The code below registers two handlers for the click event on a `<button>` element. Note the differences between the two techniques used:

```
<button id="mybutton">Click me</button>
<script>
var b = document.getElementById("mybutton");
b.onclick = function() { alert("Thanks for clicking me!"); };
b.addEventListener("click", function() { alert("Thanks again!"); }, false);
</script>
```

Calling `addEventListener()` with “click” as its first argument does not affect the value of the `onclick` property. In the code above, a button click will generate two `alert()` dialog boxes. More importantly, you can call `addEventListener()` multiple times to register more than one handler function for the same event type on the same object. When an event occurs on an object, all of the handlers registered for that type of event are invoked, in the order in which they were registered. Invoking `addEventListener()` more than once on the same object with the same arguments has no effect—the handler function remains registered only once, and the repeated invocation does not alter the order in which handlers are invoked.

`addEventListener()` is paired with a `removeEventListener()` method that expects the same three arguments but removes an event handler function from an object rather than adding it. It is often useful to temporarily register an event handler and then remove it soon afterward. For example, when you get a mousedown event, you might register temporary capturing event handlers for mousemove and mouseup events so that you can see if the user drags the mouse. You’d then deregister these handlers when the mouseup event arrives. In such a situation, your event handler removal code might look like this:

```
document.removeEventListener("mousemove", handleMouseMove, true);
document.removeEventListener("mouseup", handleMouseUp, true);
```

17.2.4 attachEvent()

Internet Explorer, prior to IE9, does not support `addEventListener()` and `removeEventListener()`. In IE5 and later, it defines similar methods `attachEvent()` and `detachEvent()`.

The `attachEvent()` and `detachEvent()` methods work like `addEventListener()` and `removeEventListener()`, with the following exceptions:

- Since the IE event model does not support event capturing, `attachEvent()` and `detachEvent()` expect only two arguments: the event type and the handler function.
- The first argument to the IE methods is an event handler property name, with the “on” prefix, rather than the unprefixed event type. For example, pass “onclick” to `attachEvent()` where you would pass “click” to `addEventListener()`.
- `attachEvent()` allows the same event handler function to be registered more than once. When an event of the specified type occurs, the registered function will be invoked as many times as it was registered.