

jQuery Effects

Buddhika Gayashani
Department of ICT
Faculty of Technology
University of Ruhuna



Objectives

- Identify and use different jQuery effects

Hide() and Show()

Syntax

- `$(selector).hide(speed, callback);`
- `$(selector).show(speed, callback);`
- The optional speed parameter specifies the speed of the hiding/showing,
 - values: "slow", "fast", or milliseconds
- The optional callback parameter is a function to be executed after the hide() or show() method completes

Example

```
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
    <script>
      $(document).ready(function() {
        $("#b1").click(function() {
          $(".para").hide(1000);
        });
        $("#b2").click(function() {
          $(".para").show();
        });
      });
    </script>
  </head>
  <body >
    <button id="b1">Hide</button>
    <button id="b2">Show</button>
    <p class="para1">This is my first JQuery code</P>
    <p class="para">This is my second JQuery code</P>
  </body>
</html>
```

jQuery toggle()

- You can toggle between the hide() and show() methods with the toggle() method
- Shown elements are hidden and hidden elements are shown

Syntax:

- `$(selector).toggle(speed, callback);`

Example

```
$("#button").click(function() {  
    $("#p").toggle();  
});
```

Example

```
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/
jquery/3.6.0/jquery.min.js"></script>
    <script>
      $(document).ready(function() {
        $("#b1").click(function() {
          $(".para").toggle();
        });
      });
    </script>
  </head>
  <body >
    <button id="b1">Hide/Show</button>
    <p class="para1">This is my first JQuery code</P>
    <p class="para">This is my second JQuery code</P>
  </body>
</html>
```

jQuery Effects -Fading

- With jQuery you can fade an element in and out of visibility
- jQuery fade methods:
 - `fadeIn()`
 - `fadeOut()`
 - `fadeToggle()`
 - `fadeTo()`

jQuery fadeIn() Method

- Used to fade in a hidden element

Syntax

- `$(selector).fadeIn(speed,callback);`

Note

- The optional callback parameter is a function to be executed after the fading completes
- Example

```
$("#button").click(function() {  
    $("#div1").fadeIn();  
    $("#div2").fadeIn("slow");  
    $("#div3").fadeIn(3000);  
});
```


Jquery fadeOut() Method

- Used to fade out a visible element

Syntax

- `$(selector).fadeOut(speed,callback);`

Example

```
$("#button").click(function() {  
    $("#div1").fadeOut();  
    $("#div2").fadeOut("slow");  
    $("#div3").fadeOut(3000);  
});
```

jQuery fadeToggle() Method

- Toggles between the fadeIn() and fadeOut() methods
- If the elements are faded out, fadeToggle() will fade them in
- If the elements are faded in, fadeToggle() will fade them out

Syntax:

- `$(selector).fadeToggle(speed,callback);`

Example

```
$("#button").click(function() {  
    $("#div1").fadeToggle();  
    $("#div2").fadeToggle("slow");  
    $("#div3").fadeToggle(3000);  
});
```

Jquery Effects - Sliding

- With jQuery you can create a sliding effect on elements
- jQuery has the following slide methods
 - `slideDown()`
 - `slideUp()`
 - `slideToggle()`

jQuery slideDown() Method

- Used to slide down an element

Syntax

- `$(selector).slideDown(speed,callback);`
- Example

```
$("#flip").click(function() {  
    $("#panel").slideDown();  
});
```

jQuery slideUp() Method

- Used to slide up an element

Syntax

- `$(selector).slideUp(speed,callback);`

Example

```
$("#flip").click(function() {  
    $("#panel").slideUp();  
});
```

jQuery slideToggle() Method

- Toggles between the slideDown() and slideUp() methods
- If the elements have been slid down, slideToggle() will slide them up
- If the elements have been slid up, slideToggle() will slide them down
- `$(selector).slideToggle(speed,callback);`

Example

```
$("#flip").click(function() {  
    $("#panel").slideToggle();  
});
```

jQuery Effects - Animation

- jQuery animate() method is used to create custom animations
- Four kinds of animate() method,
 - Manipulate multiple properties
 - Using relative values
 - Using pre-defined values
 - Using queue functionality

jQuery animate() Method

Syntax:

- `$(selector).animate({params}, speed, callback);`
- The required params parameter defines the CSS properties to be animated

Example

```
$("#button").click(function() {  
    $("#div").animate( {left: '250px'} );  
});
```


Manipulate Multiple Properties

- Multiple properties can be animated at the same time

- **Example**

```
$("#button").click(function() {  
    $("#div").animate({  
        left: '250px',  
        opacity: '0.5',  
        height: '150px',  
        width: '150px'  
    });  
});
```

Using Relative Values

- It is also possible to define relative values (the value is then relative to the element's current value)
- This is done by putting += or -= in front of the value

Example

```
$("#button").click(function() {  
    $("#div").animate({  
        left: '250px',  
        height: '+=150px',  
        width: '+=150px'  
    });  
});
```

Using Pre-defined Values

- Specify a property's animation value as "show", "hide", or "toggle"

Example

```
$("#button").click(function() {  
    $("#div").animate({  
        height: 'toggle'  
    });  
});
```

Queue Functionality

- By default, jQuery comes with queue functionality for animations
- Queue functionality : If you write multiple `animate()` calls after each other, jQuery creates an "internal" queue with these method calls.
- Then it runs the animate calls ONE by ONE
- This functionality can be used , if you want to perform different animations after each other

Queue Functionality....

Example

```
$("#button").click(function() {  
    var div = $("#div");  
    div.animate({height: '300px', opacity: '0.4'}, "slow");  
    div.animate({width: '300px', opacity: '0.8'}, "slow");  
    div.animate({height: '100px', opacity: '0.4'}, "slow");  
    div.animate({width: '100px', opacity: '0.8'}, "slow");  
});
```

jQuery Callback Functions

- A callback function is executed after the current effect is 100% finished
- With effects, the next line of the script code can be run even though the effect is not finished
- To prevent this, callback functions can be used
- A callback function is executed after the current effect is finished

Syntax

\$(selector).hide(speed,callback);

Jquery Callback Functions

- The example below has a callback parameter that is a function that will be executed after the hide effect is completed:
- Example with Callback

```
$("#button").click(function() {  
    $("#p").hide("slow", function() {  
        alert("The paragraph is now hidden");  
    });  
});
```

Questions...

