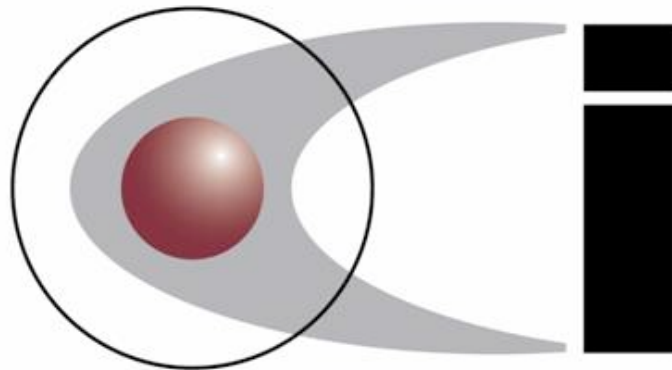


jQuery - RIA Miracle!

R. Mark Volkmann
mark@ociweb.com



OBJECT COMPUTING, INC.

jQuery is ...

- A JavaScript library for building web apps

- hides browser differences
- accesses DOM elements with CSS-like selectors
- simplifies traversing and modifying DOM
- simplifies responding to mouse and keyboard events
- animates changes such as color, opacity, size and position
- sends data to a server and retrieves data from it (Ajax)
- reduces need for loops
 - many jQuery functions act on each selected element (functional)

Document Object Model (DOM) is in-memory representation of HTML

Cascading Style Sheets (CSS) specify formatting including colors, fonts and sizes

- Dual licensed

- GNU Public License and MIT License

<http://jquery.com>

jQuery UI is ...

- A large jQuery plugin that adds

There are many more jQuery plugins.
jQuery UI is the most commonly used.

- widgets
 - **Accordion, Autocomplete, Button, Datepicker, Dialog, Progressbar, Slider, Tabs**
- interactions
 - **Draggable, Droppable, Resizable, Selectable, Sortable**
- effects
 - Effect, **Hide, Show, Toggle**, Color animation, **Add/Remove/Toggle/Switch class**
- utilities
 - Position

- Themeable

- can select from several prebuilt themes and build custom themes

<http://jqueryui.com>

Selecting DOM Elements

- Three primary ways
 - by tag name - *name*
 - by id attribute value - *#id*
 - by CSS class - *.class-name*
- Also by DOM tree hierarchy
 - direct children - *a > b*
 - descendants - *a b*

Support for jQuery “selector” syntax was split into a separate library and named “Sizzle”.

<http://sizzlejs.com/>

W3C DOM has adopted this syntax in the “Selectors API”.

<http://www.w3.org/TR/selectors-api/>

Magical jQuery \$ Function

- Selects DOM elements with CSS-like selectors
 - and encapsulates them in a jQuery object
 - many methods on this object act on each encapsulated DOM element
- Creates DOM elements from HTML text
- Encapsulates a reference to a DOM element, or array of them, in a jQuery object
- Specifies ready function
 - `$ (some-function)` is the same as
`$ (document) .ready (some-function)`

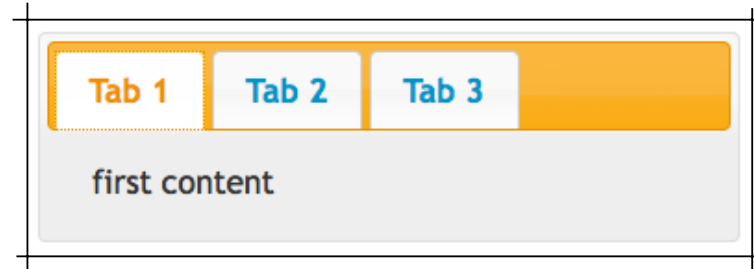
Let's Build Something!

- A meal planner
 - provide a tab for each day of the week
 - select a specific week from a calendar
 - display meals for each day in an accordion (breakfast, lunch, dinner)
 - allow items to be entered in a text field with autocomplete using Ajax
 - update progress bar to indicate progress in planning 21 meals for week
 - support drag and drop of items to a different meal, different day and trash can
 - provide a view for printing all planned meals

Demo Time!

