MIDDLESEX Community College

Tools and Technologies for Tech Writers 2023

Week 2

Progressive Information Disclosure

Notices

This document was prepared as a handout for the Middlesex Community College Tools and Technologies for Technical Writers class, Winter semester 2023.

Prepared by Zoë Lawson, course instructor.

Contents

Further reading on Progressive Information Disclosure

Officially a user interface design theory, it has a lot of practical implications with technical writing.

- Applying Progressive Information Disclosure: User Interface Content Design by Andrea Ames
- https://www.nngroup.com/articles/progressive-disclosure/
- https://www.interaction-design.org/literature/topics/progressive-disclosure - More UI focused
- Developing Quality Technical Information: A Handbook for Writers and Editors, Third Edition

Recommended tools

While text editors can be the most basic editor out there, they can have a lot of features that make your life easier.

There are numerous text editors out there. Both Windows and Macs have built in editors. However, these are very rudimentary. There are others that can parse and highlight common coding languages, offer robust in file and cross-file search, and many other features.

I'm not a Mac person, so I'm not sure what to recommend. There's Sublime (https://www.sublimetext.com/), but it requires a license. I remember not hating Atom (https://atom.io/), but it was discontinued in December of 2022. My favorite for the PC is Notepad++.

Notepad++

You have to edit text files. Or tidy up stuff that's easier to work with as text. Or change a file's encoding. Notepad++ is a fantastic text editor.

You are going to need a text editor. Sometimes you will be working with text files, such as properties files or raw HTML. Other times you will be cutting and pasting between applications that add extra bits. (Have you ever copied something from a web page and pasted it into a Word file and got extra junk?) Yes, all computers have some sort of text editor, but plain old Notepad doesn't have extra features. Having a good text editor is invaluable.

Notepad++ is my favorite text editor. It's familiar with a whole bunch of common coding languages and provides highlighting. You can change file encoding (e.g. English vs Russian) and also change line endings. You can search and replace across files. You can perform searches using regular expressions.

Reputable download

https://notepad-plus-plus.org/

Pencil

This tool was introduced to me as a UI mockup tool. You can also use it to make basic diagrams.

Being able to throw together a UI design is sometimes part of your job. Being able to put together basic workflow diagrams is a common requirement for techdocs. This is a nice tool to be aware of and can be worth using.

Reputable download

https://pencil.evolus.vn/

Week 2 Homework

Re-write the text of a UI using a properties file.

1. Make sure you have the latest and greatest from the mcc_tools_tech project.

   See the GitCheatsheet.pdf in the Handouts directory, or the video I made.
2. In the `Week02-ProgressiveInfo\Homework` folder of the class repo, review the `login.html` file. This is the mockup of a UI you are working on.
3. Make a copy of `login.properties`, changing the name to be *`FirstNameLastName`*`.properties`.

   You must use this naming convention. Automation has been set up using a specific file name.

   Valid file names are:

   - `DecioLeonardo.properties`
   - `ElisabethChaput.properties`
   - `MelindaHobausz.properties`
   - `PaulStevens.properties`
   - `SarimProeung.properties`
4. Revise the text of the UI by making changes to your copy of the *`YourName`*`.properties` file.
5. Add your copy of the *`YourName`*`.properties` to Git and make a pull request.

There is a GitHub action setup to read in your *`YourName`*`.properties` and generate a *`YourName`*`.html` file.

Note: This is configured using GitHub Actions. I am not sure if it will work in your forked repo or not. Whether or not it works may depend on whether or not you chose to enable actions when setting up your account. It should work in the class repo.

If the automation doesn't work, I will transform the files for you by hand.

Configurations for GitHub Actions

There are configuration settings you may need to make to allow GitHub Actions to work in your fork of the class repository.

GitHub is constantly updating it's security. Therefore, what worked last week, may not work next week.

To enable GitHub Actions to be able to have the properties file to HTML transform happen in your fork, you may need to change some settings in your GitHub account.

1. Go to your fork of the class repository on github.com.
   My "student" demo account (mcc-demo) goes to https://github.com/mcc-demo/mcc_tools_tech.
2. Click on **Settings** for the repository.

   It's the last of the tabs, after Insights. Do not go to the settings for your profile.
3. In the list of settings on the left, select **Actions** > **General**.
4. Scroll to the bottom of the page.

5. Under Workflow Permissions, change the following:
   - Select **Read and write permissions**.
   - Select the **Allow GitHub Actions to create and approve pull requests** check box.
6. Click **Save**.

If all goes well, if you make changes to your properties file, it will automatically generate an HTML file. You can then pull from your fork ("Pull from origin") to see the changes to the HTML file locally.

Don't forget to make a pull request to get your changes into the class repository. (Click the **Contribute** button in your fork in GitHub.)

Optional work

You are not required to do this, but if you'd like to explore more.

My little properties file is very limited. I'm sure you can do better.

Using a UI mockup tool like Pencil, or even something like PowerPoint if you don't want to try a new tool, and mock up a UI with PID.

Pick something simple. Once you start getting into PID, you can go very, very deep down the rabbit hole of different things you can do. (This is why I picked a log in screen.)

1. What text would you like on the screen to help people figure out what to do?
2. Where would you like hover text?
3. Where would you like more interactive "help" buttons? What do you want them to do?
4. Does it have a help system? Is it context sensitive?

This is something you could eventually polish up and have as part of a portfolio, if you're interested.

Having some experience with a mockup tool doesn't hurt the resume, either.