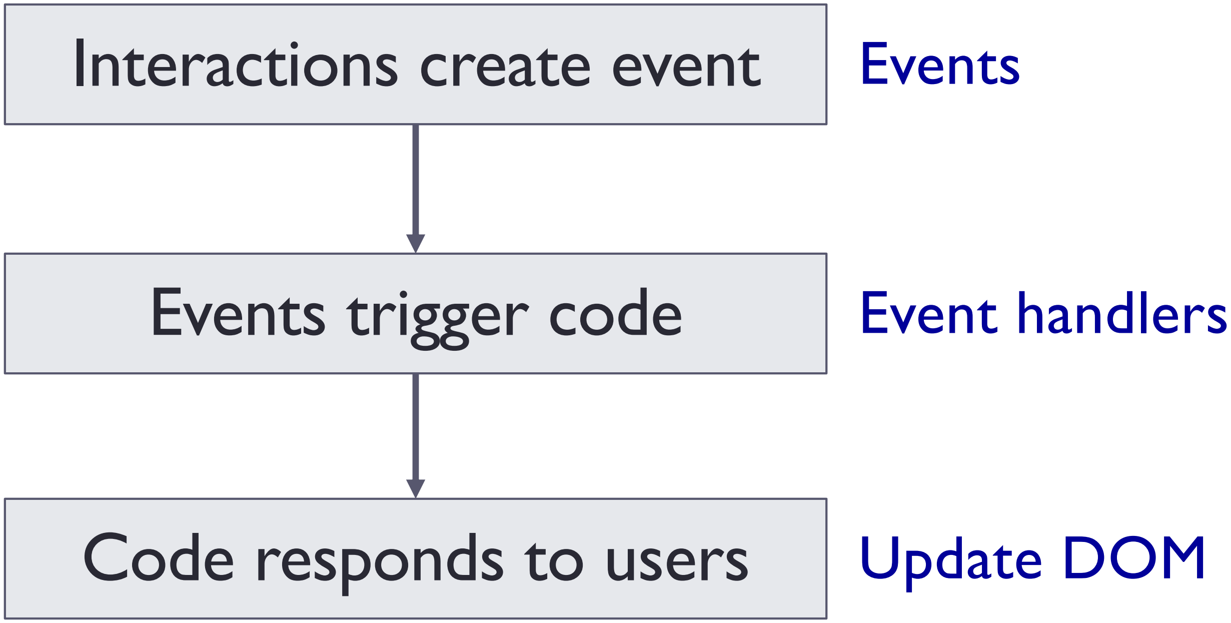
JavaScript and Events

CS 4640
Programming Languages for Web Applications

[Robert W. Sebesta, "Programming the World Wide Web
Jon Duckett, Interactive Frontend Web Development]

Events



Interactions create event — Events

Events trigger code — Event handlers

Code responds to users — Update DOM

Scripts often respond to events by updating the content of the page (via DOM) which makes the page appear interactive

(Some) Event Types

UI events – occur when a user interacts with the browser's user interface (UI) – work with window object

UI Events	Description
load	Web page has (just) finished loading
unload	Web page is unloading
error	Browser encounters a JavaScript error or unavailable web resources
resize	Browser window has been resized
scroll	User has scrolled up or down the page

(Some) Event Types

Focus events – occur when an element gains or loses focus

Focus Events	Description
focus / focusin	Element gains focus
blur / focusout	Focus loses focus

Keyboard events – occur when a user interacts with the keyboard

Keyboard Events	Description
keydown	User first presses a key
keyup	User releases a key
keypress	Character is being inserted

(Some) Event Types

Mouse events – occur when a user interacts with a mouse, trackpad, or touchscreen

Mouse Events	Description
click	User presses and releases a button over the same element
dbclick	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 a mouse (not on a touchscreen)
mouseover	User moves a mouse over an element (not on a touchscreen)
mouseout	User moves a mouse off an element (not on a touchscreen)

