

Introduction to GitHub



Image: <https://github.com>

Sebastian Schwindt

sschwindt@ucdavis.edu | <https://sebastian-schwindt.org>

Kenneth Larrieu

kglarrieu@ucdavis.edu

Prof. Gregory B. Pasternack lab – <http://pasternack.ucdavis.edu>

Department of Land, Air, and Water Resources
239 Veihmeyer Hall, University of California, Davis
Davis, CA 95616-8628

Davis, CA | August 28, 2019

NOT SO LONG AGO,
IN A GALAXY CLOSE BY-

HEY GEORGE
WHAT'S UP?

I ACCIDENTILY
DELETED ANOTHER
PAGE OF MY
MANUSCRIPT--

NOT THIS STUPID
'SUN BATTLE' THING
AGAIN--

OH WELL...
YOU HAD IT ALL
UNDER VERSION
CONTROL RIGHT?

IT'S NOT STUPID!
- YOUR STUPID!

VERSION CON-WHAT?

UGH...

Table of contents

1. Creating, managing, and updating repositories

- Create a new repository
- Clone repositories
- Make, commit, and push changes
- Update (pull) local repositories
- Work with branches

2. Markdown and documentation

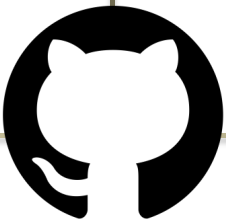
- Markdown principles
- Write a README.md
- Write Wikis

3. GitHub Pages

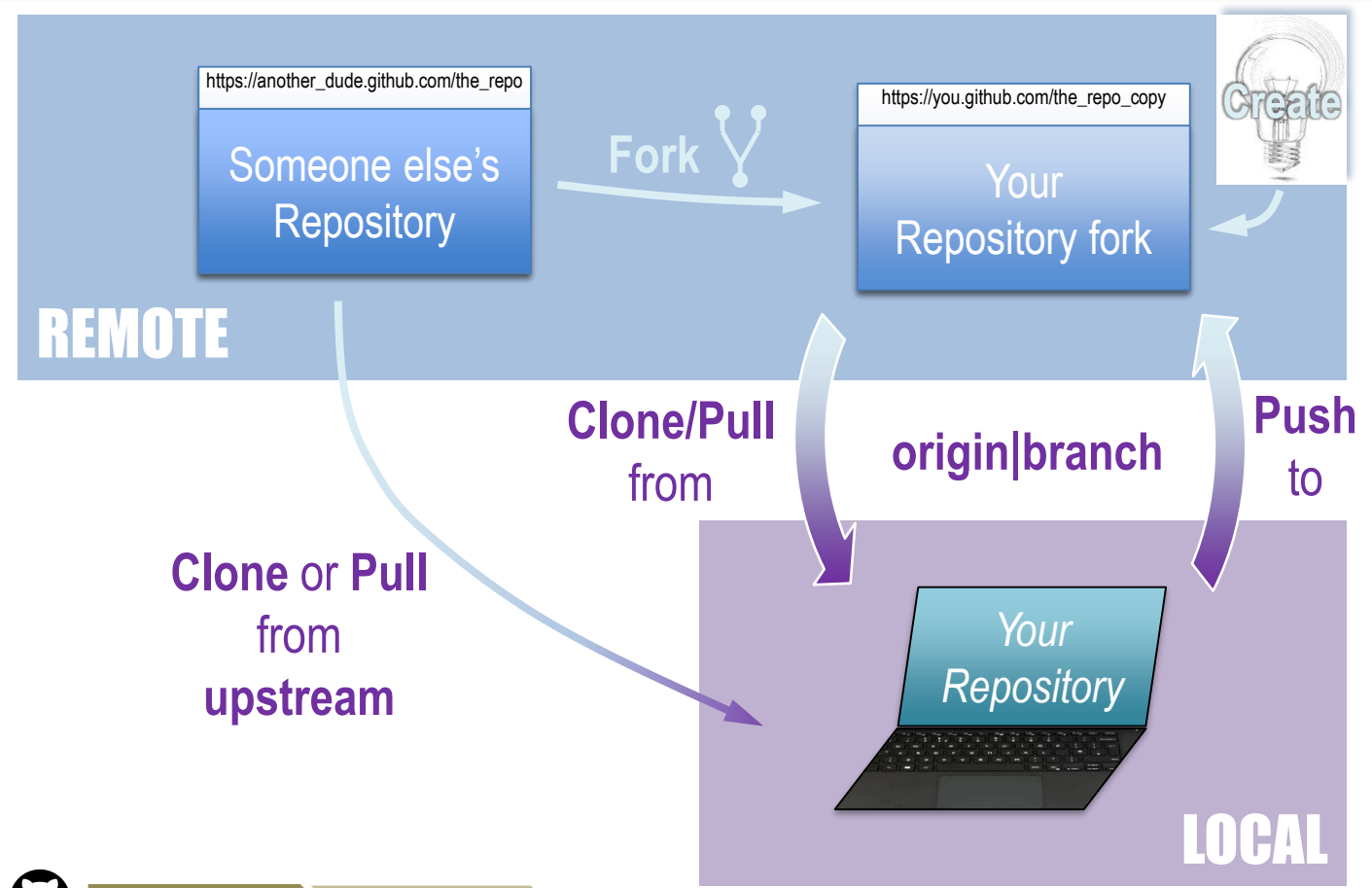
- Install and modify (basic) Themes
- Setup master (home) pages: <https://my-master.github.io>



Creating, managing, and updating repositories



How it works & git vocabulary



Create a **remote** Repository

Option 1: Fork an existing repository.

