

Dynamic HTML and jQuery, jQuery events and effects

Lesson 4

Plan for this lesson

The DOM revisited

Differences between regular javascript and jQuery

... and drop

Code Along

The jQuery UI library

Adding more visual effects to jQuery

Resizable

Code Along

Code along

Selectable

Code Along

Accordion

Code Along

Progressbar

Autocomplete

Code Along

Code Along

Datepicker

Code Along

Drag...

Code Along

Let's build a game using jQuery

The DOM revisited

The DOM [Changing CSS](#)

Last lesson we looked at using jQuery to add and change CSS properties of elements. You can also do it in pure javascript, but it is a bit more cumbersome which is why I chose to introduce the area with jQuery, but you may come across projects where using a framework is not desired.

Using regular javascript to change css:

```
document.getElementById("p2").style.color = "blue";
```

Using jQuery to change css:

```
$("#p2").css("color", "blue");
```

The jQuery UI library

The jQueryUI library [Adding to more visual effects to jQuery](#)

A set of user interface interactions, effects, widgets, and themes built on top of the jQuery JavaScript Library.

jQuery UI contains many widgets that maintain state and therefore have a slightly different usage pattern than typical jQuery plugins. All of jQuery UI's widgets use the same patterns, so if you learn how to use one, then you'll know how to use all of them.

The jQueryUI library [Adding to more visual effects to jQuery](#)

jQuery UI consists covers the following areas:

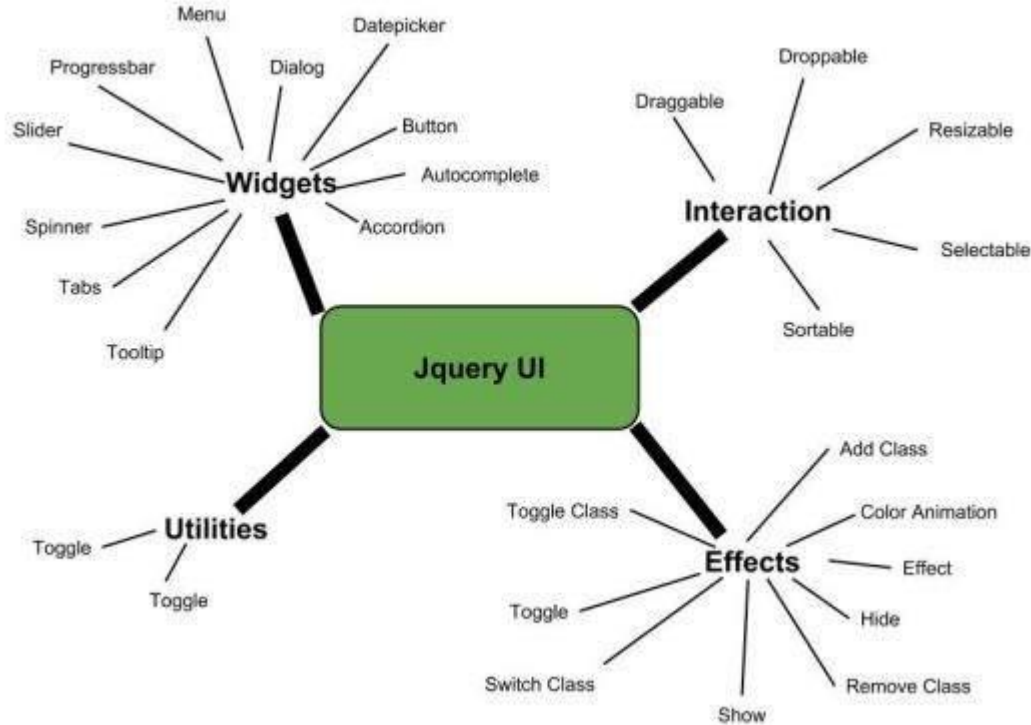
Interactions - Interactive plugins like drag, drop, resize and more which give the user the ability to interact with DOM elements.

Widgets- Via widgets, which are jQuery plugins, you can create user interface elements like accordion,datepicker etc.

Effects - Built on internal jQuery effects. They contain a full suite of custom animations and transitions for DOM elements.

Utilities - These are a set of modular tools the jQueryUI library uses internally.