Tabs Example

```
1 <script>
2   $(document).ready(function() {
3     $("#tabs").tabs();
4   });
5 </script>
6 ...
7 <div id="tabs">
8   <ul>
9     <li><a href="#t1">Tab 1</a></li>
10    <li><a href="#t2">Tab 2</a></li>
11    <li><a href="#t3">Tab 3</a></li>
12  </ul>
13  <div id="t1">first content</div>
14  <div id="t2">second content</div>
15  <div id="t3">third content</div>
16 </div>
```

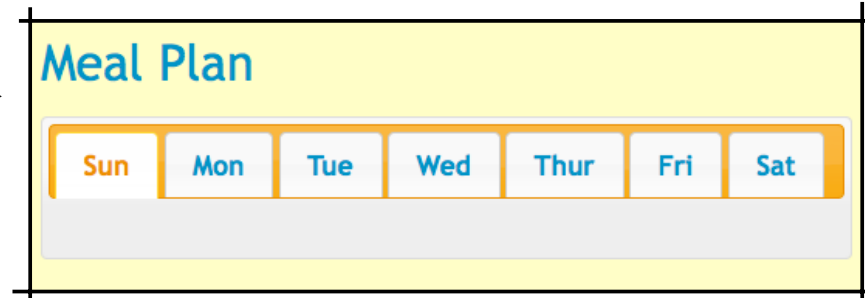


Initial HTML

```
1 <html>
2   <head>
3     <title>Meal Plan</title>
4     <link rel="stylesheet" type="text/css"
5       href="../../jquery-ui-1.8.12.custom/css/ui-lightness/jquery-ui-1.8.12.custom.css"/>
6     <link rel="stylesheet" type="text/css" href="mealplan.css"/>
7     <script
8       src="../../jquery-ui-1.8.12.custom/js/jquery-1.5.1.min.js"></script>
9     <script
10      src="../../jquery-ui-1.8.12.custom/js/jquery-ui-1.8.12.custom.min.js"></script>
11     <script src="arrayextras.js"></script>
12     <script src="mealplan.js"></script>
13   </head>
14   <body>
15     <div class="title">Meal Plan</div>
16     <div id="tabs"></div>
17   </body>
18 </html>
```

for production deployment, consider combining, minimizing and compressing the JavaScript files

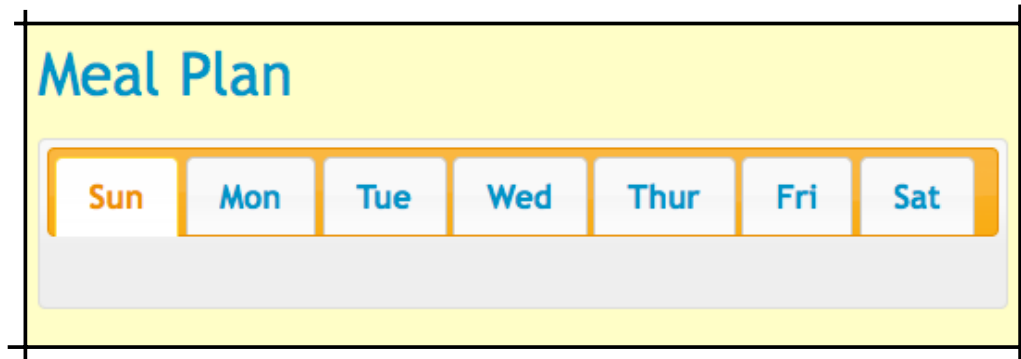
← provides ES5 array functions in ES3 environments; not needed in newer environments



Initial CSS

```
1  body {
2      /* This color was derived using http://colorblenny.com from
3         the dodge of orange and blue shades in jQuery UI Lightness theme.
4         The shade of orange is #e78f08.
5         The shade of blue is #1c94c4. */
6      background-color: #ffffcb; /* light yellow */
7  }
8
9  .title {
10     color: #1c94C4; /* blue */
11     font-family: Trebuchet MS, Tahoma, Verdana, Arial, sans-serif;
12     font-size: 24pt;
13 }
14
15 #tabs {
16     margin-top: 15px;
17 }
```

copied from "UI Lightness" theme



Initial JavaScript ...

```
1 var mp = {}; // namespace
2 mp.days = ['Sun', 'Mon', 'Tue', 'Wed', 'Thur', 'Fri', 'Sat'];
3
4 $(document).ready(function () {
5     mp.addTab('tabs', mp.days, mp.days);
6 });
7
8 mp.addTab = function (containerId, tabIds, tabLabels) {
9     var container = $('#' + containerId);
10    container.append('<ul>');
11
12    tabIds.forEach(function (tabId, index) {
13        mp.addTab(container, tabId, tabLabels[index]);
14    });
15
16    container.tabs();
17 };
```

forEach is an ES5 array function



... Initial JavaScript

```
1 mp.addTab = function (container, tabId, tabLabel) {
2   // Add li for tab.
3   container.children('ul').append(
4     '<li><a href="#" + tabId + '>' + tabLabel + '</a></li>');
5
6   // Add div for tab content.
7   container.append('<div id="' + tabId + '>');
8 };
```

Date Picker

Meal Plan

Start Date: ...

Note dates added to tabs!

