JQuery

Write less, do more js library

What Is JQuery

- JQuery is a javascript library that supports following features
- DOM manipulation
- Event handling
- AJAX Support
- Animations
- Lightweight: The jQuery is very lightweight library - about 85KB in size
- Cross Browser Support
- Latest Technology: The jQuery supports CSS3 selectors and basic XPath syntax.

Basics

- JQuery is a javascript library and hence to use the library we need to include a JS file in our pages
- And most jquery code should execute when the page is done loading.
    - Window onload waits for all images to load, document ready fires as soon as the DOM is ready

```
<script src="scripts/jquery-2.1.3.js"></script>
<script>
    window.onload = function() {
        alert( "Window.onload" );
    };

    $( document ).ready(function() {
        alert("Document ready");
    });
</script>
```

JQuery is Non-intrusive

```
<script>

  $( document ).ready(function() {
      $( "a" ).click(clickFunction);
  });

  function clickFunction(){
    alert("Link clicked");
  }

</script>
<a href="#" onclick="clickFunction();">Link 1</a>

<a href="#">Link 2</a>
```

Selectors

- JQuery uses selectors to identify and select elements
  - Selection strings are compatible with CSS 1-3
  - JQuery has added its own selection strings
- $(“selector string”) - this statement leads to a collection or a single element depending on the selection operation

Some Common Selectors

- $('*'):
  - This selector selects all elements in the document.
- $("p > *"):
  - This selector selects all elements that are children of a paragraph element.
- $("#specialID"):
  - This selector function gets the element with id="specialID".
- $(".specialClass"):
  - This selector gets all the elements that have the class of specialClass.
- $("li:not(.myclass)"):
  - Selects all elements matched by <li> that do not have class="myclass".

More Selectors

- $("a#specialID.specialClass"):
  - This selector matches links with an id of specialID and a class of specialClass.
- $("p a.specialClass"):
  - This selector matches links with a class of specialClass declared within <p> elements.
- $("ul li:first"):
  - This selector gets only the first <li> element of the <ul>.
- $("#container p"):
  - Selects all elements matched by <p> that are descendants of an element that has an id of container.
- $("li > ul"):
  - Selects all elements matched by <ul> that are children of an element matched by <li>

More Selectors

- $("strong + em"):
  - Selects all elements matched by <em> that immediately follow a sibling element matched by <strong>.
- $("p ~ ul"):
  - Selects all elements matched by <ul> that follow a sibling element matched by <p>.
- $("code, em, strong"):
  - Selects all elements matched by <code> or <em> or <strong>.
- $("p strong, .myclass"):
  - Selects all elements matched by <strong> that are descendants of an element matched by <p> as well as all elements that have a class of myclass.
- $(":empty"):
  - Selects all elements that have no children.

More Selectors

- $("p:empty"):
  - Selects all elements matched by <p> that have no children.
- $("div[p]"):
  - Selects all elements matched by <div> that contain an element matched by <p>.
- $("p[.myclass]"):
  - Selects all elements matched by <p> that contain an element with a class of myclass.
- $("a[@rel]"):
  - Selects all elements matched by <a> that have a rel attribute.
- $("input[@name=myname]"):
  - Selects all elements matched by <input> that have a name value exactly equal to myname.

More Selectors

- $("input[@name^=myname]"):
  - Selects all elements matched by <input> that have a name value beginning with myname.
- $("a[@rel$=self]"):
  - Selects all elements matched by <a> that have rel attribute value ending with self
- $("a[@href*=domain.com]"):
  - Selects all elements matched by <a> that have an href value containing domain.com.
- $(":button")
  - Selects all <button> elements and <input> elements of type="button"
- $("tr:even")
  - Selects all even <tr> elements

Lets Try It

- In the page provided, retrieve the text “The quick brown fox jumps over the lazy dog” and print it in an alert (ex1)

DOM Manipulation

- After selecting elements we can manipulate their content.
    - Change attributes, nested values
- Retrive Data
    - text() - Sets or returns the text content of selected elements
    - html() - Sets or returns the content of selected elements (including HTML markup)
    - val() - Sets or returns the value of form fields

Using Callback Approach To Deal with Collections

- Callback approach is a little more elegant than running a loop through

- Looping

```
var deletedElements = $(".deleted");
for(var i=0; i<deletedElements.length; i++){
    if($(deletedElements[i]).text()=='Active'){
        $(deletedElements[i]).text('Deleted');
    }
}
```