The jQueryUI library Adding to more visual effects to jQuery



The jQueryUI library Adding to more visual effects to jQuery

Just like jQuery you can download it from the official jQuery project site (<http://jqueryui.com>) or use the official free CDN.

You usually “build” your own custom package of jQueryUI elements you want to use.

1. jQuery itself
2. The jQueryUI additional library
3. Often a theme, so you have a defined style readymade for your elements.

You make them via a visual interface here: <http://jqueryui.com/themeroller/>

The package you download comes complete with the jQuery and UI-files you need too.

Code along

jQueryUI part one

Code from the code along [Roll your own preferred jQueryUI theme package](#)

“Roll” a jQueryUI package using the ThemeRoller (<http://jqueryui.com/themeroller/>).

It does not have to look like the one in these examples as it is “just” visual styles, play around and find your own preferred style.

Often it can save time to use such visual tools instead of coding all the looks from the bottom and hoping it will look nice by using trial and error.

We do not need all files you might end up downloading for now, only the CSS files. In the following example I’m going with the minified CSS files for faster loading (as you should in a on-the-web scenario), but obviously those are not ideal for changing stuff in should you want that at some point.

Code from the code along [HTML and javascript/jQuery](#)

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <title>Example jQuery UI Page</title>
6      <link href="css/jquery-ui.structure.min.css" rel="stylesheet">
7      <link href="css/jquery-ui.theme.min.css" rel="stylesheet">
8      <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
9      <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.min.js"></script>
10     <script type="text/javascript">
11         $(function () {
12             $('#dialogMsg').dialog();
13         });
14     </script>
15 </head>
16 <body>
17     <form id="form1" runat="server">
18         <div id="dialogMsg" title="First JQueryUI Example">
19             Hello this is my first JQueryUI example.
20         </div>
21     </form>
22 </body>
23 </html>
```

The jQueryUI library [Accordion](#)

Accordion Widget in jQueryUI is a jQuery based expandable and collapsible content holder that is broken into sections and probably looks like tabs. jQueryUI provides accordion() method to achieve this.

See the full list of option at

https://www.tutorialspoint.com/jqueryui/jqueryui_accordion.htm

Code along

jQueryUI part two

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="utf-2">
5     <title>Example jQuery UI Page</title>
6     <link href="css/jquery-ui.structure.min.css" rel="stylesheet">
7     <link href="css/jquery-ui.theme.min.css" rel="stylesheet">
8     <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
9     <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.min.js"></script>
10    <script>
11        $(function() {
12            $( "#accordion-1" ).accordion();
13        });
14    </script>
15    <style>
16        #accordion-1{font-size: 14px;}
17    </style>
18 </head>
19 <body>
20 <div id="accordion-1">
21     <h3>Tab 1</h3>
22 <div>
23 <p>
24     Lorem ipsum dolor sit amet, consectetur adipisicing elit,
25     sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
```

```
26         Ut enim ad minim veniam, quis nostrud exercitation ullamco
27         laboris nisi ut aliquip ex ea commodo consequat.
28     </p>
29 </div>
30 <h3>Tab 2</h3>
31 <div>
32     <p>
33         Lorem ipsum dolor sit amet, consectetur adipisicing elit,
34         sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
35         Ut enim ad minim veniam, quis nostrud exercitation ullamco
36         laboris nisi ut aliquip ex ea commodo consequat.
37     </p>
38 </div>
39 <h3>Tab 3</h3>
40 <div>
41     <p>
42         Lorem ipsum dolor sit amet, consectetur adipisicing elit,
43         sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
44         Ut enim ad minim veniam, quis nostrud exercitation ullamco
45         laboris nisi ut aliquip ex ea commodo consequat.
46     </p>
47 </div>
48 </div>
49 </body>
50 </html>
```


The jQueryUI library [Autocomplete](#)

Autocompletion is a mechanism frequently used in modern websites to provide the user with a list of suggestions for the beginning of the word he has typed in a text box. The user can then select an item from the list, which will be displayed in the input field. This feature prevents the user from having to enter an entire word or a set of words.

jQueryUI provides an autocomplete widget—a control that acts a lot like a `<select>` dropdown, but filters the choices to present only those that match what the user is typing into a control. jQueryUI provides the **`autocomplete()`** method to create a list of suggestions below the input field and adds new CSS classes to the elements concerned to give them the appropriate style.