(Some) Event Types

Form events – occur when a user interacts with a form element

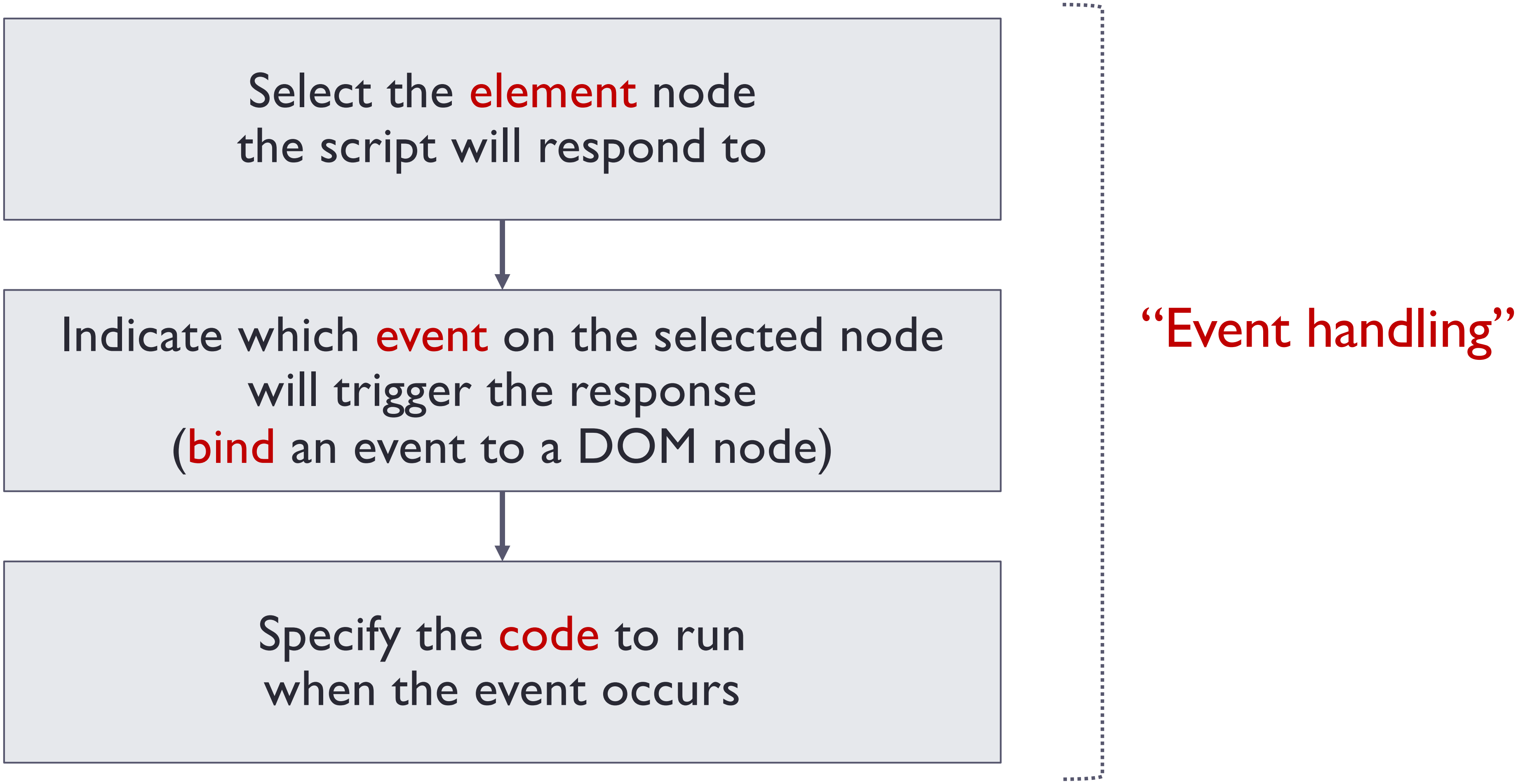
Form Events	Description
input	Value in any <input> or <textarea> element has changed or any element with t he contenteditable attribute
change	Value in select box, checkbox, or radio button changes
submit	User submits a form (using a button or a key)
reset	User clicks on a form's reset button
cut	User cuts content from a form field
paste	User pastes content into a form field
select	Users selects some text in a form field

(Some) Event Types

Mutation events – occur when the DOM structure has been changed by a script

Mutation Events	Description
DOMSubtreeModified	Change has been made to a document
DOMNodeInserted	Node has been inserted as a direct child of another node
DOMNodeRemoved	Node has been removed from another node

