

MADCAP FLARE 2024

Responsive Output

Copyright © 2024 MadCap Software. All rights reserved.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of those agreements. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of MadCap Software.

MadCap Software
9171 Towne Center Drive, Suite 335
San Diego, California 92122
858-320-0387
www.madcapsoftware.com

THIS PDF WAS CREATED USING MADCAP FLARE.

CONTENTS

CHAPTER 1

Welcome	5
---------------	---

CHAPTER 2

Creating a Project From a Template	6
How to Create a New Project From a Template	7

CHAPTER 3

Responsive Skins	8
How to View Responsive Skins and Media Queries	9
How to Create a New Skin	12
How to Associate a Skin With a Target	15

CHAPTER 4

Responsive Conditions	18
How to Apply Responsive Conditions to Content	19
How to Associate Responsive Conditions to a Target	21
How to View Responsive Conditions in Output	22

CHAPTER 5

Responsive Layouts	26
How to Create a Table With Content	27

How to Create a Responsive Layout29
How to Edit the Responsive Layout Style35

APPENDIX

PDFs39
Tutorials39
Cheat Sheets40
User Guides41

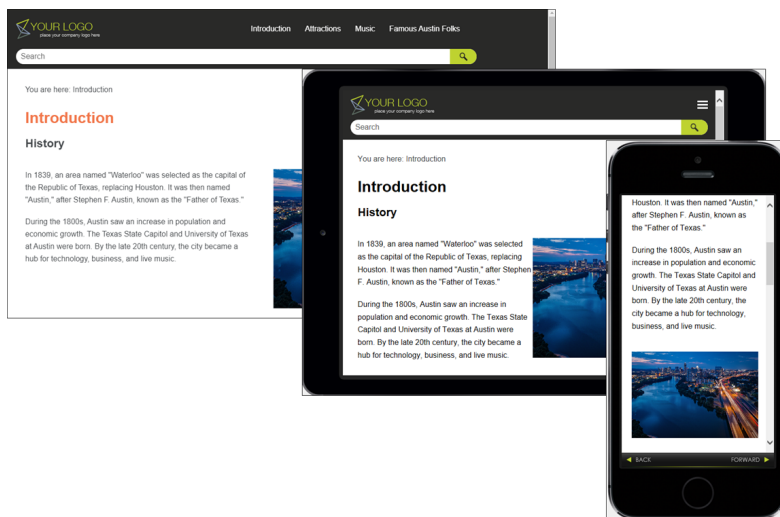
CHAPTER 1

Welcome

Approximate Time: 20 minutes

Welcome to the Flare Responsive Output Tutorial.

- **Tutorial Goal** Learning about responsive framework and content concepts to increase online output usability.
- **Key Concepts** Getting familiar with how responsive skins and media queries affect output, creating responsive content and conditions based on screen size, and interacting with the Responsive Layout window pane for optimizing responsive design.



In another tutorial, we touched on how responsive output is displayed based on CSS style settings, as well as tablet and mobile breakpoints. It is recommended to work through the Using Media Queries tutorial before starting this one.

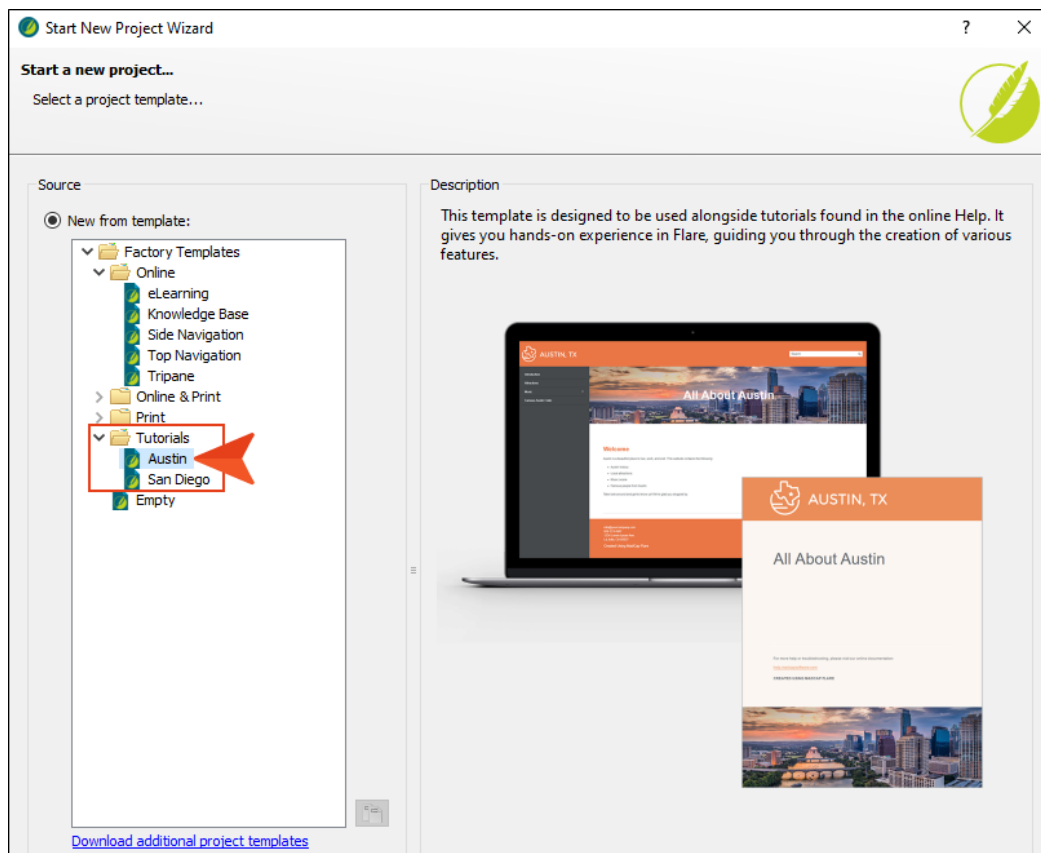
CHAPTER 2

Creating a Project From a Template

To start, let's create a new project based on Flare's Austin template.

I How to Create a New Project From a Template

1. Select **File > New Project**. The Start New Project Wizard opens.
2. In the **Project name** field, give your project any name you want.
3. (Optional) In the **Project folder** field, you can enter a location where you want to save your project. By default, projects are saved in your Documents > My Projects folder.
4. Click **Next**.
5. Under **Factory Templates**, expand the **Tutorials** folder. Select the **Austin** template.




6. Click **Next**.
7. For branding, keep the default settings. (This is where you could customize your project's look with branding, such as colors and logo.) Click **Next**.
8. Click **Finish**. The project is created and loaded into Flare.

Responsive Skins

Responsive content, layouts, and skins all contribute to the responsive design of your output; meaning, the way it renders and how it adjusts automatically for different device sizes. Responsive design techniques use HTML and CSS to increase usability while making your HTML5 output look good on any screen.

Consider the skin when thinking about output. A skin is a file that contains information about the appearance of an online output window. When a skin is responsive, the navigation elements (e.g., menu/table of contents items) are automatically adjusted depending on the size of the screen.

 **NOTE** Responsive output is always enabled for Side and Top Navigation skin types, and skinless output. Keep in mind that the Austin template includes a Side Navigation skin.

This chapter discusses the following:

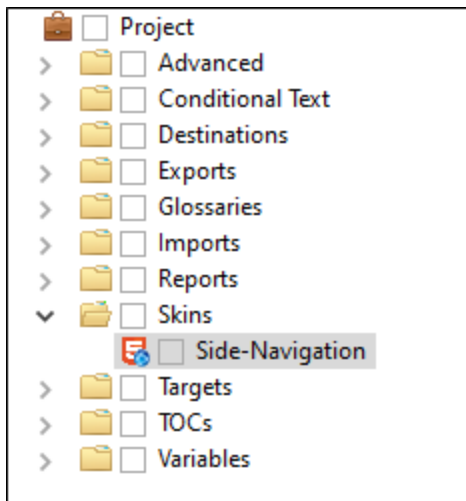
How to View Responsive Skins and Media Queries	9
How to Create a New Skin	12
How to Associate a Skin With a Target	15

I How to View Responsive Skins and Media Queries

To support responsive output, HTML5 skins come equipped with three mediums/media queries—Web, Tablet, and Mobile. Web kind of serves as both a medium and media query in Flare (controlling how skins and content look on a large screen), whereas tablet and mobile are technically both media queries, not mediums. You can use these media queries to specify different looks when necessary for the various devices people might be using to view your output.