The jQueryUI library `Autocomplete`

Any field that can receive input can be converted into an Autocomplete, namely, `<input>` elements, `<textarea>` elements, and elements with the `contenteditable` attribute.

See the full list of option at https://www.tutorialspoint.com/jqueryui/jqueryui_autocomplete.htm

Code along

jQueryUI part three

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-2">
5   <title>Example jQuery UI Page</title>
6   <link href="css/jquery-ui.structure.min.css" rel="stylesheet">
7   <link href="css/jquery-ui.theme.min.css" rel="stylesheet">
8   <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
9   <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.min.js"></script>
10  <script>
11    $(function() {
12      $( "#autocomplete-4" ).autocomplete({
13        source: [
14          { label: "Iceland", value: "ICE" },
15          { label: "Austria", value: "AUS" }
16        ]
17      });
18    });
19  </script>
20 </head>
21 <body>
22   <!-- HTML -->
23   <div class="ui-widget">
24     <p>Type I OR A</p>
25     <input id="autocomplete-4">
26   </div>
27 </body>
28 </html>
```

The jQueryUI library [Datepicker](#)

Datepickers in jQueryUI allow users to enter dates easily and visually. You can customize the date format and language, restrict the selectable date ranges and add in buttons and other navigation options easily.

jQueryUI provides a **datepicker()** method that creates a datepicker and changes the appearance of HTML elements on a page by adding new CSS classes.

Transforms the `<input>`, `<div>`, and `` elements in the wrapped set into a datepicker control.

By default, for `<input>` elements, the datepicker calendar opens in a small overlay when the associated text field gains focus. For an inline calendar, simply attach the datepicker to a `<div>`, or `` element.

See the full list of option at

https://www.tutorialspoint.com/jqueryui/jqueryui_datepicker.htm

Code along

jQueryUI part four

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-2">
5   <title>Example jQuery UI Page</title>
6   <link href="css/jquery-ui.structure.min.css" rel="stylesheet">
7   <link href="css/jquery-ui.theme.min.css" rel="stylesheet">
8   <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
9   <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.min.js"></script>
10  <script>
11    $(function() {
12      $( "#datepicker-13" ).datepicker();
13      $( "#datepicker-13" ).datepicker("show");
14    });
15  </script>
16 </head>
17 <body>
18   <!-- HTML -->
19   <p>Enter Date: <input type="text" id="datepicker-13"></p>
20 </body>
21 </html>
```

The jQueryUI library [Drag...](#)

Being able to move elements on the screen gives the user a more tactile experience of your web-page.

jQueryUI provides **draggable()** method to make any DOM element draggable. Once the element is draggable, you can move that element by clicking on it with the mouse and dragging it anywhere within the viewport.

You can even add options using the syntax
\$(selector, context).draggable(options);

See the full list of options at https://www.tutorialspoint.com/jqueryui/jqueryui_draggable.htm