Option 2: **Create a new repository.**

1. Go to <https://github.com> and sign in.

2. In the upper left corner click on Repositories  .

3. Name your *public* repository (e.g., *juices*), add a description, check *Initialize ... with a README*, **select** *Add .gitignore: ...*, and *Add a license: ...*

Skip this step if you're importing an existing repository.

☒ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer.

Add .gitignore: **Python** ▼

Add a license: **MIT License** ▼



*Liability for
your work!*

*Exclude file types from
push/pull actions*

4. Click on  .



Repository

Create

Create a Repository

sschwindt / fruits

Unwatch1

Star0

Fork0

<> Code

Issues0

Pull requests0

Projects0

Wiki

Security

Insights

Settings

A tasty training repository.

Edit

Manage topics

1 commit

1 branch

0 releases

1 contributor

Branch: masterNew pull request

Create new fileUpload filesFind FileClone or download

sschwindt Initial commit

Latest commit 89c1a52 2 minutes ago

.gitignore

LICENSE

README.md

Initial commit

Initial commit

Initial commit

2 minutes ago

2 minutes ago

2 minutes ago

Commit messages

README.md

Remote editing

fruits

A tasty training repository.

Only visible to you (owner)

Commit messages

Remote editing



Clone a Repository (make **local**)

Option 1: Download as .zip file (quick-&-dirty).

Option 2: **Pull with git bash.**

1. **Create a new folder for GitHub repositories (e.g., `D:\GitHub\`).**
2. **Open Git Bash  (Windows) / command line (Linux).**
→ Requires that git is installed (<https://git-scm.com/downloads>)
3. **Navigate to the GitHub folder (git bash):** `cd "D:/GitHub/"`
4. **Clone the before created repository:**

```
git clone https://github.com/sschwindt/fruits
```

*Keep Git Bash open
– we'll need it!*

Verify that the folder `D:\GitHub\fruits` contains the repository (explorer).



Repository



Clone

Modify a Repository (**local**)

Modify *D:\GitHub\fruits \README.md* **by using any text editor:**



The screenshot shows a Typora text editor window titled "README.md - Typora". The window has a menu bar with "File", "Edit", "Paragraph", "Format", "View", "Themes", and "Help". The main editing area contains two lines of text: "# Welcome to my digital fruit stand" and "Anything but icecream.". The text is in a pinkish-red color. Line numbers "1" and "3" are visible on the left side of the editor. At the bottom of the window, there is a status bar that says "</> Exit Source Code Mode".

```
# Welcome to my digital fruit stand
Anything but icecream.
```



Push **local** changes to a **remote** Repository

Go back to Git Bash :

1. **Go to local copy:** `cd fruits`

2. **Verify status of local copy:** `git status`

→ Git Bash should prompt that README.md was modified (nothing else).

3. Optional: see what has been changed: `git diff`

4. **Add (stage) changes:** `git add` 

Determines what is added.

git add README.md would be sufficient here.

.gitignore provides enhanced control

5. **Commit changes:**

`git commit -m "Modify Readme"`

Keep track of commits by adding a message (-m).

5. Optional: update local branch (more later on): `git pull --rebase`

6. **Push changes:** `git push` (user name and password will be queried)

It may be necessary to set the upstream origin : `--set-upstream origin master`



Push local changes to a remote Repository

7. Optional: View commit history: *git log*

OR

View detailed commit history on GitHub

sschwindt / fruits

Watch

1

Star

0

Fork

0

Code

Issues

Pull requests

Projects

Wiki

Security

Insights

A tasty training repository.

5 commits

1 branch

0 releases

1 contributor

MIT

Branch: master

New pull request

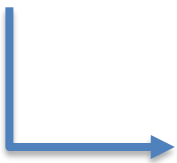
Create new file

Upload files

Find File

Clone or download

sschwindt	Update _config.yml	Latest commit 480346a 22 hours ago
gitignore	Initial commit	5 days ago
LICENSE	Initial commit	5 days ago
README.md	Modify Readme	5 days ago
_config.yml	Update _config.yml	22 hours ago



Modify Readme

Browse files

master

sschwindt committed 5 days ago

1 parent B9c1a52 commit B64da9dc50eb3e95820a180b32d26975fbf4e06f

Showing 1 changed file with 2 additions and 2 deletions.

Unified Split

4 README.md

... @@ -1,2 +1,2 @@

1 - # fruits

2 - A tasty training repository.

1 + # Welcome to my digital fruit stand

2 + Anything but icecream.



