Maqetta: Visual Authoring of HTML5 User Interfaces

Log in
to participate

Maqetta Widget Spotlight: HTML Tables

tonyerwin | *Oct 2 2012* | *Visits (2173)*

0
Like 0
Share
Tweet 0

Introduction

The `<table>` element is one of the most powerful of the HTML 5 elements. But, it can also be one of the most difficult to master because of its many attributes, style properties, and applicable child elements. Luckily, Maqetta provides easy-to-use facilities so a designer can quickly build tables without a deep understanding of their internal nuts and bolts. This post will explore these time-saving features. In addition, some more advanced topics related to issues concerning styling of table elements (caused by Dojo's *document.css* stylesheet) will be discussed at the end of the article.

When to Use `<table>` Elements?

Before diving into `<table>` manipulation in the Maqetta UI, let's briefly touch on what the `<table>` element is used for and what alternatives exist. `<table>`'s are primarily intended for the display of data in an n-by-n grid format. However, without getting into the deep philosophical debates regarding `<table>` vs. `<div>` layouts, they can also be used to layout elements within a page.

In my opinion, `<table>`'s as layout elements have particular value when doing rapid prototyping of simple layouts. For one thing, tables tend to have a shallower learning curver. In addition, sometimes a table is an extraordinarily natural fit for a desired layout (for example, to layout a name/address form which has labels on the left and text fields on the right).

However, before you start relying too much on tables as layout elements (a state *Smashing Magazine* would call "table hell") please note that Maqetta offers full support for `<div>` elements. In addition, Maqetta supports the various Dojo containers (e.g., `AccordionContainer`, `BorderContainer`, `TabContainer`, etc.) to help you manage the most complex page layouts.

I should also point out that even if your goal is just to display tabular data, there are cases where you'd be better served using specially tailored Dojo widgets like `GridX` rather than `<table>`. For example, if your data is stored in a file or comes from a web service or if you want to provide advanced functionality for your users like selection, sorting, filtering, etc., then you should use `GridX`.

Creating a Basic Default Table

To get started working with HTML tables in Maqetta, let's first create a new HTML file by choosing the **Desktop Application...** option from the **Create** menu. When the dialog comes up, you can enter any file name you wish and then click the **Create** button. This will cause a new file to be created and opened in the page editor with a blank canvas.

Next, let's drag and drop a `<table>` element from the Widget Palette onto the canvas. The Widget Palette can be found on the left side of the Maqetta workspace, and the `<table>` element is one of many HTML elements found inside of the *HTML* section in the palette (shown in the screenshot below).

About this blog

In this blog, Tony Erwin covers a range of topics related to Maqetta. Maqetta is an open source project that provides WYSIWYG visual authoring of HTML5 user interfaces (desktop and mobile).