Code along

jQueryUI part five

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>Example jQuery UI Page</title>
6   <link href="css/jquery-ui.structure.min.css" rel="stylesheet">
7   <link href="css/jquery-ui.theme.min.css" rel="stylesheet">
8   <style>
9     #draggable { width: 150px; height: 150px; padding: 0.5em; background:#eee;}
10  </style>
11  <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
12  <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.min.js"></script>
13  <script>
14    $(function() {
15      $( "#draggable" ).draggable();
16    });
17  </script>
18 </head>
19 <body>
20   <form id="form1" runat="server">
21     <div id="draggable" class="ui-widget-content">
22       <p>I got to move it move it!</p>
23     </div>
24   </form>
25 </body>
26 </html>
```

The jQueryUI library ...and drop

jQueryUI provides `droppable()` method to make any DOM element droppable at a specified target (a target for draggable elements).

You have 8 options, but the following example demonstrates the usage of three options (a) `addClass` (b) `disabled` and (c) `tolerance` in the drop function of jQueryUI.

See the full list of options at https://www.tutorialspoint.com/jqueryui/jqueryui_droppable.htm

Code along

jQueryUI part six

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>Example jQuery UI Page</title>
6   <link href="css/jquery-ui.structure.min.css" rel="stylesheet">
7   <link href="css/jquery-ui.theme.min.css" rel="stylesheet">
8   <style>
9     #draggable { width: 150px; height: 150px; padding: 0.5em; background:#eee;}
10  </style>
11  <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
12  <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.min.js"></script>
13  <style>
14    .wrap{
15      display: table-row-group;
16    }
17    #japanpeople,#germanypeople, #javatutorial,#phptutorial {
18      width: 120px; height: 70px; padding: 0.5em; float: left;
19      margin: 0px 5px 10px 0;
20    }
21
22    #japan,#germany,#java,#php {
23      width: 140px; height: 100px;padding: 0.5em; float: left;
24      margin: 10px;
25    }
26  </style>
27  <script>
28    $(function() {
29      $( "#japanpeople" ).draggable();
30      $( "#germanypeople" ).draggable();
31
32      $( "#japan" ).droppable({
33        accept: "#japanpeople",
34        drop: function( event, ui ) {
35          $( this )
```

```
36         .addClass( "ui-state-highlight" )
37         .find( "p" )
38         .html( "Dropped!" );
39     }
40 });
41 $( "#germany" ).droppable({
42     accept: "#germanypeople",
43     drop: function( event, ui ) {
44         $( this )
45         .addClass( "ui-state-highlight" )
46         .find( "p" )
47         .html( "Dropped!" );
48     }
49 });
50
51 $( "#javatutorial" ).draggable({scope : "java"});
52 $( "#phptutorial" ).draggable({scope : "php"});
53 $( "#java" ).droppable({
54     scope: "java",
55     drop: function( event, ui ) {
56         $( this )
57         .addClass( "ui-state-highlight" )
58         .find( "p" )
59         .html( "Dropped!" );
60     }
61 });
62 $( "#php" ).droppable({
63     scope: "php",
64     drop: function( event, ui ) {
65         $( this )
66         .addClass( "ui-state-highlight" )
67         .find( "p" )
68         .html( "Dropped!" );
69     }
70 });
```

```
71 - });
72 -     </script>
73 - </head>
74 - <body>
75 -     <div class="wrap" >
76 -         <div id="japanpeople" class="ui-widget-content">
77 -             <p>People to be dropped to Japan</p>
78 -         </div>
79 -         <div id="germanypeople" class="ui-widget-content">
80 -             <p>People to be dropped to Germany</p>
81 -         </div>
82 -         <div id="japan" class="ui-widget-header">
83 -             <p>Japan</p>
84 -         </div>
85 -         <div id="germany" class="ui-widget-header">
86 -             <p>Germany</p>
87 -         </div>
88 -     </div>
89 -     <hr/>
90 -     <div class="wrap" >
91 -         <div id="javatutorial" class="ui-widget-content">
92 -             <p>People who want to learn Java</p>
93 -         </div>
94 -         <div id="phptutorial" class="ui-widget-content">
95 -             <p>People who want to learn PHP</p>
96 -         </div>
97 -         <div id="java" class="ui-widget-header">
98 -             <p>Java</p>
99 -         </div>
100 -        <div id="php" class="ui-widget-header">
101 -            <p>PHP</p>
102 -        </div>
103 -    </div>
104 - </body>
105 - </html>
```

The jQueryUI library `Resizable`

jQueryUI provides `resizable()` method to resize any DOM element. This method simplifies the resizing of element which otherwise takes time and lot of coding in HTML. The `resizable ()` method displays an icon in the bottom right of the item to resize.

Again there is many options, 19 to be exact, but we will only use two in the example.

See the full list of options at

https://www.tutorialspoint.com/jqueryui/jqueryui_resizable.htm

Code along

jQueryUI part seven

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>Example jQuery UI Page</title>
6   <link href="css/jquery-ui.structure.min.css" rel="stylesheet">
7   <link href="css/jquery-ui.theme.min.css" rel="stylesheet">
8   <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
9   <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.min.js"></script>
10  <style>
11    .ui-widget-header {
12      background:#b9cd6d;
13      border: 1px solid #b9cd6d;
14      color: #FFFFFF;
15      font-weight: bold;
16    }
17    .ui-widget-content {
18      background: #cedc98;
19      border: 1px solid #DDDDDD;
20      color: #333333;
21    }
22    #resizable-2,#resizable-3 {
23      width: 150px;
24      height: 150px;
25      padding: 0.5em;
26      text-align: center; margin: 0; }
27  </style>
28  <script>
29    $(function() {
30      $( "#resizable-1" ).resizable({
31        animate: true
32      });
33      $( "#resizable-2" ).resizable({
34        ghost: true
35      });
36    });
```