How Events Trigger Code



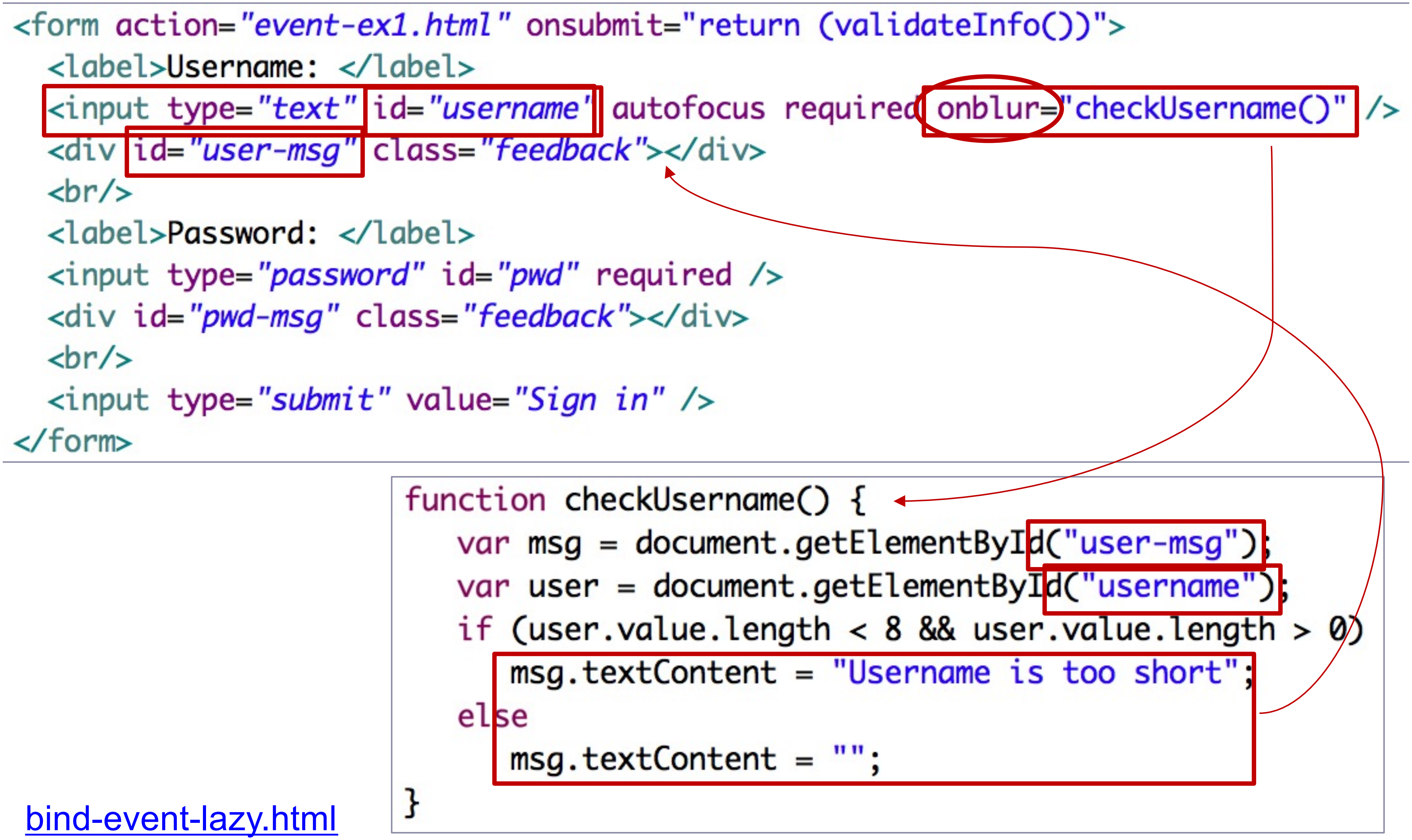
Select the element node the script will respond to

Indicate which event on the selected node will trigger the response (bind an event to a DOM node)

Specify the code to run when the event occurs

“Event handling”

Example: Event Handling (Lazy)