Related posts

Fragmentation Contin...
UpdatedYesterday 2:52 PM 0 0

The Importance of Ev...
UpdatedYesterday 2:51 PM 0 0

Insights from Innova...
UpdatedJune 12 1 1

Maqetta Means Mock-u...
UpdatedJune 12 1 0

TWOdW: Why HTML5 gra...
UpdatedJune 11 0 0

Tags

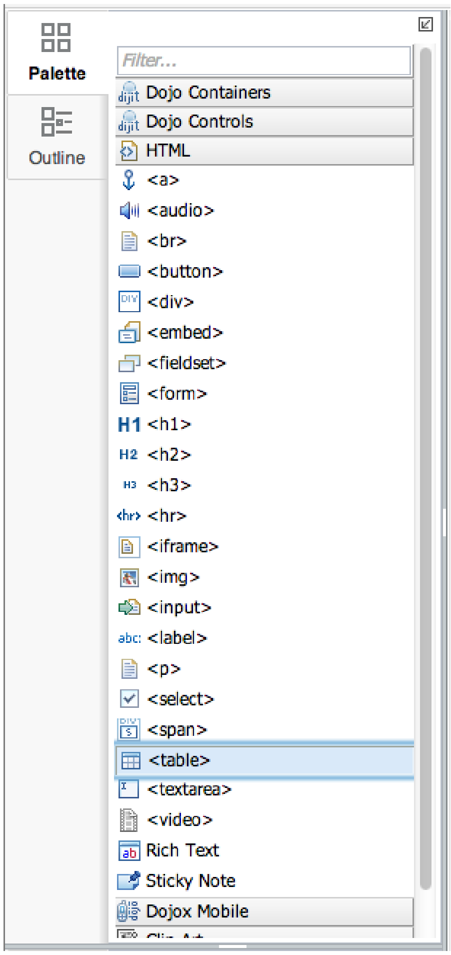
Find a Tag

css design dijit dojo gridx html5 javascript maqetta mobile phonegap prototype rational_application_developer rational-application-developer welcome workflow

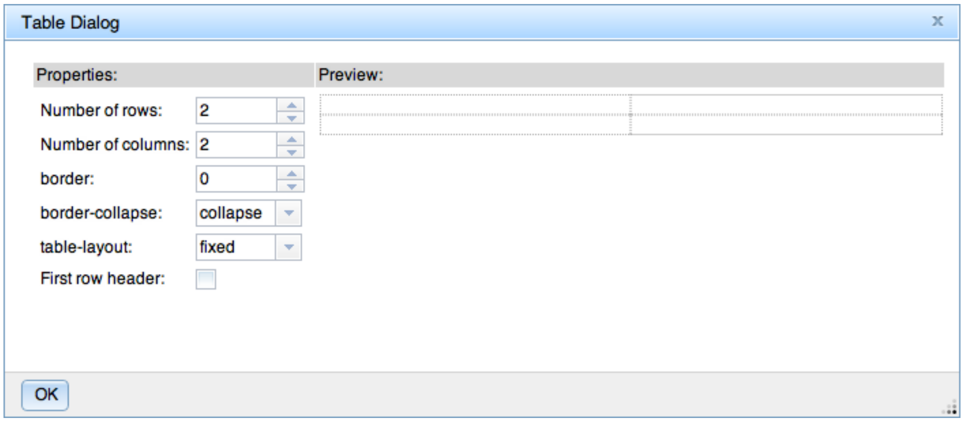
Cloud | List

Recent tweets

Follow @tonyerwin



When you drop the `<table>` element onto the canvas, you will see a dialog like the one below:

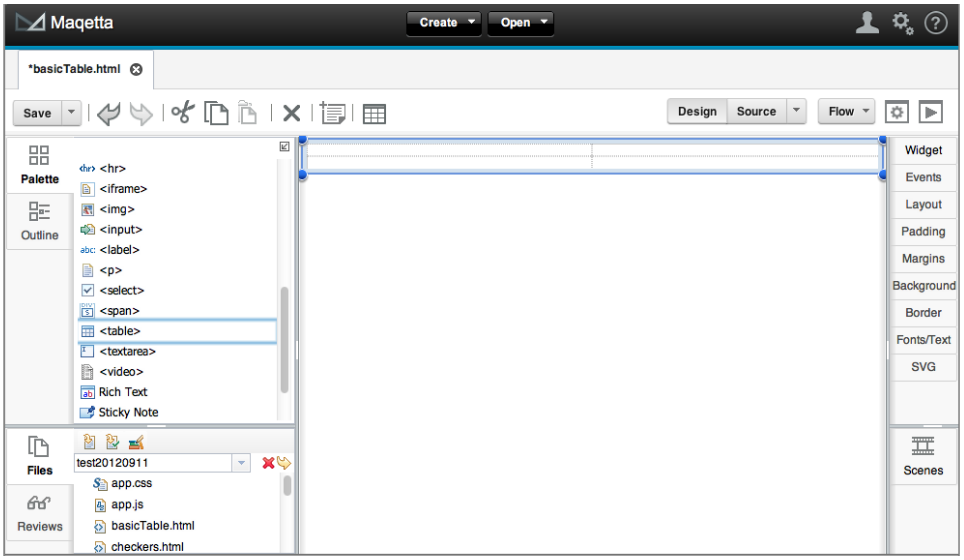


Number of rows: 2
Number of columns: 2
border: 0
border-collapse: collapse
table-layout: fixed
First row header: □

