

Office of the Chancellor Web Templates Training and Usage Guide

Basic Rules

1. Do not alter top two levels of folder structure
 - a. You may add HTML files to the second tier of the site's folder structure and folders and HTML files to the third tier of the site's folder structure, but do not add or delete folders or files from the top two levels
2. Do not open or alter anything in the "css" folder
3. Do not open or alter anything in the "templates" folder
4. Never detach a page from a template once it has been attached

Page-by-Page Transition Process

STEP ONE: Creating new pages (for content not currently represented in the folder structure)

1. Locate the folder in the file structure in which you would like to create a new page
2. Open an existing file in that folder
 - a. If there is not an existing file, let me know and I will help you create one
3. "Save as" and rename the file. Make sure there are no capital letters or spaces in the title. You may use a dash (-) or an underline (_) to represent a space.
 - a. Make sure you type in the complete ".html" file extension. Dreamweaver will default to ".htm".
 - b. Make sure the title is simple and closely related to the main topic of the page. If you are creating a section with many, similar types of documents, come up with a sensible organizational system for naming the documents. Dates, document numbers or other conventions may be appropriate.
4. Once the file is saved, it should retain a few important features from the original document. These include:
 - a. Correct template, including right-hand menus
 - b. Correct Page ID number: The page ID identifies the current state, so that the correct right navigation item is highlighted when a user navigates the site
 - c. If you "save as" into a new folder deeper in the site structure, you may be prompted to update links. Click "Yes."
5. Type in new page title and eliminate any content that was specific to the old page.
 - a. If it is helpful, you may retain HTML code structure (A list or table setup, H1 tags, etc.). But you must be sure to get rid of all of the old content.
 - b. If, for some reason, you need assign the page a different page ID, type in the new page ID. Page IDs are the numbers one through twenty spelled out. These numbers correspond with the order of right-hand navigation items counting from top to bottom
6. Reassign the correct template to the page: (Modify → Templates → Apply Template to Page). You must do this to force the new page title and page ID to update. This is a bug in Dreamweaver.
7. Make sure to toggle the view to the "code" view and find the editable region where you can create content for the page.
 - a. The code looks like this:

```
<!-- InstanceBeginEditable name="copy" --><!-- InstanceEndEditable -->
```
 - b. You get to use the space between the ">" and the "<"
8. You are now ready to copy and paste content into your new page.

STEP TWO: Copying and pasting old content into new page

1. Since we have decided to copy pages from the live site rather than from the old site structure, it is important for you to have a method for making sure that you capture everything you need from the old site.
 - a. I recommend that you hit every page in a particular section before moving on to new sections, but you can figure out the method that suits you and the information you are transitioning best.
2. Make decisions NOW about what should be transitioned and what should not. Take notes of these decisions, so you remember what areas you chose to pass over.
 - a. If you are having difficulty deciding where the information belongs in the new site structure, bring it up to the group, so that we can make our best assessment for its new home.
3. Navigate to the page you wish to copy in IE or Firefox, highlight only the relevant content on the page you wish to move into the new site structure and copy that content (CTRL + c)
4. Return to Dreamweaver and paste the text into the CODE view
 - a. Do not paste text into the DESIGN view; Dreamweaver will attempt to recreate the code from the original page, which we do not want
 - b. You may also use "notepad" or "wordpad" to strip code out of content by pasting into the editor and reselecting the text and copying and pasting it into Dreamweaver. In this case, you may copy it into either the CODE or the DESIGN view
5. You are now ready to recode the page

STEP THREE: Recoding pages in XHTML format

1. You may choose to layout pages in either the code view or the design view in Dreamweaver. In some cases, it may be necessary to rely on the code view, so we will go through both methods.
2. You are looking for the process that works the fastest for you but still respects clean and valid coding conventions.
3. We will go over tables in a future session. If you need to create tables today, ask and I will work with you individually on it.
4. It is a good idea to keep the split view open even if you prefer to work in the design view. You need to make sure that you are not accidentally coding the page incorrectly or causing Dreamweaver to create new, inline styles for you (I will show you an example of this).
5. Please respect the fact that you have many more options available to you in Dreamweaver than are necessary to code pages correctly. Some of these options will code pages incorrectly. Coding pages incorrectly will create orphan copy that will not be updated during future site redesigns because the coding choices will remain on an individual HTML page and not in the global style sheets
6. You are now ready to learn to write valid XHTML Code. Please refer to the coding manual on the next page.