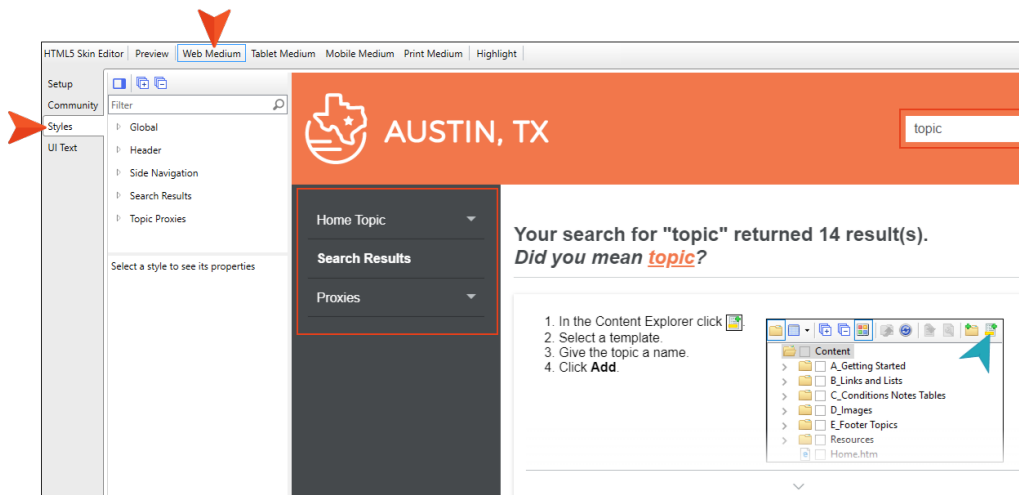
 **NOTE** Sometimes the terms mediums and media queries are used interchangeably.

1. From the Project Organizer, expand the **Skins** folder, and double-click **Side-Navigation**.

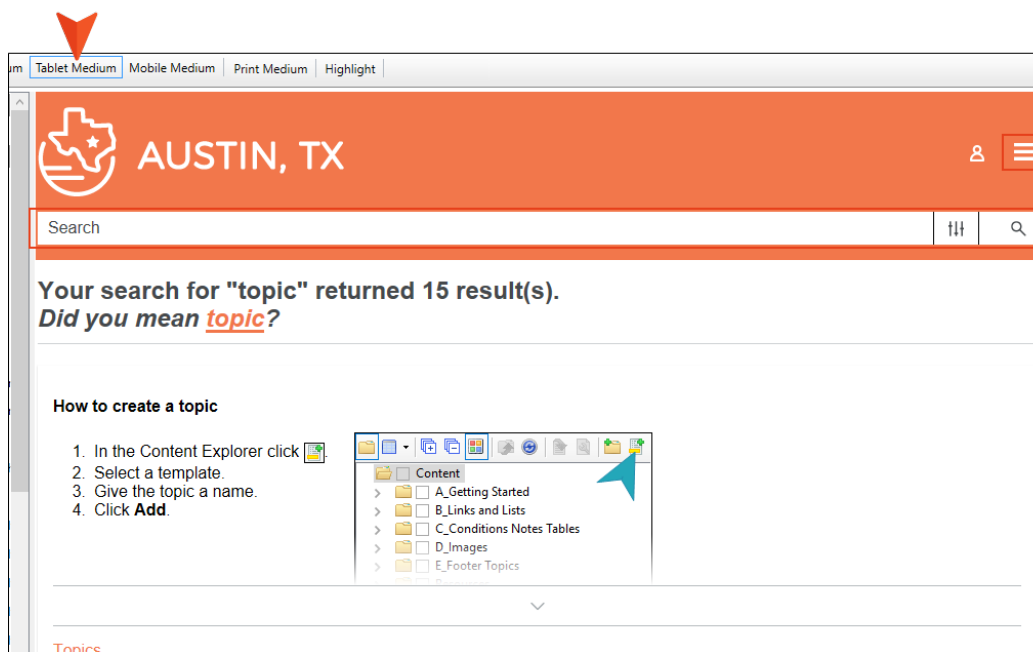


2. In the HTML5 Skin Editor, select the **Styles** tab (if not already selected).

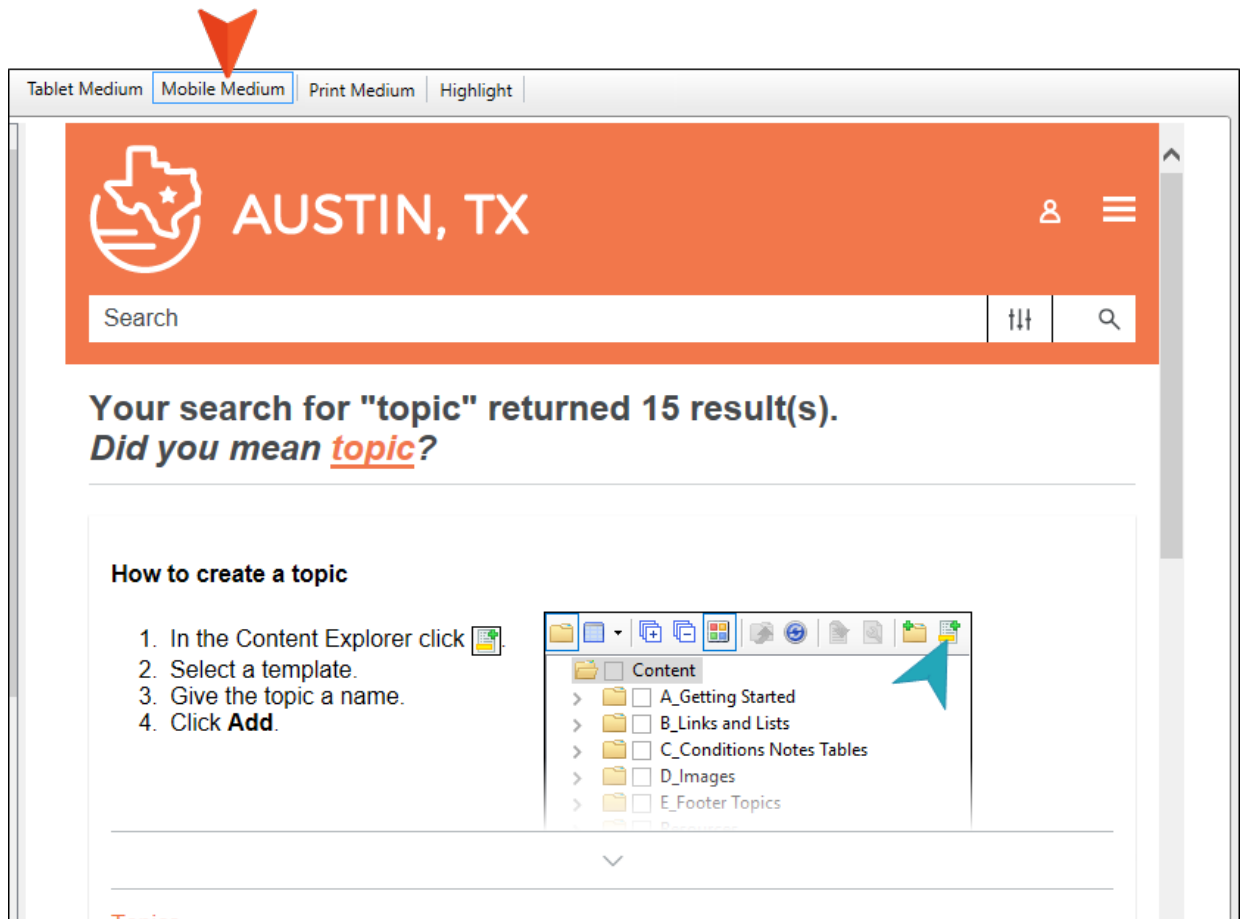
3. From the local toolbar, the default **Web Medium** should be selected. This is what you will see if viewing output on a large screen. In the preview area to the right, notice the menu on the left side, and the search bar in the upper-right corner.




4. Select **Tablet Medium**. This is what you will see if viewing output on a tablet, or when the width of the window is narrower and reaches a certain breakpoint. Notice the decreased width, the flyout menu in the upper-right corner, and the repositioning of the search bar.



5. Select **Mobile Medium**. This is what you will see if viewing output on a mobile device, or when the window gets narrow enough for the mobile breakpoint. Notice the decreased width.

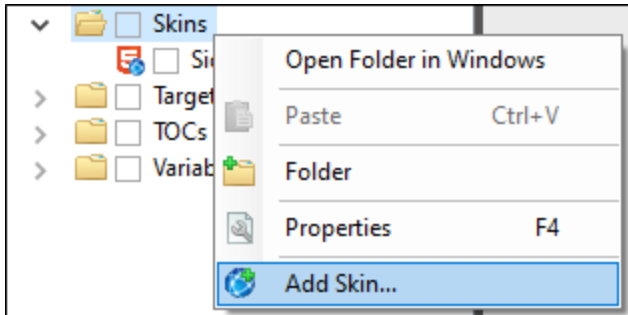


 **NOTE** In the HTML5 Skin Editor, the user interface displays a Print medium button next to the Web, Tablet, and Mobile mediums. The main purpose of the Print medium is for controlling how the skin looks when printed from an online source. The Print medium in the skin is its own entity, separate from the other mediums, and it does not share the same purpose as mediums in the stylesheet or in responsive output.