Updating & merging local repositories

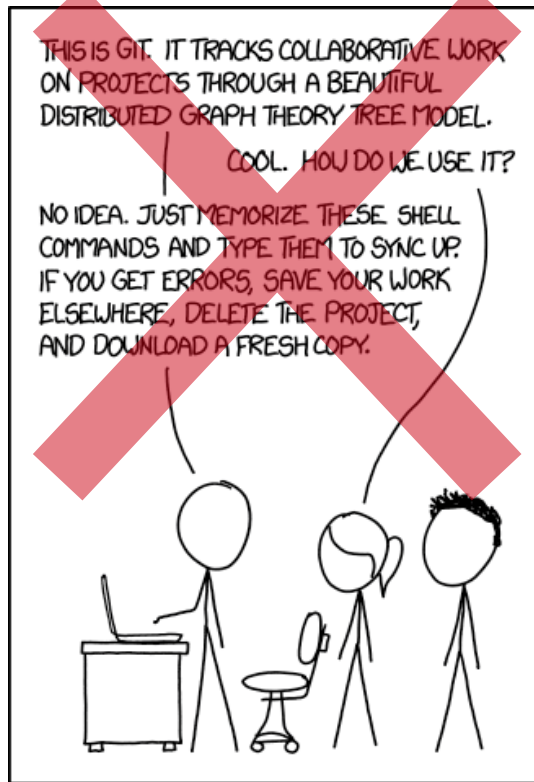


Image: xkcd 1579 CC BY-NC 2.5

No!



*Let's do an
exercise ...*



Repository

Update

Pull/merge changes from a **remote** Repository

Partner exercise

Partner A

Invite a partner to your repository

1. **Go to your repo on GitHub:** e.g., `https://github.com/klarrieu/veggies`
2. **Select** *Settings* → *Collaborators* **and add partner B.**

Partner B

Accept the invitation to collaborate & clone the repository

1. **Go to local folder:** `cd "D:/GitHub"`
2. **Clone the repository:** `git clone https://github.com/klarrieu/veggies`
3. **Go to local copy:** `cd veggies`

Now both partners should have a local copy of A's repository.



Pull/merge changes from a **remote** Repository

Both partner A and B

→ **Modify** *README.md* in your **local** repository

→ **Add** (*git add .*), **commit** (*git commit -m "I did...omg"*), and **push** (*git push*) your changes

... and see what happens.



Pull/merge changes from a **remote** Repository

Modifying your local copy, it appears nothing is wrong...

```
C:\Users\Neth\Code\veggies>git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\Neth\Code\veggies>git add .

C:\Users\Neth\Code\veggies>git commit -m "ate my veggies"
[master 4764fa5] ate my veggies
1 file changed, 4 insertions(+)
```



Pull/merge changes from a **remote** Repository

...but when you try to push your changes to **remote**, you see this!

```
C:\Users\Neth\Code\veggies>git push
To https://github.com/klarrieu/veggies.git
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://github.com/klarrieu/veggies.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

While we were modifying the **local** repo, changes were made to **remote**.

We need to pull the **remote changes before we can integrate our **local** changes** (*git pull --rebase*).



Pull/merge changes from a **remote** Repository

Merge conflict: Multiple contributors pushed changes to the same lines in the same file.

→ Git does not know which changes should be kept.

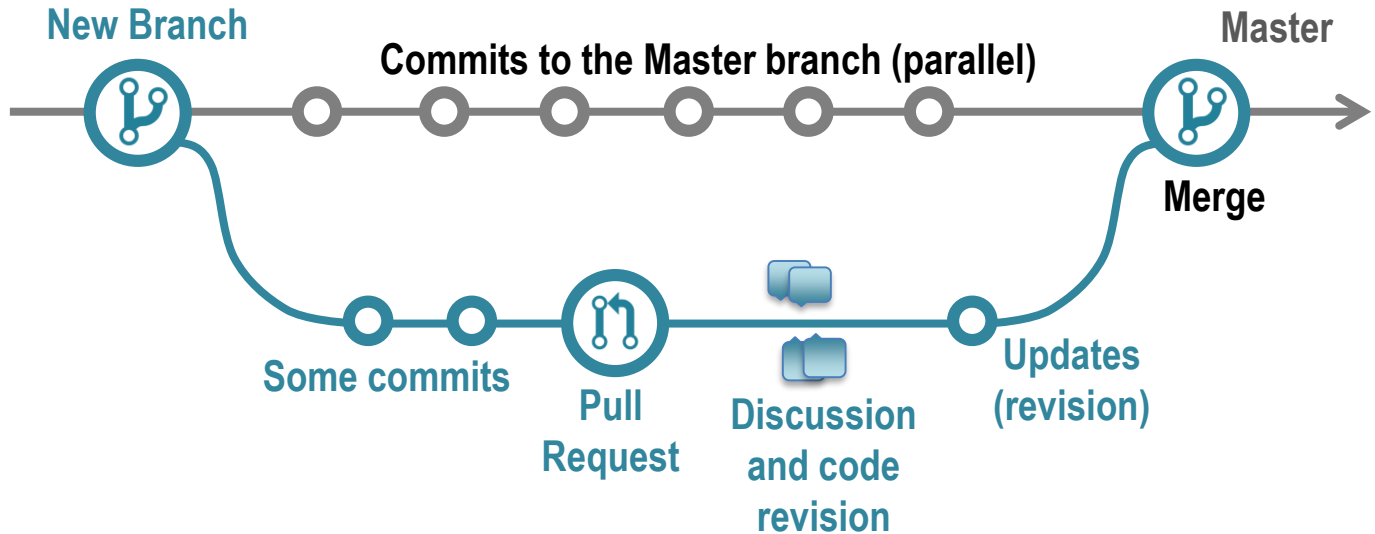
```
C:\Users\Neth\Code\veggies>git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/klarrieu/veggies
   f45ef6d..72938bd  master    -> origin/master
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
```

Remedy:

- 1. Open the file where the merge conflict occurred (*README.md*)**
- 2. Edit the file, deciding which lines of code to keep in the process.**
- 3. Add (*git add .*), commit (*git commit -m "Fixed merge conflict"*), and push (*git push*) your changes.**



Working with branches



Further reading: <https://help.github.com/en/articles/about-branches>



Markdown & Documentation



What is Markdown

Created in 2004, Markdown became a popular **markup** language.

Syntactically distinguishable computer text

Syntactic text