Coding XHTML pages

The formats and styles below are found in the “format” and “styles” menus on the Properties toolbar in Dreamweaver. The samples below show how these formats and styles are written in the HTML code. It is important to know what each format and style looks like so that you can tell if you are using the correct tags in the code and that the tags are not accidentally repeated or overlapping inappropriately.

FORMATS

For the purposes of this discussion, formats and styles are virtually interchangeable; however, formats represent structural blocks of text or other information that are native to HTML/XHTML code. These blocks have been reformatted in an external CSS style sheet, so that they display in a manner that is graphically consistent with the look and feel of the Web site, but even without the style sheet, a heading 1 tag or <h1> would be larger and bolder than a paragraph tag or <p>. Moreover, lists or , , etc. and tables or <table> would also look like lists and tables, but without special font sizes, colors, line spacing, etc.

The following are some of the format or block-level elements you will need to understand to work with the new templates.

Headings (sample: <h1></h1>)

Paragraphs (sample: <p></p>)

Lists (sample:)

Tables (sample: <table><tr><td></td></tr></table>)

Divisions (sample: <div></div>)

The meaning of these cryptic tags will be discussed below. Just know that you are dealing with a limited set of options and that, for the most part, you will only be formatting headings, paragraphs and lists. You can either format blocks of text by highlighting it and selecting an appropriate format from the format menu or you can type the appropriate formatting tags around the text you want formatted in the actual HTML code. For most formatting, either process takes about the same amount of time. The important thing to remember here is not to overlap tags or add additional styles to formats that are already styled in the cascading style sheet.

For example, all heading tags are already pre-styled as bold, so it is not necessary to add a “bold” style to them. They only need the opening and closing <h?></h?> tags to get their color, size, text weight, line-spacing, padding on top and bottom, etc.

HOW TO SET UP BLOCKS OF TEXTS IN FORMATS

Here are some sample format or block-level elements, how you would code them, and what goes inside of them.

Headings

h1 - <h1>around title text</h1>

h2 - <h2>around subhead text</h2>

h3 - <h3>around subhead text</h3>

h4 - <h4>around subhead text</h4>

h5 - <h5>around subhead text</h5>

h6 - <h6>around subhead text</h6>

Paragraphs

<p>around paragraph text</p>

<p class="indent">around paragraphs you want to indent</p>

<p class="news">around paragraphs you want smaller font for</p>

Lists

Lists follow the following structure in almost every instance you will encounter. A set of `` or `` tags open and close the entire list. These tags tell whether the list is a numbered list `` or a bulleted list ``. `` stands for ordered list and `` stands for unordered list. Lists can be styled further and can be used for many different visual elements using style sheets, but we will stick to simple lists for the time being.

Within the opening and closing tags in a list are the actual tags for each list item. Each list item will receive either a number or a bullet (or a roman numeral and an arrow, etc.). List items are represented by the following tags: ``.

How Numbered (or “ordered”) lists are structured in HTML:

```
<ol>
  <li>This is the setup for a numbered list; each </li> represents an item in the list and wraps around a
    numbered list item. HTML will automatically generate the list number. </li>
  <li>Item 2</li>
</ol>
```

This list would display like this in a browser:

1. This is the setup for a numbered list; each `` represents an item in the list and wraps around a numbered list item. HTML will automatically generate the list number.
2. Item 2

How Bulleted (or “unordered”) lists are structured in HTML

```
<ul>
  <li>this is the setup for a bulleted list; each </li> represents an item in the list and wraps around a
    bulleted list item</li>
</ul>
```

This list would display like this in a browser:

- This is the setup for a bulleted list; each `` represents an item in the list and wraps around a bulleted list item.
- Item 2

STYLES and CLASSES

Styles, which are called classes in CSS lingo, are special style elements that are not recognized by browsers without the embedded or attached styles that define them. Consequently, these are not structural elements that make up a basic HTML page, but rather extra elements that help to generate the users visual experience of a page. The importance of this distinction is that an HTML page should still be readable even if these styles are missing, such that a text reader for the visually impaired can still read the page and know which things are titles, which things are copy text and which things are navigation links, etc.