I How to Create a New Skin

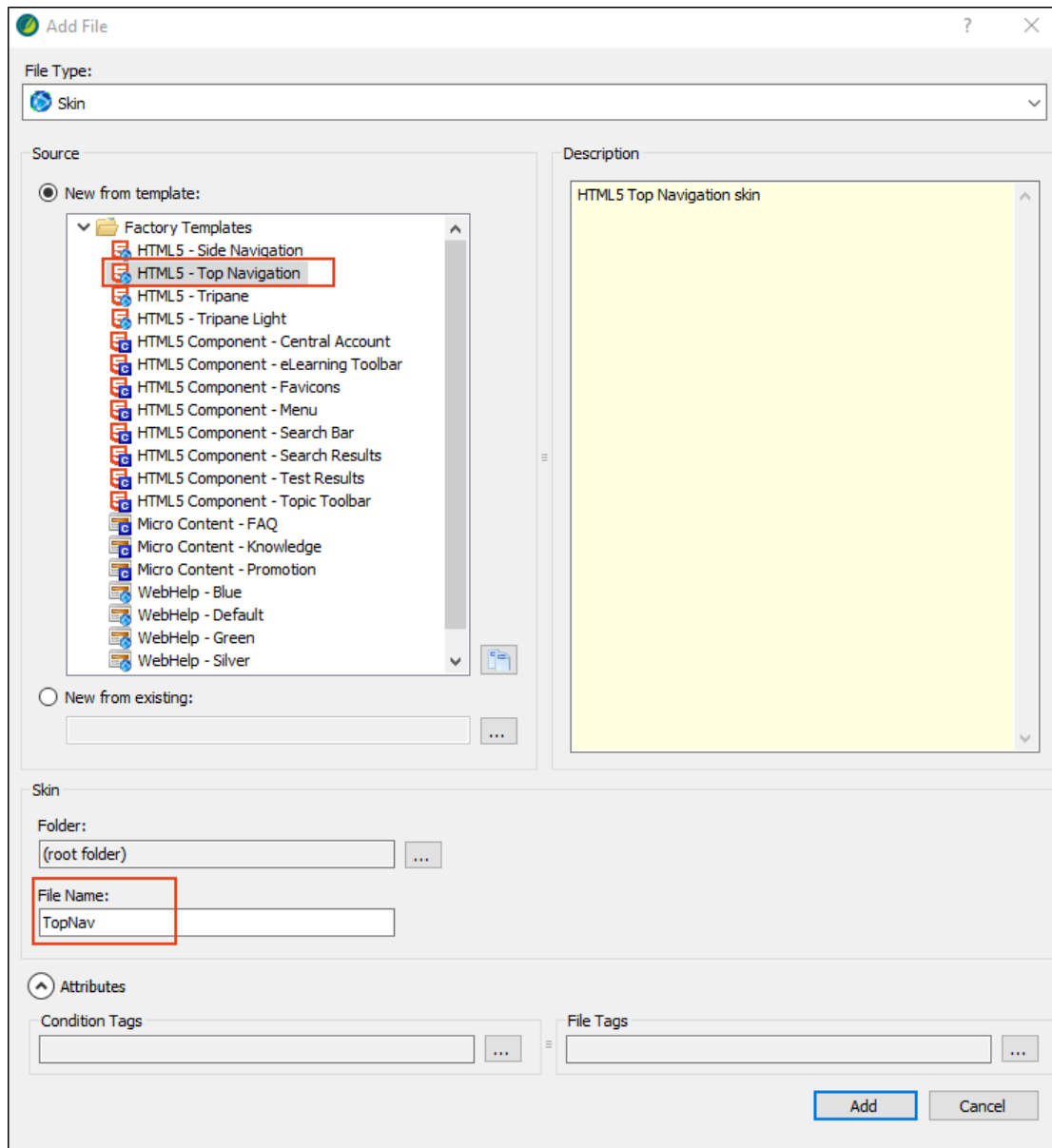
For the Austin project, let's create a new skin. Like the Side Navigation skin, the Top Navigation skin is frameless, flexible, and looks modern (where the main menu is positioned at the top).

1. From the Project Organizer, right-click the **Skins** folder.
2. From the context menu, select **Add Skin**. The Add File dialog opens.



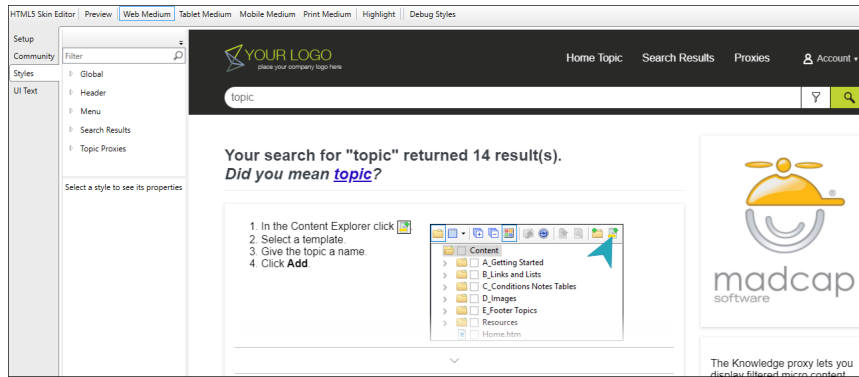
3. In the **Source** area, select **New from template**, and select **HTML5 – Top Navigation**.

4. In the **Skin** area, in the **File Name** field, type `TopNav`.

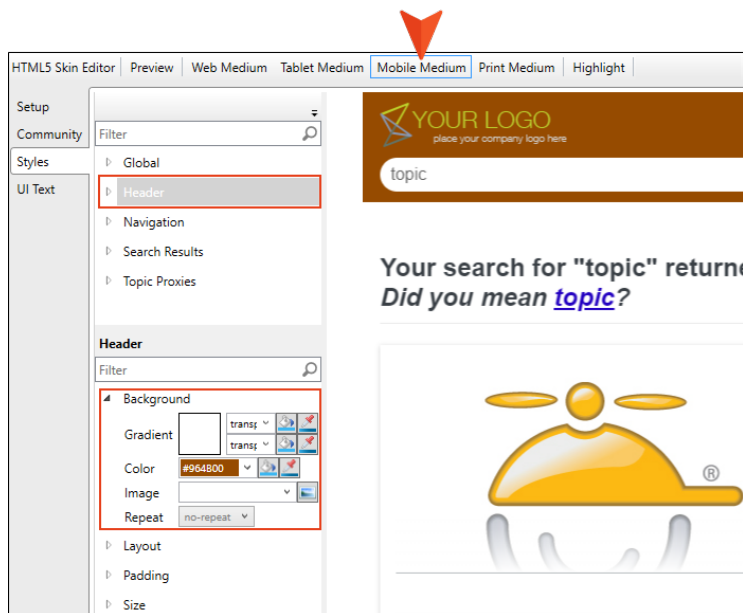


5. Click **Add**. The new skin is added to the Skins folder, and it opens in the HTML5 Skin Editor.

6. With the **Styles** tab selected, click through the different mediums to get familiar with the new skin.



7. From the local toolbar, click **Mobile Medium**.
8. Let's make a quick style change to the skin. Select the **Header** style group, and expand the **Background** property.
9. In the **Color** field, type #964B00. (This will make the header background for the mobile medium brown.)

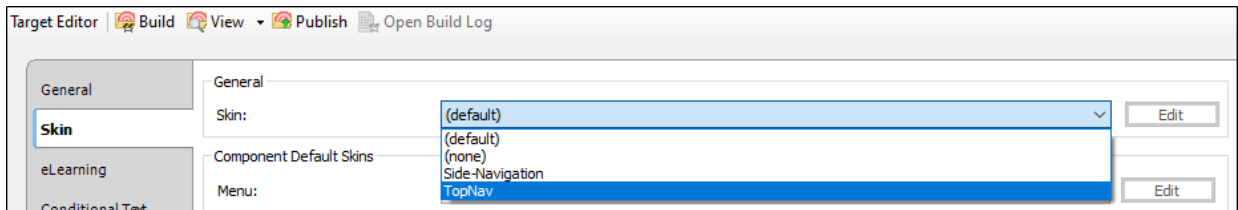



10. Click  to save your work.

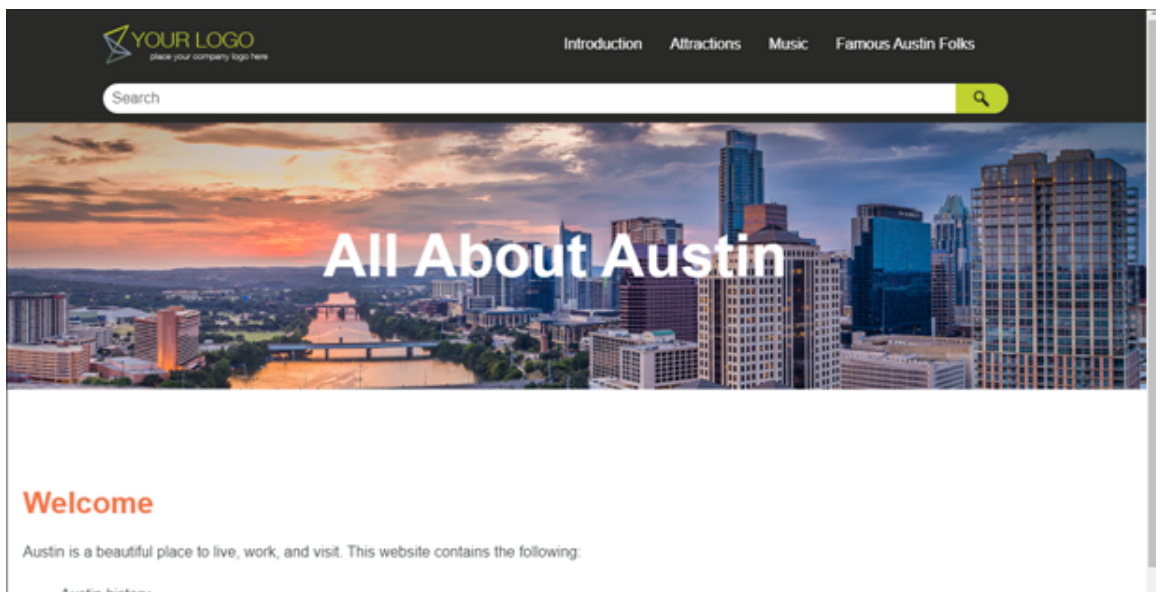
I How to Associate a Skin With a Target

In order to see a new skin in generated output, you can associate it with the target.

