Manual Editing

The Complete Guide for Editing STS Manuals

Classroom Course Manual

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

Written, designed, and produced by:
DoIT Software Training for Students
Last Updated 6/15/2016

About Software Training for Students

Software Training for Students is an organization on campus that provides free software training to all students and faculty. Our services include custom workshops, open-enrollment classes, one-on-one project help, and access to Lynda.com. For more information on the Software Training for Students (STS) program, visit our website at **wisc.edu/sts**.

STS is part of the Division of Information Technology (DoIT) - Academic Technology at UW-Madison. For more information regarding DoIT Academic Technology, visit **at.doit.wisc.edu**.

Topics Outline

1. Introduction
2. File Structure
3. Essential Elements
4. Character Formatting
5. Tools and Instructions
6. Image Placement
7. Image Conventions
8. Sidebar Elements
9. Code Blocks
10. Saving and Exporting
11. Best Practices

Introduction

Software manuals written by Software Training for Students are used all over campus to teach trainers, faculty, and students alike. They are structured and designed with HTML, CSS, and JavaScript. However, to edit the manuals, you only need a bit of knowledge with HTML.

Each manual exists as an editable HTML file. This HTML file is linked to a single master CSS file which provides formatting for all the manuals. Likewise, each manual is linked to a series of JavaScript files, provides features like an automated table of contents.

This handy little guide contains everything you possibly need to know about the structure of our manuals, how to edit them, and the best practices to keep in mind during the editing process. As long as you adhere to the guidelines defined by this manual, everything will be just fine.

A few things to keep in mind as you enter the wonderful world of manual editing:

- Do not add your own CSS to the document. You are only allowed to use the features and funcitons outlined in this guide.
- Feel free to download the manual files and edit them using Dreamweaver, or your own text editor. (Be careful, Dreamweaver likes to add its own custom CSS everywhere.)
- Edit only the main content of each manual. You'll notice there are two tags in each manual that mark the beginning and end of the editable content.

Editing manuals can be a pretty dry task. There's not really any way around that. I would reccommend putting on some good electronic or deep house music while working, at least that's what I do to make it through. I also talk to Trevor, he's a nice guy.

File Structure

Basic Manual Structure

Each manual is a series of files, self-contained within a single folder. Inside the folder, you should find the following things:

- The main HTML file containing the contents of the manual.
- A keyboard icons folder containing a series of SVG files representing the most common keys that can be found on standard keyboards. They are intended for use with the keyboard shortcut element, discussed later in this document.
- A class files folder, used for storing any exercise files.
- An images folder, containing screenshots and icons that are referenced within the HTML.

Within the main HTML file, you'll find two comments, marking the beginning and end of the manual's main contents. You may only edit the HTML contained within these two comments. Outside the comments, there is important information that must be consistent across all manuals.

Manual Metadata

The information in this section is not necessary to know for manual editing, it is merely for informational purposes only. In the `<head>` of the `manual.html` file, you will find metadata, links, and information required to make the manual function. Here is a list of functions performed by the head of the manual.

- Sets page title
- Provides meta description
- Links to main stylehsheet
- Links to and initializes code highlight script
- Links to jQuery hosted by Google
- Links to main JavaScript effects file
- Imports custom fonts from Google Fonts

Our manuals rely on resources that are stored outside of the manual's main folder. Our custom fonts and jQuery files are hosted by Google. Our master files (like the stylesheet, effects script, etc) are hosted in a GitHub repository controlled by STS. If Google or GitHub ever goes down, then our manuals crash and burn with them!

Master Files

If you ever get a chance to sneak around the STS GitHub repository, be sure to check out the `master-files` folder. This folder contains the master CSS stylesheet, JavaScript files, and images that are linked to every manual. Be careful, editing one of these files will immidiatley change every manual instantaneously.

Here's an explanation of each of the important files.

- **The Master Stylesheet:** Titled as `stylesheet.css`, this file contains the styling for ALL manuals. It is very robust, and should only be handled by expert hands.
- **The Highlighting Script:** Titled as `code-highlight-script.js`, this file contains the JavaScript neccessary to automatically add color to code blocks used within the manuals.
- **The Effects Script:** Titled as `effects.js`, this file contains any other JavaScript effects we wish to add to the manuals. Right now, it automatically generates the topics outline at the beginning of each manual, and nothing else.

