



# Domino Volt

## Building Apps with Style



# – Things we will cover

---

- Style Editor
- Using Fonts
- Documented Classes
- Custom Classes
- Dynamically setting styles with JavaScript
- Show / Hide Sections
- Navigation – Tricks and Techniques
- Dynamically setting content
- Responsive Web Settings
- Text items and images
- HTML Fragments
- Using a button instead of an Action button

# Style Editor

The screenshot displays the 'Style Editor' application with a 'Theme Editor' dialog box open. The main application window has a menu bar with 'Forms', 'Access', 'Stages', 'Style', 'Events', 'Settings', and 'Validation'. The 'Style' menu is active, showing 'Preview', 'Save', and 'Close' options. The 'Theme Editor' dialog box has a title bar with 'Theme Editor' and a close button. On the left side of the dialog, there are fields for 'Theme name:' (containing 'MyTheme') and 'Theme:' (a dropdown menu set to 'Flat Blue'). Below these are expandable sections: 'General' (with sub-items 'Page Layout', 'Fonts', 'Required Indicator'), 'Buttons', 'Sections', and 'Tabs'. The main area of the dialog is a 'Sample' preview of a form. The form has a blue header 'Sample' and contains several fields: a required 'Name' field with a hint and an error indicator, a required 'Address' field with a hint, a required 'Date of Birth' field with a calendar icon and a hint, and a 'Button' field. Below the form is a tabbed interface with 'Tab 1' and 'Tab 2'. Under 'Tab 1', there is a required 'Employment' field with radio buttons for 'Full Time', 'Part Time', and 'Contract'. At the bottom of the dialog, there are 'Submit', 'Cancel', 'Back', and 'Next' buttons. A 'Preview' button is located at the bottom left of the dialog, and 'OK' and 'Cancel' buttons are at the bottom right. A 'Show CSS' link is visible at the bottom right of the sample preview area.


# Using Fonts

Files

Upload or manage any files in your application.

[Add](#)

Sort by: [Type](#) | [Date](#) | [Reference](#) | [Relationship](#)

▼  **Gotham-Book.woff2**  
Updated on 5/5/20, 4:27 PM

[Copy](#) | [Update](#) | [Download](#) | [Delete](#)

**Theme name:**

**Theme:**

---

▼ General

- > Page Layout
- ▼ Fonts
  - General >
  - Label Font
  - Error Mess
  - Hints >
  - Input Text
  - > Required In
- > Buttons
- > Sections
- > Tabs

Size  px

Color

Family  Default  
 Times New Roman  
 Comic Sans MS  
 Courier New  
 **Gotham-Book**

Gotham-Book

# Documented (& undocumented) classes

[https://help.hcltechsw.com/domino\\_volt/1.0/ref\\_customized\\_css.html](https://help.hcltechsw.com/domino_volt/1.0/ref_customized_css.html)

## ✓ Class Names

### View Responses layout

Class names for the View Responses layout are:

- .lfAppForm
- .lfAppFormArea
- .lfAppMainArea
- .lfAppOverviewArea
- .lfBanner
- .lfFormTitleBar
- .lfMn.lfAppArea

The following graphic shows where the class names for various parts of the **View Responses** section are located.



# When in doubt use the inspector and style editor in your browser

The screenshot displays a web browser window with a form titled "Styling with CSS". The form contains the following fields and buttons:

- First Name
- Last Name
- Street Address
- City
- State (dropdown menu)
- Email Address
- Submit button
- Cancel button

Below the form, a warning message states: "Style sheet could not be loaded." The browser's developer tools are open, showing the "Style Editor" tab. The left sidebar lists several CSS files, with the selected one being an inline style sheet with 6 rules. The main pane shows the following CSS code:

```
1
2
3 .lfMn .dijitPlaceHolder {
4   margin: 10px 0px 10px 1px;
5   padding: 3px 0px 0px 5px;
6   white-space: nowrap;
7   overflow: hidden;
8   color: black;
9   font-style: inherit;
10  font-size: 16px;
11 }
12
13 .lfMn.lotusui30dojo .dijitTextBox {
14   border: none;
15   padding-left: 5px;
16   margin: 10px 0px -5px 0px;
17   font-size: 16px;
```