1. From the Project Organizer, expand **Targets**, and double-click **All-About-Austin-HTML5**.
2. Select the **Skin** tab.
3. In the **General** area, from the **Skin** drop-down, select **TopNav**.




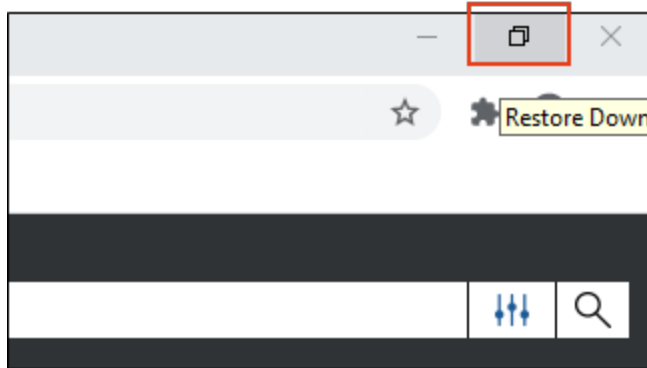
4. Click  to save your work.
5. Select the **Project** ribbon. Then from the **Build Primary** drop-down, select **Build All-About-Austin-HTML5**. The Builds window pane opens at the bottom of the Flare interface, and the target begins to generate. When the build finishes successfully, the Build Progress cell turns dark green.
6. Double-click the row to open the HTML5 output, and view it in a regular browser with the window maximized. Notice the new look that the TopNav skin design provides.




7. Click and drag the browser window to reduce it. Notice the header turns from black to brown when it reaches the responsive setting breakpoint for the mobile screen size.



 **NOTE** With a maximized window, you might have to click the **Restore Down** icon first, in order to click and drag to reduce the size of the browser window.



 **NOTE** In the **Target Editor > Skin** tab, you can view the default Responsive Output Settings; that is, the Tablet Breakpoint, and the Mobile Breakpoint.

CHAPTER 4

Responsive Conditions

Depending on the output type, you can make your content responsive to different devices and screen sizes by using responsive conditions and responsive layouts, in conjunction with media queries.

For HTML5 targets, conditions can be used to make content responsive (e.g., content changes automatically according to the size of the screen, rather than simply based on “include” or “exclude”). Associating conditions with factory media queries (Web, Tablet, Mobile) can help make some content more appropriate or easier to read on different devices.

This chapter discusses the following:

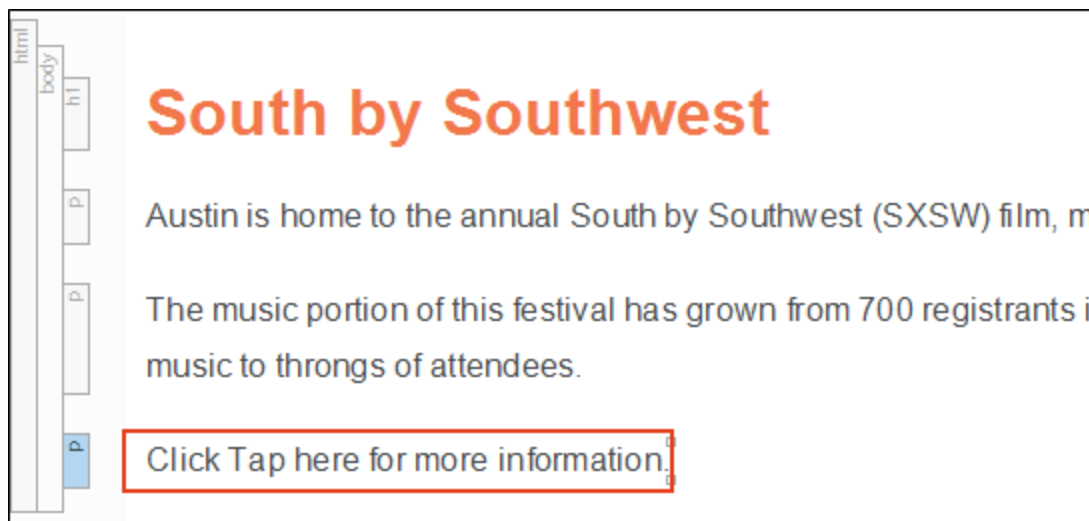
How to Apply Responsive Conditions to Content	19
How to Associate Responsive Conditions to a Target	21
How to View Responsive Conditions in Output	22


How to Apply Responsive Conditions to Content

Let's create content in a topic as an appropriate sample of responsive content.

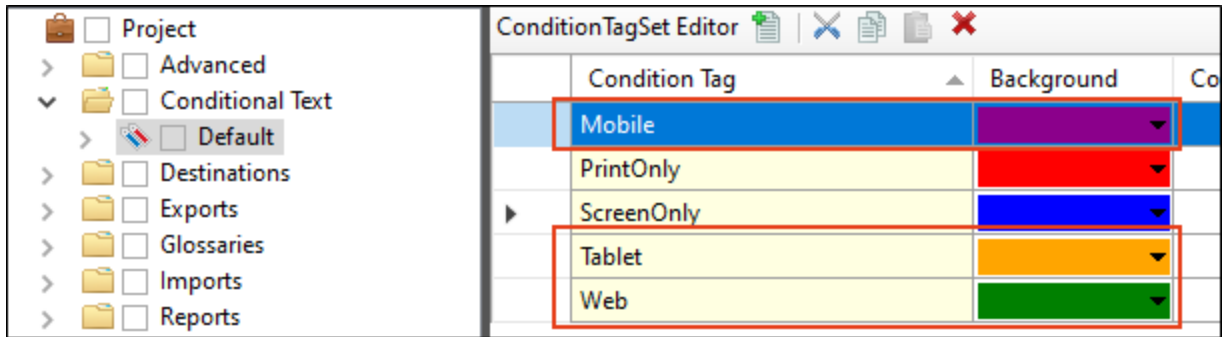
1. From the Content Explorer, double-click **South-by-Southwest.htm**.
2. At the end of the existing content, create an empty paragraph.
3. Copy and paste this example text to the line:

`Click Tap here for more information.`



 **NOTE** Because you can interact with UI in several ways, it is acceptable to change your terminology. For example, using “click” makes sense for larger screens, and using “tap” makes sense for tablets or mobile screens.

- From the Project Organizer, expand the **Conditional Text** folder, and then double-click **Default**. The project's default conditions open in the Condition Tag Set Editor. You can see that condition tags already exist for the Web, Tablet, and Mobile media queries.



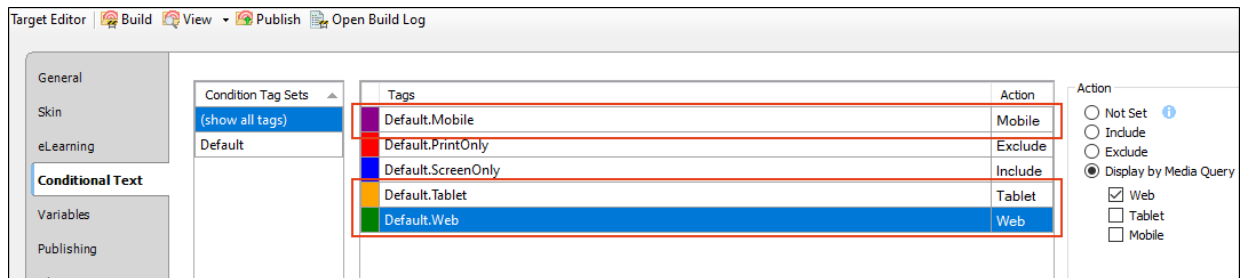
- Go back to the **South by Southwest** topic.
- Select the word **Click** plus one space beyond it, and then right-click.
- From the context menu, select **Conditions**.
- From the Condition Tags dialog, select **Web**. Then click **OK**.
- Do the same for the word **Tap**, but instead apply both the **Tablet** and **Mobile** conditions to it. You should end up with the following conditions applied in the topic.




- Click  to save your work.
- Close the Condition Tag Set Editor and the South by Southwest file.

I How to Associate Responsive Conditions to a Target

1. Open the All-About-Austin-HTML5 target.
2. In the Target Editor, select the **Conditional Text** tab.
3. In the **Tags** area, select the **Default.Mobile** tag.
4. In the **Action** area, select **Display by Media Query**, and then select **Mobile**.
5. In the **Tags** area, select the **Default.Tablet** tag.
6. In the **Action** area, select **Display by Media Query**, and then select **Tablet**.
7. In the **Tags** area, select the **Default.Web** tag.
8. In the **Action** area, select **Display by Media Query**, and then select **Web**. The tags should look like this when you are done.



