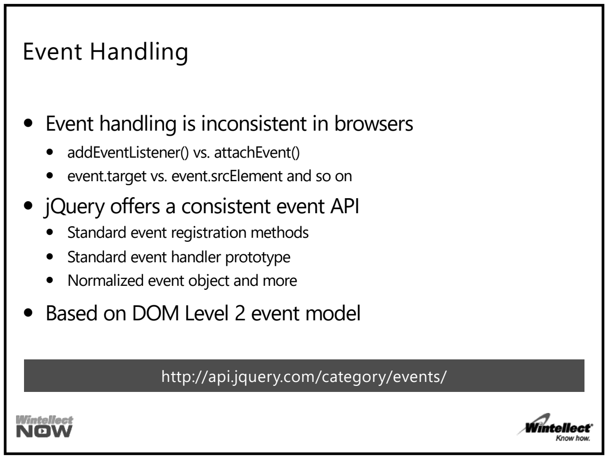
jQuery Events

Jeff Prosise
jeffpro@wintellect.com

Consulting/Training

1

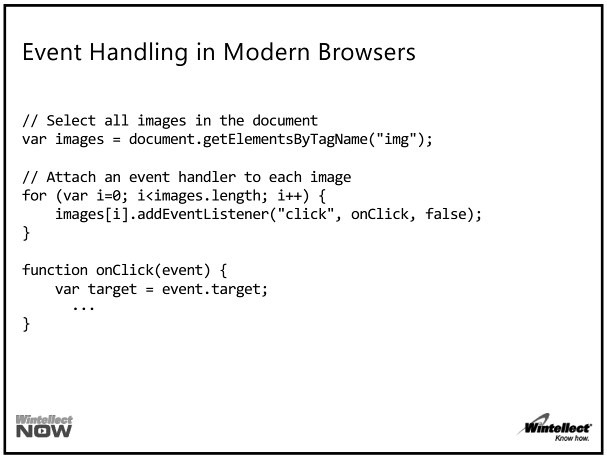


Event Handling

- Event handling is inconsistent in browsers
  - addEventListener() vs. attachEvent()
  - event.target vs. event.srcElement and so on
- jQuery offers a consistent event API
  - Standard event registration methods
  - Standard event handler prototype
  - Normalized event object and more
- Based on DOM Level 2 event model

http://api.jquery.com/category/events/

2



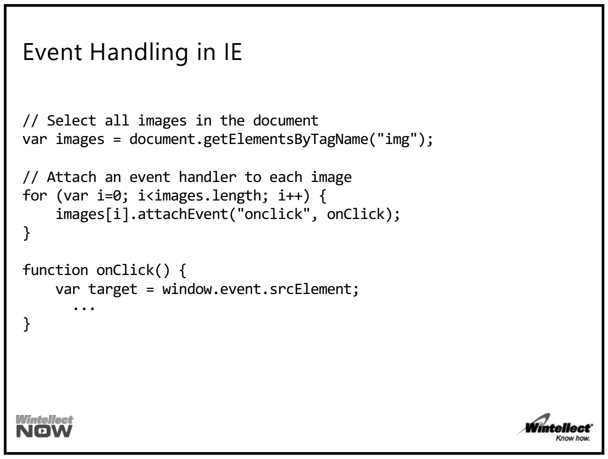
Event Handling in Modern Browsers

```
// Select all images in the document
var images = document.getElementsByTagName("img");

// Attach an event handler to each image
for (var i=0; i<images.length; i++) {
    images[i].addEventListener("click", onClick, false);
}

function onClick(event) {
    var target = event.target;
      ...
}
```

3



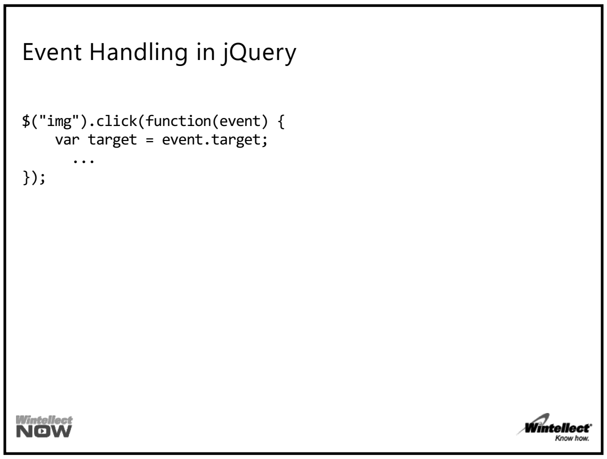
Event Handling in IE

```
// Select all images in the document
var images = document.getElementsByTagName("img");

// Attach an event handler to each image
for (var i=0; i<images.length; i++) {
    images[i].attachEvent("onclick", onClick);
}

function onClick() {
    var target = window.event.srcElement;
      ...
}
```

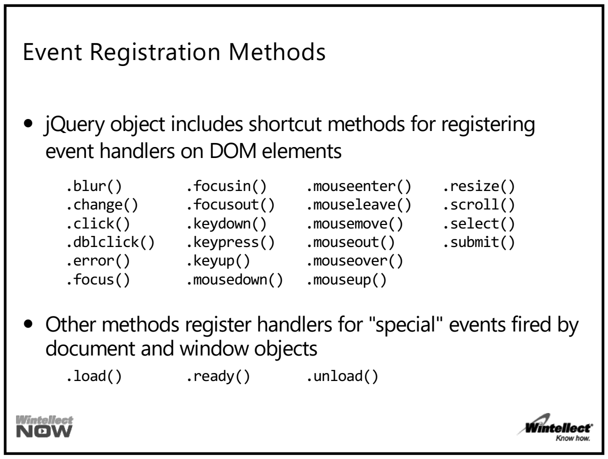
4



Event Handling in jQuery

```
$("img").click(function(event) {
    var target = event.target;
      ...
});
```