Sun 9/19	Mon 9/20	Tue 9/21	Wed 9/22	Thur 9/23	Fri 9/24	Sat 9/25
----------	----------	----------	----------	-----------	----------	----------

Meal Plan

Start Date: ...

Sun 9

Sep 2010

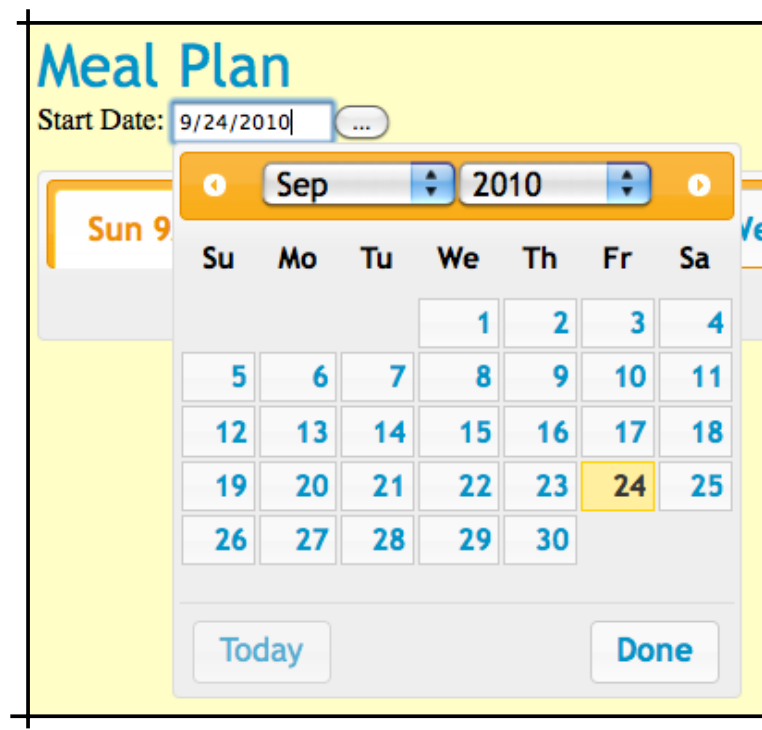
Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

Today Done

Date Picker HTML

- 1 `<label for="startDate">Start Date:</label>`
- 2 `<input id="startDate" type="text" size="10"/>`

← container for a label associated with an input element

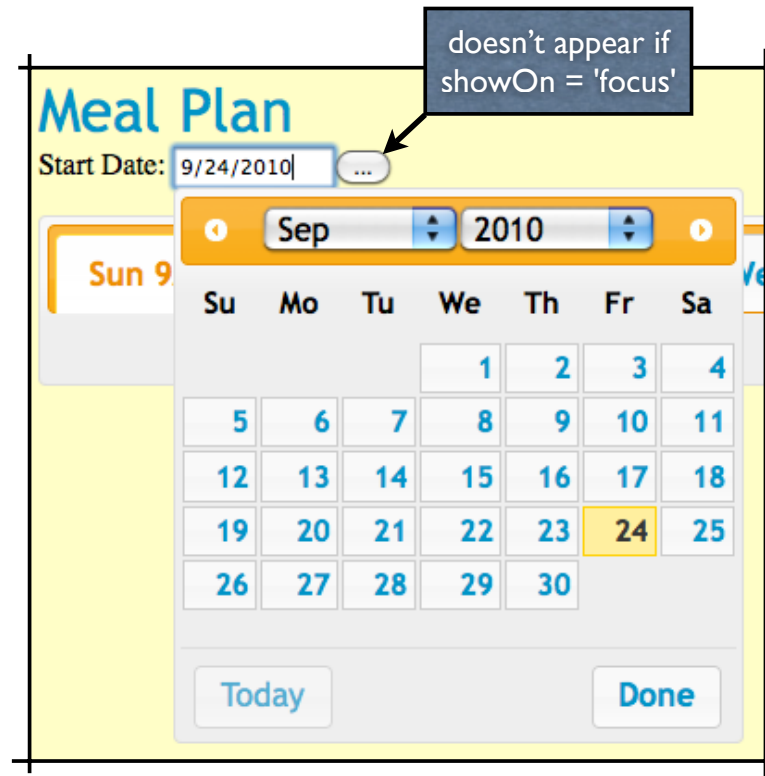


Date Picker JavaScript ...

```
1 $('#startDate').datepicker({
2   changeMonth: true,
3   changeYear: true,
4   dateFormat: 'm/d/yy',
5   onClose: mp.addDates, 16
6   showButtonPanel: true,
7   showOn: 'both' // focus and button
8 });
9
10 $('#startDate').val(
11   mp.getFormattedDate() ); 15
12
13 mp.addDates( 16
14   new Date().toString() ); // to tabs
15
```

date string is passed

current date



... Date Picker JavaScript ...

```
1 mp.getFormattedDate = function (date) {  
2   if (!date) {  
3     date = new Date();  
4   }  
5   return (date.getMonth() + 1) + '/' +  
6     date.getDate() + '/' + date.getFullYear();  
7 };
```