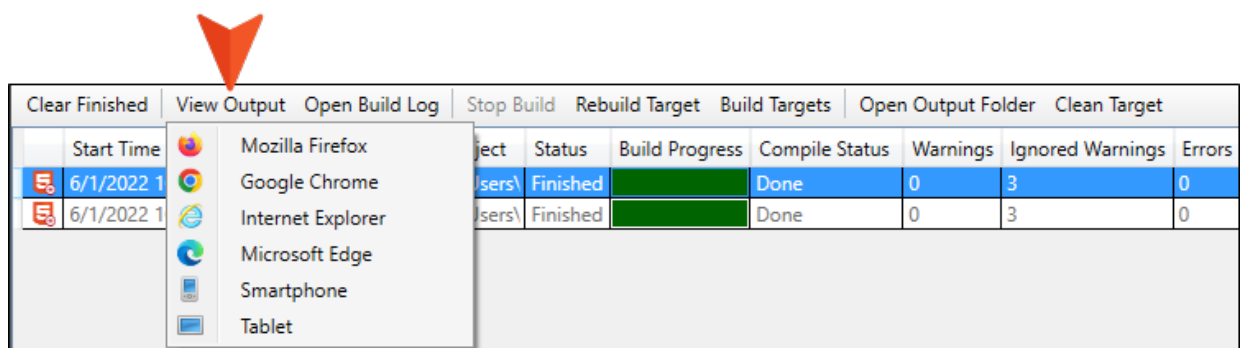
 **NOTE** A condition tag set is used to hold condition tags you create for your project. Flare provides you with an initial condition tag set, which contains some condition tags to help get you started. You can rename the default condition tag set or any others that you add to your project.

9. Click  to save your work.

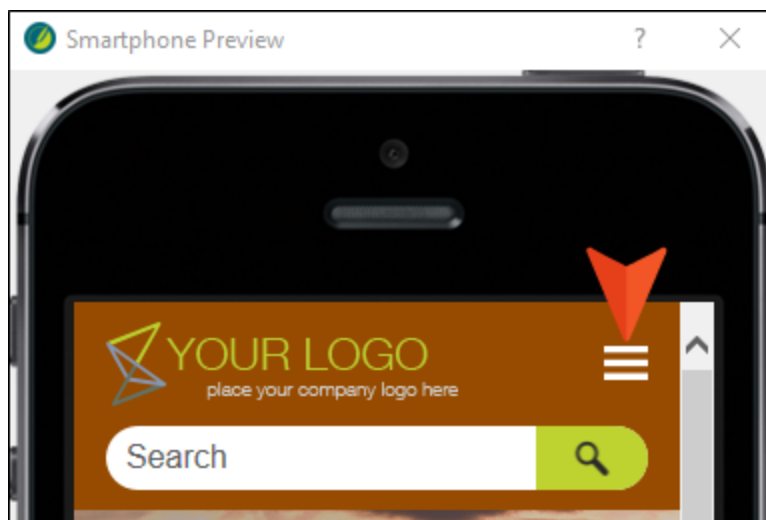
How to View Responsive Conditions in Output

Responsive output works by automatically changing the display once the viewer reaches a certain width.

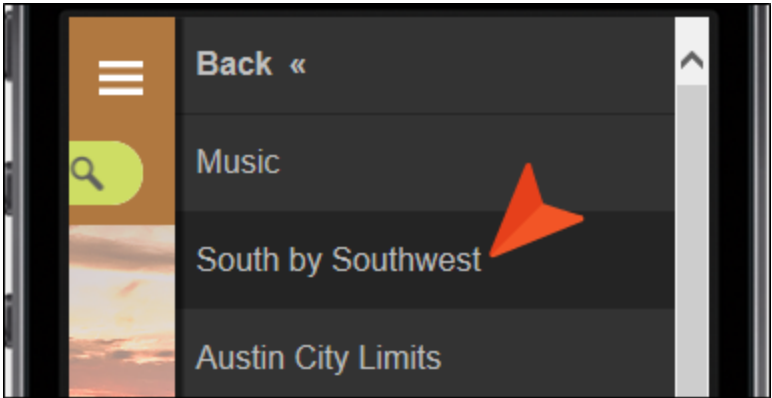
1. Select the **Project** ribbon. Then from the **Build Primary** drop-down, select **Build All-About-Austin-HTML5**. The Builds window pane opens at the bottom of the Flare interface, and the target begins to generate. When the build finishes successfully, the Build Progress cell turns dark green.
2. Select the build row, and from the **Builds** local toolbar, click **View Output**.



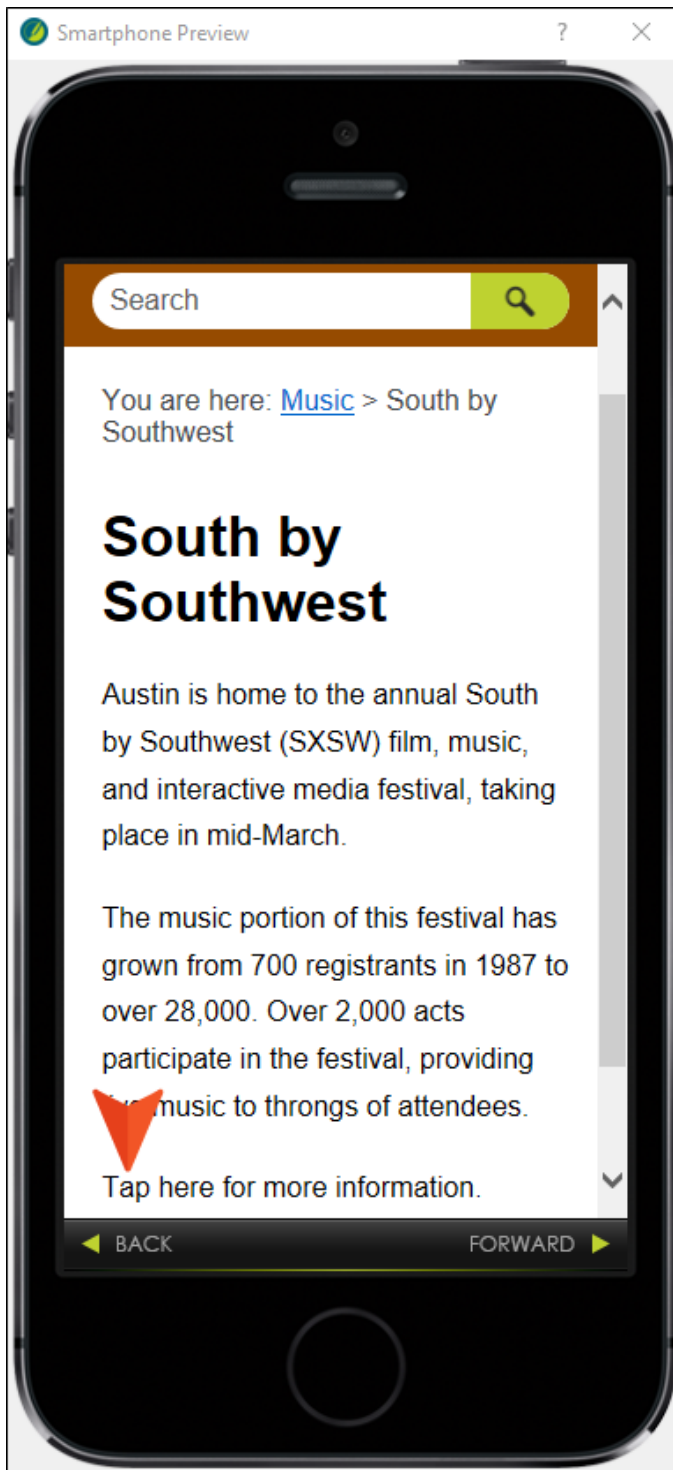
3. From the context menu, select **Tablet** or **Smartphone**. Since the device screen size is smaller, navigate to the main menu by clicking the icon in the upper-right corner.



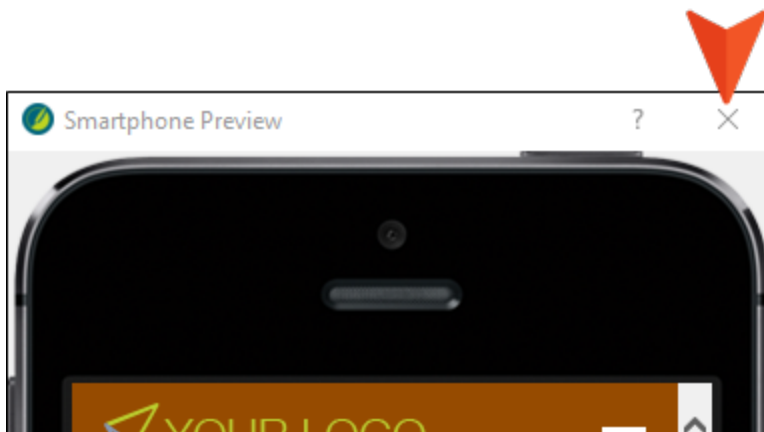
4. From the flyout menu, select the **South by Southwest** topic.