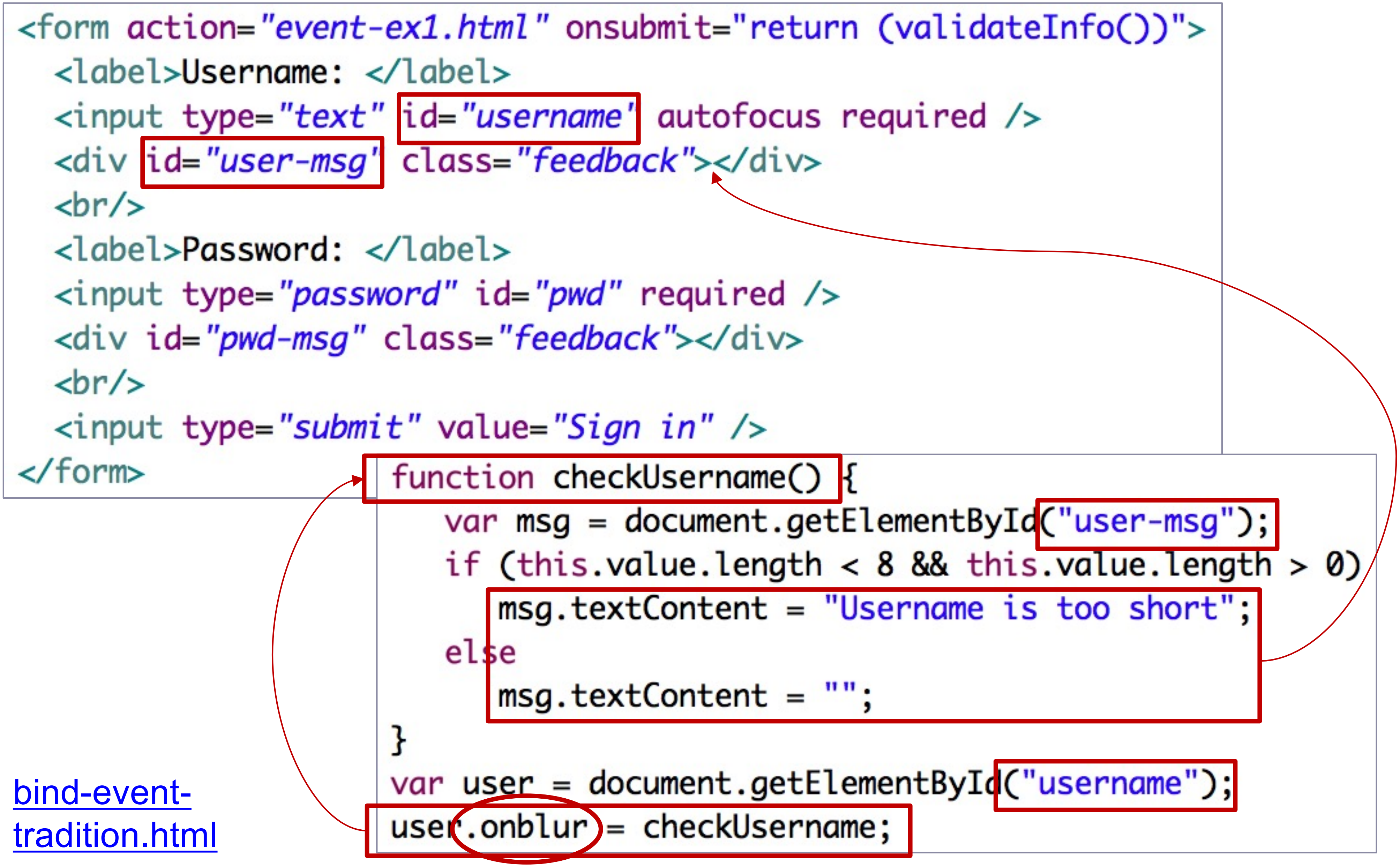


```
<form action="event-ex1.html" onsubmit="return (validateInfo())">
  <label>Username: </label>
  <input type="text" id="username" autofocus required onblur="checkUsername()" />
  <div id="user-msg" class="feedback"></div>
  <br/>
  <label>Password: </label>
  <input type="password" id="pwd" required />
  <div id="pwd-msg" class="feedback"></div>
  <br/>
  <input type="submit" value="Sign in" />
</form>
```

```
function checkUsername() {
   var msg = document.getElementById("user-msg");
   var user = document.getElementById("username");
   if (user.value.length < 8 && user.value.length > 0)
      msg.textContent = "Username is too short";
   else
      msg.textContent = "";
}
```

bind-event-lazy.html

Example: Event Handling (Tradition)