```
# Better than Word/Richtext
*OS*-independent
functionality:
- Avoid formatting of the
  same kind of thing
  redundantly (and
  inconsistently)
- Backwards compatibility
- Formulae handling ... etc.
```

Marked up

Better than Word/Richtext

OS-independent functionality:

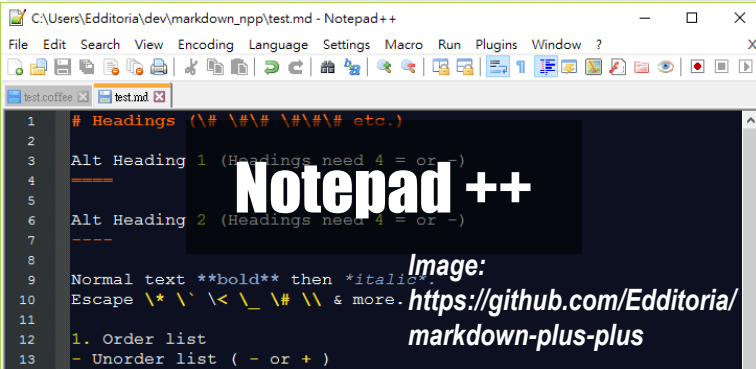
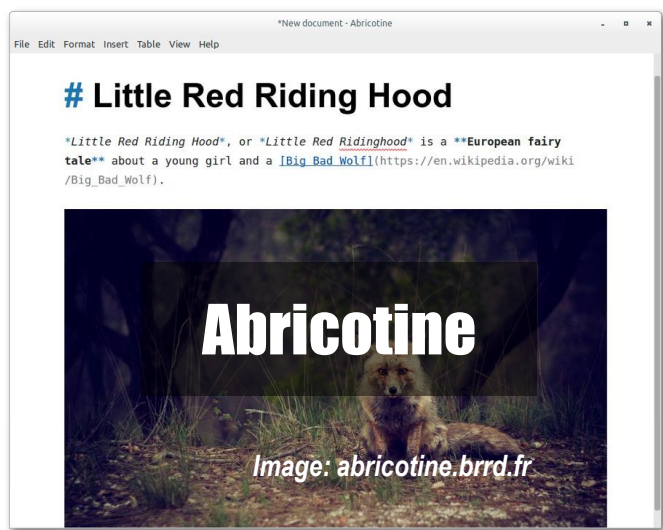
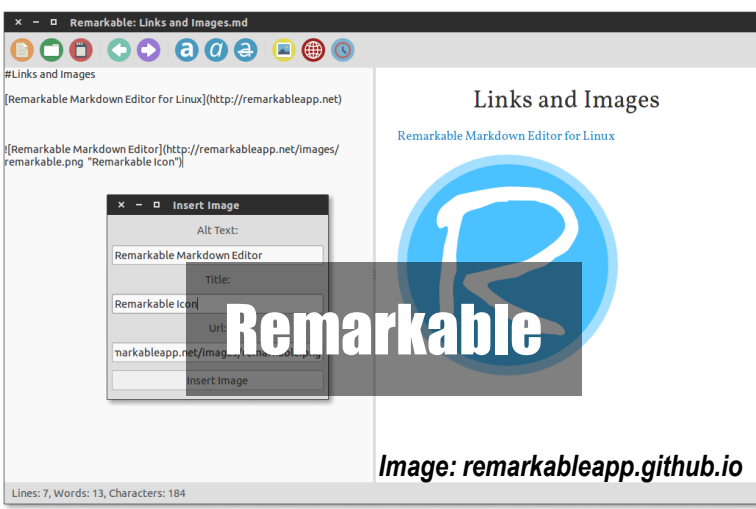
- Avoid formatting of the same kind of thing redundantly (and inconsistently)
- Backwards compatibility
- Formulae handling ... etc.

► Popular markup languages: LaTeX, XML, HTML, **Markdown** (simplified HTML)

► Popular editors: Remarkable, AbriCotine, Mark Text, Typora



Get your favorite ...



How to use Markdown (.md) (1)

Italic

****Bold****

~~~~Strikethrough~~~~

# Heading 1

## Heading 2

[Link] (<https://fruitsinfo.com>)

![ImgName] ([https://raw.githubusercontent.com/RiverArchitect/Media/master/images/RA\\_icon.png](https://raw.githubusercontent.com/RiverArchitect/Media/master/images/RA_icon.png))