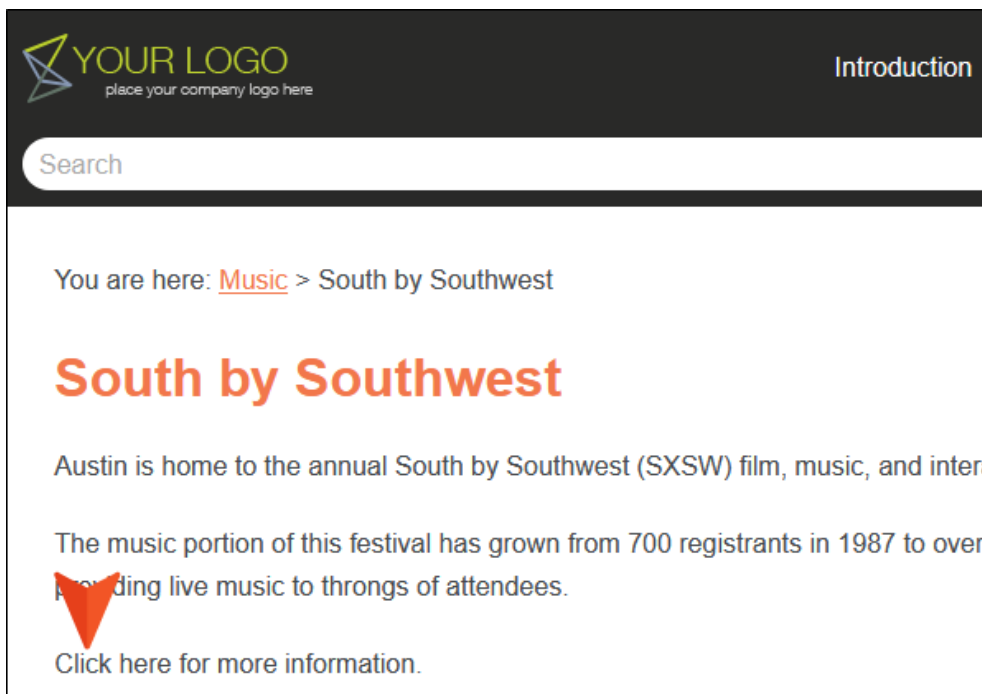
5. Notice the word “tap” appears for the smaller screen sizes.



6. Close the Smartphone (or Tablet) Preview window pane.



7. From the Builds window pane, double-click the build row to open the HTML5 output, and view it in a regular browser with the window maximized.
8. Select **Music**, and then choose **South by Southwest**. Notice the word "click" appears when viewing at a larger screen size.



Responsive Layouts

Styles and media queries can be used to make your content responsive in HTML5 output. This lets you present information—both its substance and structure—differently depending on the size of the screen or device. Flare provides a Responsive Layout window pane that helps you create this kind of content more easily. You also have the option of using third-party solutions (e.g., Zurb Foundation grid system) by adding the appropriate styles in your stylesheets and topics.

For this tutorial, let's focus on Flare's responsive layout system.

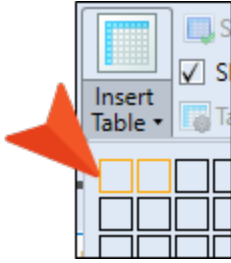
This chapter discusses the following:

How to Create a Table With Content	27
How to Create a Responsive Layout	29
How to Edit the Responsive Layout Style	35

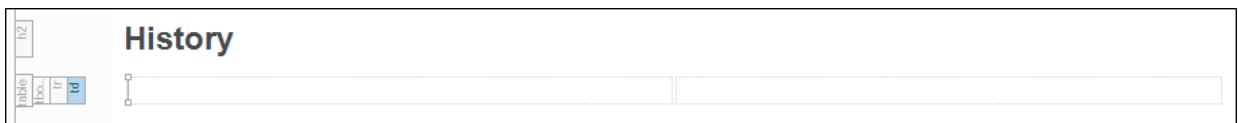
How to Create a Table With Content

First, let's create some content in a table to illustrate why a responsive layout is valuable, and why using a table might not meet your needs.

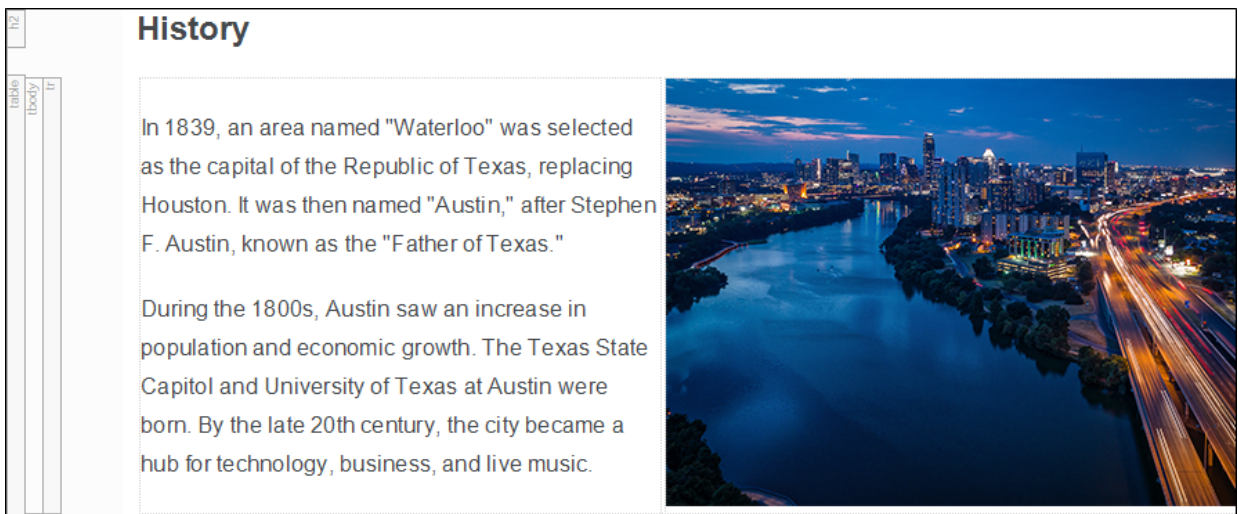
1. From the Content Explorer, open the **Introduction** topic.
2. After the first h2 tag (History), create an empty paragraph.
3. From the **Table** ribbon, click **Insert Table** (down-arrow).
4. Hover over the grid to choose one row and two columns.




The table should look like this.




5. Copy the image, and paste it into the second column. And then copy the two paragraphs of text, and paste them into the first column.



6. To clean up the topic, delete the original image and text.
7. Click  to save your work.
8. Build the HTML5 output, and open it in a browser.
9. From the main menu, select the **Introduction** topic.
10. Click and drag the browser window to reduce it. In this example, the image gets smaller as the screen size is reduced. Ideally, the image size should stay the same, and stack underneath the text when the screen size is reduced.

In 1839, an area named "Waterloo" was selected as the capital of the Republic of Texas, replacing Houston. It was then named "Austin," after Stephen F. Austin, known as the "Father of Texas."

During the 1800s, Austin saw an increase in population and economic growth. The Texas State Capitol and University of Texas at Austin were born. By the late 20th century, the city became a hub for technology, business, and live music.

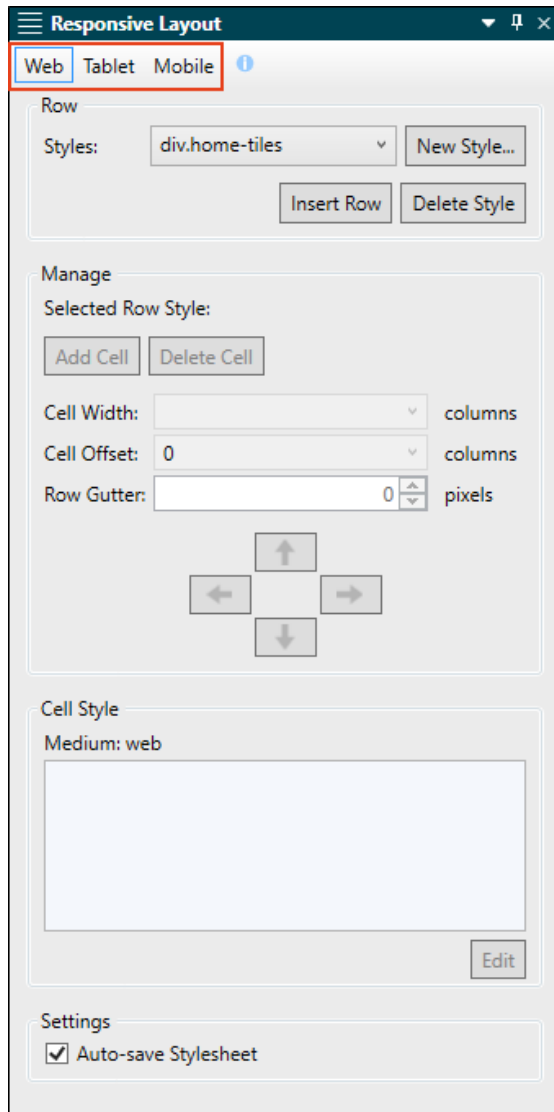


11. Close the HTML5 output.

How to Create a Responsive Layout

Let's create a responsive layout using the Responsive Layout window pane. This window pane lets you create a "one-row grid" to hold content. The benefit of putting your content into such a grid is that it allows it to shift and be responsive to screens of different sizes.

1. Go back to the **Introduction** topic. Place the cursor at the end of the **History** heading.
2. From the **Home** ribbon, select **Responsive Layout**. The Responsive Layout window pane opens. Notice the Web, Tablet, and Mobile media query buttons at the top.