Alright, now that you've got a bit of background knowledge on the inner workings of our manuals, let's get to the fun stuff!

Essential Elements

Headings

The stylesheet associated with the STS manuals only supports three different levels of headings, demonstrated below.

```
<!-- heading examples -->
<h1> First Level Title </h1>
<h2> Second Level Title </h2>
<h3> Third Level Title </h3>
```

Lists

Unordered and ordered lists work the same way as they alway do, however **do not forget to add `<p>` tags inside of each list element.**

```
<!-- list example -->
<ul>
    <li><p> List item </p></li>
</ul>
```

Tables

Building a table follows the exact same structure as your standard HTML table. So if you already know how to make tables in HTML, you can skip this I suppose.

First, you'll need to use the `<table>` tag to begin and end the table. Inside the table, use `<tr>` to specify a new row. Each row has multiple elements, enclosed with the `<td>` tag. If you are creating a heading for the table (usually the top row) you use the `<th>` tag instead.

Confused? No problem. Just check out this example of a 2x2 table below.

```
<!-- table example -->
<table>
    <tr>
        <th> First column heading </th>
        <th> Second column heading </th>
    </tr>
    <tr>
        <td> Stuff in first column </td>
        <td> Stuff in second column </td>
    </tr>
</table>
```

Character Formatting

There are a few different ways you can format regular paragraph text:

- **Bold text**
- *Italisized text*

- Highlighted text
- `Type text`

All basic text should be encolsed in `<p>` tags. You can also use `<em>` and `<b>` tags to *italisize* and **bold** text. Along with this, you can highlight text using `<span class = "highlight">`.

Finally, you can also format words as type text. Type text is when you change the font of a few words `to make it look like this.` This should be used when referencing something code-related in the manual, or when referring to a file name, file path, or menu bar item. You can create type text by using `<span class = "type-text">`.

```
<!-- bold and italics example -->
<p> A sentence with <b>bold</b> and <em>italicized</em> words. </p>

<!-- highlight example -->
<p> Text with <span class = "highlight"> highlighted </span> words. </p>

<!-- type text example -->
<p> Text with <span class = "type-text"> typed </span> words. </p>
```

Tools and Instructions

In STS manuals, we often showcase a variety of different tools in a single piece of software. We also walk users through exercises and examples, using step-by-step instructions. There's a couple different elements we'll be utilizing to help us with these tasks:

- **Toolbox:** A small blue box (not a Tardis) that allows you to showcase different tools and provide a brief description of what they do.
- **Tool Icon:** When referencing a specific tool or button during an exercise, you can have it show up as a small circular icon in the sidebar.
- **Instruction List:** A list with special styling, used to walk users through step-by-step examples. The styling of this list distinguishes it from a normal list, and makes step-by-step procedures easier to follow.

Toolbox and Tool Icons

A **toolbox** contains a list of tools being introduced to the user. Each tool has a small icon with an image of that tool, along with the tool name and a brief description. Here is an example of a toolbox with two tools listed.

```
<!-- toolbox example -->
<div class = "toolbox">
    <div>
        <img src = "images/icon-name.jpg">
        <p><b> Tool Name: </b> A brief description of the tool. </p>
    </div>
    <div>
        <img src = "images/icon-name.jpg">
        <p><b> Tool Name: </b> A brief description of the tool. </p>
    </div>
</div>
```

Finally, **tool icons** are small circular placed in the sidebar. They are helpful for referencing specific buttons or tools during the exercise or instructions.

```
<!-- tool icon example -->
<img class = "sidebar-icon" src = "images/icon-name.jpg">
```

Please note, when taking screenshots of tools and buttons, make sure the image is proportionally square. If not, the screenshot will become distored.

Instruction Lists

Instruction lists are specially styled lists used to walk users through an example or exercise. They are styled to be more visually appealing easier to follow. Do not confuse these with regular lists; they serve different purposes. These lists are *only* to be used with giving step-by-step instructions.