... Date Picker JavaScript

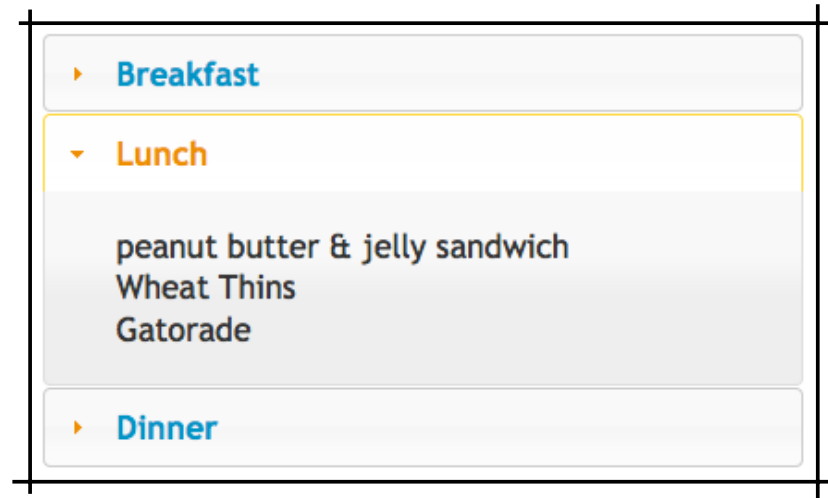
```
1 mp.addDates = function (dateText) {
2   var date = new Date(dateText);
3
4   // Change date to previous Sunday.
5   date.setDate(date.getDate() - date.getDay());
6
7   // For each tab anchor ...
8   $('#tabs > ul > li > a').each(function (index, a) {
9     var anchor = $(a), day;
10    day = anchor.text().split(' ')[0]; // get day abbreviation
11    dateText = (date.getMonth() + 1) + '/' + date.getDate();
12    anchor.text(day + ' ' + dateText);
13    date.setDate(date.getDate() + 1); // advance by one day
14  });
15  };
```

Annotations:

- of week (points to `date.getDay()`)
- a DOM element (points to `a`)
- see slide 7 (points to `$('#tabs > ul > li > a')`)

Accordion Example

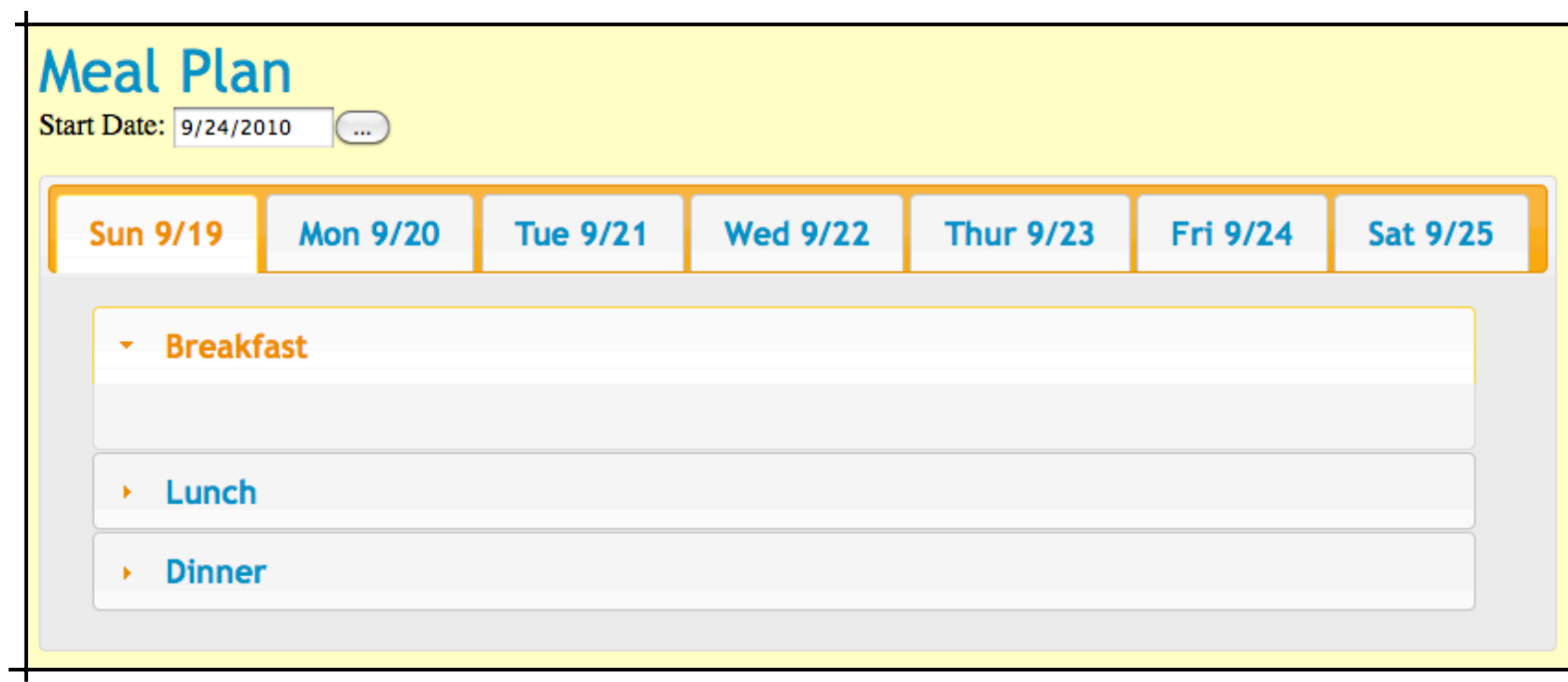
```
1 <script>
2   $(document).ready(function() {
3     $("#meals").accordion();
4   });
5 </script>
6 ...
7 <div id="meals">
8   <h3><a href="">Breakfast</a></h3>
9   <div>
10    cereal<br/>banana<br/>orange juice
11  </div>
12  <hr/>
13  <h3><a href="">Lunch</a></h3>
14  <div>
15    peanut butter & jelly sandwich<br/>Wheat Thins<br/>Gatorade
16  </div>
17  <hr/>
18  <h3><a href="">Dinner</a></h3>
19  <div>spaghetti<br/>French bread<br/>lemonade</div>
20 </div>
```