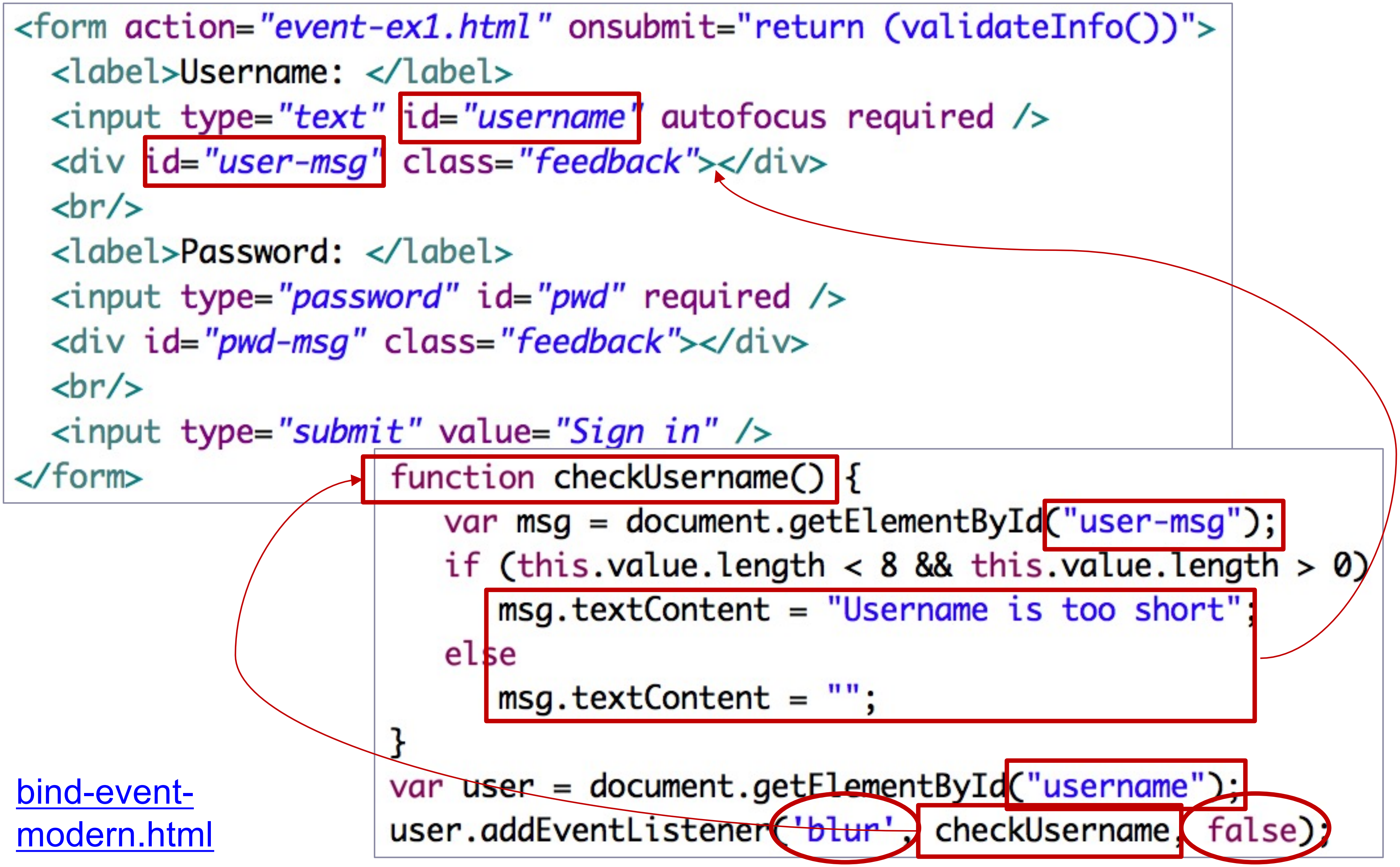


```
<form action="event-ex1.html" onsubmit="return (validateInfo())">
  <label>Username: </label>
  <input type="text" id="username" autofocus required />
  <div id="user-msg" class="feedback"></div>
  <br/>
  <label>Password: </label>
  <input type="password" id="pwd" required />
  <div id="pwd-msg" class="feedback"></div>
  <br/>
  <input type="submit" value="Sign in" />
</form>
```

```
function checkUsername() {
    var msg = document.getElementById("user-msg");
    if (this.value.length < 8 && this.value.length > 0)
        msg.textContent = "Username is too short";
    else
        msg.textContent = "";
}
var user = document.getElementById("username");
user.onblur = checkUsername;
```

bind-event-tradition.html

Example: Event Handling (Modern)

