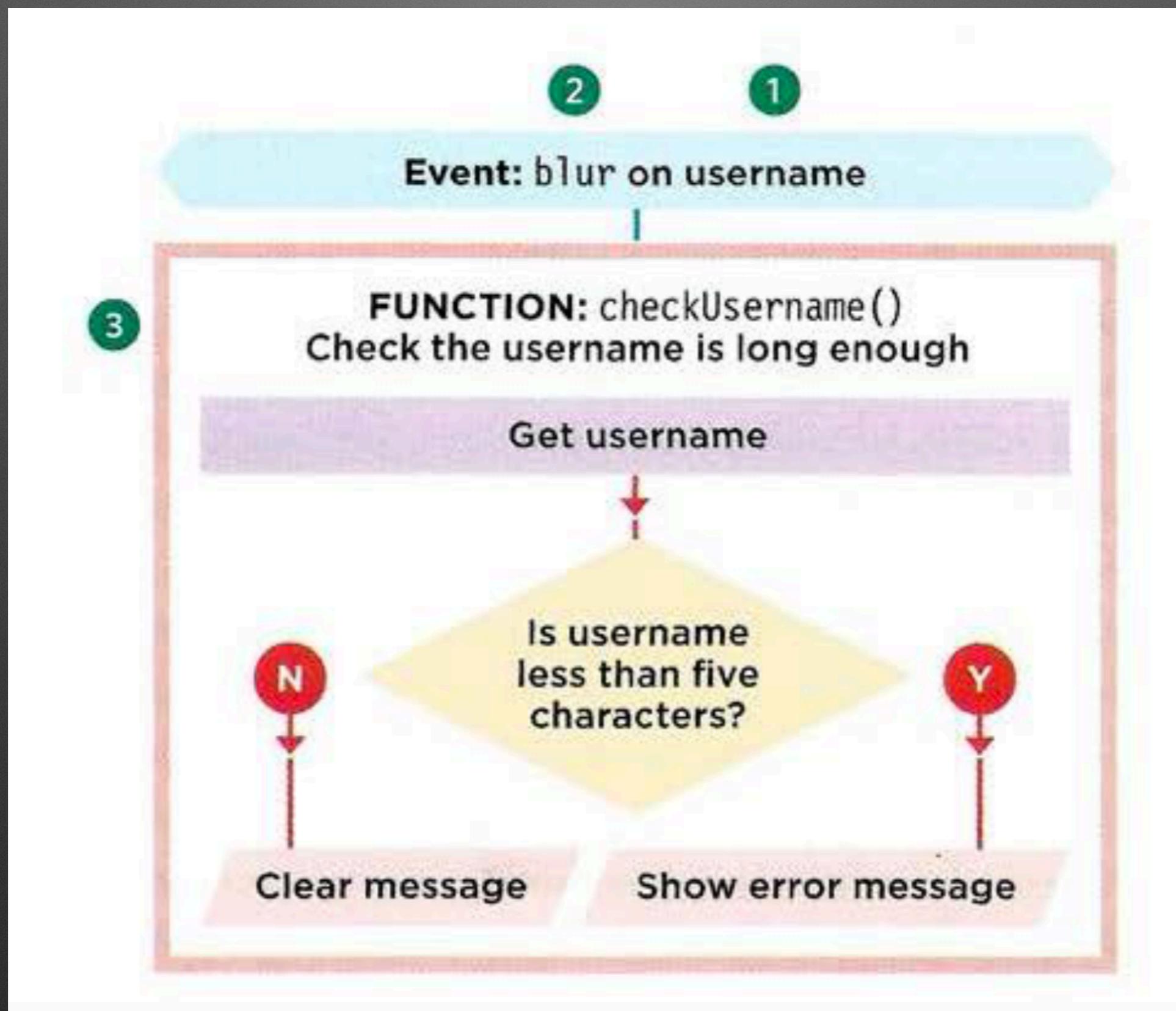
The previous two pages showed a selection of the popular events that you can monitor for.

3

State the **code** you want to run when the event occurs.