- Using callback

```
$(".deleted").text(function(i,origText){
    if(origText=='Active'){
        return "Deleted";
    }
});
```

each()

- $.each() is a generic iterator function for looping over object, arrays, and array-like objects. Plain objects are iterated via their named properties while arrays and array-like objects are iterated via their indices.

```
var arr = [ 1, 2, 3, 4, 5 ];
$.each( arr, function( index, value ){
    console.log(value);
});

var obj = {
        foo: 1,
        bar: 2
}
$.each( obj, function( key, value ) {
    console.log(value);
});

$( "li" ).each( function( index, element ){
    console.log( $( this ).text() );
});
```

Other Manipulations

- css() - get or set a certain css style value

```
$(“.row").css("background-color","blue");
```

- attr()- get or set a certain attribute
- addClass() - Adds one or more classes to the selected elements
- removeClass() - Removes one or more classes from the selected elements
- toggleClass() - Toggles between adding/removing classes from the selected elements

Lets try it

- In the provided html, remove all records from view that have been deleted (ex2)
- Define a class and highlight the “important” records with an yellow background

Adding New Content

We will look at four jQuery methods that are used to add new content:

- append() - Inserts content at the end of the selected elements
- prepend() - Inserts content at the beginning of the selected elements
- after() - Inserts content after the selected elements
- before() - Inserts content before the selected elements

Removing Content

To remove elements and content, there are mainly two jQuery methods:

- remove() - Removes the selected element (and its child elements)
- empty() - Removes the child elements from the selected element