For the purposes of content maintenance, styles are used to further define the formats or block level elements above.

For example, a style can be used to make a title of a book italic within a paragraph or list or to bold someone's name in a paragraph, etc.

Here is what those two examples might look like in HTML code:

Italic style within a paragraph

`<p>Have you read Steinbeck's`

In a browser, this sentence would look like this:

Have you read Steinbeck's *Of Mice and Men*?

Bold text within a paragraph

`<p>John Smith was fired recently for not understanding how to properly code an HTML page.</p>`

In a browser, this sentence would look like this:

John Smith was fired recently for not understanding how to properly code an HTML page.

Other List Classes

Alphabetical list

`<ol class="alphalist">`only use with ul tag; will create a list that is sorted alphabetically rather than by number``

Alphabetical list with capital letters

`<ol class="alphalistcaps">`same as .alphalist but with capital letters``

List with lowercase roman numerals

`<ol class="lowerromanlist">`list with lowercase roman numerals``

List with uppercase roman numerals

`<ol class="upperromanlist">`list with uppercase roman numerals``

How to use class styles, e.g., .bold or .ital, with paragraphs `<p>`, list items ``, table cells `<td>`, table rows `<tr>`, and divisions `<div>`

`<p class="bold">`all styles like .bold can also be applied at the 'block' level to all the text within a given block level item. Paragraphs, lists, table cells, divisions (divs) are all block level items`</p>`

`<ul class="bold">`for entire list to be bold``

`<li class="bold">`to bold a particular list item``

`<td class="bold">`to bold all the text in a table cell`</td>`

`<tr class="bold">`to bold all the text in a table row`</tr>`

`<th class="bold">`to bold all the text in a table header`</th>`

`<div class="bold">`to bold all the text in a div`</div>`

Other class styles

(this is a growing list – please submit requests for new styles to Jeff Crandall)

You may select these styles from the “styles” menu on the properties bar in Dreamweaver or you may code them yourselves in the code view.

1. **.bold** - `tags go around only the text you want to be bold`
2. **.bold_large** - `slightly larger bold for titles that can have text right under them. Also, colored same as heading tags`
3. **.boldital** - `bold and italics combined`
4. **.asterick** - `special color and size for asterick`
5. **.ital** - `italics only`
6. **.topnavtext** - `will style like links across top of page`
7. **.topnavlines** - `will style like lines between links across top of page`
8. **.cutline** `small text under photos`
9. **.cut_ital** `small italic text under photos`
10. **.quote_left** - `normal sized text, italic, aligned left`
11. **.quote_center** - `normal sized text, italic, aligned center`
12. **.quote_right** - `normal sized text, italic, aligned right`

Additional new styles available on the finance fonts style sheet:

1. **.indent** – indents first line of paragraph
2. **.news** – smaller font size
3. **.right** – text align right
4. **.center** – text align center (use sparingly on things like table headers)
5. **.reversebold** – bold white text
6. **.bold_black** – bold black text (instead of blue default)
7. **.alphalist** – alphabetical list instead of numbered list or bullet list
8. **.cutline** – small, centered text for photo cutlines
9. **.cutlineleft** – cutline text, left-justified

Additional rules for new templates

The following are a few rules that have come up during the development of these templates. These need to be adhered to so that the templates don't display incorrectly in browsers and so that we maintain fully accessible Web sites that are compliant with the Web standards developed by the Worldwide Web consortium (<http://www.w3.org/>).

These rules are subject to change at this point, but they are the trouble spots we've identified as we work with these templates and Dreamweaver.

Rule One:

As a practice, do not display full URLs (web addresses) in the content area. Instead, describe the link in normal language.

Issue: Long URLs and unbroken strings of text (without a space) extend into the right column and either push the right column's dividing line to the right or extend over it.

Example:

Instead of a URL like this: http://www.mnscu.edu/media/newsreleases/2006/020406_teachercenter.html

Type these words and link to the URL above: "View Teacher Center news release"

NOTE: E-mail addresses can typically be displayed in full (as long as they are not incredibly long) and full e-mail addresses can be helpful to users whose mail clients are not properly hooked up to their Web browsers.