The use the instruction list, simply add the `instruction-list` class to your `<ol>` tag.

```
<!-- instruction list example -->
<ol class = "instruction-list">
    <li><p> This is an item in my list </p></li>
</ol>
```

You can also create an instruction list inside of an instruction list, adding sub-steps to your exercise. The inner instruction lists will take on special styling to help distinguish the sub steps from the main steps. Take a look at the example below.

```
<!-- instruction list with substeps example -->
<ol class = "instruction-list">
    <li>
        <p> This is a step </p>
    </li>
    <li>
        <p> This is a step list.</p>
        <ol class = "instruction-list">
            <li><p> This is a sub-step! </li></p>
        </ol>
    </li>
</ol> </li>
</ol>
```

Image Placement

There are three sets of classes that can be used to place images in a variety of different ways. These include:

- Images that are centered in the page.
- Images that are placed in the righthand sidebar.
- Images that can be placed side by side.

Each image *must* placed inside of its own `<div>` container labeled with the proper class. Also, you may optionally add a caption below your image by adding a `<p>` tag immidiately below the image. Take a look at the generic structure below.

```
<!-- image without a caption -->
<div class = "image-class-name">
    <img src = "images/image-name.jpg">
</div>

<!-- image with a caption -->
<div class = "image-class-name">
    <img src = "images/image-name.jpg">
    <p> This is my optional caption! </p>
</div>
```

Along with this, all images (including pictures, screenshots, etc) should be stored inside the specific `images` folder for that manual. For more information on saving and naming images, check out our image conventions section in this guide.

Centered Images

Centered images will exist in the center of the page, but can also be flexible and sit next to sidebar elements. There are five different classes of sidebar images, each with a different size.

Class	Description
`centered-image-tiny`	Places image in center, tiny width.
`centered-image-small`	Places image in center, small width.
`centered-image-medium`	Places image in center, medium width.
`centered-image-large`	Places image in center, large width.
`full-width-image`	Allows image to take on full width of page.

Sidebar Images

Sidebar images are useful for displaying images alongside text, rather than after the text. There are only three classes, providing three different sizes of sidebar images.

Class	Description
`side-image-small`	Places image in righthand sidebar, gives extremely small width.
`side-image-medium`	Places image in righthand sidebar, gives medium amount of width.
`side-image-large`	Places image in righthand sidebar, gives large amount of width.

Side by Side Images

Did you know you can have two images displayed side by side, in tandem? All you need to do is add the proper class to your container, then chuck two image tags inside, like you see below.

```
<!-- side by side images -->
<div class = "sidebyside-image-class">
    <img src = "images/image-name.jpg">
    <img src = "images/image-name.jpg">
    <p> This is my optional caption! </p>
</div>
```

Both images that are being placed side by side need to be the same size. While this is not precisely required, it will help keep the manual visually appealing. Here's the table of side by side image classes that are available to use:

Class	Description
`side-image-small`	Places two small images together, side by side.
`sidebyside-image-large`	Places two large images together, side by side.

Adding Borders

Occasionally, you'll notice the screenshots of your program have a white background that blends in with the page behind it. If you wish, you can add a light border around the image to avoid this issue. To add a light border, add the `border` class to the `<div>` element that holds the image. Take a look at the example below.

```
<!-- image with border -->
<div class = "image-class-name border">
    <img src = "images/image-name.jpg">
</div>
```

Image Conventions

Throughout the manual creation process, we accrue many images and screenshots that reside in the `images` folder. It's important to keep these images organized with good naming conventions. These manuals are also designed to be displayed on the web, which means it has to have a quick loading time. In order to attain this, we will limit the size and file type of our screenshots and images.

A few general guidelines:

- All filenames should be lowercase, with a dash in-between words.
- Each screenshot should be saved as a PNG file.
- Screenshots should be cropped and annotated before being placed in manual.
- Screenshots of tool icons should have a square aspect ratio.
- Images should not exceed **1200px** in width.

Naming General Screenshots

The file name of each screenshot has three portions: a keyword assocating it to a specific section of the manual, a small description of what it shows, and an optional number if there are multiple screenshots of the same kind. A dash separates each section. Here's what the structure looks like:

`sectionkeyword-descriptor-number.png`

And if you're looking for a few examples to work off of. Again, the number is optional.