The dialog contains the following options:

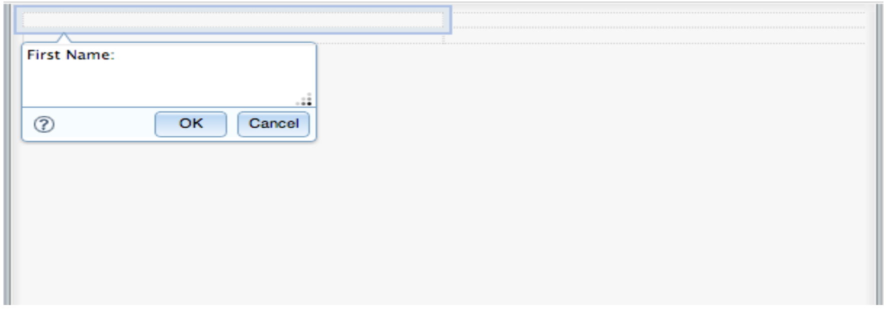
- *Number of rows*
- *Number of columns*
- *border* — determines the thickness of the border around table cells (0 means no border)
- *border-collapse* — corresponds to the `border-collapse` CSS property and determines whether the border is represented (or, *collapsed*) as a single border. The default for this option is `collapse`.
- *table-layout* — corresponds to the `table-layout` CSS property and determines the algorithm used by the browser to layout the table. The possible values are:
  - *auto* — means the browser will inspect all of the cells in the table to determine the widest unbreakable content in order to set column widths
  - *fixed* — means the browser will use the table and column widths only and not inspect cell contents. While `auto` is the browser default, we've chosen to make `fixed` the default in Maqetta because it performs better and (most importantly) leads to the most predictable results as you place data in the table cells (e.g., column sizes will remain stable and most closely reflect any explicit widths you've set).
  - *inherit* — means the algorithm to use is determined by the parent element
- *First row header* — determines whether the first row in the table should be a header row (i.e., use `<th>` elements instead of `<td>` elements)

If you change any of these values, the preview table in the dialog will automatically update to reflect the change. For our purposes, you can just accept the defaults and click **OK**. After doing so, you will see a 2x2 table in the page editor. Note that if you ever want to change any of these options, you can relaunch this dialog by selecting the table and then double-clicking on it.



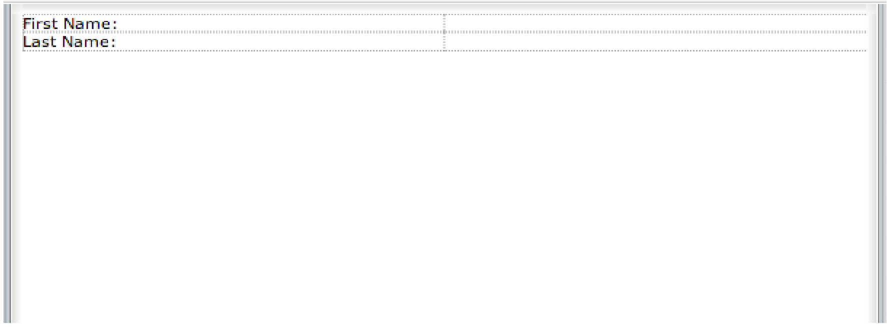
Adding Text and Widgets to Table Cells

Now, let's build on what we have done so far to create the start of a simple name/address form like I alluded to earlier. Let's start by adding some labels in the left column. Double-click on the first cell in the table and a dialog will appear with an editable text box. Type "First Name:" as shown below:

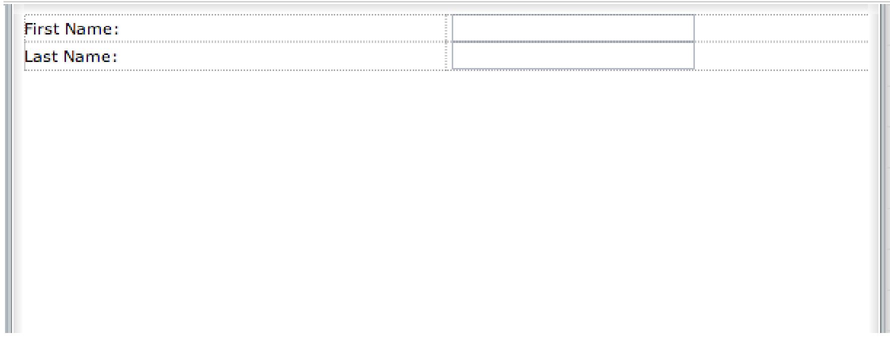

First Name:

When you click **OK**, the text will appear within the cell in the page editor. *NOTE: In Chrome, you may see the column size change when you insert the text into the first cell (despite the fact we have set* `table-layout` *to* `fixed`*). This is a known bug with the page editor in Release 7 of Maqetta. If you are using Chrome and you want the page editor to properly reflect the column width, then you can save, close, and reopen your file.*

Next, double-click on the 2nd cell in the left column, enter "Last Name:", and click **OK**. You should see something like the following in the page editor:


First Name:
Last Name:

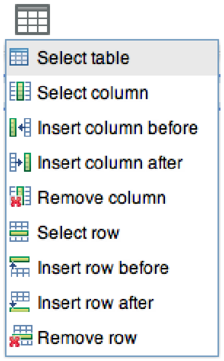
Next, let's add text fields to the right column of the table. Open up the *Dojo Controls* section in the Widget Palette, find the `TextBox` widget, and drag and drop it into the first cell of the 2nd column (the cell just to the right of "First Name:"). Then, repeat for the 2nd cell in the column. After doing so, you should see something like the following in the page editor:



First Name:
Last Name:

Table Commands

In the next few sections, we're going to make further modifications to the table and its children. In order to do so, it's important that you be aware of the *Table commands* menu in the toolbar:



It contains the following actions that become enabled depending on what child elements of the table are selected:

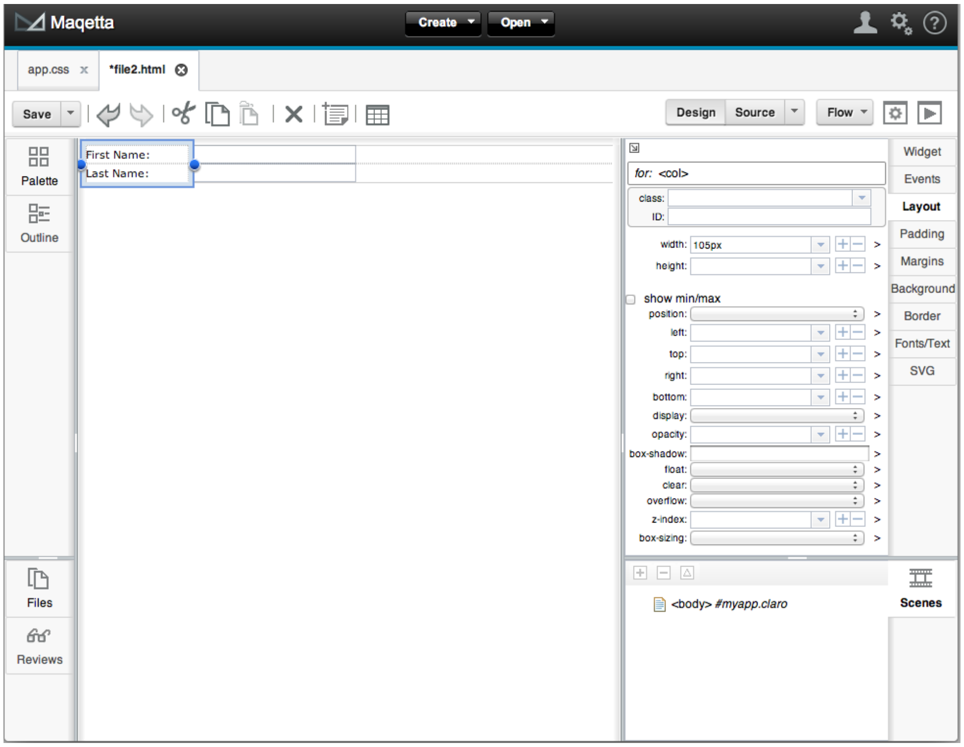
- **Select table** — provides a short-cut to select the parent table of the selected element
- **Select column** — selects the parent column of the selected cell
- **Insert column before** — adds a new column to the left of the currently selected cell or column
- **Insert column after** — adds a new column to the right of the currently selected cell or column
- **Remove column** — deletes the currently selected column (or the parent column if a cell is selected)
- **Select row** — selects the parent row of the selected cell
- **Insert row before** — adds a new row above the currently selected cell or row
- **Insert row after** — adds a new row below the currently selected cell or row
- **Remove row** — deletes the currently selected row (or the parent row if a cell is selected)