5



Event Registration Methods

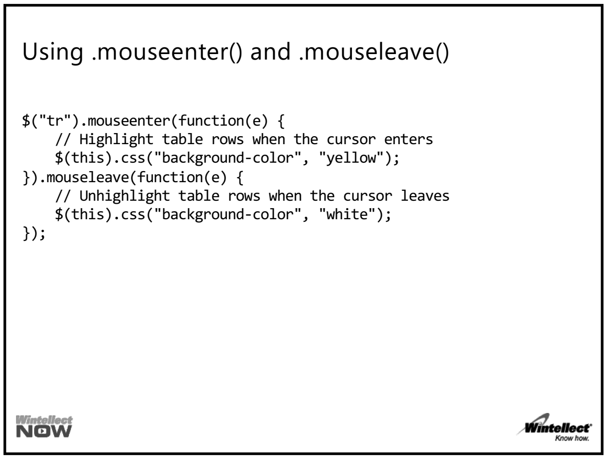
- jQuery object includes shortcut methods for registering event handlers on DOM elements

```
.blur()       .focusin()     .mouseenter()   .resize()
.change()     .focusout()    .mouseleave()   .scroll()
.click()      .keydown()     .mousemove()    .select()
.dblclick()   .keypress()    .mouseout()     .submit()
.error()      .keyup()       .mouseover()
.focus()      .mousedown()   .mouseup()
```

- Other methods register handlers for "special" events fired by document and window objects

```
.load()       .ready()       .unload()
```

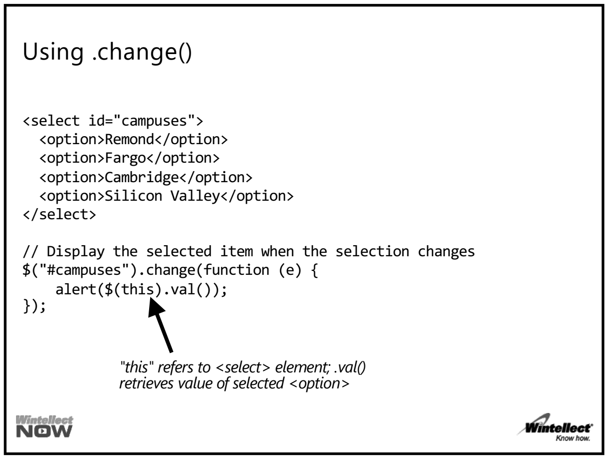
6



Using .mouseenter() and .mouseleave()

```
$("tr").mouseenter(function(e) {
    // Highlight table rows when the cursor enters
    $(this).css("background-color", "yellow");
}).mouseleave(function(e) {
    // Unhighlight table rows when the cursor leaves
    $(this).css("background-color", "white");
});
```

7



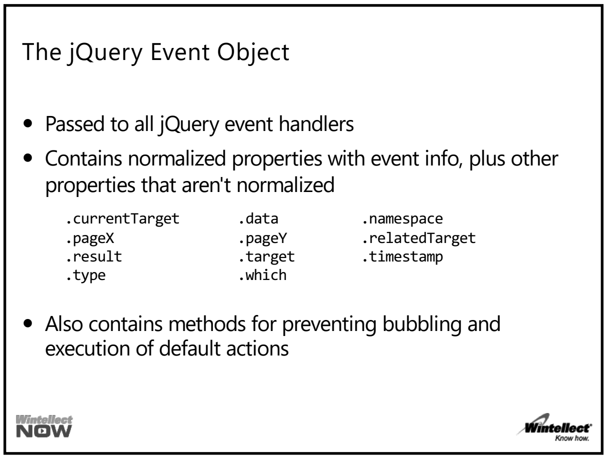
Using .change()

```
<select id="campuses">
  <option>Remond</option>
  <option>Fargo</option>
  <option>Cambridge</option>
  <option>Silicon Valley</option>
</select>

// Display the selected item when the selection changes
$("#campuses").change(function (e) {
    alert($(this).val());
});
```

"this" refers to <select> element; .val() retrieves value of selected <option>

8



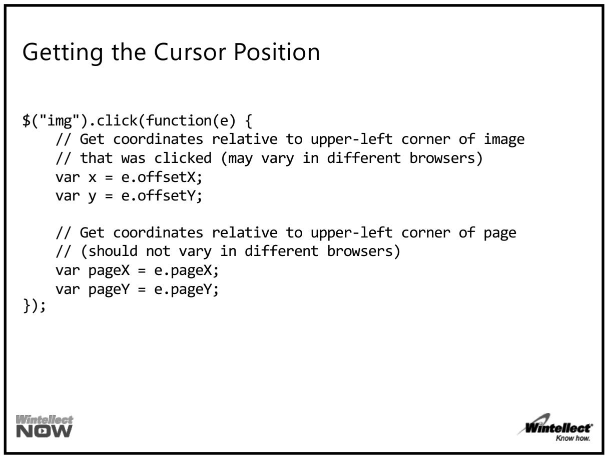
The jQuery Event Object

- Passed to all jQuery event handlers
- Contains normalized properties with event info, plus other properties that aren't normalized

```
.currentTarget      .data         .namespace
.pageX              .pageY        .relatedTarget
.result             .target       .timestamp
.type               .which
```