Accordion CSS

(no HTML changes)

```
1 .ui-accordion {  
2   width: 750px;  
3 }
```



The screenshot shows a jQuery UI accordion widget. The title bar is yellow and contains the text "Meal Plan" in blue. Below the title bar is a "Start Date:" label followed by a text input field containing "9/24/2010" and a small grey button with three dots. The accordion body is light grey and contains seven panels, each representing a day of the week from "Sun 9/19" to "Sat 9/25". The "Sun 9/19" panel is expanded, showing three sub-panels: "Breakfast" (with a downward arrow), "Lunch" (with a rightward arrow), and "Dinner" (with a rightward arrow). The other panels are collapsed.

Accordion JavaScript ...

```
1 mp.day = 'Sun'; // default used later
2 mp.days = ['Sun', 'Mon', 'Tue', 'Wed', 'Thur', 'Fri', 'Sat']; repeated from earlier slide
3 mp.meal = 'Breakfast'; // default used later
4 mp.meals = ['Breakfast', 'Lunch', 'Dinner'];
5
6 mp.days.forEach(function (day) {
7     mp.addAccordion(day, mp.meals);
8 });
```

... Accordion JavaScript ...

```
1 mp.addAccordion = function (containerId, labels) {
2   var container = $('#' + containerId); // for tab content
3
4   labels.forEach(function (label) {
5     var anchor, div, divId;
6     divId = containerId + '-' + label; // of an accordion panel
7     div = $('<div id="' + divId + '">'); // for accordion panel content
8
9     anchor = $('<a>' + label + '</a>');
10    container
11      .append($('<h3>').append(anchor))
12      .append(div);
13  });
```

continued on next slide

... Accordion JavaScript

```
14 container.accordion({
15     autoHeight: false, // each panel can have a different height
16     change: function (event, ui) {
17         mp.meal = ui.newHeader.text(); // currently displayed meal
18     }
19 });
20 };
```

used on slide 27
to open the accordion to the current meal
when a different day tab is selected

Adding Items

Meal Plan

Start Date: ...

New Item:

Sun 9/19	Mon 9/20	Tue 9/21
----------	----------	----------

- ▶ Breakfast
- ▼ Lunch
 - pizza
- ▶ Dinner

Adding Items HTML

```
1 <table>
2   <tr>
3     <td class="label"><label for="startDate">Start Date:</label></td>
4     <td><input id="startDate" type="text" size="10"/></td>
5   </tr>
6   <tr>
7     <td class="label"><label for="itemText">New Item:</label></td>
8     <td>
9       <input id="itemText" type="text" size="20"/>
10      <button id="addButton">Add</button>
11    </td>
12  </tr>
13 </table>
```

I know, I know!
I'm using a table for layout
instead of CSS.
<http://giveupandusetables.com/>