```
36 ┌     });
37 ┌     </script>
38 ┌ </head>
39 ▼ <body>
40 ▼     <div id="resizable-1" class="ui-widget-content">
41 ▼         <h3 class="ui-widget-header">
42             Pull my edges and Check the animation!!
43 ┌         </h3>
44 ┌     </div><br>
45 ▼     <div id="resizable-2" class="ui-widget-content">
46         <h3 class="ui-widget-header">I'm ghost!!</h3>
47 ┌     </div>
48 ┌ </body>
49 ┌ </html>
```

The jQueryUI library [Selectable](#)

jQueryUI provides `selectable()` method to select DOM element individually or in a group. With this method elements can be selected by dragging a box (sometimes called a lasso) with the mouse over the elements. Also, elements can be selected by clicking or dragging while holding the Ctrl/Meta key, allowing for multiple (non-contiguous) selections.

Again there is options, six to be exact, but we will only use *selected*.

See the full list of options at

https://www.tutorialspoint.com/jqueryui/jqueryui_resizable.htm

Code along

jQueryUI part eight

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-2">
5   <title>Example jQuery UI Page</title>
6   <link href="css/jquery-ui.structure.min.css" rel="stylesheet">
7   <link href="css/jquery-ui.theme.min.css" rel="stylesheet">
8   <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
9   <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.min.js"></script>
10  <style>
11    #selectable-1 .ui-selecting { background: #707070 ; }
12    #selectable-1 .ui-selected { background: #EEEEEE; color: #000000; }
13    #selectable-1 { list-style-type: none; margin: 0;
14      padding: 0; width: 20%; }
15    #selectable-1 li { margin: 3px; padding: 0.4em;
16      font-size: 16px; height: 18px; }
17    .ui-widget-content {
18      background: #cedc98;
19      border: 1px solid #DDDDDD;
20      color: #333333;
21    }
22    .resultarea {
23      background: #cedc98;
24      border-top: 1px solid #000000;
25      border-bottom: 1px solid #000000;
26      color: #333333;
27      font-size:14px;
28    }
29  </style>
```

```
30 <script>
31   $(function() {
32     $( "#selectable-1" ).selectable({
33       selected: function() {
34         var result = $( "#result" ).empty();
35         $( ".ui-selected", this ).each(function() {
36           var index = $( "#selectable-1 li" ).index( this );
37           result.append( " #" + ( index + 1 ) );
38         });
39       }
40     });
41   });
42 </script>
43 </head>
44 <body>
45   <h3>Events</h3>
46   <ol id="selectable-1">
47     <li class="ui-widget-content">Product 1</li>
48     <li class="ui-widget-content">Product 2</li>
49     <li class="ui-widget-content">Product 3</li>
50     <li class="ui-widget-content">Product 4</li>
51     <li class="ui-widget-content">Product 5</li>
52     <li class="ui-widget-content">Product 6</li>
53     <li class="ui-widget-content">Product 7</li>
54   </ol>
55   <span class="resultarea">Selected Product</span>>
56   <span id=result class="resultarea"></span>
57 </body>
58 </html>
```

The jQueryUI library [Progressbar](#)

Progress bars indicate the completion percentage of an operation or process. We can categorize progress bar as **determinate progress bar** and **indeterminate progress bar**.

A **determinate progress bar** should only be used in situations where the system can accurately update the current status. A determinate progress bar should never fill from left to right, then loop back to empty for a single process.

If the actual status cannot be calculated, an **indeterminate progress bar** should be used to provide user feedback.

The jQueryUI library `Progressbar`

jQueryUI provides an easy-to-use progress bar widget that we can use to let users know that our application is hard at work performing the requested operation.

jQueryUI provides `progressbar()` method to create progress bars.

