

# Publishing

# The basics of web hosting

# Files are real

TheVerge

SUBSCRIBE 

+ TECH + REPORT + LAPTOPS

## File not found

A generation that grew up with Google is forcing professors to rethink their lesson plans

by + Monica Chin  
*Illustrations by Micha Huigen*  
Sep 22, 2021, 8:00 AM EDT

 0 Comments



Monica Chin, "File not found," *The Verge*, September 22, 2021

# Files live on computers



# Websites are files



```
<!DOCTYPE html>
<html>
<head>
  <title>Page title</title>
</head>
<body>
  <h1>A heading</h1>
  <p>Some text</p>
</body>
</html>
```



```
h1 {
  color: "red";
  text-align: center;
  margin-bottom: 10px;
}
```



Other files



```
const listItems = document.querySelectorAll('ul li');

function toggleDone(e) {
  if (!e.target.classList.contains('done')) {
    e.target.classList.add('done');
  } else {
    e.target.classList.remove('done');
  }
}

listItems.forEach((item) => {
  item.addEventListener("click", toggleDone);
});
```

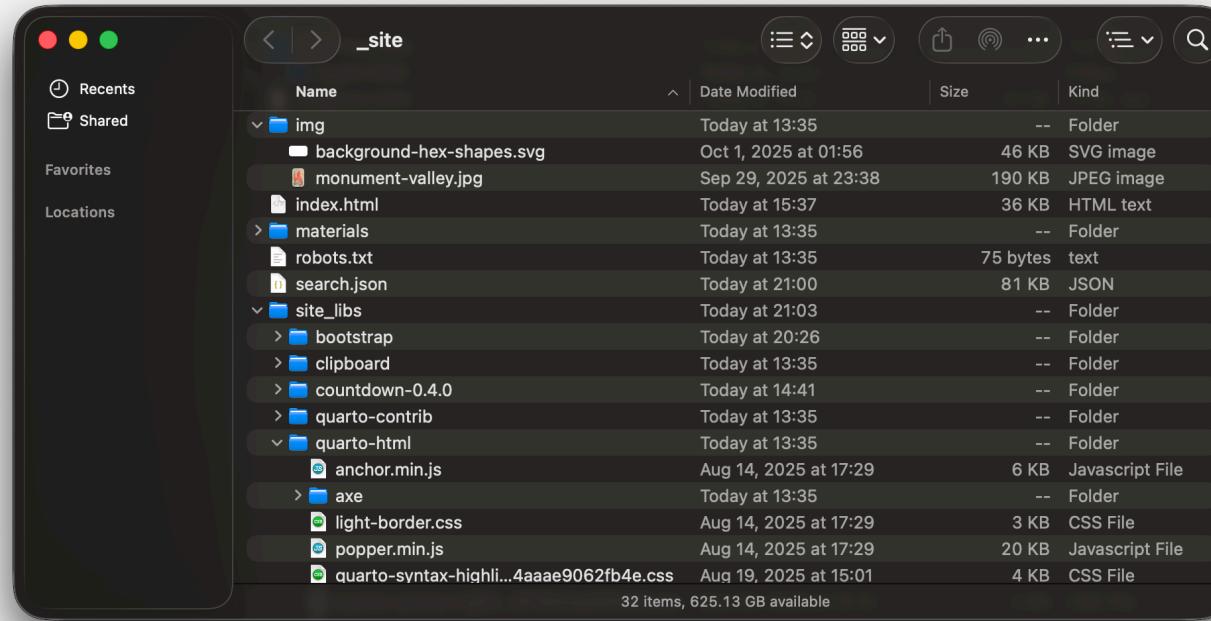
# (Some are databases + files)

The screenshot shows the WordPress dashboard interface. The left sidebar contains navigation links: Home, Updates, Posts, Media, Pages, Comments, Appearance, Plugins, Users, Tools, Settings, and a Collapse menu. The main content area includes:

- Site Health Status**: Shows a green "Good" status with a message about site health and a link to the Site Health screen.
- At a Glance**: Displays 1 Post, 1 Page, and 1 Comment.
- Activity**: Shows a recently published post titled "Hello world!" from "A WordPress Commenter". It also lists recent comments and provides links to manage them.
- Quick Draft**: A form for creating a new draft post, including fields for Title and Content, and a "Save Draft" button.
- WordPress Events and News**: A section listing upcoming events, such as the "WordPress 6.5 Brighton Launch Party" on Tuesday, March 26, 2024, and "Secure Your Site: Join Cambridge WordPress Backups & Security Meetup Apr8th 7pm" on Monday, April 8, 2024.