When the event occurs, on a specified element, it will trigger a function. This may be a named or an anonymous function.

Ví dụ



3 cách để Bind 1 Event đến một thẻ

HTML EVENT HANDLERS

See p251

**This is bad practice, but you
need to be aware of it because
you may see it in older code.**

Early versions of HTML included
a set of attributes that could
respond to events on the

TRADITIONAL DOM EVENT HANDLERS

See p252

DOM event handlers were
introduced in the original
specification for the DOM.
They are considered better than
HTML event handlers because
they let you separate the
JavaScript from the HTML.

DOM LEVEL 2 EVENT LISTENERS

See p254

Event listeners were introduced
in an update to the DOM
specification (DOM level 2,
released in the year 2000).
They are now the favored way of
handling events.

Cách 1 (KHÔNG NÊN XÀI) - HTML EVENT HANDLERS

HTML

c06/event-attributes.html

```
<form method="post" action="http://www.example.org/register">
  <label for="username">Create a username: </label>
  <input type="text" id="username" onblur="checkUsername()" />
  <div id="feedback"></div>

  <label for="password">Create a password: </label>
  <input type="password" id="password" />

  <input type="submit" value="Sign up!" />
</form>
...
<script type="text/javascript" src="js/event-attributes.js"></script>
```

JAVASCRIPT

c06/js/event-attributes.js

```
function checkUsername() {                                // Declare function
  var elMsg = document.getElementById('feedback');      // Get feedback element
  var elUsername = document.getElementById('username'); // Get username input
  if (elUsername.value.length < 5) {                    // If username too short
    elMsg.textContent = 'Username must be 5 characters or more'; // Set msg
  } else {                                              // Otherwise
    elMsg.textContent = '';                            // Clear message
  }
}
```

Cách 2 - TRADITIONAL DOM EVENT HANDLERS

- Tất cả trình duyệt đều hiểu cách này, tuy nhiên chỉ có thể gắn 1 function đến một Event Handlers

element.onevent = functionName;

ELEMENT

EVENT

CODE

DOM element
node to target

Event bound to node(s)
preceded by word "on"

Name of function to call (with
no parentheses following it)

A reference
to the DOM
element node
is often stored
in a variable.

```
function checkUsername() {  
    // code to check the length of username  
}  
[var el = document.getElementById('username');  
el.onblur = checkUsername;]
```

The event name is preceded by the word "on."

The code starts
by defining the
named function.

The function
is called by the
event handler on
the last line, but
the parentheses
are omitted.