Rule Two:

Do not use tables for content or photos. Tables should only be used for tabular data. If you do need to use a table, do not specify a width or height either in a percentage or a fixed number (for the table as a whole or for any cell, column, or row). Instead only specify cell spacing or cell padding. I've been using a cell spacing setting of "8" and that seems to create nice tables in these templates.

Issue: Tables with fixed widths can easily break the display of these templates. If cell spacing and/or cell padding is used, then the content container will decide how wide they should be. There will also be an issue if too many columns are used, so if you need to display a table with many columns, you should consider one of the following options: Use a "fullwidth" template (610 pixel wide page content area), create a PDF from an Excel spreadsheet, or use a blank white HTML file with a link back to the last Web page.

Also, the use of tables for non-tabular data (i.e., page layout) is not in keeping with current accessibility standards, because it creates item order issues for text-readers and other alternative browsers, especially when table headers and labels are not clearly defined.

If you require table styling (borders and cell fills), please contact me and I will develop table styles for you. We will need to create these anyway and I have it on my list of to-dos, but I can get to it faster if you have an actual need that comes up.

Rule Three:

Never select a color for a fill or background color in Dreamweaver. Colors should only be generated by a style from the style sheet we've provided.

Issue: All of the typical user-friendly features provided in Dreamweaver create inaccessible code on the backend. It winds up writing HTML tags and/or CSS styles into the file that is currently open (but not globally).

Aside from creating accessibility issues, it also means that these tags will not be updated when we make global style sheet changes. If you use the provided CSS styles we provide, each HTML page will always be updated and current whenever we roll out revisions to the primary CSS files or Web templates.

Rule Four:

Never select a font, font size, font color or font styling (e.g., Bold, Italic, Underline, etc.) in Dreamweaver. As with colors, all font styles are determined by styles in the style sheet. If we don't have enough styles to cover all of your needs, then contact me and I will develop more for you.

Issue: Again, selecting a font size, style, etc., in Dreamweaver will write a special tag for it on the page, thus, creating the same issues noted above. We have already pre-selected many styles for fonts, so you either need to use the default style for all copy, which is automatically set, a heading 1-6 style for headers from the heading menu, or special styles such as "bold" or "ital", etc. that have been set up for you and which display in a style menu that I will show you.

Rule Five:

Special class styles have been created to include photos in the content area of Web pages. These styles take care of alignment and text-wrapping. Photo styles are meant to be placed in a division tag that surrounds both the photo and any photo cutline; moreover, the cutline class style should be placed in a paragraph tag.

In the future, we will need to come up with a protocol for image handling, but since these templates are touchier than table-based templates, we need to make sure that the size, dimensions, and clarity of imagery are correct. Once we have trained a handful of content maintainers on these practices, we may be able to decentralize photo and image handling again.

In these templates, no image or table wider than approximately 400 pixels can be used.

Class styles for photos and cutlines:

```
.photoright {position: relative; float: right; margin-left: 10px; margin-bottom: 10px; margin-top: 0px; margin-right: 0;}
```

```
.photoleft {position: relative; float: left; margin-left: 0; margin-bottom: 10px; margin-top: 0px; margin-right: 10px;}
```

```
.cutline {font-size: .85em; color: #000000; padding-bottom: 8px; padding-top: 2px; padding-right: 0px; padding-left: 0px; text-align: center;}
```

Rule Six:

Do not display clip art, animated gifs or attempt to replace the headers we've supplied. If you have a need for a new header, clip art, or animated gifs, please contact Jeff Crandall (jeff.crandall@so.mnscu.edu), so that your particular needs can be addressed. This is largely a taste/sophistication issue and there may be exceptions, but we'd like to keep an eye on the type of imagery we are using on these sites to make sure that they are representing the system well.

The use of charts and graphs to support content are exceptions to the rule above, but I would still like to make sure that display standards are being followed for that sort of visual information as well.

Rule Seven:

Do not include any logos on these templates other than the system logo that has been provided in the upper left hand corner. If you believe you have a special case in which another organization's logo is appropriate, please contact Jeff Crandall (jeff.crandall@so.mnscu.edu) to discuss its use.