*Italic*

**Bold**

~~Strikethrough~~

**Heading 1**

**Heading 2**

[Link](#)



# How to use Markdown (.md) (2)

> Blockquote

- \* List item 1
- \* List item 2

1. Numbered list item 1
2. Numbered list item 2

\*\*\* (Horizontal Rule)

Embedded ``inline code``

```

# code block

print ('To be or not to be')

```

| Blockquote

- List item 1
- List item 2

1. Numbered list item 1
2. Numbered list item 2

---

(Horizontal Rule)

Embedded `inline code`

# code block

print ('To be or not to be')



# How to use Markdown (.md) (3)

A table

```
| Fruit | Kingdom | Genus |  
|-----|:-----:|-----:|  
|Banana | Plantae | Musa |  
|Jackfruit|Plantae|Artocarpus|
```

More linking

```
[[InternalGithubPage]]  
[InternalPageSection] (Internal  
PageSection#Header-name)  
Go to section    Space = "-"  
[Defined Reference][1]  
  
[1]: https://wikipedia.org
```



A table

| Fruit     | Kingdom | Genus      |
|-----------|---------|------------|
| Banana    | Plantae | Musa       |
| Jackfruit | Plantae | Artocarpus |

More linking

```
InternalGithubPage  
InternalPageSection
```

Defined Reference



# Edit README.md

Go back to Git Bash :

1. Go to local copy of Repository: `cd fruits`
2. Verify status of local copy: `git status`
3. In the explorer: Open README.md with Markdwon Editor and rewrite it: README.md = your automated welcome page



The screenshot shows a window titled 'README.md - Typora' with a standard menu bar (File, Edit, Paragraph, Format, View, Themes, Help). The editor content shows a Markdown heading on line 1: '# Welcome to my digital fruit stand' in pink, followed by the text 'Anything but icecream.' on line 2. Line 3 is empty. At the bottom, a status bar indicates '</> Exit Source Code Mode'.



# Push **local** changes to **remote**

Go back to Git Bash :

**1. Verify status of local copy:** *git status*

→ Git Bash should prompt that README.md was modified (nothing else).

**2. Add (stage) changes:** *git add .*

**3. Commit changes:**

*git commit -m "Make Readme Great Again"*

**5. Update local branch** (just good practice...): *git pull --rebase*

**6. Push changes:** *git push*



## Working collaboratively is more than sharing codes ...



### RIVER ARCHITECT

Wiki & User Guide

Download ZIP

Go to Main

View On GitHub

Home

- ▶ Installation
- ▶ Get Started
- ▶ Lifespans
- ▶ Morphology (Terraforming)
- ▶ Ecohydraulics
- ▶ ProjectMaker

Add new | Edit | History | Source

Search...

## RIVER ARCHITECT WIKI

▶ Table of Contents

Please refer to the [installation guide](#) for first-time installation, update, and launch assistance.

*River Architect* serves for the GIS-based planning of habitat enhancing river design features regarding their lifespans, parametric characteristics, optimum placement in the terrain, and ecological benefit. A main graphical user interface (GUI) provides five modules for generating lifespan and design maps, action (optimum lifespan) maps, terrain modification (terraforming) assessment of digital elevation models (DEM), habitat evaluation, and project cost-benefit analyses.

*River Architect* invites to analyses and modifications of the longevity and ecological quality of riverscapes. Different planning bases ("conditions") can be easily created using an introductory module called *GetStarted*. Lifespan, Morphology (Terraforming) and Ecohydraulic assessments can then be created based on the *Conditions*, including the creation of project plans and cost-benefit maps, a Project Maker module.

Lifespan maps indicate the expected longevity of restoration features as a function of terrain change, hydrology, and 2D hydrodynamic modeling results. Design maps are a side product of lifespan and design mapping. Feature dimensions for stability, such as the minimum required size of angular boulders to avoid them (more information in [Schwindt et al.2019](#)). Best lifespan maps result from the comparison of lifespan of restoration features and assign features with the highest longevity to each pixel of a raster. This allows for the optimum features as a function of highest lifespans among comparable feature groups such as different species.

Morphology (Terraforming) includes routines to Modify Terrain for river restoration purposes. Two algorithms are implemented: (1) Threshold value-based terrain modifications in terms of riparian forest establishment; and (2) River Builder for the creation of synthetic river valleys. It compares an original (pre-project or pre-terraforming application) and a modified DEM (post-project or post-terraforming application) to determine required earth movement (terraforming volumes) works.



## Enable Wiki for a repository

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

Settings

Options

Collaborators

Branches

Webhooks

Notifications

Integrations & services

Deploy keys

Moderation

Interaction limits

Settings

Repository name

fruits

Rename

☐ Template repository

Template repositories let users generate new repositories with the same directory structure and files. I used as a template for creating other repositories.

Social preview

Upload an image to customize your repository's social media preview.

Images should be at least 640×320px (1280×640px for best display).