`recoloring-dropshadow-1.png`
`recoloring-dropshadow-2.png`
`goalseek-menubutton.png`
`cropping-exampleimage-1.png`
`pivottables-settings.png`

Naming Tool Icons

Small, square screenshots of tool icons have slightly different naming conventions. The name will consist of two words, the first being "icon" followed by the name of the tool. Here are a few examples:

`icon-movetool.png`
`icon-paintbucket.png`
`icon-formatpainter.png`

Sidebar Elements

Other than images, there are few different things that can live in the sidebar of the manual. Right now, we can add notes and keyboard shortcuts.

Notes

A sidebar note can be used for adding tips, suggestions, or anything else that's useful information, but not crucial to the exercise. Sidebar notes are `<div>` containers with a heading and text inside. You can also (optionally) add an image as well. The structure is shown below.

```
<!-- sidebar note example -->
<div class = "sidebar-note">
    <h2> Title of Note </h2>
    <p> Note text goes here </p>
    <img src = "images/image-name.jpg">
</div>
```

Keyboard Shortcuts

Keyboard shortcut boxes are handy to inform users of the quickest and most commonly used shortcuts for their program. Their formatting is a little more complicated than the rest, but they look really cool.

Inside each manual folder, is a folder named `keyboard-icons`. This folder contains images of every single keyboard key for both Mac and Windows. Inside of the keyboard shortcut element, we will be referencing these different icons.

Here's an example keyboard shortcut element.

```
<!-- keyboard shortcut example -->
<div class = "keyboard-shortcut">
    <h2> Title of Shortcut </h2>
    <p> Description of shortcut </p>
    <img src = "keyboard-icons/icon-control">
    <img src = "keyboard-icons/icon-alt">
    <img src = "keyboard-icons/icon-delete">
</div>
```

If you have multiple keyboard shortcuts you would like to share in the same container, or you would like to specify differences between Mac and Windows shortcuts, you can use the `<h3>` tag to add a label.

Code Blocks

Code blocks are used to display code of any sort. This includes JavaScript, HTML, CSS, Python, SQL, and a few others. When writing in code blocks, all whitespace and indentation is displayed. So be careful with how you write your code inside the code block itself.

The code block uses two tags, the `<pre>` and `<code>` tags. Inside both of these tags is where you write whatever code snippet you wish to share.

```
<!-- code block example -->
<pre><code>Everything written in here is displayed verbatim,
including whitespaces and indents.</code></pre>
```

Saving and Exporting

So you have this stunningly beautiful HTML file that you've spent hours working on. You're in love with it. And of course, you want to export it into its final form: a PDF. Well, to ensure consistency across manuals, there's a few steps we have to take to export this as a PDF exactly how we want it.

1. Open up `manual.html` in Google Chrome.
2. Go to `File > Print`.
3. Change the destination to save as a PDF.
4. Under margins, choose custom. Then, manually set the margins to be **0.6in** on each side.
5. Under options, make sure "simplify page" is **unchecked**, "headers and footers" is **unchecked**, and "background graphics" is **checked**.
6. Hit that big blue save button and hopefully it looks right.

Best Practices

The following section contains a serires of best pratices to use while editing an STS manual.

Exercises and Instructions

- Exercises should be less than 10 steps long. If an exercise is longer, create sub-steps by placing another instruction list inside of the first one.
- Small, single-step instructions don't need an instruction list. An instruction list is only for a multiple-step exercise. For example, if I want to tell the user to open a file, I don't need an instruction list. I can highlight the text instead.
- Don't be afraid of adding paragrahs of information to break up an exercise into smaller parts. If the majorithy of the manual is exercise steps, then we've done something wrong.

Utilizing Elements

- The stylesheet allows for many different elements, including keyboard shortcuts, sidebar notes, icons, toolboxes, and much more. Make sure to actually utilize these features whenever you can!
- With too many elements, a the manual may lose its visual appeal. Be sure to balance text, images, and other elements in an appealing way.