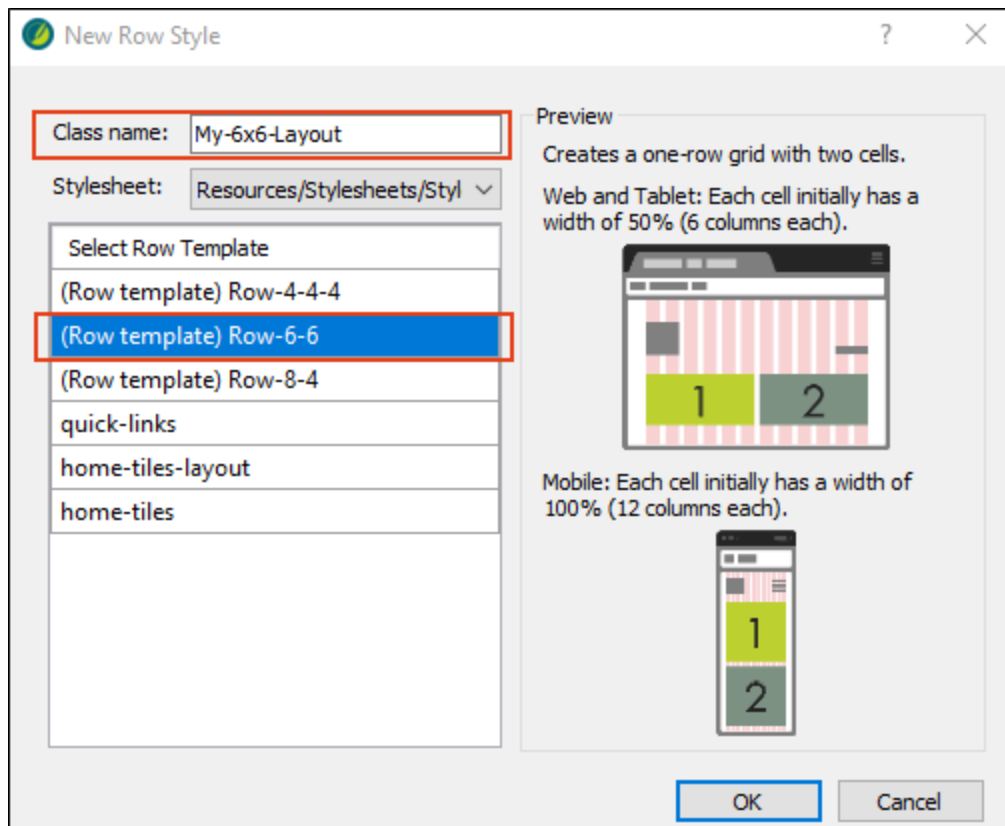
3. Make sure **Web** is selected.

NOTE It is recommended that you start with Web to edit a layout for large screens. After you finish those settings, you can move on to Tablet, and then finally Mobile.

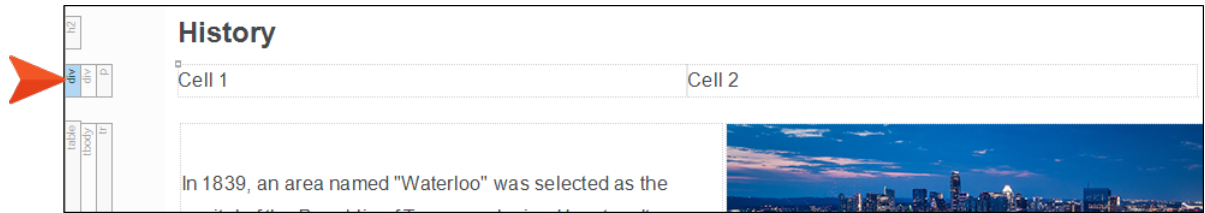
4. In the **Row** area, click **New Style**. The New Row Style dialog opens.

Flare considers the entire width of the screen displaying your output, and it divides this into 12 imaginary columns. The maximum width is 100% of the space, so 12 columns equals 100%. Half the width of the available screen space is 50%, or 6 imaginary columns. One-third of the width is 33.333%, or 4 columns. It is no more complicated than that. For more information see the online Help.

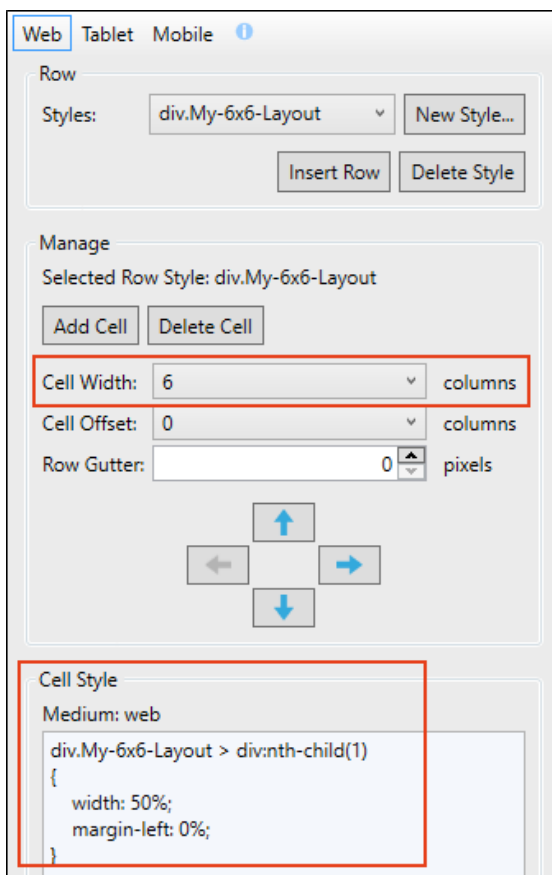
5. In the **Class Name** field, type `My-6x6-Layout`.
6. In the **Select Row Template** area, select **(Row template) Row-6-6**. Click **OK**.



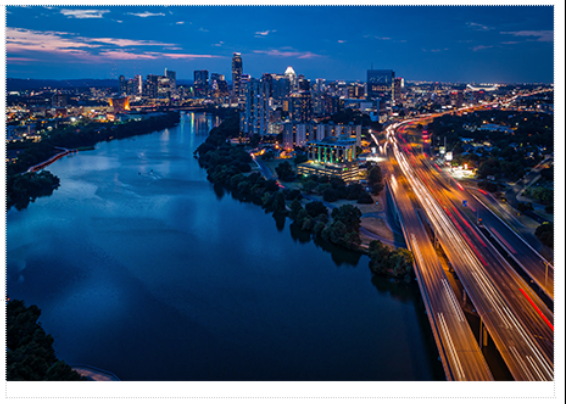
- From the Responsive Layout window pane, click **Insert Row**. In the XML Editor, the new row (a div style) gets inserted after the History heading. Click in **Cell 1**. In the Responsive Layout window pane notice that the **Cell Width** is 6 columns. This equates to 50%, or half the maximum width for a row.




When you create a responsive layout in Flare, you are asked to create a new "row" style, or use an existing one. It's called a row style because it controls everything that is going on inside that one-row grid. However, in CSS it is actually a div style. In the **Cell Style** area at the bottom of the Responsive Layout window pane, notice the div style, `div.My-6x6-Layout`.

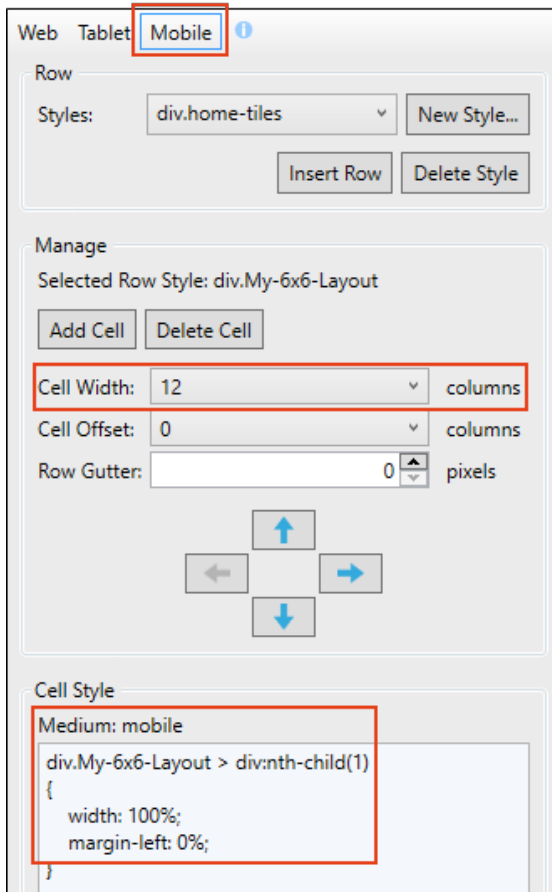


- From your previous table, copy the image, and paste it into **Cell 2**, and then copy the corresponding text paragraphs into **Cell 1**. (Replace the default cell text in the layout grid.)