- Also contains methods for preventing bubbling and execution of default actions

9

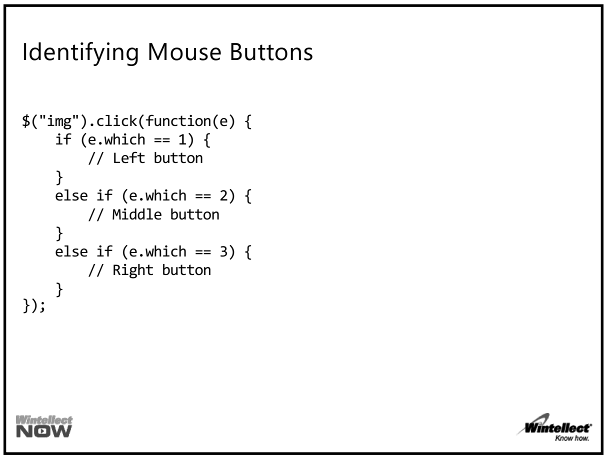


Getting the Cursor Position

```
$("img").click(function(e) {
    // Get coordinates relative to upper-left corner of image
    // that was clicked (may vary in different browsers)
    var x = e.offsetX;
    var y = e.offsetY;

    // Get coordinates relative to upper-left corner of page
    // (should not vary in different browsers)
    var pageX = e.pageX;
    var pageY = e.pageY;
});
```

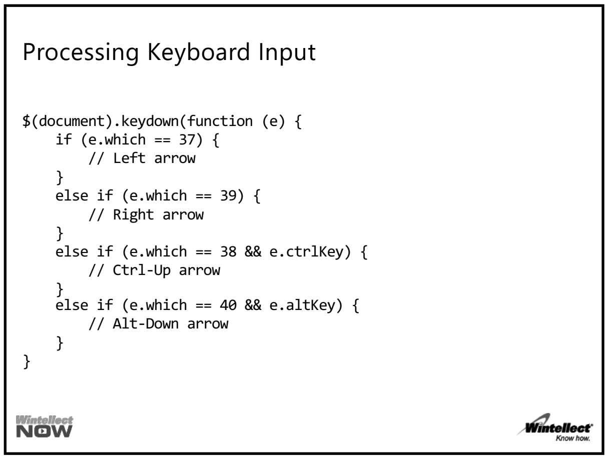
10



Identifying Mouse Buttons

```
$("img").click(function(e) {
    if (e.which == 1) {
        // Left button
    }
    else if (e.which == 2) {
        // Middle button
    }
    else if (e.which == 3) {
        // Right button
    }
});
```

11



Processing Keyboard Input

```
$(document).keydown(function (e) {
    if (e.which == 37) {
        // Left arrow
    }
    else if (e.which == 39) {
        // Right arrow
    }
    else if (e.which == 38 && e.ctrlKey) {
        // Ctrl-Up arrow
    }
    else if (e.which == 40 && e.altKey) {
        // Alt-Down arrow
    }
}
```

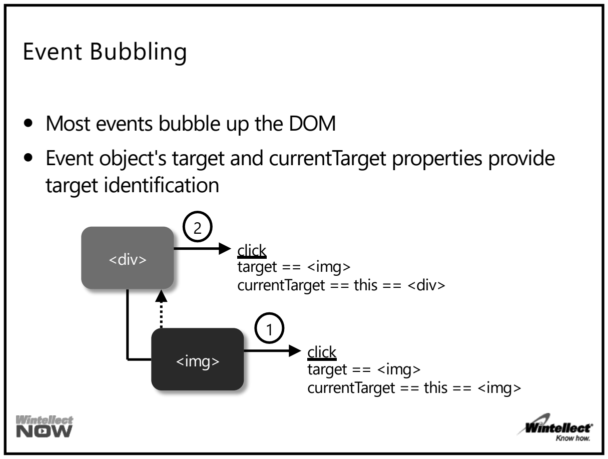
12

DEMO

Event Handling

Consulting/Training

13

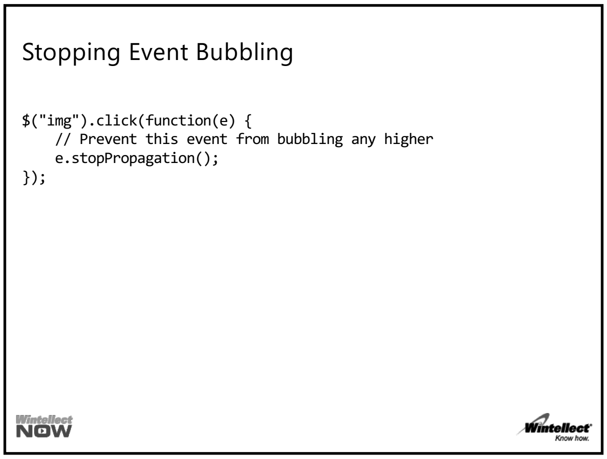


Event Bubbling

- Most events bubble up the DOM
- Event object's target and currentTarget properties provide target identification

<div>
2
click
target == <img>
currentTarget == this == <div>
<img>
1
click
target == <img>
currentTarget == this == <img>

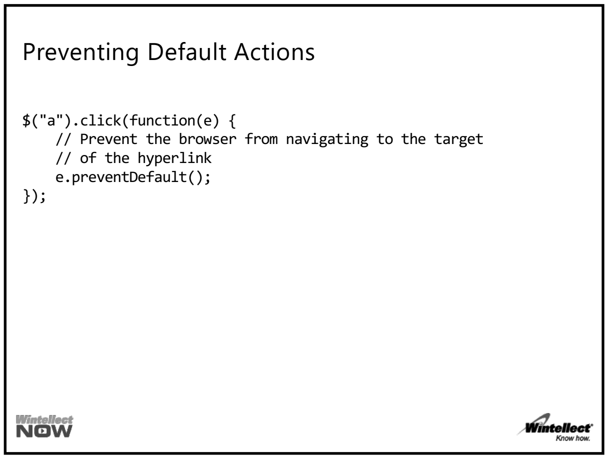
14



Stopping Event Bubbling

```
$("img").click(function(e) {
    // Prevent this event from bubbling any higher
    e.stopPropagation();
});
```