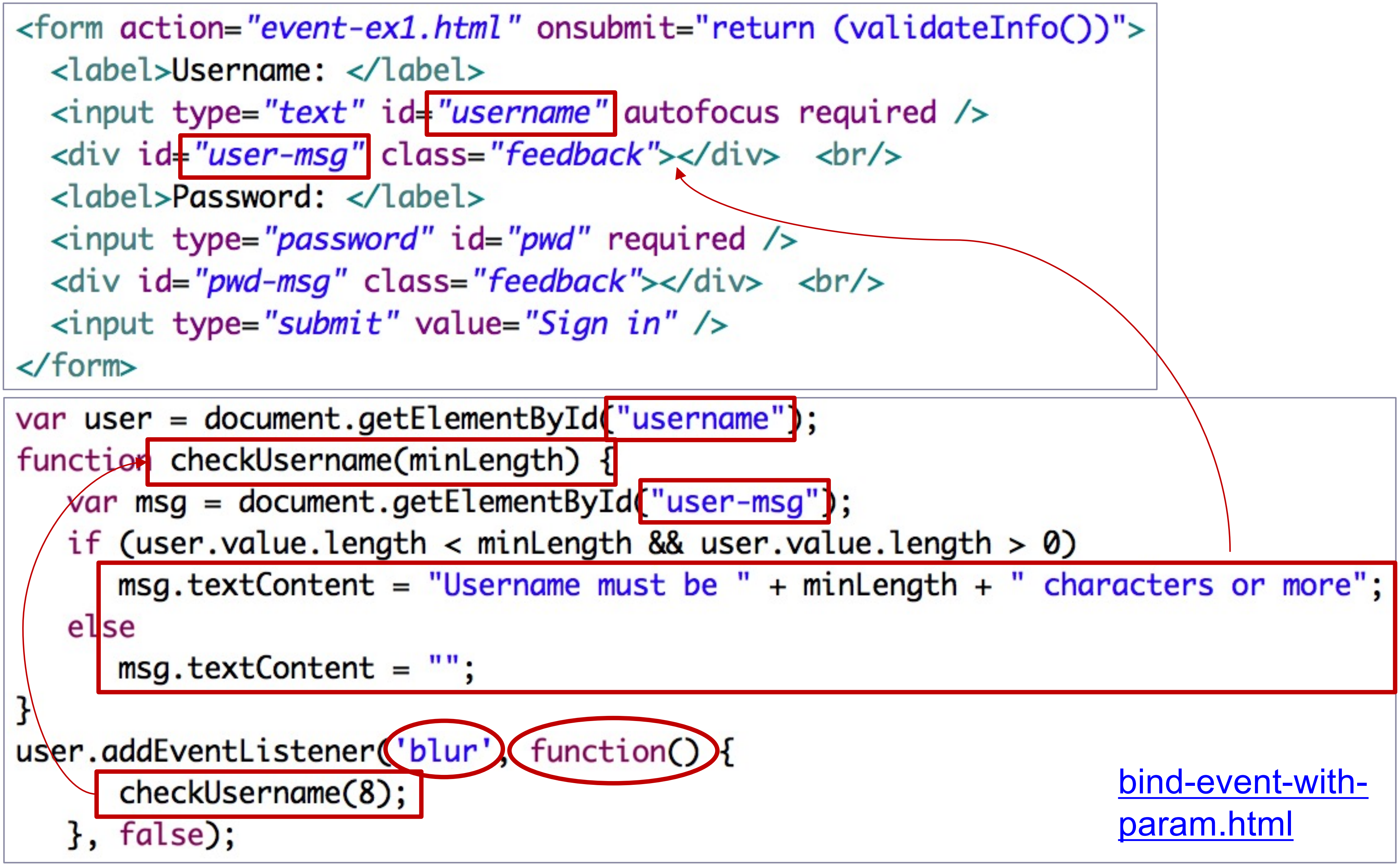


```html
<form action="event-ex1.html" onsubmit="return (validateInfo())">
  <label>Username: </label>
  <input type="text" id="username" autofocus required />
  <div id="user-msg" class="feedback"></div>
  <br/>
  <label>Password: </label>
  <input type="password" id="pwd" required />
  <div id="pwd-msg" class="feedback"></div>
  <br/>
  <input type="submit" value="Sign in" />
</form>
```

```javascript
function checkUsername() {
    var msg = document.getElementById("user-msg");
    if (this.value.length < 8 && this.value.length > 0)
        msg.textContent = "Username is too short";
    else
        msg.textContent = "";
}
var user = document.getElementById("username");
user.addEventListener('blur', checkUsername, false);
```

