Tech Help With Twine (Sugarcube)

# Workflow/Organisation

* **A Modern Workflow For Twine:** <https://dev.to/lazerwalker/a-modern-developer-s-workflow-for-twine-4imp>
* **Twee and VS**: <https://github.com/JoshuaGrams/tiny-qbn/blob/master/doc/tweego.md>
  + Setting up VS to call TweeGo as a compiler
* **Twee/VS video tutorial:** <https://www.youtube.com/watch?v=u2LPu4efU5U>
* **Making Twine Games, programmer edition:** <https://jonc.dev/twine-games-tutorial>
  + Useful info on file structure
* **Structuring for bigger visual novels** <https://www.youtube.com/watch?v=sy2lye_3wpg>
  + Essentially, a quickstart guide to tweego/VSCode/Sugarcube
* **Embed a Twine 2 file in Wordpress:** <https://opensource.com/article/20/2/embed-twine-wordpress>
* **GitHub/Git** Beginner's guide: <https://www.youtube.com/watch?v=PWqS4NBhEY8>
  + Git cheatsheet <https://education.github.com/git-cheat-sheet-education.pdf>
    - git add .
    - git commit –m “commit notes”
    - git push [alias] [branch]

# Twine2/Twee3/SugarCube Tutorials

## Sugarcube/Twine2

* **Sugarcube 2.35 docs:** <https://www.motoslave.net/sugarcube/2/docs/>
* **Twine 2 (fairly recent) video tutorials**: <https://www.youtube.com/playlist?list=PLlXuD3kyVEr5jWoG0oDygKWOgFC3qrKN->
* **Edit the sidebar**: <https://twinery.org/cookbook/sidebar_left/sugarcube/sugarcube_sidebar_left.html>
* **Twine Tools:** <https://twinelab.net/twine-resources/#/>
* **Setting variables across Javascript and Twinescript:** <https://discord.com/channels/389867840406159362/389868418855075840/873131723620245515>
  + **State.variables** <https://www.motoslave.net/sugarcube/2/docs/#state-api-method-getvar>
* Setting variables as passage names: <https://discord.com/channels/389867840406159362/389868418855075840/872922293423177819>

## Sugarcube: Macros

* **Typespeed:** <https://www.motoslave.net/sugarcube/2/docs/#macros-macro-type>
* **Button:** <https://www.motoslave.net/sugarcube/2/docs/#macros-macro-button>
* **Audio:** <https://www.motoslave.net/sugarcube/2/docs/#macros-audio>
* **Media passages:** <https://www.motoslave.net/sugarcube/2/docs/#guide-media-passages>

## Twee3

* **Twee 3 Standards:** <https://twinery.org/cookbook/terms/terms_twee.html>
* **TweeGo Docs:** <https://www.motoslave.net/tweego/docs/#introduction>
  + **Useful commands:**
    - **Log stats about a story:**  
      -l, --log-stats
    - **Output dir to file:**   
      tweego -o example\_2.html example\_dirtwectory\_2
    - **Watch file changes, recompile:**  
      tweego -o example.html example.twee --watch

## Markup:

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Syntax & Example** | **Rendered As** | **Displays As (*roughly*)** |
| Emphasis | //Emphasis// | <em>Emphasis</em> | *Emphasis* |
| Strong | ''Strong'' | <strong>Strong</strong> | **Strong** |
| Underline | \_\_Underline\_\_ | <u>Underline</u> | Underline |
| Strikethrough | ==Strikethrough== | <s>Strikethrough</s> | ~~Strikethrough~~ |
| Superscript | Super^^script^^ | Super<sup>script</sup> | Superscript |
| Subscript | Sub~~script~~ | Sub<sub>script</sub> | Subscript |

# Web design

## CSS

* **CSS For Twine (a bit old but probably fine)**: <http://www.adamhammond.com/wp-content/uploads/2016/03/hammond_twineguide_3_css.pdf>
* **Beginner’s CSS For Twine:** <https://twinery.org/forum/discussion/1528/css-is-your-friend-the-basics-of-changing-twines-default-appearance-for-newbs>
* **Importing Google Fonts:** <https://twinery.org/cookbook/googlefonts/sugarcube/sugarcube_googlefonts.html>
* **Button CSS** <http://twinery.org/questions/4/can-i-change-the-appearance-of-button-macros>
* **Button CSS designer** <https://www.bestcssbuttongenerator.com/>
* **CSS Color Scheme** <https://palette.ninja/#c1fa62-#c7ac4e-#e0c258-#080703-#fad862>
* **Smooth Scrolling Accessibility**: <https://css-tricks.com/smooth-scrolling-accessibility/>
* **CSS Lint** (tells you if your CSS has bad practices) <http://csslint.net/>
* **Twine Horizontal UI:** <https://ccrberus.itch.io/twine-sugarcube-template>
* **Unicode dingbats:** <https://unicode-table.com/en/#dingbats>
  + Note: you need to insert the CSS symbol into the stylesheet using the ‘content’ property.
  + Arrows: <https://unicode-table.com/en/sets/arrow-symbols/#down-arrows>
  + Up Arrowhead: <https://unicode-table.com/en/2B9D/>
* **CSS Gradient generator:** <https://cssgradient.io/>
* CSS Patterns: <https://bansal.io/pattern-css#font>