Changing Element Properties

Next, let's say we want to change the size of the first column to eliminate some of the white space between the labels and the text fields. To do so, we'll need to select the first column and modify its properties:

1. Select the first cell in the table.
2. From the *Table commands* menu described in the previous section, choose **Select column**.
3. With the column selected, you can make the column smaller by using the resize handles on the left and right sides of the column. You can also use the *Layout* section of the Properties Palette if you prefer to enter an explicit value for the width.

The screenshot below shows how the table might look once you've changed the column size:

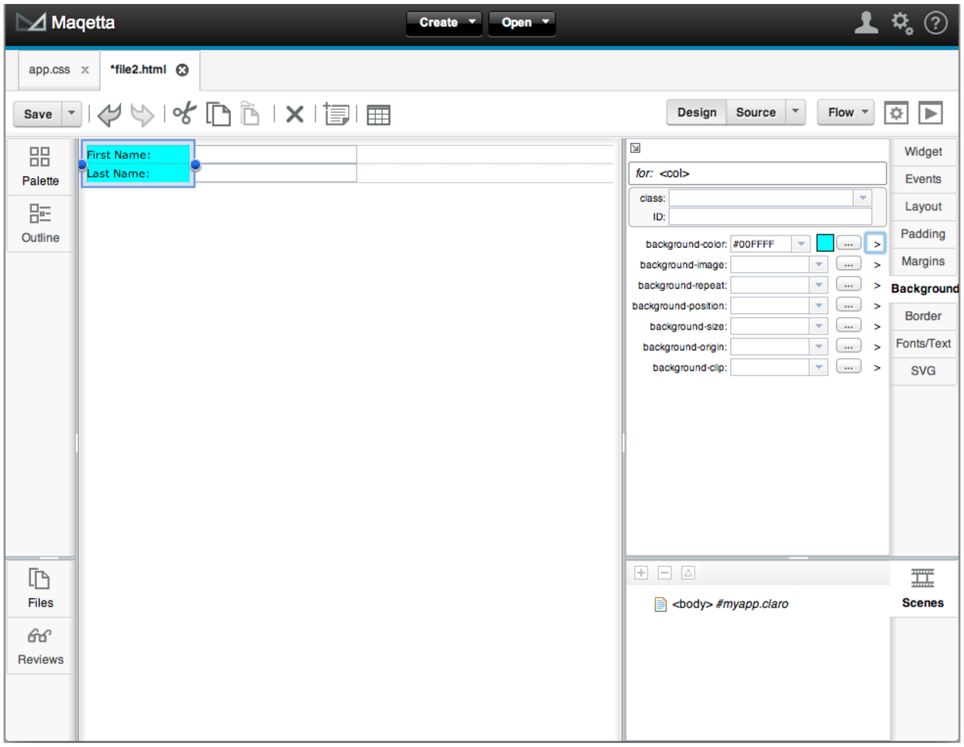


First Name:
Last Name:

for: <col>

width: 105px

You may also wish to change the background color of the first column to better differentiate it from the 2nd column of text fields. In the screenshot below, I selected the column and changed the background color to `#00FFFF` (kind of a light blue) in the *Background* section of the Properties Palette:



First Name:
Last Name:

for: <col>

background-color: #00FFFF

NOTE: Under the hood, as we change column properties the modifications are reflected in the CSS properties of the associated `<col>` *element. If we were to select a row and change its properties, the changes would be reflected in the CSS properties of the associated* `<tr>` *element.*

Adding Table Elements

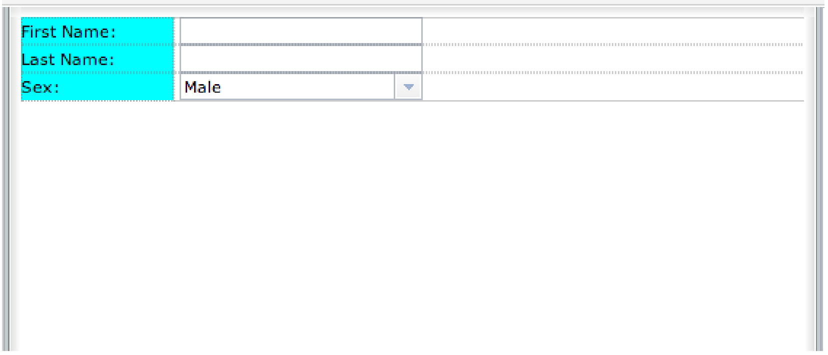
Now, let's say we want to collect additional data in our form. For example, we may want to collect the user's sex. To do so:

1. Select a cell in the 2nd row (for example, the one that says "Last Name:").
2. From the *Table commands* menu described in the last section, choose **Insert row after**. This will insert a new, empty row at the bottom of the table. *Alternatively, we could have opened up the table configuration dialog and changed the row count there.*
3. Double-click on the first cell in the last row, enter "Sex:" into the dialog, and click **OK**.
4. From the *Dojo Controls* section of the Widget Palette, drag and drop a `ComboBox` widget into the cell just to the left of the "Sex:" cell.

5. Enter the following in the input dialog for the `ComboBox` and click **OK**.

```
Male
Female
```

After doing this, your table should look something like the following:



First Name:
Last Name:
Sex: Male

A Word on Cell Padding (Advanced)

At this point, we've touched on all of the main features for creating and manipulating tables in Maqetta, and we've created a nice little table. But, you may also notice that the table contents are a bit crammed together (e.g., the two text fields and the combo box are almost touching). This is because of style rules found in the *document.css* style sheet required by Dojo. These rules cause there to be no spacing on `<td>` elements (even if the `cellpadding` property is set directly on the `<table>` element).

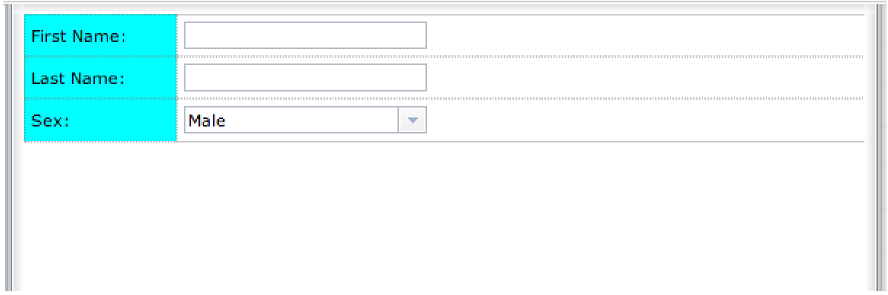
One way to alleviate this (without selecting each cell in the table and manually setting the padding) is to add a general style rule to *app.css*. There is more information about what *app.css* does and how to work with it in the Maqetta documentation, but for the purposes of this article you can do the following:

1. Find *app.css* in the Files Palette and open it by double-clicking on it.
2. Add the style rule below to the *app.css* file:

```
#myapp td, #myapp th {
  padding:5px;
}
```

