Front-end development

Module 1

Websites 101 and Twine Basics

Bits and Bots
study group

Content

Websites 101

We thought that it might be useful to do some exploring and reading before starting out with HTML in module 2. A good introduction to website development that includes clear definitions on the different coding languages and a bit more of an overview of how website development works can be found here. Based on this resource as a starting point please feel free to search and explore these concepts further as suits you. It would be great to share any other resources you find helpful as a starting point. A rather less serious, fun and only slightly relevant video about country internet codes can be found here for general enjoyment.

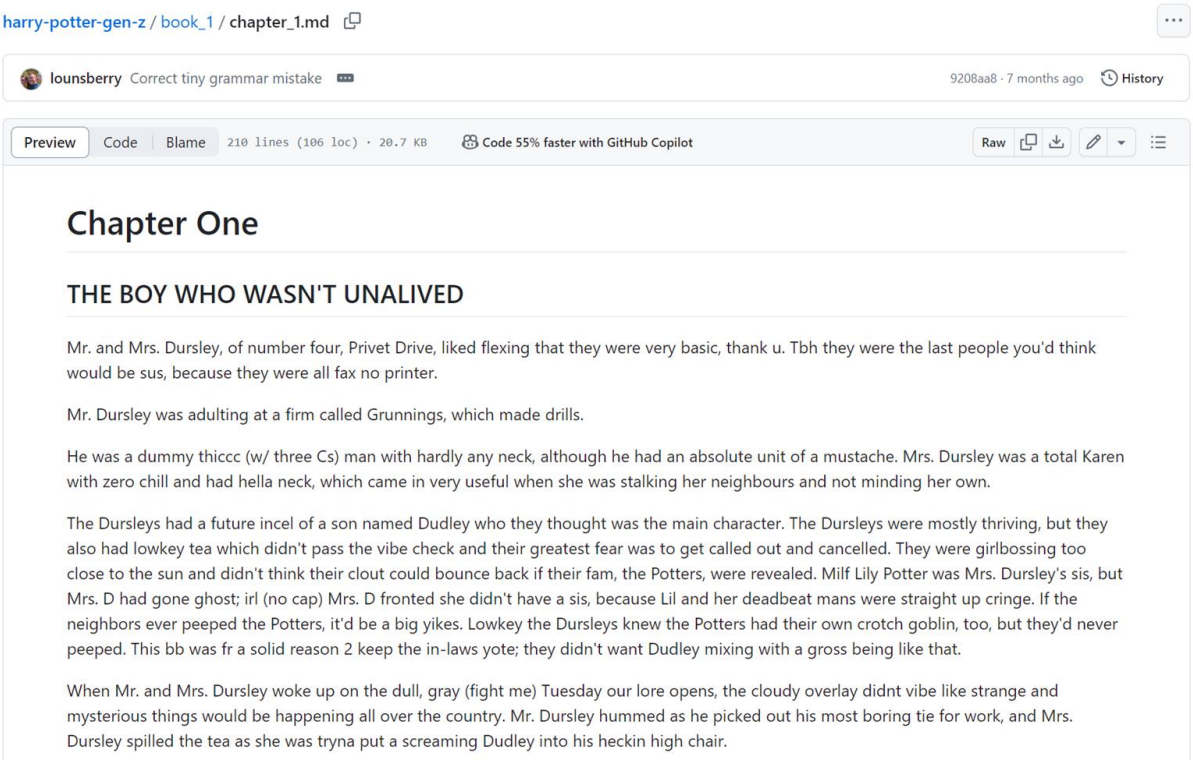
GitHub

During this course we will be using GitHub… A lot! Why? GitHub is used by so many software developers and even if you hate it and don't wish to use it, it really helps to at least understand the context in which developers are working.

GitHub is the ultimate collaboration tool for coding projects but can be used as a platform for any collaborative work. It's also really useful to store your work and share it. GitHub also allows you to create free webpages which we will look at later! In this part of the module we will understand more about why GitHub is so useful, create your account and try out some of the features. To do this we will mostly be following some of the tutorials provided by GitHub itself for learning.