bind-event-modern.html

Example: Using Parameters with Event Listener

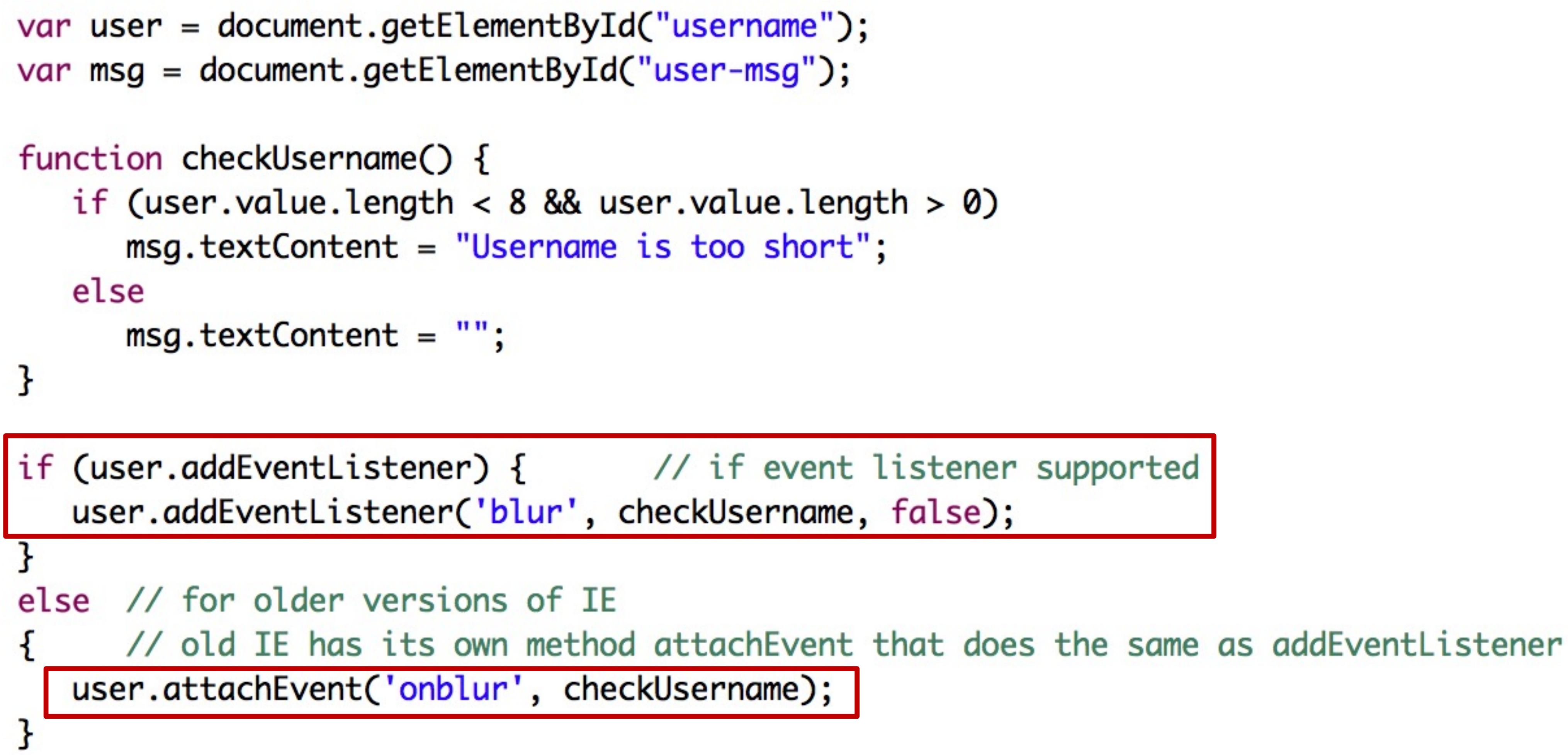


```
<form action="event-ex1.html" onsubmit="return (validateInfo())">
  <label>Username: </label>
  <input type="text" id="username" autofocus required />
  <div id="user-msg" class="feedback"></div>  <br/>
  <label>Password: </label>
  <input type="password" id="pwd" required />
  <div id="pwd-msg" class="feedback"></div>  <br/>
  <input type="submit" value="Sign in" />
</form>
```

```
var user = document.getElementById("username");
function checkUsername(minLength) {
   var msg = document.getElementById("user-msg");
   if (user.value.length < minLength && user.value.length > 0)
      msg.textContent = "Username must be " + minLength + " characters or more";
   else
      msg.textContent = "";
}
user.addEventListener('blur', function() {
     checkUsername(8);
   }, false);
```

bind-event-with-param.html

Example: Supporting Older Version of IE (w/o Param)