Adding Items CSS

```
1  .item { copied from "UI Lightness" theme used on slide 28  
2      font-family: Trebuchet MS, Tahoma, Verdana, Arial, sans-serif;  
3      font-size: 12pt;  
4  }
```


Adding Items JavaScript ...

```
1  var items, itemText;
2
3  //items = ['banana', 'burritos', 'cereal', 'gatorade', 'lemonade',
4  // 'orange juice', 'lasagna', 'pizza', 'spaghetti'];
5  itemText = $('#itemText');
6  itemText.autocomplete({
7    //source: items // to get items from array
8    source: mp.getPossibleItems // to get items from Ajax service 31
9  });
10
11 itemText.keyup(function (event) {
12   if (event.which === 13) { // enter key
13     mp.addItem();
14   }
15 });
16
17 itemText.focus();
```

... Adding Items JavaScript ...

```
1 $('#addButton').button(); // makes it look like a jQuery UI button
2 $('#addButton').click(mp.addItem);
3
4 $('label').addClass('ui-widget'); // to use font of jQuery UI widgets
5
6 mp.addItem = function () {
7     var itemDiv, itemText = $('#itemText');
8     itemDiv = mp.createItem(itemText.val()); 28
9     mp.dropItem(itemDiv, mp.day); 30
10    itemText.val('');
11    itemText.focus();
12 };
```

day associated with
currently selected tab

... Adding Items JavaScript ...

- Modify mp.addTab

```
1 container.tabs({  
2   select: function (event, ui) {  
3     mp.day = ui.panel.id;  
4  
5     // Open the accordion for this day to the current meal.  
6     $('#'+ mp.day).accordion(  
7       'activate', mp.meals.indexOf(mp.meal));  
8   }  
9 });
```

... Adding Items JavaScript ...

```
1 mp.createItem = function (text) {  
2   var itemDiv = $('<div class="item">');  
3   itemDiv.text(text);  
4   return itemDiv;  
5 };
```

...Adding Items JavaScript ...

```
1 // Returns a boolean indicating if an element is in an array.
2 // $.isArray isn't reliable in IE!
3 mp.inArray = function (element, array) {
4     for (var index in array) {
5         if (array[index] === element) {
6             return true;
7         }
8     }
9     return false;
10 };
```

used on next slide

... Adding Items JavaScript ...

```
1  mp.dropItem = function (itemDiv, dayOrMeal) { called from slides 26 & 42
2    if (mp.inArray(dayOrMeal, mp.days)) { 29
3      mp.day = dayOrMeal; // use current meal
4    } else if (mp.inArray(dayOrMeal, mp.meals)) {
5      mp.meal = dayOrMeal; // use current day
6    } else {
7      throw "'" + dayOrMeal + "' is an invalid drop target';
8    }
9
10   var target = $('#' + mp.day + '-' + mp.meal); // an accordion panel
11   target.append(itemDiv);
12 };
```

...Adding Items JavaScript

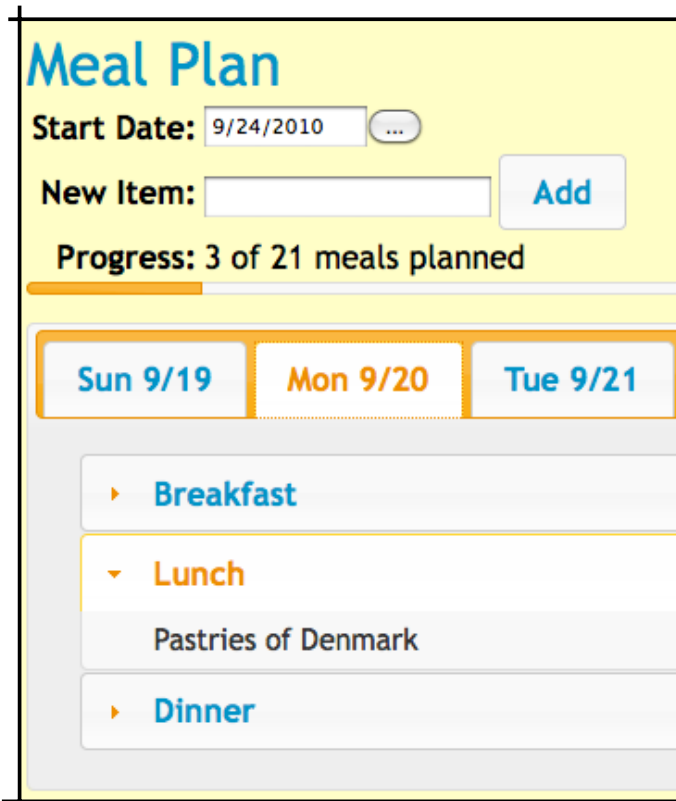
generated by autocomplete



```
1 mp.getPossibleItems = function (req, callback) called from slide 25
2   $.ajax({
3     url: 'http://localhost:3000',
4     dataType: 'jsonp', // JSON with Padding to get around same origin policy
5     data: {
6       term: req.term // what the user typed so far
7     },
8     success: callback
9   });
10 // TODO: Make this use a local array of items
11 // TODO: if the service isn't available.
12 };
```