GitHub is a platform that allows anyone to create, store, manage, and share content. Oftentimes this is code. But that is not the only thing present in GitHub. Here are some of the use cases for GitHub:

- Software development
- Fanfiction
- Zines
- Community/crowdsourcing projects
- Finding file samples
- Educational resources
- Reading lists
- Datasets
- Blog posts
- Documentation



harry-potter-gen-z / book_1 / chapter_1.md

lounsberry Correct tiny grammar mistake 9208aa8 · 7 months ago History

Preview Code Blame 210 lines (106 loc) · 20.7 KB Code 55% faster with GitHub Copilot Raw

Chapter One

THE BOY WHO WASN'T UNALIVED

Mr. and Mrs. Dursley, of number four, Privet Drive, liked flexing that they were very basic, thank u. Tbh they were the last people you'd think would be sus, because they were all fax no printer.

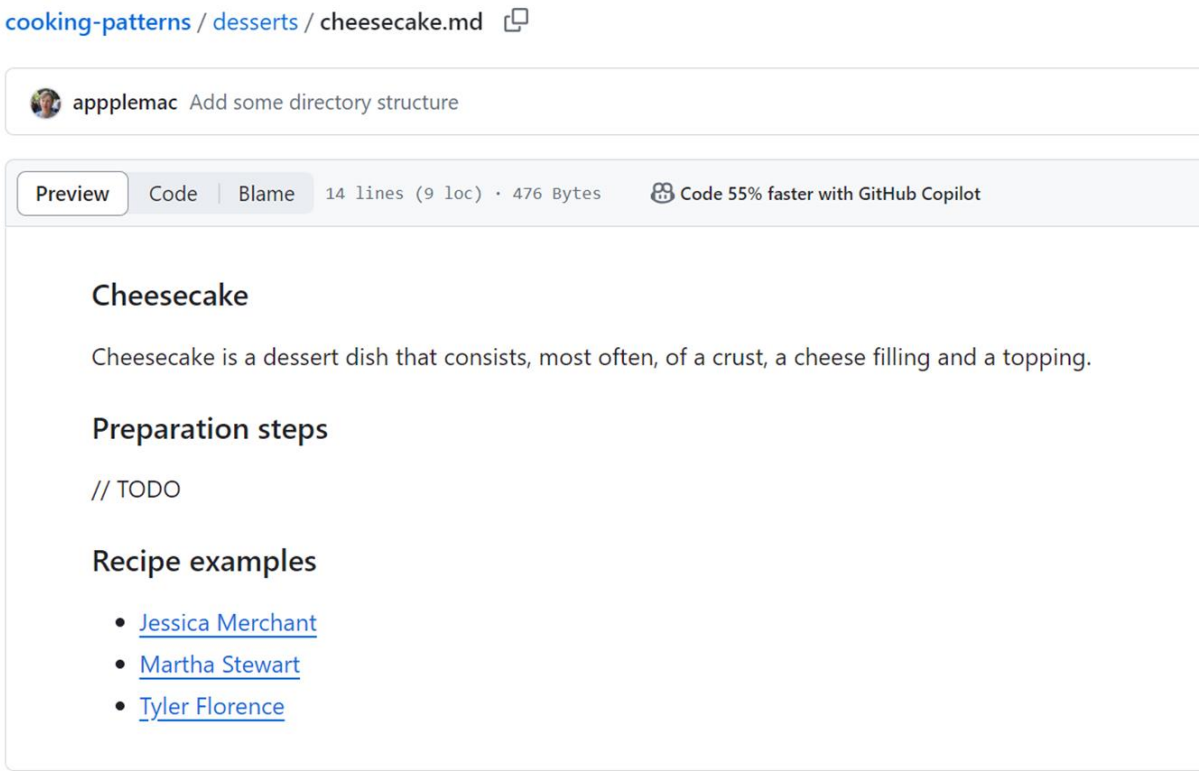
Mr. Dursley was adulting at a firm called Grunnings, which made drills.