Download template

Edit

Features

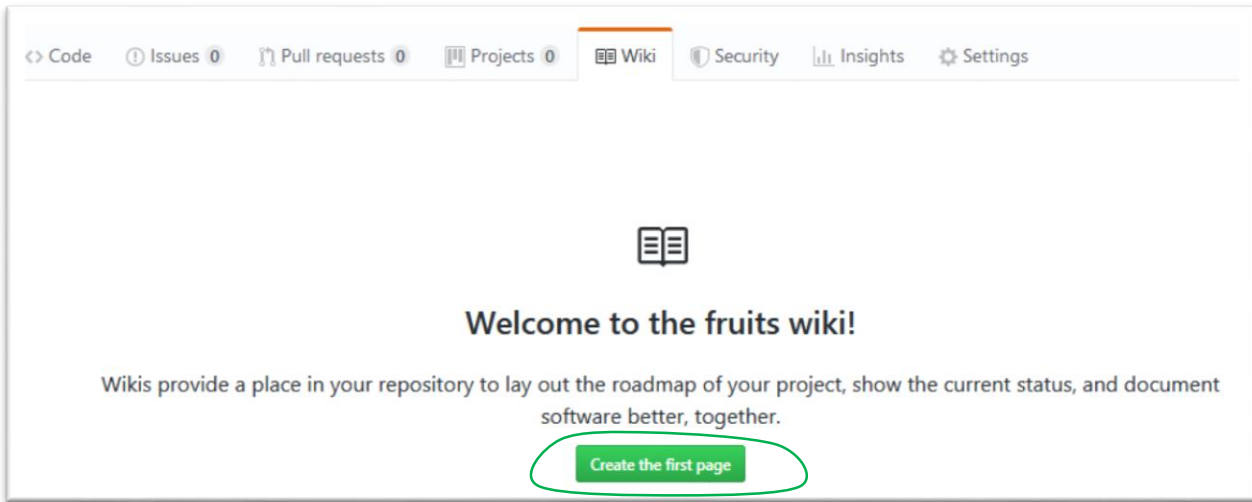
☒ Wikis

GitHub Wikis is a simple way to let others contribute content. Any GitHub user can create and edit documentation, examples, support, or anything you wish.



# Wikis

## Get started



Note: The Wiki is a repository within the repository

→ It can be cloned like any other repository:

```
https://github.com/YOUR_ID/REPO.wiki.git
```



## Get started: Create “Home”.md

Home

Define page name  
(Home.md)

Write

Preview

Verify modifications

h1

h2

h3

B

i

<>

Edit mode: Markdown

Welcome to the fruits wiki!

Facts about digital fruits, their methods (?) and usage (well... just eat it).

Markdown text

Edit message

Initial Home page

“Commit” message



## Get started: Add a sidebar

<> Code

🔔 Issues 0

🔗 Pull requests 0

📁 Projects 0

📖 Wiki

🛡 Security

📊 Insights

⚙ Settings

Home

EditNew Page

Welcome to the fruits wiki!

Facts about digital fruits, their methods (?) and usage (well... just eat it).

+ Add a custom footer

\_Sidebar

WritePreview

h1h2h3

Bi<>

Edit mode: Markdown

Menu

- [[Home]]

- [2beCreated](2beCreated#A-section)

▼ Pages 1

Find a Page...

Home

+ Add a custom sidebar

Clone this wiki locally

https://github.com/sschwi

Wikis

Setup

31

## Get started: Add another page

Home

EditNew Page

Welcome to the fruits wiki!

Facts about digital fruits, their methods (?) and usage (well... just eat it).

+ Add a custom footer

Pages 1

Menu

- Home
- 2beCreated

Create new page

2beCreated

WritePreview

h1h2h3

**B***i*`<>`

- 

-

Edit mode: Markdown

# The Section

People throughout the ages have eaten some things that may seem strange to modern palates. Let's start with ancient Egypt. Which of these animals was considered a delicacy in the land of the Nile?

- \* Vultures
- \* Cats
- \* Hedgehogs
- \* Rats

# A section

The Inuit people of Greenland have made few contributions to the world of haute cuisine. One of their delicacies is a food called kiviat. What is it?

Remember

[2beCreated] (#A-section)

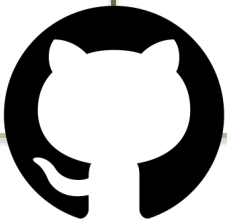
Alternative section link name

<a name="other"></a>





## **GitHub pages**



# GitHub Pages

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

Settings

A tasty training repository.

Manage topics

Edit

2 commits

1 branch

0 releases

1 contributor

MIT

Branch: master

New pull request

Create new file

Upload files

Find File

Clone or download

sschwindt

Modify Readme

Latest commit 864dabd 4 days ago

.gitignore

Initial commit 4 days ago

LICENSE

Initial commit 4 days ago

README.md

Initial commit 4 days ago

README.md

https://github.com/YOUR\_ID/REPO

✎

Welcome to my digital fruit stand

Anything but icecream.



# PYDROSCAPE


Python Tools for Riverscape and Hydraulic Data Analyses

[View On GitHub](#)

DOWNLOAD:

ZIP

TAR



[Introduction](#)  
[Requirements](#)  
[Usage](#)  
[Documentation \(Wiki\)](#)

## Introduction

`pydroscape` provides Python3 tools for many sorts of riverscape-related analyses, including tools for geodata statistics, plotting and processing of other data.