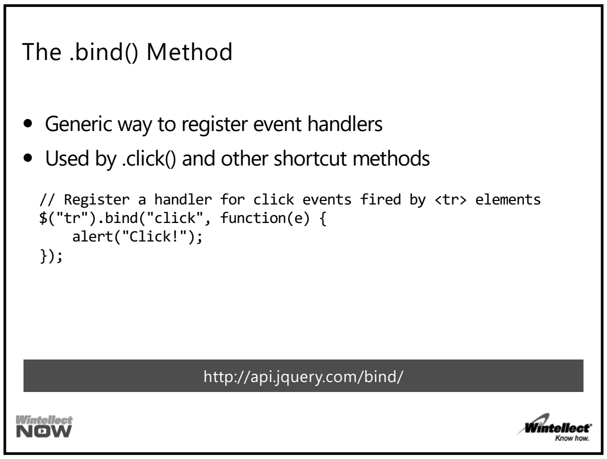
15



Preventing Default Actions

```
$("a").click(function(e) {
    // Prevent the browser from navigating to the target
    // of the hyperlink
    e.preventDefault();
});
```

16



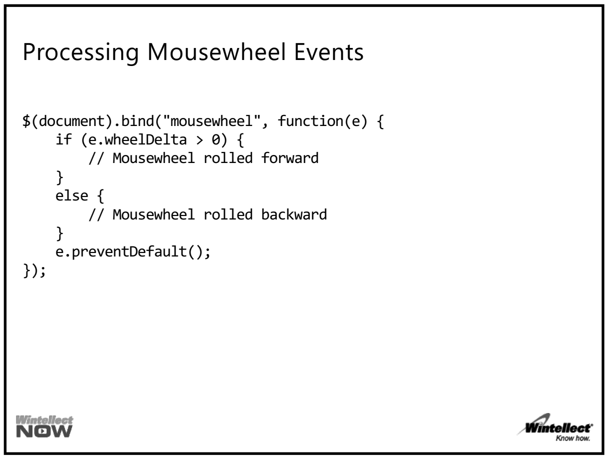
The .bind() Method

- Generic way to register event handlers
- Used by .click() and other shortcut methods

```
// Register a handler for click events fired by <tr> elements
$("tr").bind("click", function(e) {
    alert("Click!");
});
```

http://api.jquery.com/bind/

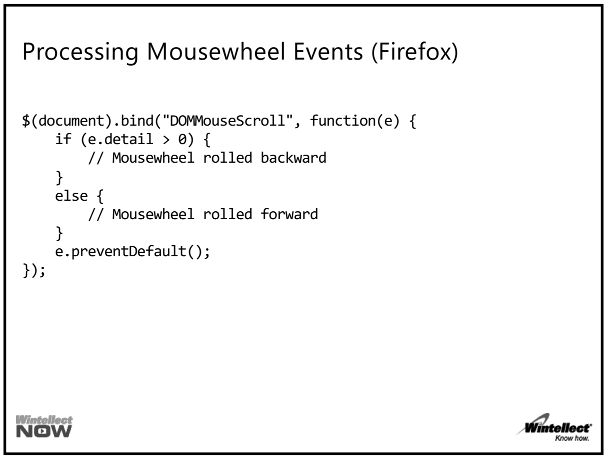
17



Processing Mousewheel Events

```
$(document).bind("mousewheel", function(e) {
    if (e.wheelDelta > 0) {
        // Mousewheel rolled forward
    }
    else {
        // Mousewheel rolled backward
    }
    e.preventDefault();
});
```

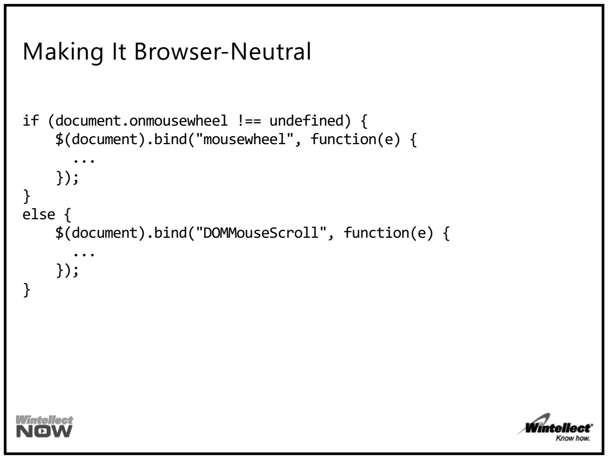
18



Processing Mousewheel Events (Firefox)

```
$(document).bind("DOMMouseScroll", function(e) {
    if (e.detail > 0) {
        // Mousewheel rolled backward
    }
    else {
        // Mousewheel rolled forward
    }
    e.preventDefault();
});
```

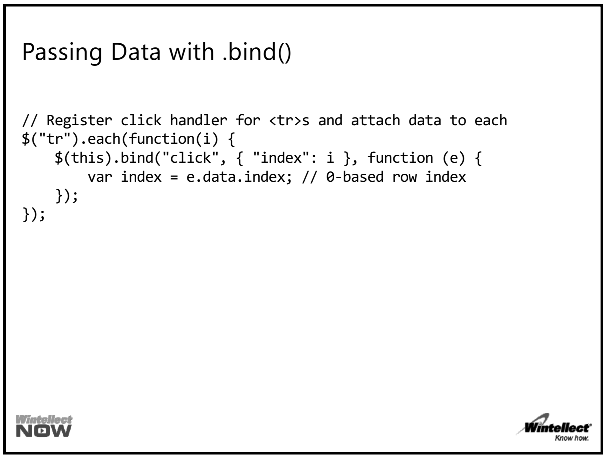
19



Making It Browser-Neutral

```
if (document.onmousewheel !== undefined) {
    $(document).bind("mousewheel", function(e) {
      ...
    });
}
else {
    $(document).bind("DOMMouseScroll", function(e) {
      ...
    });
}
```

