9.3 Events

Callback functions

Web developers may use jQuery to write code that executes in response to an event. The `on()` method attaches a callback function to the selected elements of a `jQuery` object. A ***callback function*** is a function that is executed when an event occurs. A callback function is also called an ***event listener*** or ***event handler***. jQuery code for registering callback functions is more concise and easier to read than using plain JavaScript.

Figure 9.3.1: Registering callback functions in JavaScript and jQuery.

```
// Example callback function
function myCallback() {
  console.log("You clicked the button");
}

// Registering a click callback function with plain JavaScript
document.getElementById("mybutton").addEventListener("click",
myCallback);

// Registering a click callback function with jQuery
$("#mybutton").on("click", myCallback);

// or simply
$("#mybutton").click(myCallback);
```

Feedback?

jQuery also provides a number of shortcut methods for handling common events. Ex: `click()` can be used to register a click callback function instead of the more verbose `on("click")` method.

Table 9.3.1: Mouse events and shortcut methods.

Event	Shortcut	Description
click	`click()`	Triggered when the mouse clicks on an element.
dblclick	`dblclick()`	Triggered when the mouse double-clicks on an element.

Event	Shortcut	Description
mouseover / mouseout	`mouseover() / mouseout()`	Triggered only once when the mouse pointer moves over or leaves an element and any of the element's children.

mouseenter / mouseleave	`mouseenter() / mouseleave()`
	`hover()`
		Triggered only once when the mouse pointer moves over or leaves an element. `hover()` binds a callback function, or functions, for both events

mousemove	`mousemove()`	Triggered when the mouse pointer moves over an element.

Feedback?

PARTICIPATION ACTIVITY 9.3.1: Mouse events. ✓

Given the following JavaScript, match the equivalent jQuery code segments.

```
function doThis() {
   console.log("doThis");
}

function doThat() {
   console.log("doThat");
}
```

If unable to drag and drop, refresh the page.

$("p").on("dblclick", doThis);

$("p").dblclick(doThis);

`.dblclick()` is the shortcut for the dblclick event

Correct

$("p").hover(doThis, doThat);

$("p").mouseenter(doThis).mouseleave(doThat);

`.hover()` simplifies the process of registering mouseenter and mouseleave callback functions.

$("p").mousemove(doThis);	$("p").mousemove(function() { console.log("doThis"); }); Callback functions can be anonymous functions.	Correct

Reset

Feedback?

The ready event

One of the most important jQuery events is the `ready` event. The ***ready event*** is triggered when the browser has finished loading the webpage's DOM. *Good practice is to use jQuery selectors inside the `ready` callback function to ensure the DOM is fully loaded before searching the DOM for elements.*

Figure 9.3.2: Waiting for the ready event.

```
// Wait until the document's DOM is
ready
$(document).ready(function() {
   // DOM is ready to go
   $("button").addClass("big");
});
```

Feedback?

Developers use the `ready` event so frequently that the shortcut `ready()` method is not needed. The `$()` function can be supplied a callback function that will only be executed when the `ready` event triggers on the document.

Figure 9.3.3: Registering a ready event callback function with $().

```
$(function() {
   // DOM is ready to go

$("button").addClass("big");
});
```

Feedback?

PARTICIPATION ACTIVITY

9.3.2: The ready event. ✓

1) Developers typically use a ready event callback function when the jQuery code is located at the end of the webpage.

● True

○ False

✓

Correct

The DOM will be loaded by the time the jQuery code executes at the end of the page. However, using the ready event callback ensures the DOM is ready regardless of where the code is located.

2) The code `$(document).ready(doThis);` is equivalent to `$(doThis);` where `doThis` is a function.

● True

○ False

✓

Correct

The shorter syntax is preferred by most developers.

3) The ready event is triggered after all the page's images, videos, and scripts have finished downloading.

○ True

● False

✓

Correct

The ready event is triggered when the DOM is loaded, not when all page resources are loaded. The load event is triggered after all page resources have been downloaded. Ex:

```
$(window).on("load", function() {
   // All resources are now loaded
});
```

Feedback?

