



JavaScript and jQuery Course

Class 13

1

Effects / Effect Methods

Effects are **methods** built into jQuery which create animation type effects.
These methods are **applied to a selector when an event occurs**

- ▶ showing / hiding
- ▶ sliding up / sliding down (to hide/show)
- ▶ fading in / fading out (to hide/show)

▶ 2

Effects

They take three optional parameters.

1. **speed factor** or time duration (milliseconds / value)
- defaults to 400ms or "normal" if not specified
2. **"easing" value** ('swing' - default or 'linear')
3. **"callback function"** - runs when the effect is completed.

▶ 3

Effects Must Respond to an Event

General Syntax

responds to an event
 ↓
 \$("xyz").click(function(){

3 optional parameters
 ↓
\$(selector).effect(speed, easing, callback) ;

}); // end click

▶ 4

Speed Parameter

Speed = duration 1,000 milliseconds = 1 second

- ▶ **“fast”** = 200 milliseconds
- ▶ **“normal”** = 400 milliseconds - **default**
- ▶ **“slow”** = 600 milliseconds
- ▶ **number** = milliseconds

Note: strings are enclosed in quotations, not numeric values!

▶ 5

Example – showing / hiding

.hide(“fast”) - hides the selected element

.show(“slow”) - shows the selected element

.toggle(2000) - toggles between hide() and show()

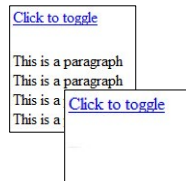
Manipulates the CSS display property

▶ 6

Syntax

```
$("#link1").click(function(){
    $("#para1").toggle(); // 400ms duration
});
```

if no parameters specified...
speed = 400ms / “normal”
easing = “swing”



▶ 7

Example – Sliding

.slideUp(“fast”) - slides up to hide

.slideDown(“slow”) - slides down to show

.slideToggle(2000) - toggles between the two

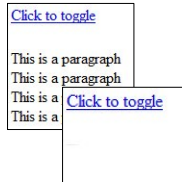
Manipulates the CSS height property (0 to actual)

▶ 8

Syntax

```
$("#link1").click(function(){
    $("#para1").slideToggle(); // 400ms duration
});
```

if no parameters specified...
speed = 400ms / "normal"
easing = "swing"



▶ 9

Example – Fading

.fadeIn("fast") - faded into view
.fadeOut("slow") - fades out of view
.fadeToggle(2000) - toggles between the two

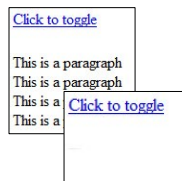
Manipulates the CSS opacity property (0 to 1)

▶ 10

Syntax

```
$("#link1").click(function(){
    $("#para1").fadeToggle(); // 400ms duration
});
```

if no parameters specified...
speed = 400ms / "normal"
easing = "swing"



▶ 11

Fading – one way

fadeTo()

fades the selection to a specific **opacity** - a number between 0.00 and 1.00

\$(selector).fadeTo(speed, opacity, easing, callback);

required parameter

▶ 12

Syntax

```
$("#link1").click(function(){
    $("#para1").fadeTo(1000, 0.4);
});
```

Click to fade from normal to .04

This is a paragraph
This is a paragraph
This is a paragraph
This is a paragraph

speed and duration
are specified
easing = "swing"

▶ 13

Effects

delay()

- sets a timer to delay the execution of the next item in the queue

```
$("#link2").click(function(){
    $("#para").delay(2000).fadeIn();
});
```

chaining

▶ 14

Animations

animate() method

- performs a custom animation using a set of **CSS properties**
- **does NOT need to be connected to an event** 🤖
- can run when the page loads or within a function call

▶ 15

Animations

animate()

- uses **CSS styles**
- the CSS property value is changed gradually, creating an animated effect
- **only numeric values can be animated** 🤖
- string values and shorthand properties cannot be animated
- **Use JS syntax, not CSS syntax, for the property names** 🤖

▶ 16

Syntax

Properties use JS syntax not CSS syntax

- ▶ `backgroundPositionX`, `backgroundPositionY`
- ▶ `borderWidth`, `borderBottomWidth`, `borderLeftWidth`, `borderRightWidth`, `borderTopWidth`
- ▶ `borderSpacing`, `margin`, `marginBottom`, `marginLeft`, `marginRight`, `marginTop`
- ▶ `padding`, `paddingBottom`, `paddingLeft`, `paddingRight`, `paddingTop`
- ▶ `height`, `width`, `outlineWidth`
- ▶ `maxHeight`, `maxWidth`, `minHeight`, `minWidth`, `fontSize`
- ▶ `bottom`, `left`, `right`, `top`
- ▶ `letterSpacing`, `wordSpacing`, `lineHeight`, `textIndent`

▶ 17

Syntax

`animate()` method

`$(selector).animate({ styles }, speed, easing, callback);`

styles are required
if no parameters specified...
 speed = 400ms / "normal"
 easing = "swing"
 no callback function

▶ 18

Animate

property name is not in quotations

`$(this).animate({height: "500px"}, "slow");`

// can use unit value if in quotations

`$(this).animate({height: 500}, "slow");`

// no quotations for numbers

▶ 19

Callback Function

`$('h1').animate({fontSize: "1.5em"}, 3000,`

`function(){`

`$('h1').addClass('red') }`

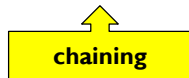
`);`

▶ 20

Delay Method

- delays an **animation or effect**
- requires a time duration

```
$('#hl').delay(3000).animate({fontSize: "1.5em"}, 3000);
```



► 21

Delay Method

Chaining methods using white space

```
$('#hl').delay(3000)
    .animate({fontSize: "1.5em"}, 3000)
    .delay(3000)
    .animate({fontSize: ".08em"}, 3000)
```

► 22

Animate

It is best to already have the **CSS styles set**

Before you animate them

So you have a baseline to return to

► 23

Animate

You must have opacity set to animate it

You must have a relative position set to move it

► 24

Stop Method

- stop()
- stops the animation **for the selected element**
- commonly used with the **“hover”** or mouse / click events
- clears the “queue” of any animations in the event that the user caused it to happen by fast mouse movements

▶ 25

Stop Method

```

$("xyz").hover( function(){
    $(this).stop(true).animate({ width: 100}, "fast");
},
function(){
    $(this).stop(true).animate({ width: 200}, "fast");
}
); // end hover

```

▶ 26

setInterval Method

- setInterval()
- method of the window object
- runs a function at a specified time interval in milliseconds

▶ 27

Syntax

```

setInterval(functionName, 5000);

```

← **named function**

```

setInterval(function() {

    // code here

}, 5000);

```

← **anonymous function**

▶ 28

Animate – using setInterval

```
function getSmall(){  
    $('h1').animate({fontSize: "22pt"}, 1000);  
}  
  
setInterval(function() {  
    $('h1').animate({fontSize: "32pt"}, 1500, getSmall);  
}, 3000);
```

callback
function