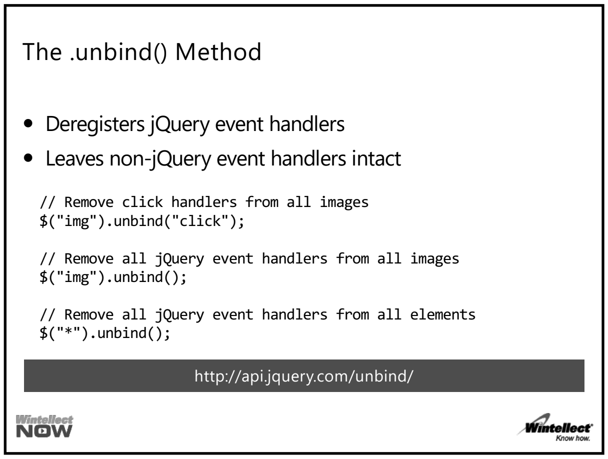
20



Passing Data with .bind()

```
// Register click handler for <tr>s and attach data to each
$("tr").each(function(i) {
    $(this).bind("click", { "index": i }, function (e) {
        var index = e.data.index; // 0-based row index
    });
});
```

21



The .unbind() Method

- Deregisters jQuery event handlers
- Leaves non-jQuery event handlers intact

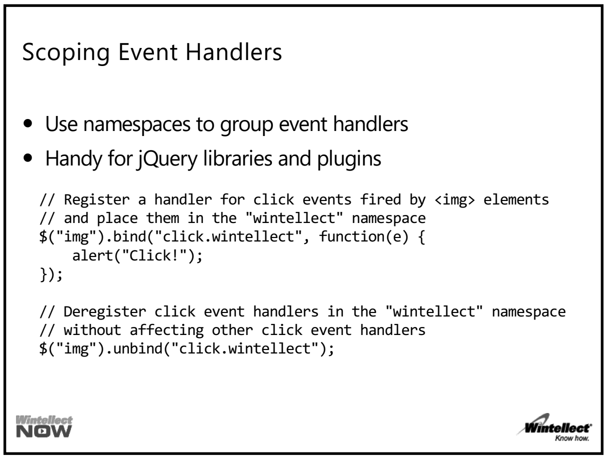
```
// Remove click handlers from all images
$("img").unbind("click");

// Remove all jQuery event handlers from all images
$("img").unbind();

// Remove all jQuery event handlers from all elements
$("*").unbind();
```

http://api.jquery.com/unbind/

22



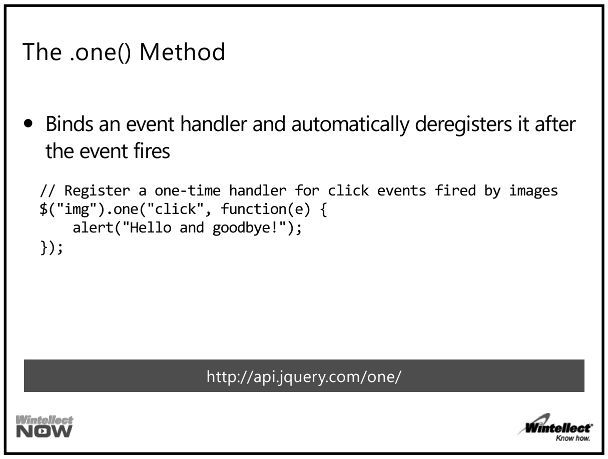
Scoping Event Handlers

- Use namespaces to group event handlers
- Handy for jQuery libraries and plugins

```
// Register a handler for click events fired by <img> elements
// and place them in the "wintellect" namespace
$("img").bind("click.wintellect", function(e) {
    alert("Click!");
});

// Deregister click event handlers in the "wintellect" namespace
// without affecting other click event handlers
$("img").unbind("click.wintellect");
```

23



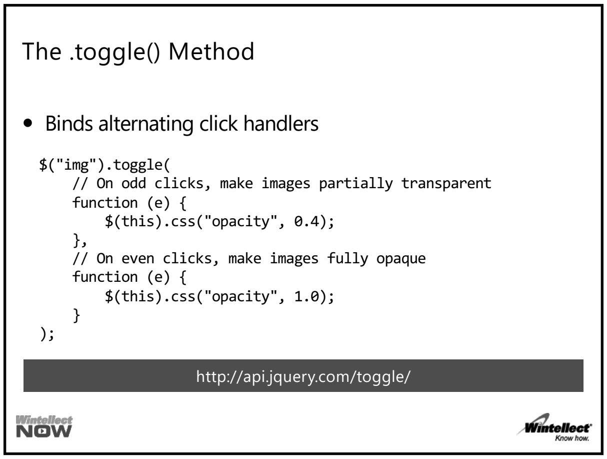
The .one() Method

- Binds an event handler and automatically deregisters it after the event fires

```
// Register a one-time handler for click events fired by images
$("img").one("click", function(e) {
    alert("Hello and goodbye!");
});
```