**Web Design For Mobile**

* **Mobile Web Development**: <https://developer.mozilla.org/en-US/docs/Web/Guide/Mobile>
* **Mobile Friendliness**: <https://developer.mozilla.org/en-US/docs/Web/Guide/Mobile/Mobile-friendliness>
* **Touch Target Sizes**: <https://www.lukew.com/ff/entry.asp?1085>
* **Best Practice for Speeding Up Your Website:** [long link](https://developer.yahoo.com/performance/rules.html?guccounter=1&guce_referrer=aHR0cHM6Ly9kZXZlbG9wZXIubW96aWxsYS5vcmcv&guce_referrer_sig=AQAAADsnLkepN1mWRi0E1OtA8PVxTa9LDbrfk1r_ZPFqvJyBFtTkln3NCChy-iHV_qzSmAB-3DJ2U6iYr_yhwaORxsDREXeqDsZaEoD2qrpoXjrSfyxOojJi32gwPEGut_QduIi8qLmsMC885JUVaeFChMt_mDK6EW9Qa-ym4UNH6a_k)

**Native HTML elements**

* **MDN- Buttons:** <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/button>
* **MDV- Embed Audio Element:** <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/audio>
  + Twine has its own audio element so implementing an Embed can be a bit weird.
* **ARIA Accessibility:** <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA>
* Alt text:
  + <img src="img\_girl.jpg" alt="Girl in a jacket" width="500" height="600">

**Javascript**

* **QR Code functionality:**
* **Initial convo about QR codes:** <https://discord.com/channels/389867840406159362/389867840972259331/869159882127970395>
* **Tutorial on writing a function that grabs a search parameter:** <https://www.youtube.com/watch?v=j3-LV3XxhVg>

## Misc:

* **Base64 Encoder:** <https://websemantics.uk/tools/image-to-data-uri-converter/>
* **Abstraction in design:** <http://worrydream.com/LadderOfAbstraction/>
* **Batch Files:** <https://www.makeuseof.com/tag/use-windows-batch-file-commands-automate-repetitive-tasks/>
* **File Structure:** <https://mitcommlab.mit.edu/broad/commkit/file-structure/>
* **Hosting a website on github** <https://www.khanacademy.org/computing/computer-programming/html-css/web-development-tools/a/hosting-your-website-on-github>
* **Web Share API:** <https://alligator.io/js/web-share-api/>
  + <https://caniuse.com/web-share>
* **Free piano keys graphic:** <https://pixabay.com/vectors/keyboard-piano-music-sound-keys-153343/>
* **Optipng Help Doc:** <https://helpmanual.io/help/optipng/>
* AXE Accessibility chrome extension: <https://www.deque.com/axe/>
* Colour contrast checker: <https://webaim.org/resources/contrastchecker/>

File Structure

* Project File: Leeds\_Piano\_Trail
  + (git)
  + Dev
    - vscode
    - img
    - audio
    - Source
      * .tw files
  + Root (the master branch for git)
    - img
    - audio
    - index.html

# Technical Help With Ink

**Calico/Catmint**

Calico: <https://github.com/elliotherriman/calico/blob/main/documentation/getting%20started.md>

* A fork of ink that allows for more functionality.

Catmint: <https://elliotherriman.itch.io/catmint>

* Prototyping tool for Calico

**SearchQuery Javascript**

* This is the functionality which will allow different QR codes to access different exercises.
* QR codes must point to the URL of the hosted Ink story, appended with a ?query
* Javascript has the ability to read a ?query from a URL. The function that does this is explained here:
  + <https://developer.mozilla.org/en-US/docs/Web/API/URLSearchParams>
  + This function needs to be placed in main.js of the Ink story.
  + It then needs to pass the value it returns to a function called story.ChoosePathString(), just before line 45 on main.js

const tag = window.location.search.substr(1);

story.ChoosePathString(tag);

* + Alternatively, for debugging:

const tag = window.location.search.substr(1);

try {

story.ChoosePathString(tag);

}

catch(e) {

alert("Unknown knot " + tag);

}

* In addition, the structure of the story should start with a page that redirects to the appropriate knot, ideally contained in a separate .ink to keep everything organised.
  + From there, the knot should divert (tunnel?) to the introduction, also contained in a separate ink file, and then divert back to its original file, based on which knot the user has just visited.

First convo about this with george and vertext-zero, regarding the SearchQuery function within JavaScript: <https://discord.com/channels/329929050866843648/329929390358265857/867024528919887872>

Convo with Nitku about how to edit the main.js file: <https://discord.com/channels/329929050866843648/329929390358265857/867670480148103178>