He was a dummy thiccc (w/ three Cs) man with hardly any neck, although he had an absolute unit of a mustache. Mrs. Dursley was a total Karen with zero chill and had hella neck, which came in very useful when she was stalking her neighbours and not minding her own.

The Dursleys had a future incel of a son named Dudley who they thought was the main character. The Dursleys were mostly thriving, but they also had lowkey tea which didn't pass the vibe check and their greatest fear was to get called out and cancelled. They were girlbossing too close to the sun and didn't think their clout could bounce back if their fam, the Potters, were revealed. Milf Lily Potter was Mrs. Dursley's sis, but Mrs. D had gone ghost; irl (no cap) Mrs. D fronted she didn't have a sis, because Lil and her deadbeat mans were straight up cringe. If the neighbors ever peeped the Potters, it'd be a big yikes. Lowkey the Dursleys knew the Potters had their own crotch goblin, too, but they'd never peeped. This bb was fr a solid reason 2 keep the in-laws yote; they didn't want Dudley mixing with a gross being like that.

When Mr. and Mrs. Dursley woke up on the dull, gray (fight me) Tuesday our lore opens, the cloudy overlay didnt vibe like strange and mysterious things would be happening all over the country. Mr. Dursley hummed as he picked out his most boring tie for work, and Mrs. Dursley spilled the tea as she was tryna put a screaming Dudley into his heckin high chair.

Here you can see some fanfiction that is being written with a community. But you can also have recipes on there:



cooking-patterns / desserts / cheesecake.md

appplemac Add some directory structure

Preview Code Blame 14 lines (9 loc) · 476 Bytes Code 55% faster with GitHub Copilot

Cheesecake

Cheesecake is a dessert dish that consists, most often, of a crust, a cheese filling and a topping.

Preparation steps

// TODO

Recipe examples

- Jessica Merchant
- Martha Stewart
- Tyler Florence

As digital archivists, we often use open-source tools to aid us with e.g. identifying and validating digital objects or harvesting websites. Here are some useful examples of GitHub repositories used in the field that you can find:

- https://github.com/openpreserve/jhove
- https://github.com/digital-preservation/droid
- https://github.com/webrecorder/browsertrix-crawler
- https://github.com/noord-hollandsarchief (see the repositories in this account)

So why is GitHub great to use? GitHub allows people to work collaboratively on any project, with tracking and control settings favouring non-linear workflows and multiple contributions at a time. By tracking everything, there is also a lot of data integrity. Additionally, you can use GitHub to let the developers know you have an issue/something is not working.

There are a lot of people working with digital archives that use GitHub. This is the reason that we found it very important for everyone to learn how to use it. Additionally, to show you that it is not just a scary programmer platform.

GitHub Exercise

For this module, we want you to:

1. Create a GitHub account if you do not have one:
    - To start with, read more about GitHub here. One thing to bear in mind is that GitHub, while an amazing tool does come with a lot of jargon. A great explanation for some of the terminology around GitHub can be found here.
    - To create your first GitHub account you will first need to register at www.github.com and follow the instructions provided here.

2. Follow the GitHub tutorials:
    - Try out a few features with your new account. This will be creating a repository, creating a pull request, creating branches and merging a pull request. If these terms seem strange then that is because they are slightly. A link to the tutorial you will be following for the first stages of GitHub can be accessed here.
    - If you are enjoying learning and wish to practice further then you can follow the tutorial on the GitHub skills page 'Introduction to GitHub'

Note that both steps can also be done after the monthly meeting on GitHub in April. In this meeting we will go over GitHub terminology and give a demo.

Building Your First Twine Game

Background

Twine is an open-source tool for telling interactive, nonlinear stories. Why would you want to use Twine? This introduction gives you good examples of why this tool can be useful.