<h3>History</h3> <p>In 1839, an area named "Waterloo" was selected as the capital of the Republic of Texas, replacing Houston. It was then named "Austin," after Stephen F. Austin, known as the "Father of Texas."</p> <p>During the 1800s, Austin saw an increase in population and economic growth. The Texas State Capitol and University of Texas at Austin were born. By the late 20th century, the city became a hub for technology, business, and live music.</p>	
---	--

- To clean up the topic, delete the table.
- Click  to save your work.
- Place the cursor in one of the new cells. (If your XML Editor is narrow, the two cells might stack one of top of the other. That is okay.)

- From the Responsive Layout window pane, click the **Mobile** media query button. Notice the **Cell Width** is 12 columns (rather than 6), and the **Cell Style** displays the width as 100% (rather than 50%). Because of the Mobile breakpoint and column size, the text and image should stack when the output is generated. (If you click on the Tablet media query button, notice that it has the same setup as the Web media query.)




- Build the HTML5 output, and open it in a browser.
- From the main menu, select the **Introduction** topic.

15. Click and drag the browser window to reduce it. Compared to the previous table, the new style in the responsive layout adjusts better as the screen gets narrower. The image retains its size and the content stacks at the appropriate breakpoint.

History

In 1839, an area named "Waterloo" was selected as the capital of the Republic of Texas, replacing Houston. It was then named "Austin," after Stephen F. Austin, known as the "Father of Texas."

During the 1800s, Austin saw an increase in population and economic growth. The Texas State Capitol and University of Texas at Austin were born. By the late 20th century, the city became a hub for technology, business, and live music.



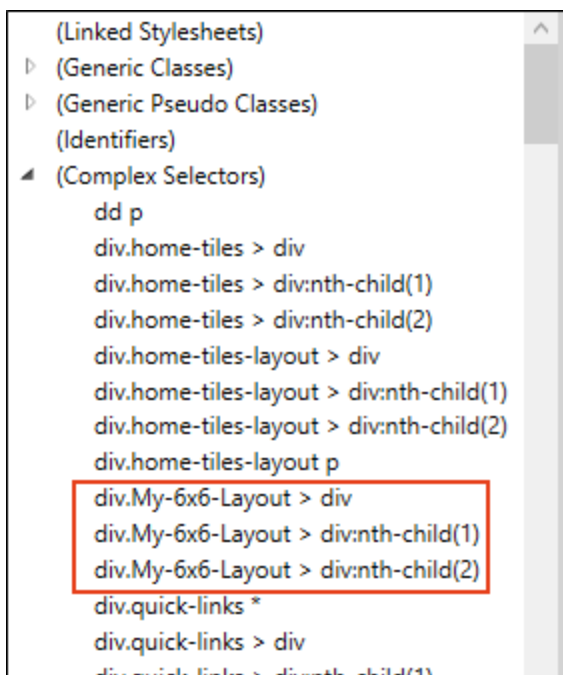
16. Close the HTML5 output.

I How to Edit the Responsive Layout Style

Let's look at the new row style for the Web media query in the stylesheet.

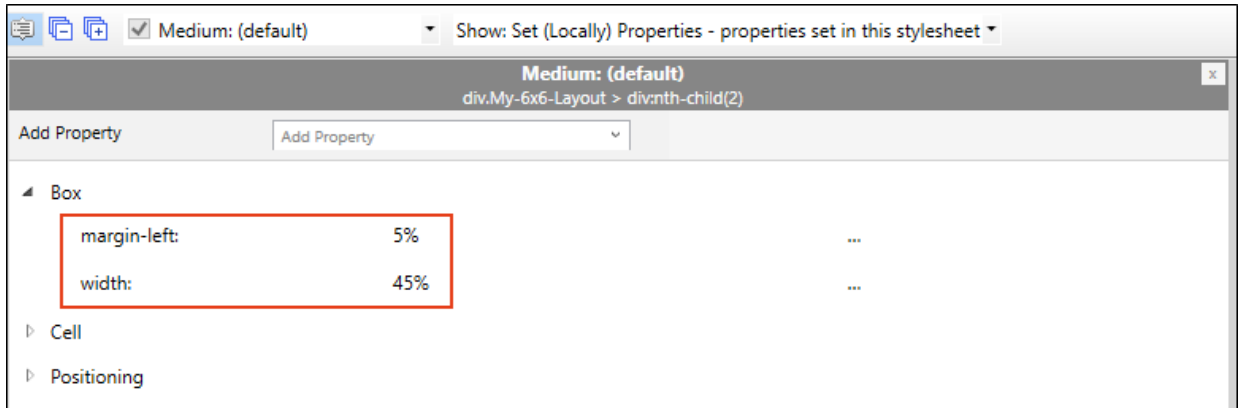
1. Click the **Styles.css** tab. (The Stylesheet Editor automatically opened when the new row style got added.)
2. In the left pane tree structure, expand the **div** element. Under it you'll see the class **My-6x6-Layout**.
3. Scroll to the top of the left pane structure tree, and expand **(Complex Selectors)**.


When you created the responsive layout, multiple complex selectors were added to the stylesheet. These selectors consist of **div** styles, using the **nth-child** pseudo class for each cell. In our case, the container holding both the cells is named **div.My-6x6-Layout > div**, Cell 1 is named **div.My-6x6-Layout > div:nth-child(1)**, and Cell 2 is named **div.My-6x6-Layout > div:nth-child(2)**.



4. Let's make a formatting adjustment to the second cell (Cell 2). Select **div.My-6x6-Layout > div:nth-child(2)**.
5. In the local toolbar, select from the **Show** drop-down, select **Show: Set (Locally) Properties - properties set in this stylesheet**.

- Expand the **Box** group (if in the Grouped view).
- Change the **margin-left** property to **5%**, and the **width** property to **45%**. Click off the field.



- Click  to save your work.
- Go back to the **Introduction** topic.

10. From the Responsive Layout window pane, click the **Web** media query button.

Notice that Cell 2 is now formatted with a 5% left margin. If you click in Cell 2, you can see the Cell Style update in the Responsive Layout window pane for the Web media query.

The change to the Cell Offset column accounts for the left margin.

Web Tablet Mobile

Row

Styles: New Style...

Insert Row Delete Style

Manage

Selected Row Style: div.My-6x6-Layout

Add Cell Delete Cell

Cell Width: columns

Cell Offset: columns

Row Gutter: pixels

Cell Style

```
Medium: web
div.My-6x6-Layout > div:nth-child(2)
{
  margin-left: 5%;
  width: 45%;
}
```

The format for Cell 2 is margin-left: 5%, and width: 45%.

Cell 1 remains at width: 50%, margin-left: 0%.

In 1839, an area named "Waterloo" was selected as the capital of the Republic of Texas. It was then named "Austin," "Father of Texas."

During the 1800s, Austin saw an increase in population and economic growth. The Texas State Capitol and University of Texas at Austin were both built in Austin. Austin became a hub for technology.

(If your XML Editor is narrow, the two cells might stack one of top of the other. That is okay.)

11. Build the HTML5 output, and open it in a browser to have a final look. For the Web screen size, notice the left margin for Cell 2.
12. Close the HTML5 output.
13. From the **Window** ribbon, select **Close All Documents**.

✓ **TIP** For optimal responsive layout appearance, set the sum of all cell widths and margins in a row to equal 100 percent.

APPENDIX

PDFs

The following PDFs are available for download from the online Help.

I Tutorials

Getting Started Tutorial

Autonumbers Tutorial

Back-to-Top Button Tutorial

Context-Sensitive Help Tutorial

Custom Toolbar Tutorial

eLearning Tutorial—Basic

eLearning Tutorial—Advanced

Image Tooltips Tutorial

Lists Tutorial

Meta Tags Tutorial

Micro Content Tutorial—Basic

Micro Content Tutorial—Advanced

Responsive Output Tutorial

Single-Sourcing Tutorial

Snippet Conditions Tutorial

Styles Tutorials

Tables Tutorial

Word Import Tutorial

| Cheat Sheets

Context-Sensitive Help Cheat Sheet

Folders and Files Cheat Sheet

Learning & Development Cheat Sheet

Lists Cheat Sheet

Micro Content Cheat Sheet

Print-Based Output Cheat Sheet

Search Cheat Sheet

Shortcuts Cheat Sheet

Structure Bars Cheat Sheet

Styles Cheat Sheet

I User Guides

Accessibility Guide

Analysis and Reports Guide

Architecture Guide

Autonumbers Guide

Branding Guide

Condition Tags Guide

Context-Sensitive Help Guide

Eclipse Help Guide

eLearning Guide

Getting Started Guide

Global Project Linking Guide

HTML5 Guide

Images Guide

Import Guide

Indexing Guide

Key Features Guide

Lists Guide

*MadCap Central Integration
Guide*

Meta Tags Guide

Micro Content Guide

Navigation Links Guide

Plug-In API Guide

Print-Based Output Guide

Project Creation Guide

QR Codes Guide

*Reviews & Contributions With
Contributor Guide*

Scripting Guide

Search Guide

SharePoint Guide

Skins Guide

Snippets Guide

Source Control Guide: Git

*Source Control Guide:
Perforce Helix Core*

*Source Control Guide:
Subversion*

*Source Control Guide: Team
Foundation Server*

Styles Guide

Tables Guide

Tables of Contents Guide

Targets Guide

Template Pages Guide

Templates Guide

Topics Guide

Touring the Workspace Guide

*Transition From FrameMaker
Guide*

*Translation and Localization
Guide*

Variables Guide

Videos Guide

What's New Guide