http://api.jquery.com/one/

24



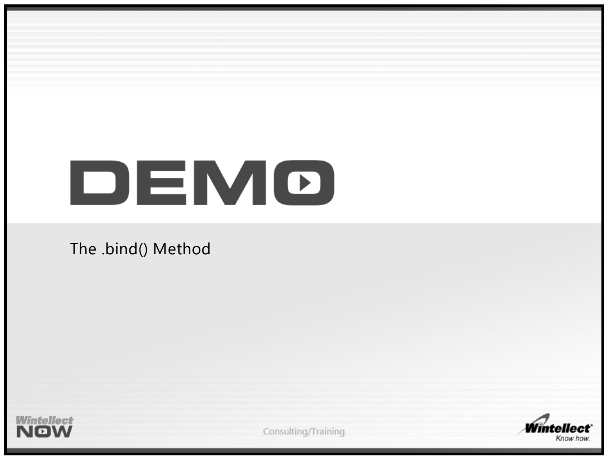
The .toggle() Method

- Binds alternating click handlers

```
$("img").toggle(
    // On odd clicks, make images partially transparent
    function (e) {
        $(this).css("opacity", 0.4);
    },
    // On even clicks, make images fully opaque
    function (e) {
        $(this).css("opacity", 1.0);
    }
);
```

http://api.jquery.com/toggle/

25

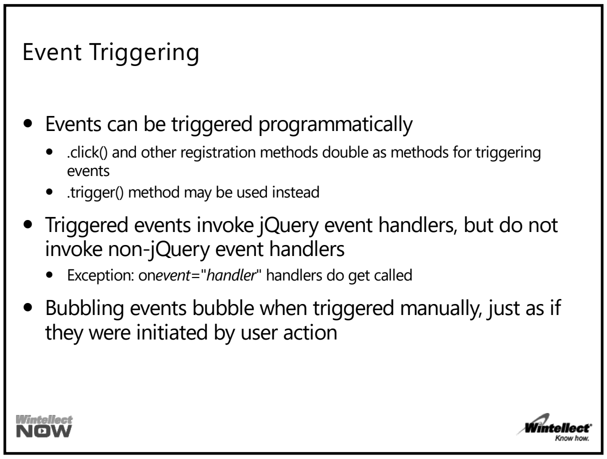


DEMO

The .bind() Method

Consulting/Training

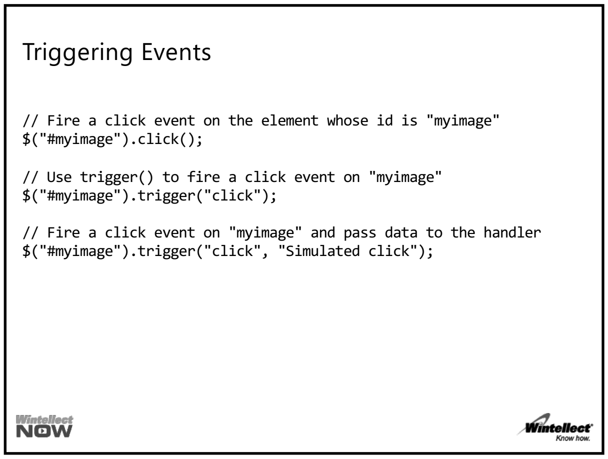
26



Event Triggering

- Events can be triggered programmatically
  - .click() and other registration methods double as methods for triggering events
  - .trigger() method may be used instead
- Triggered events invoke jQuery event handlers, but do not invoke non-jQuery event handlers
  - Exception: on*event*="*handler*" handlers do get called
- Bubbling events bubble when triggered manually, just as if they were initiated by user action

27



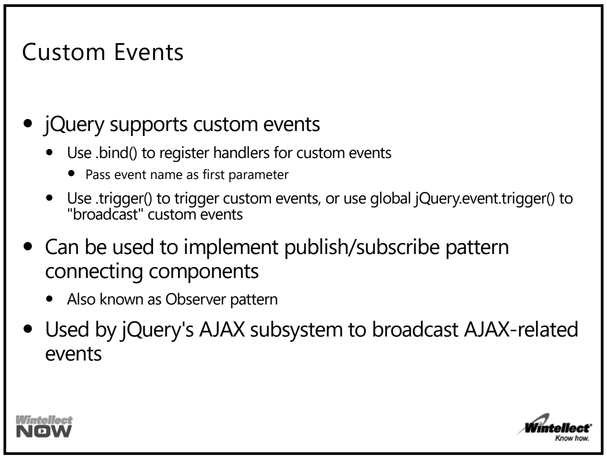
Triggering Events

```
// Fire a click event on the element whose id is "myimage"
$("#myimage").click();

// Use trigger() to fire a click event on "myimage"
$("#myimage").trigger("click");

// Fire a click event on "myimage" and pass data to the handler
$("#myimage").trigger("click", "Simulated click");
```

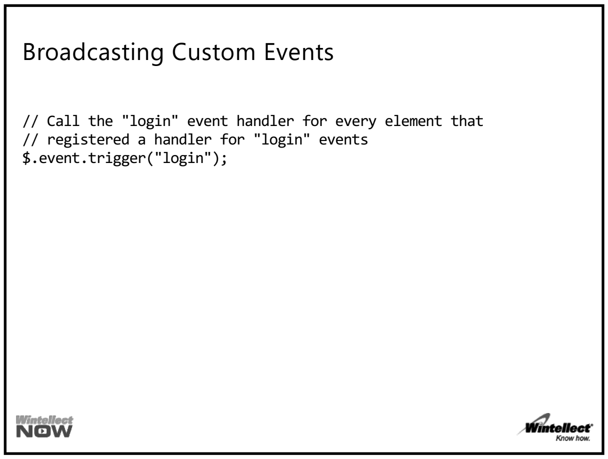
28



Custom Events

- jQuery supports custom events
  - Use .bind() to register handlers for custom events
    - Pass event name as first parameter
  - Use .trigger() to trigger custom events, or use global jQuery.event.trigger() to "broadcast" custom events
- Can be used to implement publish/subscribe pattern connecting components
  - Also known as Observer pattern
- Used by jQuery's AJAX subsystem to broadcast AJAX-related events