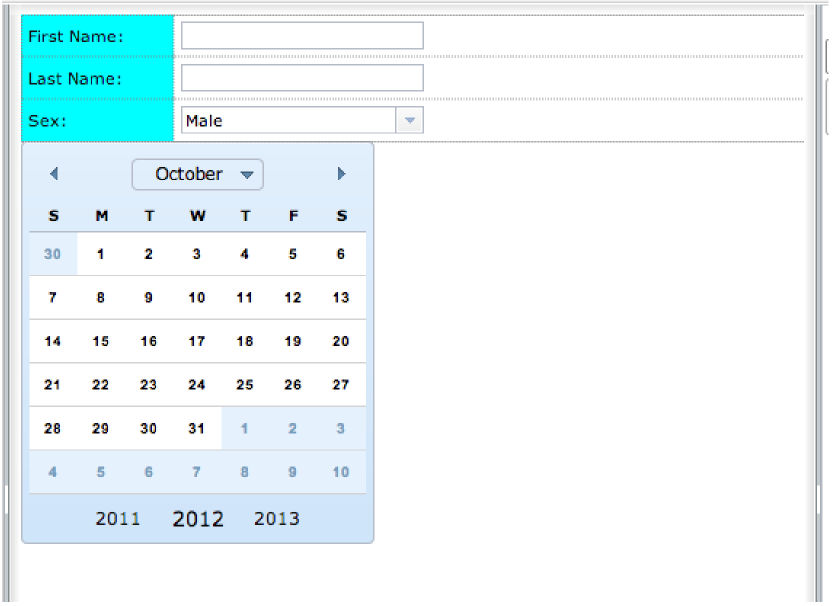
3. Save the *app.css* file.
4. Save and reopen your HTML file so that the *app.css* changes are picked up.

Now, your table should look something like the following in the page editor (with a padding of 5px added to each cell):



First Name:
Last Name:
Sex: Male

Note, however, this is not a perfect solution as this definition interferes with the the styling of Dojo widgets that create `<td>` elements behind the scenes. For example, if you were to add a `Calendar` to the page from the *Dojo Controls* section of the Widget Palette, you'd see it has a lot of extra spacing in it:



First Name:

Last Name:

Sex: Male

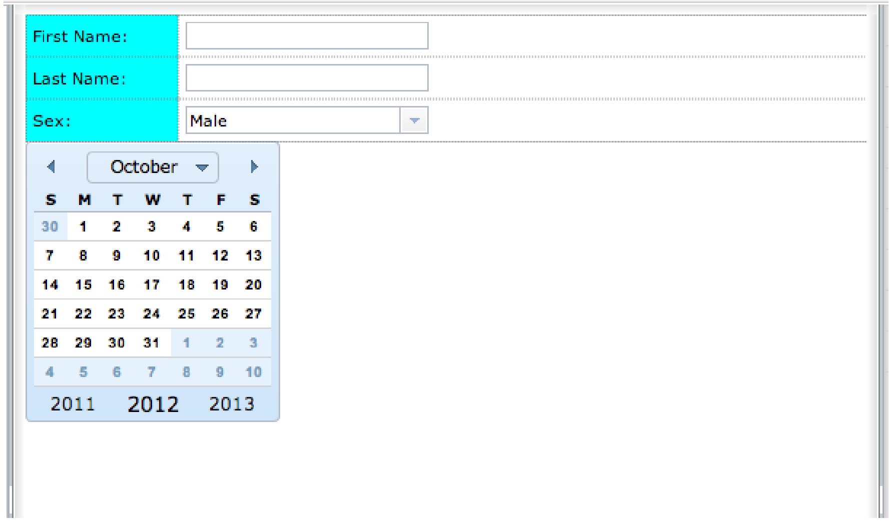
So, if you have a need to use such Dojo widgets in your page, a more robust (but slightly more complicated solution) would be to do the following:

1. Select the table. You can do this by selecting a cell in the table and then choosing **Select table** from the *Table commands* menu in the toolbar.
2. Right-click on the selected table and choose **Surround with <DIV>** from the context menu.
3. In the Properties Palette, change the id of the div to *myTableDiv*.
4. In *app.css*, edit the style rule you added earlier so that instead of being applied to *myapp* (which corresponds to the <body> element) it will be applied to *myTableDiv* (which corresponds to the new div surrounding your table):

```
#myTableDiv td, #myTableDiv th {
  padding:5px;
}
```