PARTICIPATION ACTIVITY | 9.3.3: Mouse event practice. ✓

The following HTML uses CSS and JavaScript to render the webpage. Try modifying the JavaScript so that the rendered webpage matches the expected webpage. Do not modify the HTML or CSS.

1. Make the words "Click me!" become large when the mouse is hovering over the words and return to regular size when the mouse is not hovering over the words. Do this by modifying the call to `hover()` so the `big` class is added or removed to/from the `<div>` at the right time.
2. Make the words "Click me!" alternate between red and blue when the user clicks on the words. Do this by adding a "click" event callback function to the `<div>` and calling the `toggleClass()` method to add or remove the `red` and `blue` classes to the `<div>`.

3. Although the JavaScript code will work without a ready event callback function, put all of the JavaScript code inside a ready event callback function, just for practice.

HTML | CSS | JavaScript

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jq

<div class="red">
  Click me!
</div>

```

Render webpage   Reset code

Your webpage

Click me!

Expected webpage

Click me!

▶ View solution

Feedback?

Keyboard and form events

jQuery supports keyboard events that are helpful when a webpage needs to react to the user pressing or releasing keys on a keyboard. Ex: Moving a spaceship left or right depending on what key is pressed. jQuery also supports form events that are helpful when the webpage needs to perform data validation. Ex: Ensuring the credit card field contains the correct number of digits when the user presses the submit button.

Table 9.3.2: Keyboard events and shortcuts.

Event	Shortcut	Description
keydown	`keydown()`	Triggered when the user first presses a key on the keyboard.
keyup	`keyup()`	Triggered when the user releases a key from the keyboard.
keypress	`keypress()`	Triggered when the browser registers keyboard input from printable character keys. Non-printing keys like Shift and Esc do not register keypress events.

Feedback?

Table 9.3.3: Form events and shortcuts.

Event	Shortcut	Description
focus / blur	`focus()` / `blur()`	Triggered when an element gains or loses focus.
focusin / focusout	`focusin()` / `focusout()`	Triggered when an element or any of the element's children gain or lose focus.
change	`change()`	Triggered when an element's value changes.
select	`select()`	Triggered when a user selects text in an `<input type="text">` or `<textarea>` element.
submit	`submit()`	Triggered when the user is attempting to submit a form.

Feedback?

PARTICIPATION ACTIVITY

9.3.4: Keyboard and form events.

1) What event is helpful for writing a callback function that ensures the password the user is entering is long enough while the password is being typed?

○ blur

● keyup

○ change

Correct

The keyup event is triggered each time a key is pressed and released, so the length of the text in the password field can be examined while the user types.

2) What event is helpful for writing a callback function that ensures the user typed a valid phone number after the user moves to another input field?

○ keypress

○ submit

● blur

Correct

The blur event is triggered when the phone number input field loses focus. The change event would also work since the change event is triggered when the focus is lost for text fields.

3) What event is helpful for writing a callback function that ensures a radio button was selected when the user pressed the form's submit button?

● submit

○ focusout

○ select

Correct

The submit event is triggered when the user presses the submit button. The radio button could be examined and the form submission cancelled if the radio button is not selected.

Feedback?

PARTICIPATION ACTIVITY

9.3.5: Keyboard event practice.

The following webpage converts Celsius to Fahrenheit and vice versa. The JavaScript code uses the jQuery method `val()` to extract the number that was typed into one of the text input fields when Convert is pressed. The jQuery function `$.isNumeric()` determines if what was typed is a number or not. If a number has not been typed, no conversion is performed. Otherwise `val()` is used to put the converted temperature into the appropriate text field.

Try modifying the JavaScript so the rendered webpage matches the behavior of the expected webpage. Do not modify the HTML or CSS.

1. Make the °F or °C label appear blue when focus is given to the Fahrenheit or Celsius input field, respectively. Use the focus and blur events to add or remove the CSS class `selected` to the labels.
2. Blank-out the °F input field when the user types anything into the °C field and vice versa. Use the keyup event on each of the respective input fields to change the opposing field into an empty string with the `val()` method.
3. Add the `error` class to the °F or °C text fields if the user has typed anything but a number. Remove the `error` class if the user has typed a number. Use the keyup event on each of the respective input fields to test the user input while the input is being typed.

HTML | CSS | JavaScript

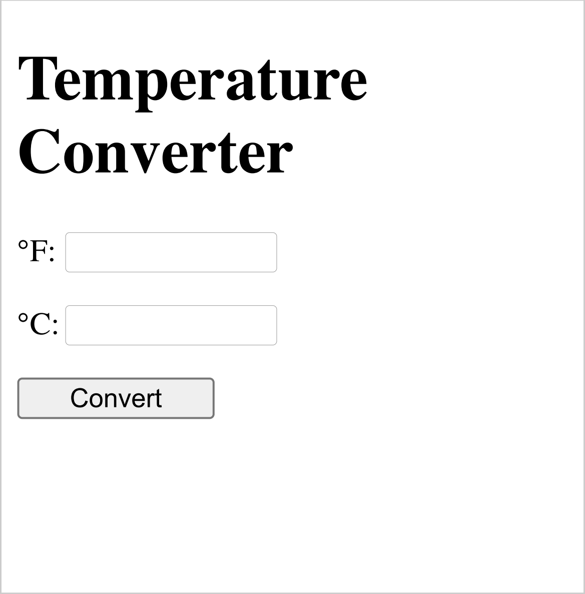
```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/j