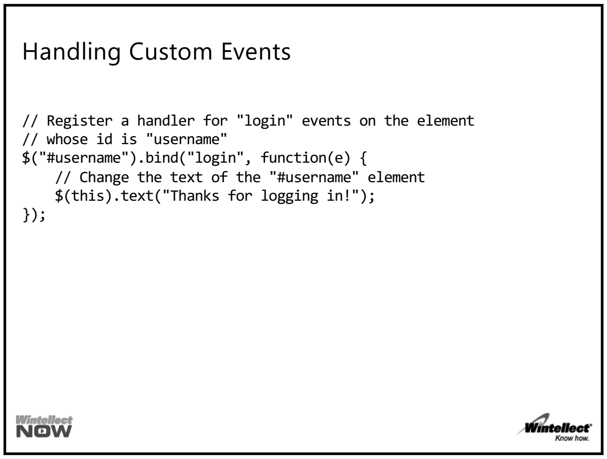
29



Broadcasting Custom Events

```
// Call the "login" event handler for every element that
// registered a handler for "login" events
$.event.trigger("login");
```

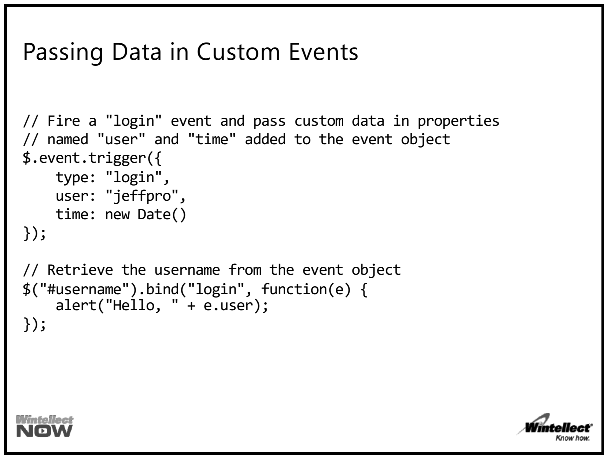
30



Handling Custom Events

```
// Register a handler for "login" events on the element
// whose id is "username"
$("#username").bind("login", function(e) {
    // Change the text of the "#username" element
    $(this).text("Thanks for logging in!");
});
```

31



Passing Data in Custom Events

```
// Fire a "login" event and pass custom data in properties
// named "user" and "time" added to the event object
$.event.trigger({
    type: "login",
    user: "jeffpro",
    time: new Date()
});

// Retrieve the username from the event object
$("#username").bind("login", function(e) {
    alert("Hello, " + e.user);
});
```

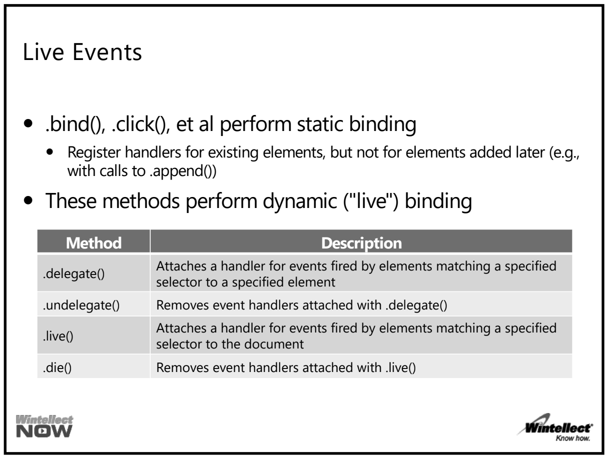
32

DEMO

Custom Events

Consulting/Training

33

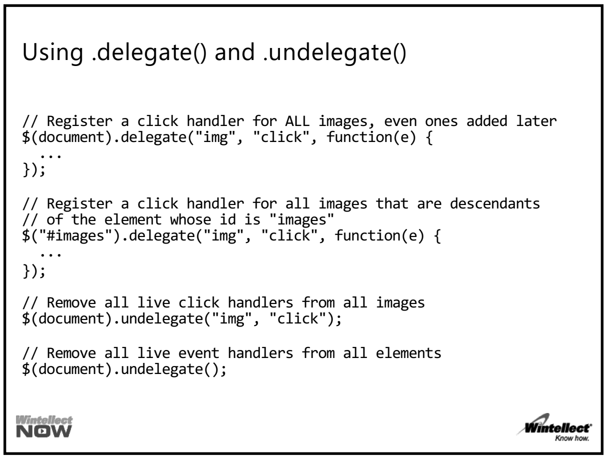


Live Events

- .bind(), .click(), et al perform static binding
  - Register handlers for existing elements, but not for elements added later (e.g., with calls to .append())
- These methods perform dynamic ("live") binding

Method	Description
.delegate()	Attaches a handler for events fired by elements matching a specified selector to a specified element
.undelegate()	Removes event handlers attached with .delegate()
.live()	Attaches a handler for events fired by elements matching a specified selector to the document
.die()	Removes event handlers attached with .live()

34



Using .delegate() and .undelegate()