5. Save the *app.css* file.
6. Save and reopen your HTML file so that the *app.css* changes are picked up.

After doing so, you will see the calendar (which is outside the *myTableDiv* <div> element) is now rendered correctly while your table cells still have the desired spacing:



First Name:

Last Name:

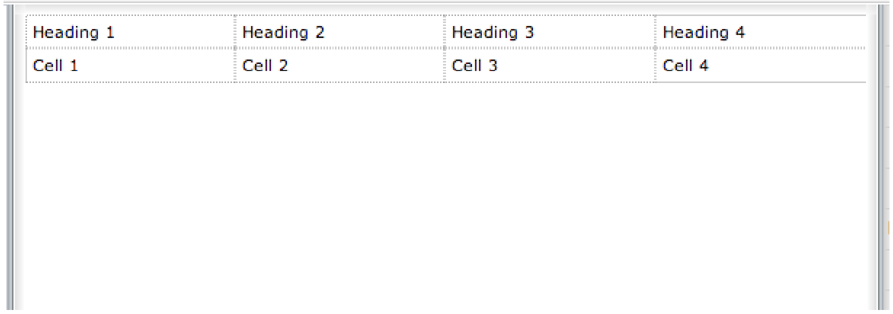
Sex: Male

A Word on Styling the Header Row (Advanced)

While we're on the topic of adding style rules to *app.css*, I wanted to briefly discuss styling of the header row. Like with cell spacing, the *document.css* stylesheet interferes with the styling of <th> elements.

So, for example, to create the table in the screenshot below I did the following:

1. Create a new HTML file.
2. Add a <table> element from the Widget Palette to the file.
3. In the table configuration dialog, set number of columns to 4 and check the box for *First row header*.
4. Enter text into all of the cells (e.g., "Header 1", etc. in the first row and "Cell 1", etc. in the second row).

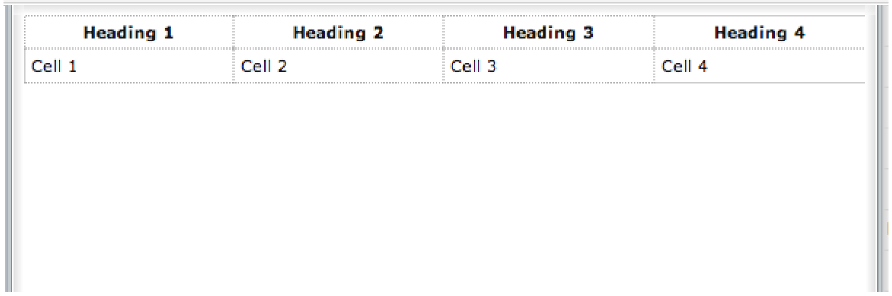


Heading 1	Heading 2	Heading 3	Heading 4
Cell 1	Cell 2	Cell 3	Cell 4

Right away, you probably notice that the styling for the header row is exactly the same as the styling for the second row. To solve this, we can add the following style rule to *app.css*:

```
#myapp th {
        font-weight: bold;
        text-align: center;
}
```

After saving *app.css* and saving/closing/reopening my HTML file, I see the following (with the contents of the header cells bold and centered):



Heading 1	**Heading 2**	**Heading 3**	**Heading 4**
Cell 1	Cell 2	Cell 3	Cell 4

NOTE: The rule is applied to `myapp` (the body) and could instead be applied to a div surrounding the table like I described for padding in the previous section.

Tags: *css prototype maqetta html5 dojo design*

Add a Comment | More Actions

Comments (0)

Add a Comment | ***More Actions***

Previous Entry | Main | Next Entry

About
Help
Contact us
Submit content

Feeds
Follow
Like

Report abuse
Terms of use
IBM privacy
IBM accessibility

Faculty
Students
Business Partners