Installed as local package, the following modules can be imported from `pydroscape.e...`:

- `e_geodata`: Performs faster calculations. [More ...](#)
- `e_plot`: x-y, 2D (heatmap), and 3D plots with numpy and matplotlib. [More ...](#)
- `e_xlsx`: Process workbooks. [More ...](#)
- `e_data`: Process experimental data recorded with any kind of data logger. [More ...](#)
- `e_sed1d`: Calculate sediment transport based on 1D cross-section-averaged hydraulic simulations. [More ...](#)

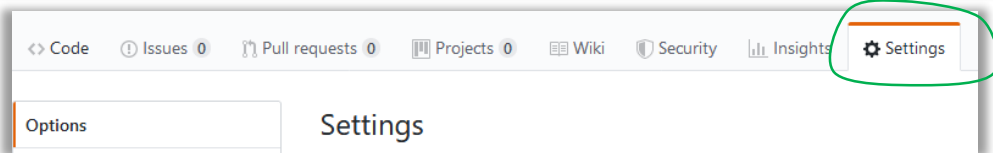
## Requirements

- Python 3.x
- Basic packages: `numpy`, `matplotlib`, `openpyxl`
- Supplemental packages (required by some functions only): `pandas`, `qgis.core`, `osgeo` ( `gdal` )