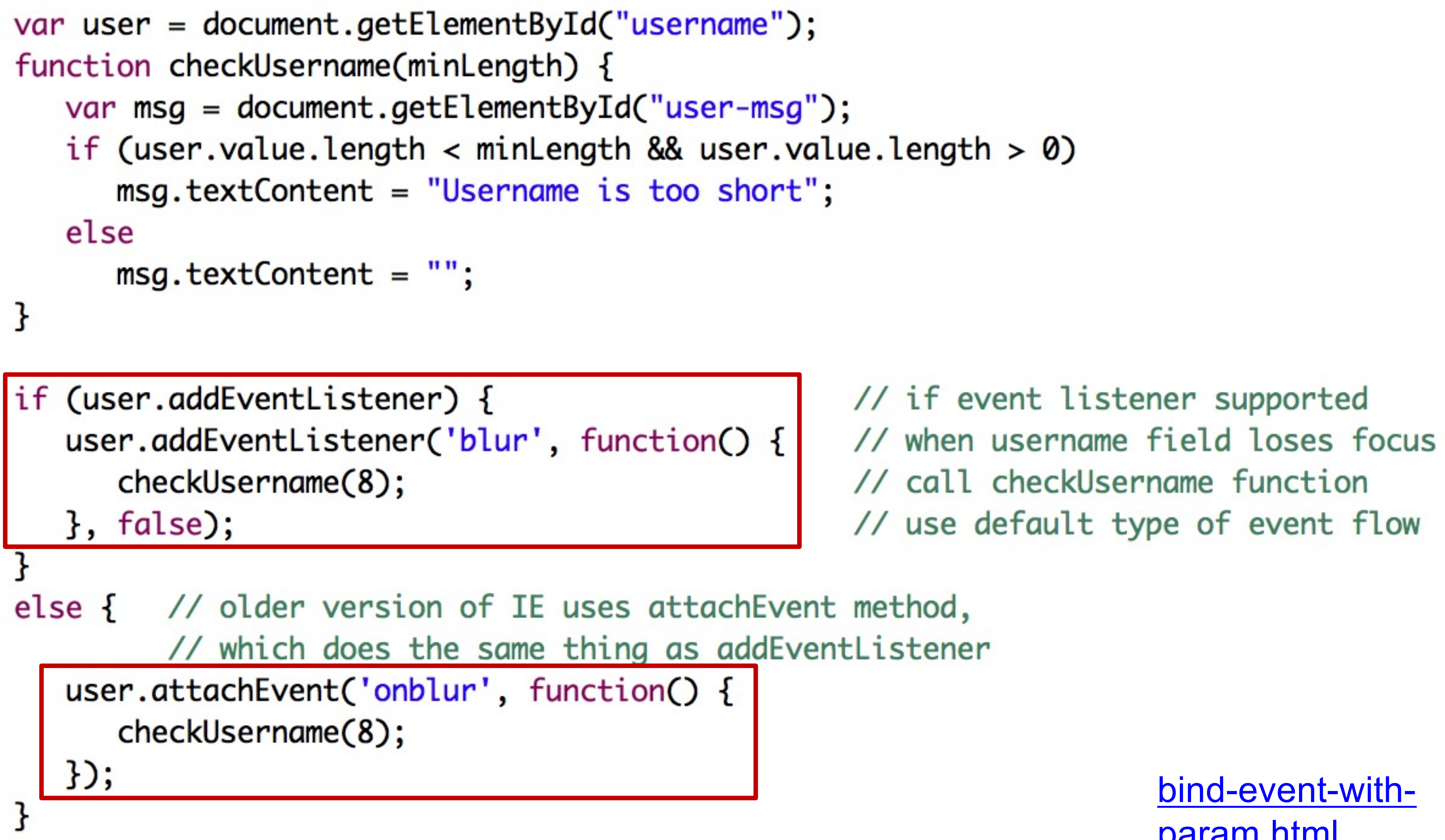
```
var user = document.getElementById("username");
var msg = document.getElementById("user-msg");

function checkUsername() {
   if (user.value.length < 8 && user.value.length > 0)
      msg.textContent = "Username is too short";
   else
      msg.textContent = "";
}

if (user.addEventListener) {       // if event listener supported
   user.addEventListener('blur', checkUsername, false);
}
else  // for older versions of IE
{     // old IE has its own method attachEvent that does the same as addEventListener
   user.attachEvent('onblur', checkUsername);
}
```

bind-event-support-multiple-browsers.html

Example: Supporting Older Version of IE (w/ Param)



```
var user = document.getElementById("username");
function checkUsername(minLength) {
   var msg = document.getElementById("user-msg");
   if (user.value.length < minLength && user.value.length > 0)
      msg.textContent = "Username is too short";
   else
      msg.textContent = "";
}

if (user.addEventListener) {                       // if event listener supported
   user.addEventListener('blur', function() {      // when username field loses focus
      checkUsername(8);                            // call checkUsername function
   }, false);                                      // use default type of event flow
}
else {    // older version of IE uses attachEvent method,
          // which does the same thing as addEventListener
   user.attachEvent('onblur', function() {
      checkUsername(8);
   });
}
```

bind-event-with-param.html

Summary

- Events occur when user interacts with a web browser or a web page (screen).

- Binding specifies the event that can happen and on which element the event happens.

- When an event occurs on an element, it can trigger a JavaScript function. The function then manipulates the page to response to the user.