See the full list of options at https://www.tutorialspoint.com/jqueryui/jqueryui_progressbar.htm

Code along

jQueryUI part nine

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-2">
5   <title>Example jQuery UI Page</title>
6   <link href="css/jquery-ui.structure.min.css" rel="stylesheet">
7   <link href="css/jquery-ui.theme.min.css" rel="stylesheet">
8   <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
9   <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.min.js"></script>
10  <style>
11    .ui-widget-header {
12      background: #cedc98;
13      border: 1px solid #DDDDDD;
14      color: #333333;
15      font-weight: bold;
16    }
17    .progress-label {
18      position: absolute;
19      left: 50%;
20      top: 13px;
21      font-weight: bold;
22      text-shadow: 1px 1px 0 #fff;
23    }
24  </style>
25  <script>
26  $(function() {
27    var progressbar = $( "#progressbar-5" );
28    progressLabel = $( ".progress-label" );
29    $( "#progressbar-5" ).progressbar({
```

```
30     value: false,
31     change: function() {
32         progressLabel.text(
33             progressbar.progressbar( "value" ) + "%" );
34     },
35     complete: function() {
36         progressLabel.text( "Loading Completed!" );
37     }
38 });
39 function progress() {
40     var val = progressbar.progressbar( "value" ) || 0;
41     progressbar.progressbar( "value", val + 1 );
42     if ( val < 99 ) {
43         setTimeout( progress, 100 );
44     }
45 }
46     setTimeout( progress, 3000 );
47 });
48 </script>
49 </head>
50 <body>
51     <div id="progressbar-5">
52         <div class="progress-label">
53             Loading...
54         </div>
55     </div>
56 </body>
57 </html>
```

Code along

jQueryUI part ten

Let's build a game using jQuery

Code from the code along [HTML](#)

```
1 <html>
2 <head>
3   <meta charset="utf-8">
4   <title>Memory Game</title>
5   <link href="css/style.css" rel="stylesheet">
6   <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
7   <script src="js/game.js"></script>
8 </head>
9 <body>
10 <div id="picbox">
11   <span id="boxbuttons">
12     <span class="button">
13       <span id="counter">0</span>
14       Clicks
15     </span>
16     <span class="button">
17       <a onclick="ResetGame();">Reset</a>
18     </span>
19   </span>
20 </div id="boxcard"></div>
21 </div>
22 </body>
23 </html>
```

```
1 * {
2     margin: 0;
3     padding: 0;
4 }
5 body {
6     font: 18px Verdana;
7     color: #FFF;
8     background: #CCC;
9 }
10 #picbox {
11     margin: 0px auto;
12     width: 640px;
13 }
14 #boxcard {
15     z-index: 1;
16     margin: 10px 0 0;
17 }
18 #boxcard div{
19     float: left;
20     width: 100px;
21     height: 100px;
22     margin: 5px;
23     padding: 5px;
24     border: 4px solid #EE872A;
25     cursor: pointer;
26     border-radius: 10px;
```

```
27     box-shadow: 0 1px 5px rgba(0,0,0,.5);
28     background: #B1B1B1;
29     z-index: 2;
30 }
31 #boxcard div img {
32     display: none;
33     border-radius: 10px;
34     z-index: 3;
35 }
36 #boxbuttons {
37     text-align: center;
38     margin: 20px;
39     display: block;
40 }
41 #boxbuttons .button {
42     text-transform: uppercase;
43     background: #EE872A;
44     padding: 5px 10px;
45     margin: 5px;
46     border-radius: 10px;
47     cursor: pointer;
48 }
49 #boxbuttons .button:hover {
50     background: #999;
51 }
```


Code from the code along JS

```
1 var BoxOpened = "";  
2 var ImgOpened = "";  
3 var Counter = 0;  
4 var ImgFound = 0;  
5  
6 var Source = "#boxcard";  
7  
8 ▼ var ImgSource = [  
9     "http://img5.uploadhouse.com/fileuploads/17699/176992640c06707c66a5c0b08a2549c69745dc2c.png",  
10    "http://img6.uploadhouse.com/fileuploads/17699/17699263b01721074bf094aa3bc695aa19c8d573.png",  
11    "http://img6.uploadhouse.com/fileuploads/17699/17699262833250fa3063b708c41042005fda437d.png",  
12    "http://img9.uploadhouse.com/fileuploads/17699/176992615db99bb0fd652a2e6041388b2839a634.png",  
13    "http://img4.uploadhouse.com/fileuploads/17699/176992601ca0f28ba4a8f7b41f99ee026d7aaed8.png",  
14    "http://img3.uploadhouse.com/fileuploads/17699/17699259cb2d70c6882adc285ab8d519658b5dd7.png",  
15    "http://img2.uploadhouse.com/fileuploads/17699/1769925824ea93cbb77ba9e95c1a4cec7f89b80c.png",  
16    "http://img7.uploadhouse.com/fileuploads/17699/1769925708af4fb3c954b1d856da1f4d4dcd548a.png",  
17    "http://img9.uploadhouse.com/fileuploads/17699/176992568b759acd78f7cbe98b6e4a7baa90e717.png",  
18    "http://img9.uploadhouse.com/fileuploads/17699/176992554c2ca340cc2ea8c0606ecd320824756e.png"  
19 ];  
20  
21 ▼ function RandomFunction(MaxValue, MinValue) {  
22     return Math.round(Math.random() * (MaxValue - MinValue) + MinValue);  
23 }
```

Code from the code along JS