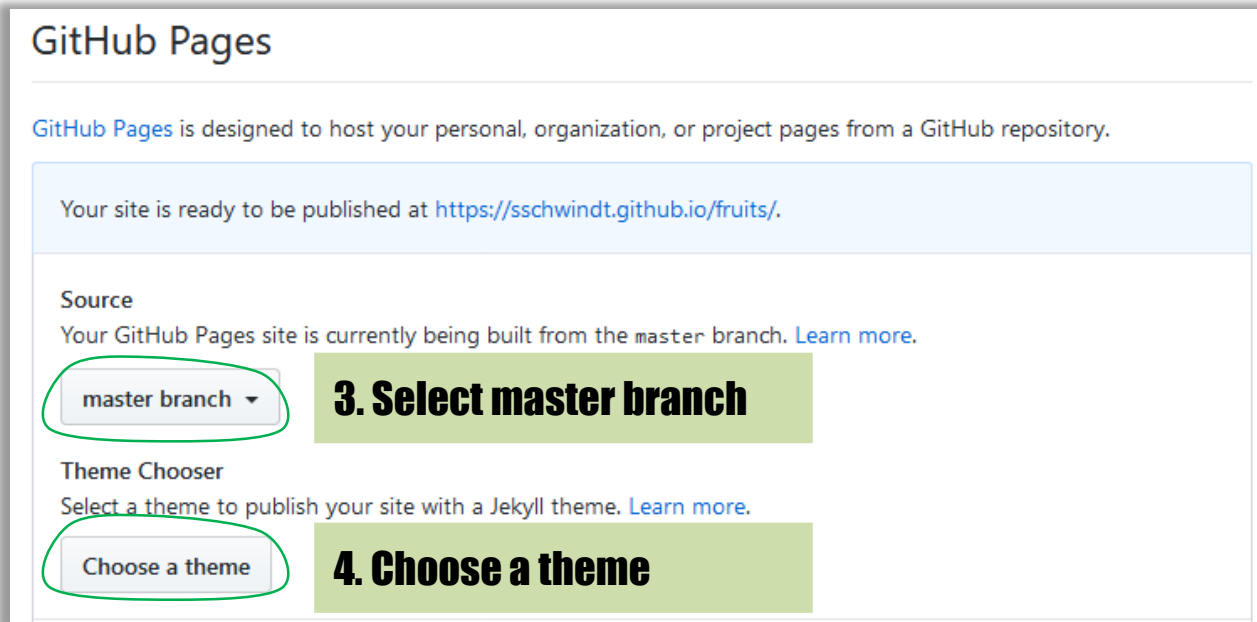


# GitHub Pages

## 1. Go to Settings

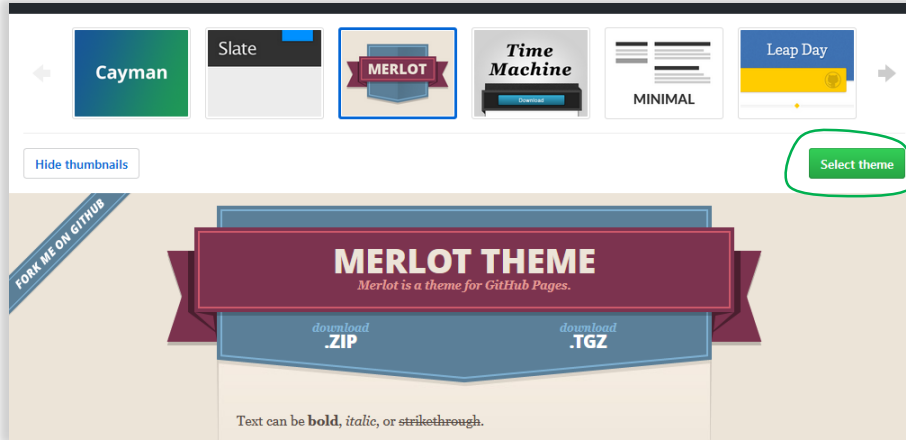


## 2. Scroll to GitHub Pages



# GitHub Pages

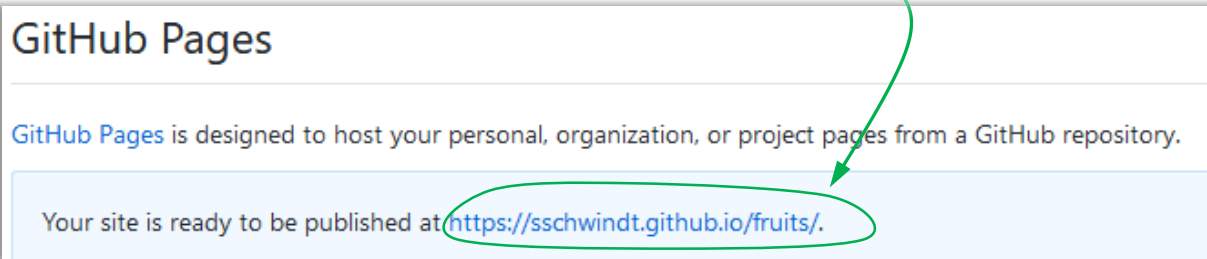
## 1. Select a Theme



*More Themes are available (maybe tricky to install...)*

<https://jekyllthemes.io/github-pages-templates>

## 2. Wait 2-3 Minutes, then check out GitHub page



# GitHub Pages

Looks nice, but there's room for optimization. The `_config.yml` file provides some options:

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

Settings

A tasty training repository.

Edit

Manage topics

3 commits

1 branch

0 releases

1 environment

1 contributor

MIT

Branch: master

New pull request

Create new file

Upload files

Find File

Clone or download

sschwindt Set theme jekyll-theme-merlot

Latest commit a085906 8 minutes ago

.gitignore

Initial commit

4 days ago

LICENSE

Initial commit

4 days ago

README.md

Modify Readme

4 days ago

\_config.yml

Set theme jekyll-theme-merlot

8 minutes ago

Open (click on) `_config.yml` & Start editing

Raw

Blame

History



# GitHub Pages

**Modify `_config.yml` & adapt it to your needs. Some options:**

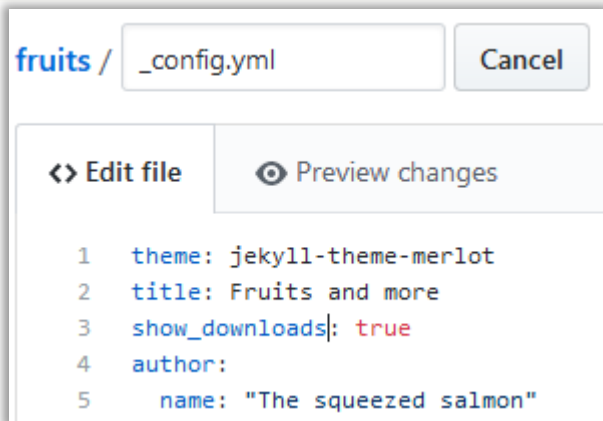
| Keyword:                                        | Possible Values                   | Description                                                                                                                                       |
|-------------------------------------------------|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>theme:</code>                             | <code>"jekyll-theme-NAME"</code>  | <i>Define / modify the installed theme.</i>                                                                                                       |
| <code>title:</code>                             | <code>"Another title name"</code> | <i>Modify the title shown on github pages.</i>                                                                                                    |
| <code>description:</code>                       | <code>"Text"</code>               | <i>Modify the repository descriptions</i>                                                                                                         |
| <code>show_downloads:</code>                    | <code>BOOL (true or false)</code> | <i>Enable download buttons.</i>                                                                                                                   |
| <code>analytics:</code><br><code>google:</code> | Google Analytics ID               | <i>Use SEO / webmaster tools (also available for bing)</i>                                                                                        |
| <code>author:</code><br><code>name:</code>      | <code>"Text"</code>               | <i>E.g., "Ms. X"</i>                                                                                                                              |
| <code>timezone:</code>                          | <code>America/Los_Angeles</code>  | <i>Uses <a href="http://en.wikipedia.org/wiki/List_of_tz_database_time_zones">http://en.wikipedia.org/wiki/List_of_tz_database_time_zones</a></i> |



# GitHub Pages

**After editing, save and wait for  
2-3 minutes.**

**Then refresh :**



## GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

✓ Your site is published at <https://sschwindt.github.io/fruits/>

*If variables were badly defined, you'll receive an email with a site generation error notice.*





# GitHub Pages

- There are more options to adapt templates

Find the default.html of the selected Jekyll-theme and copy it into your repository :

1. Create `_layouts/default.html`
2. Edit `_layouts/default.html`

*Note: HTML skill are required.*

- Creating a “home” repository: [https://YOUR\\_ID.github.io](https://YOUR_ID.github.io)

*This is how GitHub Pages can constitute you homepage, cheap and well referenced in google (SEO is another topic though).*

1. Create a new repository called [YOUR\\_ID.github.io](https://YOUR_ID.github.io)
2. Install a GitHub pages theme
3. Modify & adapt it to your needs.



**Thank you for listening.**

Get inspired by the River Architect Repository

<https://riverarchitect.github.io>

More information and contact

<https://sschwindt.github.io>

<https://github.com/klarrieu>