Cách 2 - TRADITIONAL DOM EVENT HANDLERS

- Ví dụ

JAVASCRIPT

c06/js/event-handler.js

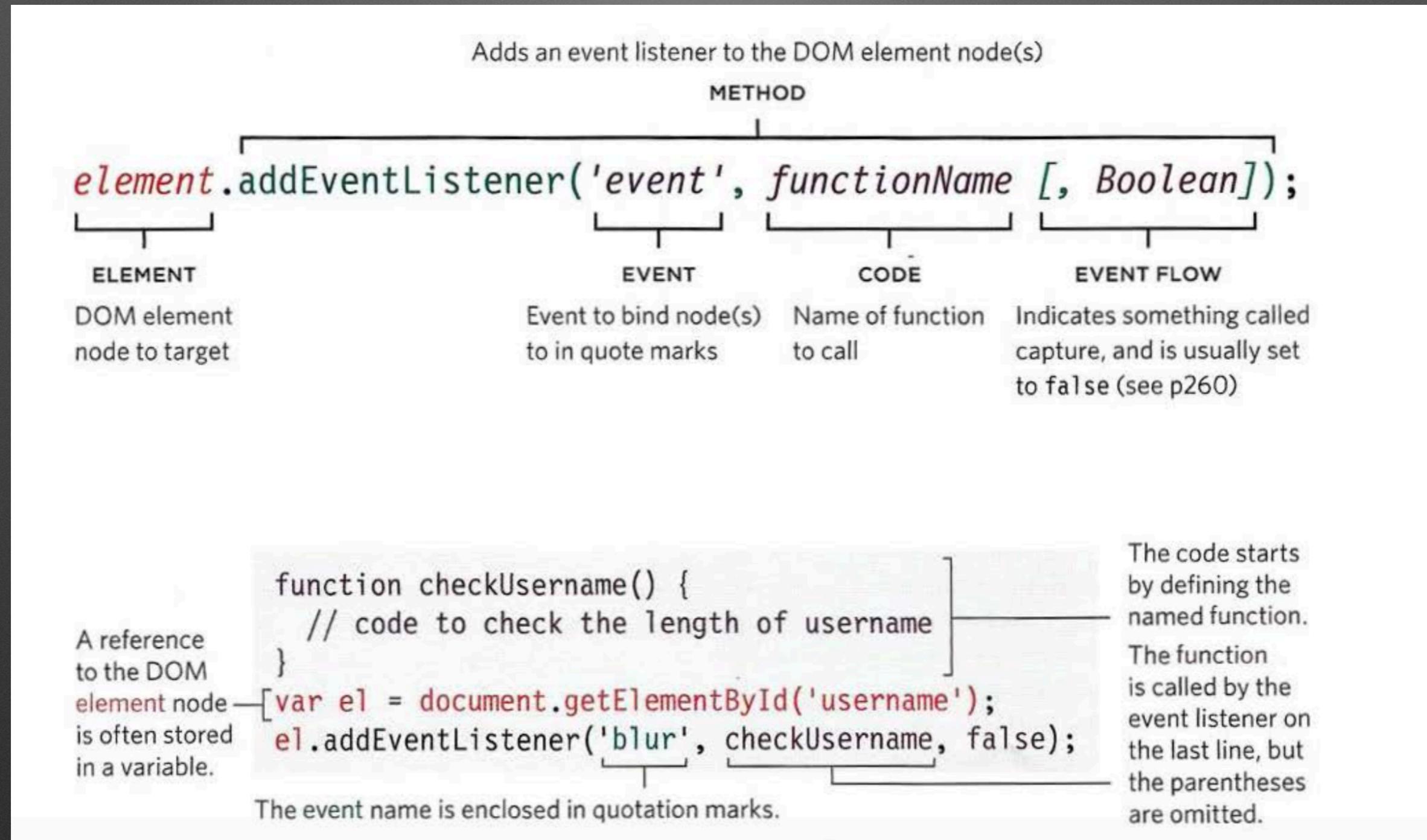
```
function checkUsername() { // Declare function
    var e1Msg = document.getElementById('feedback'); // Get feedback element
    if (this.value.length < 5) { // If username too short
        ① e1Msg.textContent = 'Username must be 5 characters or more'; // Set msg
    } else { // Otherwise
        e1Msg.textContent = ''; // Clear message
    }
}

② var e1Username = document.getElementById('username'); // Get username input
③ e1Username.onblur = checkUsername; // When it loses focus call checkUserName()
```

jquery thay thế: ***element.blur(checkUsername);***

Cách 3 - EVENT LISTENER

- Đây là cách tiếp cận mới nhất - có thể gắn nhiều hàm cùng lúc, TUY NHIÊN, không hỗ trợ các trình duyệt cũ



jquery thay thế: bind hoặc on thay addEventListener

Cách 3 - EVENT LISTENER

- Cách sử dụng

JAVASCRIPT

c06/js/event-listener.js

```
function checkUsername() {                                // Declare function
    var elMsg = document.getElementById('feedback');      // Get feedback element
    if (this.value.length < 5) {                          // If username too short
        elMsg.textContent = 'Username must be 5 characters or more'; // Set msg
    } else {                                              // Otherwise
        elMsg.textContent = '';                           // Clear msg
    }
}

② var elUsername = document.getElementById('username'); // Get username input
// When it loses focus call checkUsername()
elUsername.addEventListener('blur', checkUsername, false);
```

i ii iii

jquery thay thế: bind hoặc on thay addEventListener

Sử dụng EVENT LISTENER với THAM SỐ

The named function includes parentheses containing the parameter after the function name.