# Custom Classes

### Edit Single Line Entry Properties

Basic **Advanced** Formula Events

**ID:** ⓘ

**Data Label:** ⓘ

**Accessibility - Alternative label ID:** ⓘ

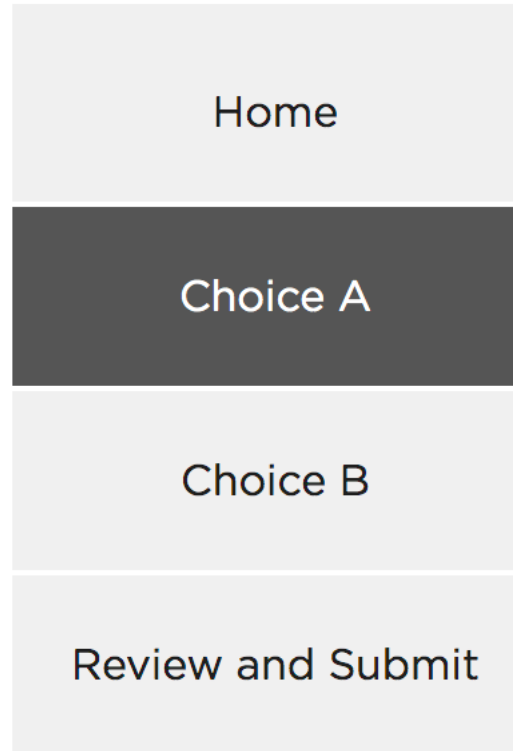
**Show in View Data** ⓘ

**Custom CSS class names:** ⓘ

**Custom Attribute** ⓘ

```
.Orange .dijitTextBox {  
  color: #f5821e;  
}  
  
.Purple .form-field-title-sign-css {  
  color: #7F00FF  
}  
  
.Bold {  
  font-weight: 900;  
}  
  
.Big {  
  font-size: 32px;  
}
```

# Dynamically setting styles with JavaScript



onClick →

```
page.F_Home.addClass('menuClick');  
page.F_ChoiceA.removeClasses('menuClick');  
page.F_ChoiceB.removeClasses('menuClick');  
page.F_ChoiceC.removeClasses('menuClick');
```

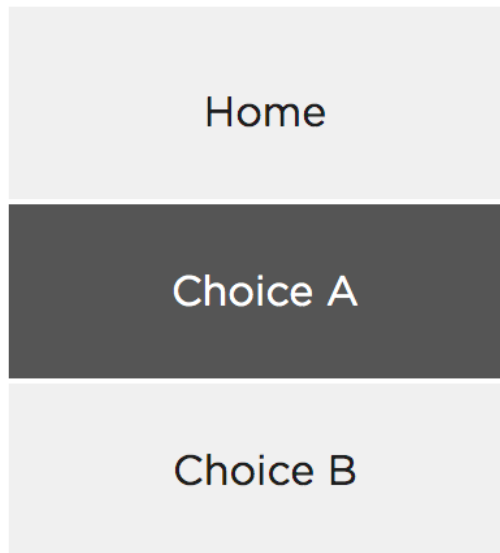
```
.menuClick{  
background-color: #555;  
color: white;  
}
```