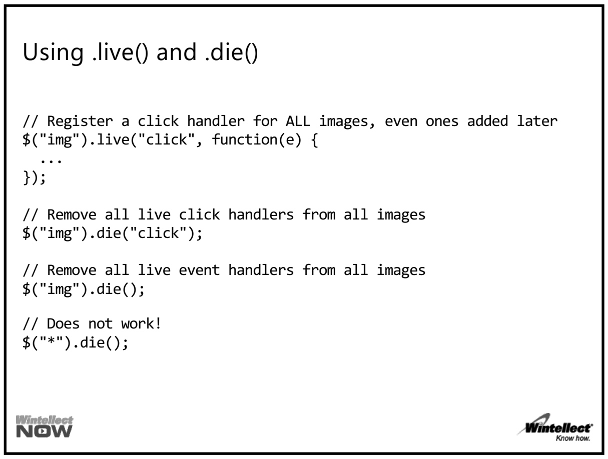
```
// Register a click handler for ALL images, even ones added later
$(document).delegate("img", "click", function(e) {
  ...
});

// Register a click handler for all images that are descendants
// of the element whose id is "images"
$("#images").delegate("img", "click", function(e) {
  ...
});

// Remove all live click handlers from all images
$(document).undelegate("img", "click");

// Remove all live event handlers from all elements
$(document).undelegate();
```

35



Using .live() and .die()

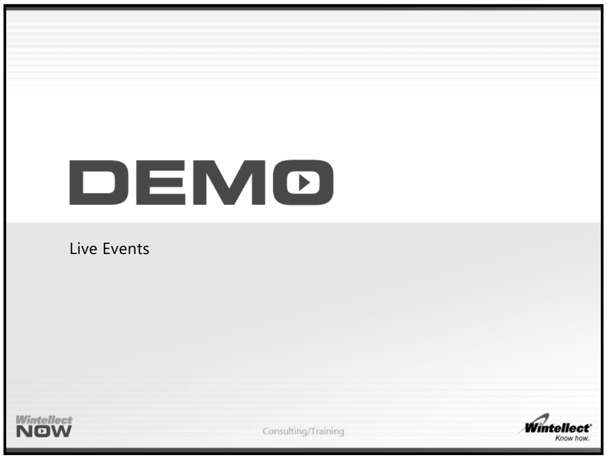
```
// Register a click handler for ALL images, even ones added later
$("img").live("click", function(e) {
  ...
});

// Remove all live click handlers from all images
$("img").die("click");

// Remove all live event handlers from all images
$("img").die();

// Does not work!
$("*").die();
```

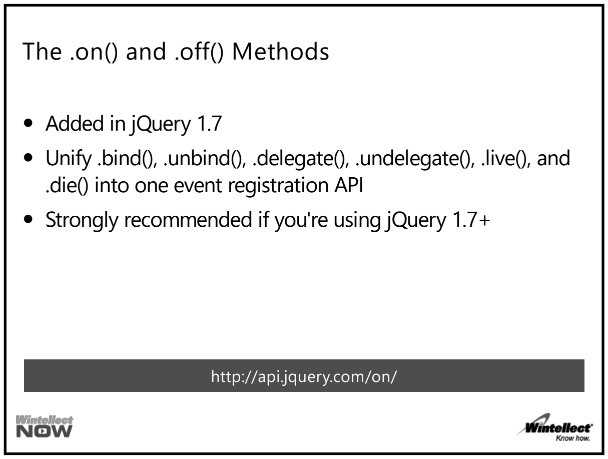
36


DEMO

Live Events

Wintellect NOW
Consulting/Training
Wintellect Know how.

37

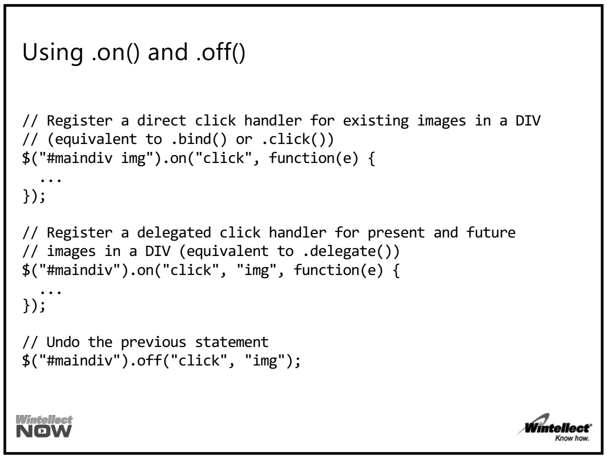

The .on() and .off() Methods

- Added in jQuery 1.7
- Unify .bind(), .unbind(), .delegate(), .undelegate(), .live(), and .die() into one event registration API
- Strongly recommended if you're using jQuery 1.7+

http://api.jquery.com/on/

Wintellect NOW
Wintellect Know how.

38


Using .on() and .off()

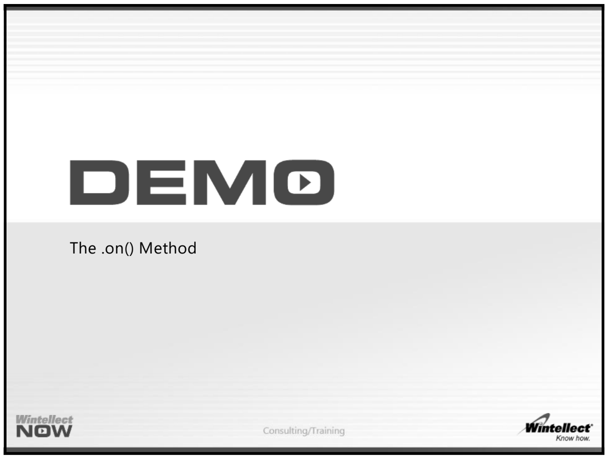
```
// Register a direct click handler for existing images in a DIV
// (equivalent to .bind() or .click())
$("#maindiv img").on("click", function(e) {
  ...
});

// Register a delegated click handler for present and future
// images in a DIV (equivalent to .delegate())
$("#maindiv").on("click", "img", function(e) {
  ...
});

// Undo the previous statement
$("#maindiv").off("click", "img");
```

Wintellect NOW
Wintellect Know how.

39


DEMO

The .on() Method

Wintellect NOW
Consulting/Training
Wintellect Know how.

40