Progress Bar HTML

```
1 <table>
2   ...
3   <tr>
4     <td class="label">
5       <label for="count">Progress:</label>
6     </td>
7     <td class="ui-widget">
8       <span id="count">0</span>
9       of 21 meals planned
10    </td>
11  </tr>
12 </table>
13
14 <div id="progress"></div>
```



The screenshot shows a web interface titled "Meal Plan" with a yellow header. It includes a "Start Date" field set to "9/24/2010" with a dropdown arrow, a "New Item:" input field, and an "Add" button. Below this, it displays "Progress: 3 of 21 meals planned" with a progress bar. A navigation bar shows "Sun 9/19", "Mon 9/20" (selected), and "Tue 9/21". The main content area lists meals: "Breakfast", "Lunch" (expanded to show "Pastries of Denmark"), and "Dinner".

Progress Bar CSS

```
1 #progress {  
2   height: 5px;  
3 }
```

Progress Bar JavaScript ...

```
1 $('#progress').progressbar();  
2  
3 mp.updateProgress = function () {  
4     var count = mp.getMealCount(); 35  
5     $('#count').text(count);  
6     $('#progress').progressbar('value', 100 * count / 21);  
7 };
```

- New code at end of mp.addTab

```
mp.updateProgress(); // count will be zero
```

... Progress Bar JavaScript ...

```
1  mp.getMealCount = function () {
2    var count = 0, items;
3
4    mp.days.forEach(function (day) {
5      mp.meals.forEach(function (meal) {
6        items = mp.getAddedItems(day, meal); 36
7        if (items.length > 0) {
8          count += 1;
9        }
10     });
11   });
12
13   return count;
14 };
```

... Progress Bar JavaScript

```
1 // Returns array of items that have been entered
2 // for a given day and meal.
3 mp.getAddedItems = function (day, meal) { called from slides 35 & 49
4     var items = [];
5     $('#' + day + '-' + meal + ' > div').each(function (index, div) {
6         items.push($(div).text());
7     });
8     return items;
9 };
```

Drag and Drop HTML

```
1 <table>
2   ...
3   <tr>
4     <td class="label"><label for="itemText">New Item:</label></td>
5     <td>
6       <input id="itemText" type="text" size="20"/>
7       <button id="addButton">Add</button>
8     </td>
9     <td></td>
11   </tr>
12   ...
13 </table>
```

Drag and Drop CSS

```
1 .compatibleTarget {  
2   background-color: lightgreen;  
3 }  
4  
5 .overTarget {  
6   background-color: green;  
7 }
```

Drag and Drop JavaScript ...

```
1 mp.droppableOptions = {  
2   activeClass: 'compatibleTarget',  
3   drop: mp.drop, 42  
4   hoverClass: 'overTarget'  
5 };
```

- New code at end of mp.addTab to make tabs be drop targets

```
$('#tabs > ul > li > a').droppable(mp.droppableOptions);
```

- New code in mp.addAccordion to make accordion headers be drop targets

```
anchor.droppable(mp.droppableOptions);
```

... Drag and Drop JavaScript ...

- To make trash can be a drop target

```
1 $('#trash').droppable({
2   drop: function (event, ui) {
3     $(ui.draggable).remove(); // actual item
4     $('.ui-draggable-dragging').remove(); // helper clone
5     $('#trash').attr('src', 'trashCanClosed.png');
6     mp.updateProgress(); // may have one less meal planned now 34
7   },
8   out: function (event, ui) {
9     $('#trash').attr('src', 'trashCanClosed.png');
10  },
11  over: function (event, ui) {
12    $('#trash').attr('src', 'trashCanOpen.png');
13  }
14 });
```


... Drag and Drop JavaScript ...