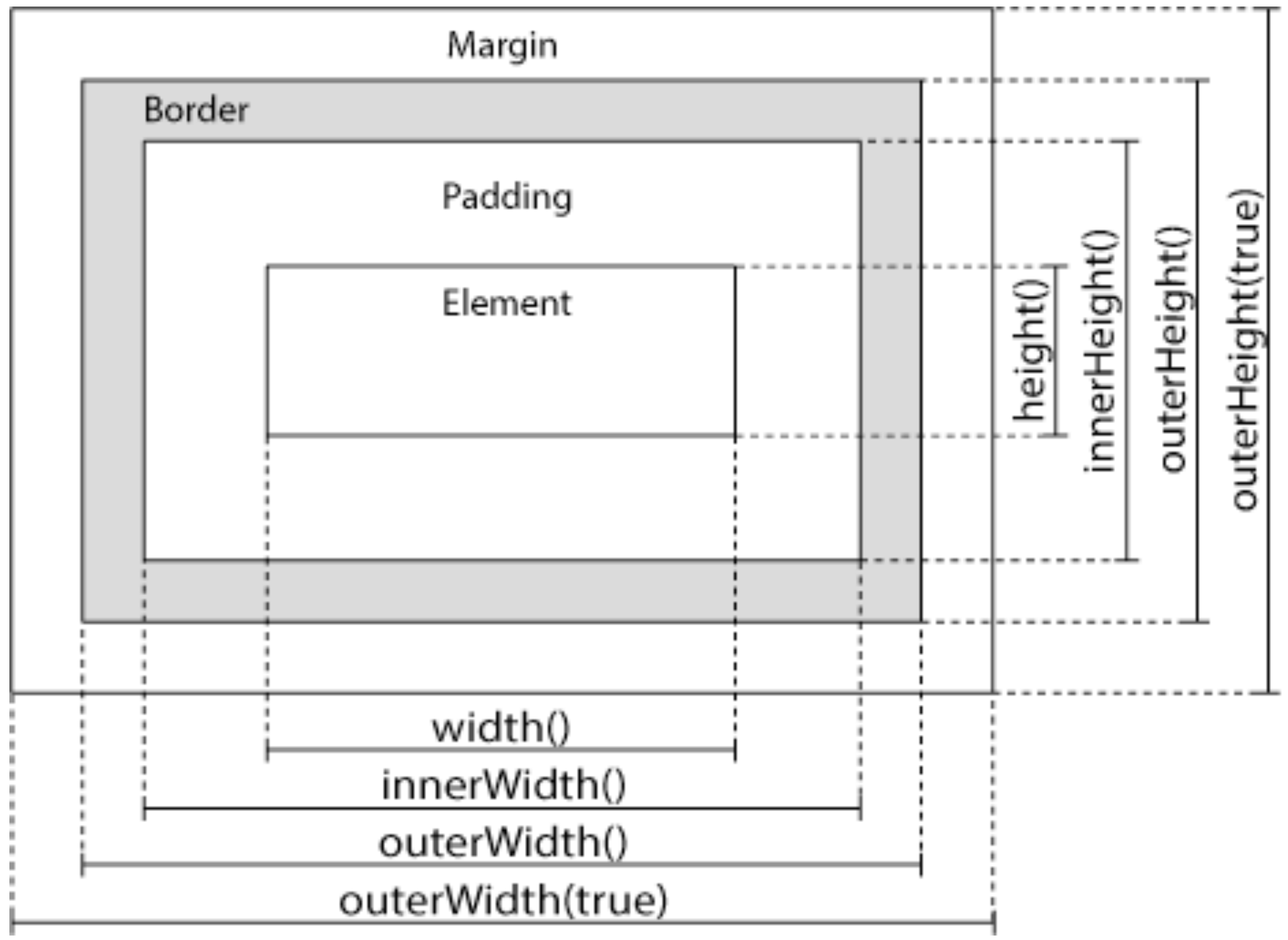
Lets try it

- In the html provided, when user clicks on “show details” load the image related to the book from adjacent table column into the results div (book_preview). (ex3)

- Things you might need:

  - next() and parent() functions exist to find sibling or parent of an element

Element Sizes



Lets Try It

- Implement a smooth zoom for the book image on the previous html (ex4)

Filters

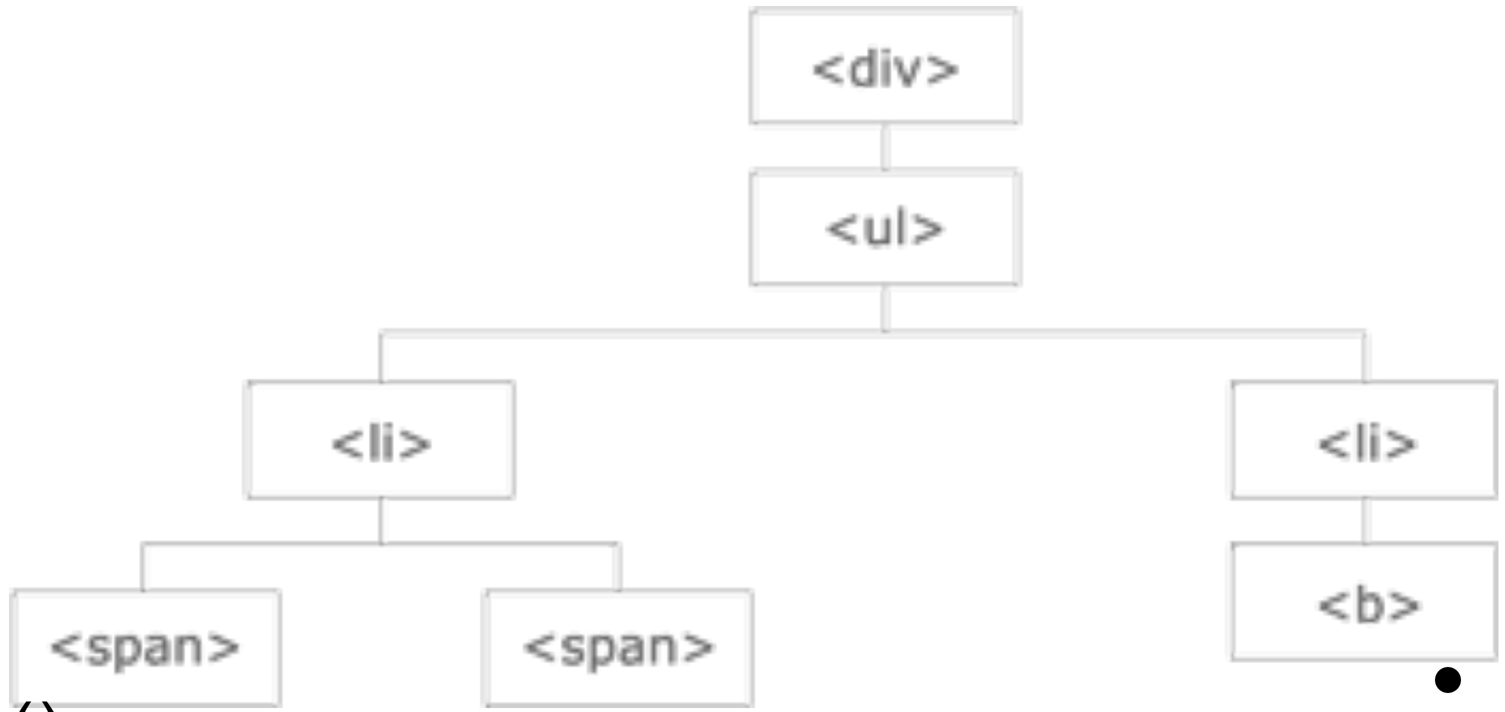
- filter() - positive selection, not() negative selection
- Filters reduce search complexity

```
$(".deleted").filter(".odd").remove();
```