<h1>Temperature Converter</h1>
<p>
   <label id="fahrLabel" for="fahr">&deg;F:</label>
   <input type="text" id="fahr">
</p>
<p>
   <label id="celsLabel" for="cels">&deg;C:</label>
   <input type="text" id="cels">
</p>
<p>
   <input type="button" value="Convert" id="convertBtn">
</p>

```

Render webpage   Reset code

Your webpage



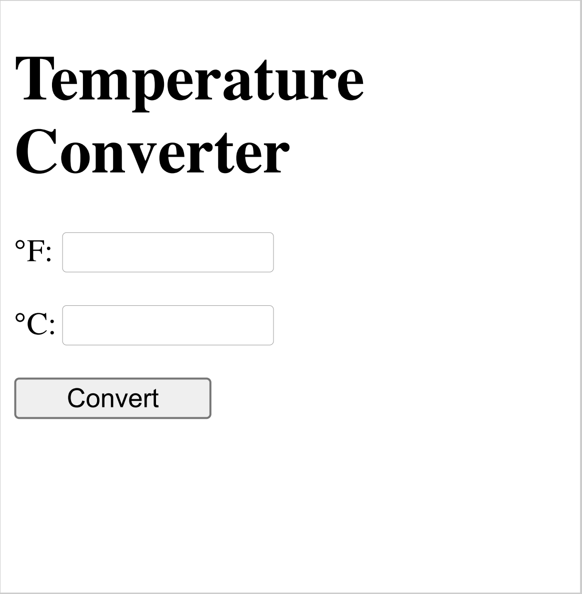
Temperature Converter

°F:

°C:

Convert

Expected webpage



Temperature Converter

°F:

°C:

Convert

▶ View solution

Feedback?

CHALLENGE ACTIVITY | 9.3.1: jQuery events. ✓

530096.4000608.qx3zqy7

Jump to level 1

☑ 1 ☑ 2 ☑ 3

Write and register an event handler that adds the class subtle to the h1 tag on mouseenter.

HTML | JavaScript

```
let $h1Element = $("h1:eq(0)");

/* Your solution goes here */
$h1Element.on('mouseenter', function() {
      // Add the class "subtle" to the specific h1 element
      $(this).addClass("subtle");
    });
```

1 | 2 | 3

Check

Try again

Done. Click any level to practice more. Completion is preserved.

✓

✓ Checking if h1 tag has the "subtle" class before any event

Yours `No`

✓ Checking if h1 has the "subtle" class after event on h1

Yours `Yes`

✓ Checking if h2 has the "subtle" class after event on h2

Yours `No`

✓ Checking if h3 has the "subtle" class after event on h3

Yours `No`

✓ Checking if p has the "subtle" class after event on p

Yours `No`

✓ Checking if a has the "subtle" class after event on a

Yours `No`

Feedback?

Exploring further:

- jQuery Events documentation
- Events - jQuery Learning Center
- $.isNumeric() documentation
- .val() documentation

How was this section?

Provide section feedback