- New code in mp.createItem

```
1 itemDiv.draggable({
2   appendTo: 'body', // can't drag out of accordion without this
3   containment: 'body',
4   cursor: 'pointer',
5   helper: 'clone', // can't drag out of accordion without this
6   revert: 'invalid', ← if not dropped on a compatible target,
7   start: function () {   slide back to starting location
8     $(this).hide(); // so only helper is visible
9   },
10  stop: function () {
11    $(this).show(); // so actual item becomes visible again
12  }
13 });
```

... Drag and Drop JavaScript ...

```
1 mp.drop = function (event, ui) {
2   var dayOrMeal, droppable;
3   droppable = $(this); // an "a" element for a tab or accordion panel
4   dayOrMeal = droppable.text().split(' ')[0];
5   mp.dropItem($(ui.draggable), dayOrMeal); // into new day/meal 30
6   mp.updateProgress(); // # of planned meals may have changed 34
7 };
```

... Drag and Drop JavaScript

- New code in mp.dropltem

```
1 // Select the tab for the day on which it was dropped.
2 $('#tabs').tabs('select', mp.days.indexOf(mp.day));
3
4 // Open the accordion panel for the meal on which it was dropped.
5 $('#'+ mp.day).accordion('activate', mp.meals.indexOf(mp.meal));
```

Print View

Meal Plan

[Print View](#)

Meal Plan

[Edit View](#)

Sun 9/19	Mon 9/20	Tue 9/21	Wed 9/22	Thur 9/23	Fri 9/24	Sat 9/25
pancakes bacon						
	pizza					
					Cheesecake Factory	

Print View HTML

- Add after title div

```
<button id="switchViewButton">Print View</button>
```

- Wrap previous content after title div in

```
<div id="editView"> ... </div>
```

- Add to end

```
<div id="printView">  
  <table id="printTable"></table>  
</div>
```

Print View CSS

```
1 #printTable {  
2   margin: 20px;  
3 }  
4  
5 #printTable th, #printTable td {  
6   padding: 10px;  
7 }  
8  
9 #printTable, #printTable th, #printTable td {  
10  border: solid red 2px;  
11  border-collapse: collapse;  
12 }  
13  
14 #switchViewButton {  
15   margin-top: 15px;  
16   margin-bottom: 15px;  
17 }
```

border is needed for #printTable so borders are drawn around empty cells

Print View JavaScript ...

```
1 var effect = 'blind';
2 $('#printView').hide();
3 $('#switchViewButton').button(); // makes it look like a jQuery UI button
4 $('#switchViewButton').toggle(
5   function () {
6     mp.fillTable(); 48
7     $(this).children('span').text('Edit View');
8     $('#editView').hide(effect);
9     $('#printView').show(effect);
10  },
11  function () {
12    $(this).children('span').text('Print View');
13    $('#printView').hide(effect);
14    $('#editView').show(effect);
15  });
```

effect is a local variable inside the anonymous ready function

wraps button text in a span tag

... Print View JavaScript ...

```
1 mp.fillTable = function () {
2   var table, td, tr;
3   table = $('#printTable');
4   table.empty();
5
6   // Create header row.
7   tr = $('<tr>');
8   table.append(tr);
9
10  // For each tab anchor ...
11  $('#tabs > ul > li > a').each(function (index, a) {
12    tr.append('<th>' + $(a).text() + '</th>');
13  });
```

the day

... Print View JavaScript


```
14 mp.meals.forEach(function (m) {
15     tr = $('<tr>');
16     table.append(tr);
17
18     mp.days.forEach(function (d) {
19         td = $('<td>');
20         tr.append(td);
21         mp.getAddedItems(d, m).forEach(function (item) {
22             td.append(item + '<br/>');
23         });
24     });
25 });
26 };
```

Done!

Meal Plan

[Print View](#)

Start Date:

New Item: [Add](#) 

Progress: 1 of 21 meals planned

Sun 10/10	Mon 10/11	Tue 10/12	Wed 10/13	Thur 10/14	Fri 10/15	Sat 10/16
					Snickers bar can of Coke	

[Edit View](#)

Sun 10/10 Mon 10/11 Tue 10/12 Wed 10/13 Thur 10/14 Fri 10/15 Sat 10/16

▼ **Breakfast**

Snickers bar
can of Coke

▶ **Lunch**

▶ **Dinner**

<http://ociweb.com/mark/programming/jquery/MealPlan/>

Learn More In OCI Classes

- Web Development Fundamentals
2 days
 - HTML 4
 - CSS 1 & 2
 - Basic JavaScript
 - JSLint
 - Advanced JavaScript
 - Firebug
 - HTML5
 - CSS3
 - ECMAScript 5
- jQuery RIA Development
2 days
 - jQuery selectors
by tag, class, id, attribute value, etc.
 - jQuery DOM traversal
parent, children, siblings, find, etc.
 - jQuery event handling
click, keypress, mouseover, live, etc.
 - jQuery effects
hide, show, fade, animate, etc.
 - jQuery DOM manipulation
attributes, classes, text, children, etc.
 - jQuery Ajax
GET, POST, XML, JSON, JSONP, etc.
 - jQuery plugins
Form, jQuery UI, Validation, Tablesorter, etc.
 - jQuery UI
new widgets, interactions and effects

<http://ociweb.com/training/>