- Filters let us write matching code

```
$(".deleted").filter(function(index){
   if(index==0)
      return true;
   else
      return false;
}).remove();
```

DOM Traversal



- parent()
- parents()
- parentsUntil()
- children()
- find()
- eq()

- siblings()
- next()
- nextAll()
- nextUntil()
- prev()
- prevAll()
- prevUntil()

AJAX with JQuery

- $(selector).load(URL,data,callback);
- $.get(URL,callback);
- $.post(URL,data,callback);
  - Load the result of the ajax call directly into the specified element
  - Callback parameters
    - responseTxt - contains the resulting content if the call succeeds
    - statusTxt - contains the status of the call
    - xhr - contains the XMLHttpRequest object

Events

- Attaching Event Handlers

```
$( "a" ).click(clickFunction);

$( "a" ).click(function(){
   alert("Link clicked");
   event.preventDefault();
});
```

- Turning on and off event handlers

```
var foo = function() {
   // Code to handle some kind of event
};
$( "body" ).on( "click", "p", foo );
$( "body" ).off( "click", "p", foo );
```

Lets try it

- On the page provided, on change of the country, change the list of states using an AJAX call. There are two files on server India.txt and US.txt by the same name as countries. Load from there (ex5)

Event Delegation

- Event delegation allows us to attach a single event listener, to a parent element, that will fire for all descendants matching a selector, whether those descendants exist now or are added in the future.

- Event delegation is a nice way of handling dynamically evolving html DOM

Delegated Events & Event Bubbling

- For this html

```
<div id="container">
    <ul id="list">
        <li><a href="http://domain1.com">Item #1</a></li>
        <li><a href="http://domain2.com">Item #2</a></li>
        <li><a href="http://domain3.com">Item #3</a></li>
        <li><a href="http://domain4.com">Item #4</a></li>
    </ul>
</div>
```

- Directly bound events - work only on currently present elements

```
$( "#list a" ).on( "click", function( event ) {
    event.preventDefault();
    console.log( $( this ).text() );
});
```

- Delegated event handler on the list will work on any newly added elements too

```
$( "#list" ).on( "click", "a", function( event ) {
    event.preventDefault();
    console.log( $( this ).text() );
});
$( "#list" ).append( "<li><a href='http://newdomain.com'>Item #5</a></li>" );
```

Event Object

- event.delegateTarget
    - The element where the currently-called jQuery event handler was attached.
- event.target
    - The DOM element that initiated the event.
- event.which
    - For key or mouse events, this property indicates the specific key or button that was pressed.
- event.stopPropagation() vs preventDefault()

Proxy

- A proxy is a copy of the function that is invoked with a different context
  - Means that “this” in the function refers to the object passed to the proxy call
  - Mostly doable using eventData when registering events

```
$('#panel').fadeIn(function(){
   // Using $.proxy to bind this:
   $('#panel button').click($.proxy(function(){
       // this points to #panel
       $(this).fadeOut();
   },this));
});
```

Form Data Binding

- Automatic data binding between form elements and an object is possible
  - Use delegated events for all input boxes inside a form
  - Use form id to identify the object
  - Bind form fields to object fields based on the name attribute

Effects

- hide()/show()/toggle()
- fadeIn()/fadeOut()/fadeTo()
- Custom animations with animate()

```
$( “#div-id" ).animate({
    opacity: 0.25,
    left: "+=50",
    height: "toggle"
}, 5000, function() {
    // Callback when animation finishes
});
```

Lets try it

- On the provided html pop up a validation message on validation error on the form. Use delegated events for form data binding. Name is mandatory, age should be above 18.(ex6)

An Useful Dialog

- Create a dialog class that supports ok/cancel buttons.
- Provide a subclass that also adds an ignore button

Create CRUD

- Using JQuery and a RESTful backend, create CRUD UI