Play Some Games

To start this module let's try playing some games that have been created by Twine to get a feel for them. Examples of games that were created with Twine:

- Rat Chaos
  - This game is about a spaceship captain choosing between going about their daily routine or "unleashing rat chaos". Fun fact: While learning the software Twine, the author created this game in a couple hours on a single day (July 18th, 2012).
  - Link: https://debacle.us/ratchaos/
- You Are Jeff Bezos
  - You start the game by waking up as Jeff Bezos. To get back to your own body, you need to spend all of Jeff Bezos' money. Then the game provides you with various choices that you can spend the money on. These choices evoke reactions.
  - Link: https://direkris.itch.io/you-are-jeff-bezos
- Finally our own games!
  - The 2024 cohort of the study group created games using twine while learning and they are really good fun to play. The games created by the group can be found in the repository. Duck Archive, Dangers in Arkam, Who wants to ruin Professor Schmoosh's Data, Stolen Archive and File Format Fling are all examples of Twine games.
  - Link: https://github.com/Lotte-W/Bits-and-Bots-study-group/tree/main/Games

Twine does not just have to be used for games. For example you can write out training modules or interactive guidance for your archive users. It could help having the users select various options rather than seeing a wall of text. Have a think about the possibilities of using twine in your work.

Try Out Twine

Twine is all about creativity and writing. Now that you've played some games the next stage of the module is to try out the software yourself. To introduce yourself to Twine we will go to https://twinery.org/ and press click on browser. For this part of the module we will be following along with the first video in Adam Hammond's https://www.adamhammond.com/twineguide/. Bear in mind what the final games you played looked like, and try and see how the techniques that you are learning may be shown in the final product of a game. If you prefer not to follow the video the Video Game

Museum has another guide to follow or there is the Twine reference guide as well (though this can be a bit overwhelming).

Developing Your Own Story

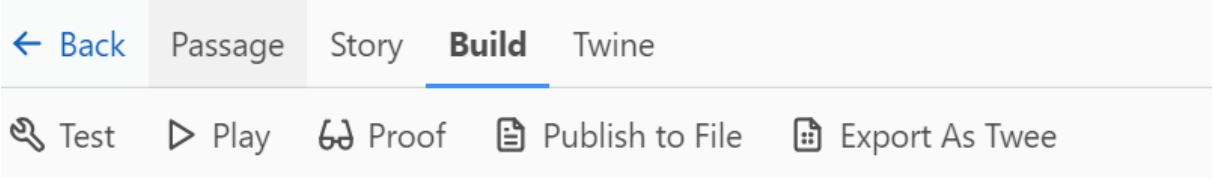
Now that you are more familiar with how Twine works it is time to take a step back and look at how you may apply it to something you would like to create. Taking into account what you have learnt, try and create a story or a game within those parameters. It may not be digital preservation related if you don't wish it to be so. Try looking up other Twine games to inspire you online.

Create Your First Story

The next stage is to create your first story using the techniques from the tutorial provided above and if you prefer reminders in written text or to explore more functionality then the Twine reference guide is also available here.

Share Your Story

For this module we will be saving the story as an HTML file. To do this go to Build at the top of the screen and then 'Publish to File'.



This will download a copy of the game. If you click on this you will be able to view and play the game in your browser! Or send it to others to play as well!

Browser vs. Desktop App

For this module we have been using the browser to create our first game. However it is also possible to install the **desktop app**. Both versions are available on the Twine Homepage. There are pros and cons of each version.

The **browser** version release uses your selected web-browser's Local Storage to store all the passages of your story, and by default the web-browser's Local Storage area is limited in size to between 5-10MBs (depends on the brand & version) which means you are limited if you want to embed media files within the passages of your story. The Local Storage are can also be affected by the "Empty Cache" (or similar concepts) feature of the web-browser (this depends on brand & settings) so it is recommend to regularly use the Twine 2 Archive option to backup all your projects in one go, or to use the Publish to File option to create locally stored HTML files for each of your projects.

The **desktop app** release stores each Story Project as a local file in a automatically predetermined location/folder on your local machine, this location should not be used by you as a destination for the Story HTML file you create using the Publish to File option or

you could end up overwriting (and losing) your projects. Because this release uses local files for storage then the limits on embedding a media file in a Passage is removed.