SQUIRREL U
QUICK CARD

http://zippyzsquirrel.github.io/squirrel-u/

Transform Your Tribal Knowledge Into Learning Trees!

1.PREREQUISITES

1. Obtain GitHub account
2. Follow steps in Squirrel U repository README.
3. IntelliJ installed (Ultimate Edition recommended)
4. Git installed to **C:\Development\Git**
   (if different location adjust settings in IntelliJ).

2.SETUP LOCAL

The following steps will automatically set up a local Squirrel U development environment for IntelliJ:

FORK SQUIRREL U REPOSITORY

Fork the **gh-pages** branch of the repository from:
https://github.com/zippyzsquirrel/squirrel-u/

CREATE INTELLIJ PROJECT

1. **File>New>Project From Version Control>GitHub**
2. Clone **squirrel-u** repository
   2.1.Git Repository URL:
   https://github.com/yourGitName/squirrel-u/
   2.2.Parent Directory:
   /Users/**userName**/IdeaProjects/
   C:\Development\workspace
   2.3.Directory Name: **squirrel-u**

SETUP ENVIRONMENT

The following steps install Ruby, Jekyll and Squirrel U repository content & code.

1. Run the **Build Jekyll** Configuration for your OS:
   Run once; when prompted in terminal:
   Homebrew install location: use **/Development**
   Run twice.
   1. First time installs Ruby. When prompted:
      1.1.Install Ruby to **C:\Development\Ruby**
      1.2.**Check checkbox**
      'Add Ruby Executables to your PATH'
   2. After first install
      2.1.Verify Ruby Devkit installs to
      C:\Development\Ruby\dev_kit
      2.2.Close terminal window, restart IntelliJ
2. Rerun **Build Jekyll-Windows** to finish setup

3. RUN LOCAL

1. IntelliJ: run **Run Jekyll** Configuration for your OS
2. View the Run console. Once you see "… done." Jekyll has loaded your site. Load time is in seconds.
3. To view Squirrel U, open **http://localhost:4000**
4. From that point any time you modify files, Jekyll will only process affected files and reloads are rapid.

4. RESOURCES

Pages	http://pages.github.com
Gem	https://github.com/github/pages-gem
Jekyll	https://jekyllrb.com
Markdown	https://daringfireball.net/projects/markdown/syntax
Emoji	http://www.webpagefx.com/tools/emoji-cheat-sheet/
Liquid	**Shopify**: https://help.shopify.com/themes/liquid/basics
	Cheatsheet: http://cheat.markdunkley.com
IDE Plugins	**IntelliJ** https://vladsch.com/product/markdown-navigator
	Atom: https://atom.io/packages/markdown-writer
	VisualStudio: http://vswebessentials.com
Scaling	**Stephan Baumgartner** https://www.smashingmagazine.com/2016/08/using-a-static-site-generator-at-scale-lessons-learned/

5. WORKFLOW

GH-PAGES

GitHub pages specifically uses **gh-pages**, not master branch. The master branch has been removed. If you are utilizing branches and push to your forked repository, make sure you are pushing to **gh-pages!**

STAY CURRENT WITH CHANGES

1. Avoid conflicts before any merge:
   1.1.Stash current work
   1.2.Select changes to merge when prompted
   1.3. Unstash
2. Update from upstream: **gh-pages** branch
3. Update Approaches:
   3.1.VCS Watch plugin prompts upstream changes
   3.2.VCS -> Git -> Pull

CONTENT DEVELOPMENT CYCLE

1. Create / Update content
2. Commit changes
3. Stash your current work
4. Update from GitHub to stay fresh with upstream
5. Unstash
6. Push changes to your public fork
7. Generate a Pull Request
8. PR is reviewed, comment, merged or closed

6. GUIDELINES

MARKDOWN FORMAT

We are utilizing a markup language called Markdown. This allows for a more concise structured document, which Jekyll converts to HTML.

HIERARCHY

Main Topic Directories

Squirrel U has been divided into 5 major sections:

1. **Quick Start** – Up and running in little time!
2. **Orientation** – history, functionality, & architecture
3. **Local IDE** – developing Squirrel U with local Jekyll
4. **GitHub** – getting started with GitHub
5. **Nuts –** Squirrel Arcana

Summary Directories

Sub-directories within each of the Main Topic Directories.

Content Directories and Files

The "meat" of Squirrel U. Contains all of the reading material, examples, videos and images that cover a particular topic.

STANDARDS

- **Summary directory**
  - Provide a **0_Summary.md** file with appropriate title in Front Matter
- **Content directory naming schema:**
  - #_TopicName
    - Prefix directory with # [1-9]
      - This automates content ordering rendering
    - TopicName should be in CamelCase
  - You must include an **index.md** file
  - Images should be in a peer directory called "images"
- **Content Markdown files**
  - On occasion you may create a single .md file for your topic.
  - Follow same naming scheme
  - An index.md file will automatically be created
  - Images should go at a peer directory level and use ../ appropriately when referencing the image
- **Global images**
  - On occasion there are global images you can use in your content. These are referenced using /images.
  - Minimize linking to Squirrel U content from other content . If you do, use "refactor" to check for references in IntelliJ to avoid breakage

PAGE LAYOUT

1. **Front Matter**
   - 1.1.Layout - "summary" or "page"
   - 1.2.Title - Use for page display
   - 1.3.Breadcrumb - Used for Hierarchy and Search
   - 1.4.Author
   - 1.5.Status - Nothing or "under construction"
   - 1.6.Content type - video/code
   - 1.7.Tags - Relevant keywords for searches
     example: keyword1, keyword2, keyword3
2. **Table of Contents**
   - 2.1.Must include **{% include toc.html %}**
     - This will build table of contents
     - To omit headers from TOC , place **{:.no_toc}** after header
3. **Introduction** - What reader will learn
4. **Prerequisites** - if needed
5. **Topic headers and Sub-headers**
   - 5.1.Organize content meaningfully
   - 5.2.Use sub-headers prudently; 3 levels recommended
6. **Steps** - If needed
7. **For Further Reading** - related material in external links

SEARCH

Accurate information in the Front Matter is extremely important. Title, Breadcrumb, Author, and Tags are all used building the search index and will provide better results for searching.

7. MARKDOWN

CONTENT BASICS

Emphasis: *italic* **bold**

Line break: end line with two or more spaces

Titled Link: [title phrase](http://example.com)

Actual Link: <http://example.com>

Image:

Emojis: **:emoji name:** Ex. :smile:

STRUCTURING CONTENT

Headers: # Head 1

Head 2, …

Unordered List:

* item 1
  * item 1a

Ordered List:

1. item 1
   * item 1a

Tables:

- "Pretty" alignment

```
| Table | Header | Row  |
|-------|--------|------|
|Col 1  |Col 2   |Col 3 |
|Col 1a |Col 2a  |Col 3a|
```

- Use colons (:) to right-align or center columns

```
| Table | Header   | Row        |
|-------|:--------:|-----------:|
|Normal |Centered  |Right Align|
|Col 1  |Col 2     |Col 3       |
```

SENTENCES AND PARAGRAPHS

Text Blocks:

- > at the beginning of each line
- `some prose or code`
- ```some lines …```

Code Syntax Highlighting

- ```***languageStyle***

  *content*

  ```

Note: ***languageStyle*** supports many styles including:
html | javascript | json | groovy | java | xml | sql | shell
  ```