```
24
25 ▼ function ShuffleImages() {
26     var ImgAll = $(Source).children();
27     var ImgThis = $(Source + " div:first-child");
28     var ImgArr = new Array();
29
30 ▼     for (var i = 0; i < ImgAll.length; i++) {
31         ImgArr[i] = $("#" + ImgThis.attr("id") + " img").attr("src");
32         ImgThis = ImgThis.next();
33     }
34
35     ImgThis = $(Source + " div:first-child");
36
37 ▼     for (var z = 0; z < ImgAll.length; z++) {
38         var RandomNumber = RandomFunction(0, ImgArr.length - 1);
39
40         $("#" + ImgThis.attr("id") + " img").attr("src", ImgArr[RandomNumber]);
41         ImgArr.splice(RandomNumber, 1);
42         ImgThis = ImgThis.next();
43     }
44 }
45
```

```
46 function ResetGame() {
47     ShuffleImages();
48     $(Source + " div img").hide();
49     $(Source + " div").css("visibility", "visible");
50     Counter = 0;
51     $("#success").remove();
52     $("#counter").html("" + Counter);
53     BoxOpened = "";
54     ImgOpened = "";
55     ImgFound = 0;
56     return false;
57 }
58
59 function OpenCard() {
60     var id = $(this).attr("id");
61
62     if ($("#" + id + " img").is(":hidden")) {
63         $(Source + " div").unbind("click", OpenCard);
64
65         $("#" + id + " img").slideDown('fast');
66
67         if (ImgOpened == "") {
68             BoxOpened = id;
69             ImgOpened = $("#" + id + " img").attr("src");
70             setTimeout(function() {
71                 $(Source + " div").bind("click", OpenCard)
72             }, 300);
```

Code from the code along JS

```
73 ▼ } else {
74     CurrentOpened = $("#" + id + " img").attr("src");
75 ▼     if (ImgOpened != CurrentOpened) {
76 ▼         setTimeout(function() {
77             $("#" + id + " img").slideUp('fast');
78             $("#" + BoxOpened + " img").slideUp('fast');
79             BoxOpened = "";
80             ImgOpened = "";
81     }, 400);
82     } else {
83         $("#" + id + " img").parent().css("visibility", "hidden");
84         $("#" + BoxOpened + " img").parent().css("visibility", "hidden");
85         ImgFound++;
86         BoxOpened = "";
87         ImgOpened = "";
88     }
89 ▼     setTimeout(function() {
90         $(Source + " div").bind("click", OpenCard)
91     }, 400);
92     }
```


Code from the code along JS

```
93     Counter++;
94     $("#counter").html("" + Counter);
95
96     if (ImgFound == ImgSource.length) {
97         $("#counter").prepend('<span id="success">You Found All Pictues With </span>');
98     }
99 }
100 }
101
102 $(function() {
103
104     for (var y = 1; y < 3 ; y++) {
105         $.each(ImgSource, function(i, val) {
106             $(Source).append("<div id=card" + y + i + ">");
107         });
108     }
109     $(Source + " div").click(OpenCard);
110     ShuffleImages();
111 });
```

Sources and further reading

- http://www.w3schools.com/js/js_htmlDOM_css.asp
- <https://jqueryui.com>
- <https://learn.jquery.com/jquery-ui/>
- <http://www.tutorialspoint.com/jqueryui/>
- <https://codepen.io/mel/pen/Brads>