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

Event name

Start of anonymous function

```
function() {
```

End of statement

End of addEventListener() method

Event flow Boolean (see p260)

End of anonymous function

The anonymous function is used as the second argument. It "wraps around" the named function.

Sử dụng EVENT LISTENER với THAM SỐ

- Cách sử dụng

JAVASCRIPT

c06/js/event-listener-with-parameters.js

```
var elUsername = document.getElementById('username'); // Get username input
var elMsg = document.getElementById('feedback'); // Get feedback element

function checkUsername(minLength) { // Declare function
    if (elUsername.value.length < minLength) { // If username too short
        // Set the error message
        elMsg.textContent = 'Username must be ' + minLength + ' characters or more';
    } else { // Otherwise
        elMsg.innerHTML = ''; // Clear msg
    }
}

elUsername.addEventListener('blur', function() { // When it loses focus
    checkUsername(5); // Pass arguments here
}, false);
```

TRÌNH DUYỆT CŨ?

KIỂM TRA HỖ TRỢ

If the browser supports
addEventListener():

Run the code inside
these curly braces

If it doesn't, do
something else:

Run the code inside
these curly braces

```
if (el.addEventListener) {  
    el.addEventListener('blur', function() {  
        checkUsername(5);  
    }, false );  
}  
else {  
    el.attachEvent('onblur', function() {  
        checkUsername(5);  
    });  
}
```

```
var elUsername = document.getElementById('username'); // Get username input
var elMsg = document.getElementById('feedback'); // Get feedback element

function checkUsername(minLength) { // Declare function
    if (elUsername.value.length < minLength) { // If username too short
        // Set message
        elMsg.innerHTML = 'Username must be ' + minLength + ' characters or more';
    } else { // Otherwise
        elMsg.innerHTML = ''; // Clear message
    }
}

if (elUsername.addEventListener) { // If event listener supported
    elUsername.addEventListener('blur', function(){// When username loses focus
        checkUsername(5); // Call checkUsername()
    }, false ); // Capture during bubble phase
} else {
    elUsername.attachEvent('onblur', function(){ // IE fallback: onblur
        checkUsername(5); // Call checkUsername()
    });
}
```

VÍ DỤ FOCUS & BLUR



VÍ DỤ FOCUS & BLUR

JAVASCRIPT

c06/js/focus-blur.js

```
function checkUsername() { // Declare function
    var username = el.value; // Store username in variable
    if (username.length < 5) { // If username < 5 characters
        elMsg.className = 'warning'; // Change class on message
        elMsg.textContent = 'Not long enough, yet...'; // Update message
    } else { // Otherwise
        elMsg.textContent = ''; // Clear the message
    }
}
function tipUsername() { // Declare function
    elMsg.className = 'tip'; // Change class for message
    elMsg.innerHTML = 'Username must be at least 5 characters'; // Add message
}

var el = document.getElementById('username'); // Username input
var elMsg = document.getElementById('feedback'); // Element to hold message

// When the username input gains / loses focus call functions above:
el.addEventListener('focus', tipUsername, false); // focus call tipUsername()
el.addEventListener('blur', checkUsername, false); // blur call checkUsername()
```

Demo KeyPress



Demo KeyPress

JAVASCRIPT

c06/js/keypress.js

```
var el; // Declare variables

function charCount(e) { // Declare function
    var textEntered, charDisplay, counter, lastKey; // Declare variables
    textEntered = document.getElementById('message').value; // User's text
    charDisplay = document.getElementById('charactersLeft'); // Counter element
    counter = (180 - (textEntered.length)); // Num of chars left
    charDisplay.textContent = counter; // Show chars left

    lastkey = document.getElementById('lastkey'); // Get last key used
    lastkey.textContent = 'Last key in ASCII code: ' + e.keyCode; // Create msg
}
el = document.getElementById('message'); // Get msg element
el.addEventListener('keypress', charCount, false); // keypress event
```