# Show / Hide Sections

Page onShow

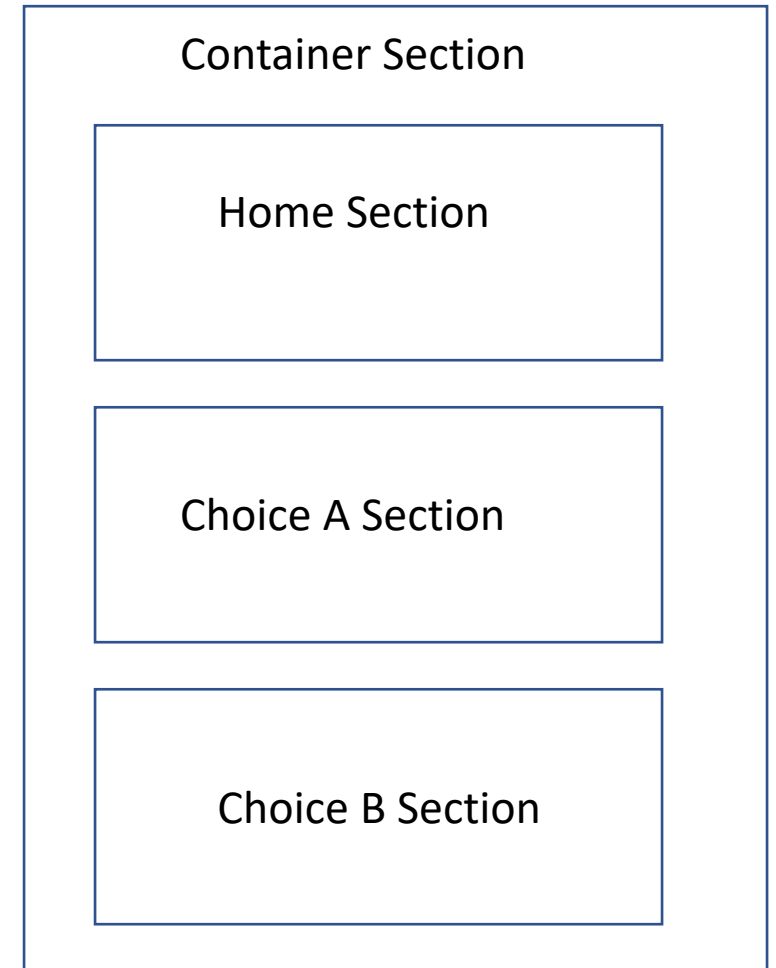
```
page.F_HomeSection.setVisible(true);  
page.F_ChoiceASection.setVisible(false);  
page.F_ChoiceBSection.setVisible(false);
```



```
page.F_HomeSection.setVisible(true);  
page.F_ChoiceASection.setVisible(false);  
page.F_ChoiceBSection.setVisible(false);
```

```
page.F_HomeSection.setVisible(false);  
page.F_ChoiceASection.setVisible(true);  
page.F_ChoiceBSection.setVisible(false);
```

```
page.F_HomeSection.setVisible(false);  
page.F_ChoiceASection.setVisible(false);  
page.F_ChoiceBSection.setVisible(true);
```



# Using a Button instead of an Action Button

Why? So you can put it anywhere you want

## On the Form onShowActionButtons

```
var actions = form.getStageActions();
for(var i=0; i<actions.length; i++)
{
if(get(actions,i).getId() === 'S_Submit')
{
get(actions,i).setVisible(false);
}
}
```

## On the Button onClick

```
var actions = form.getStageActions();
for(var i=0; i<actions.length; i++)
{
if(get(actions,i).getId() === 'S_Submit')
{
get(actions,i).activate();
break;
}
}
```

# - You can always use HTML

```
<div class="sidebar">
  <a id="home">Home</a>
  <a id="choiceA">Choice A</a>
  <a id="choiceB">Choice B</a>
  <a id="reviewSubmit">Review and Submit</a>
</div>
```

```
var home = document.getElementById('home');
home.addEventListener("click", function (event) {
  page.F_HomeSection.setVisible(true);
  page.F_ChoiceASection.setVisible(false);
  page.F_ChoiceBSection.setVisible(false);
  page.F_ChoiceCSection.setVisible(false);
})
```

```
var choiceA = document.getElementById('choiceA');
choiceA.addEventListener("click", function (event) {
  page.F_HomeSection.setVisible(false);
  page.F_ChoiceASection.setVisible(true);
  page.F_ChoiceBSection.setVisible(false);
  page.F_ChoiceCSection.setVisible(false);
})
```

etc...

# Form and Page Navigation

---

## Forms

- Each Form has its own url
- Need to open URL in place
- Links and/or JS required
- Need to be mind full of changing domains and app IDs

## Page

- `form.selectPage()`
- Navigation Item

## – Dynamically setting HTML for navigation purposes

Welcome to Form 2

[< Back](#)

Dynamically set HTML with href

```
var app = app.getUID();  
var url = '/volt-apps/secure/org/app/'  
+ app  
+ '/launch/index.html?form=F_Menu';
```

```
item.setContent('<a style="font-size:16px" href="'  
+ url  
+ '">< Back </a>');
```

# Resources

---

## Domino Volt Wiki

<https://hclwiki.atlassian.net/wiki/spaces/HDV/pages/36503585/Themes+Styles+and+Custom+CSS>

## Domino Volt Documentation

[https://help.hcltechsw.com/domino\\_volt/1.0/ref\\_customized\\_css.html](https://help.hcltechsw.com/domino_volt/1.0/ref_customized_css.html)

## Domino Volt Sample Apps

[https://hclpnpsupport.hcltech.com/community?id=community\\_question&sys\\_id=856e44d01b1c5418c48197d58d4bcb8e&view\\_source=featuredList](https://hclpnpsupport.hcltech.com/community?id=community_question&sys_id=856e44d01b1c5418c48197d58d4bcb8e&view_source=featuredList)