# Quarto websites are *static*

There's no need for fancy web servers—  
Quarto puts the whole rendered website in `_site/`



Your job is to put that `_site/` on a server.

# Servers

Web servers are computers that (1) store files and (2) show those files as websites to web browsers.

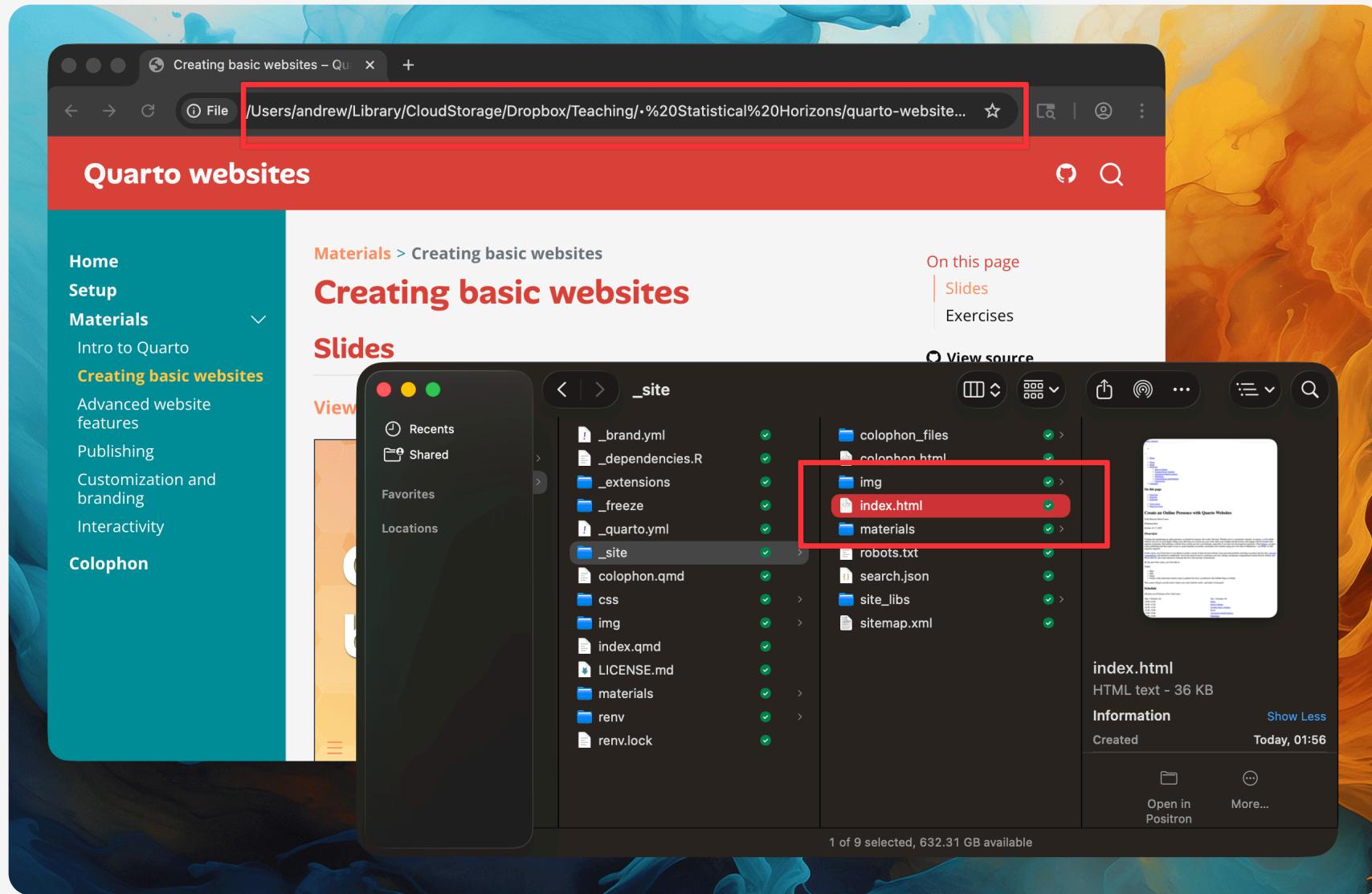


# Local files

file:///Users/andrew/Desktop/example/index.html

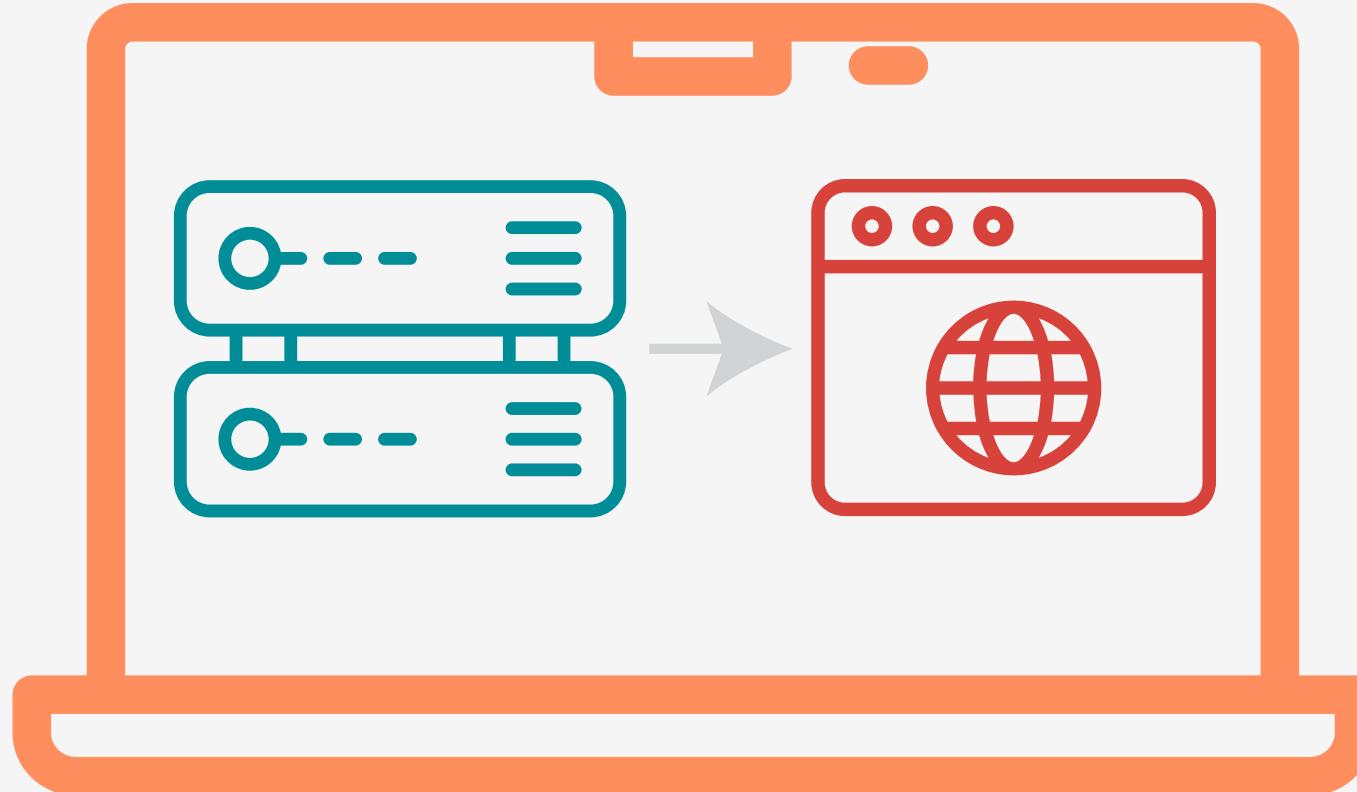


# Local files



# Local server

`http://localhost:3333/`



# Local server





Stephen A Smith @stephenasmith

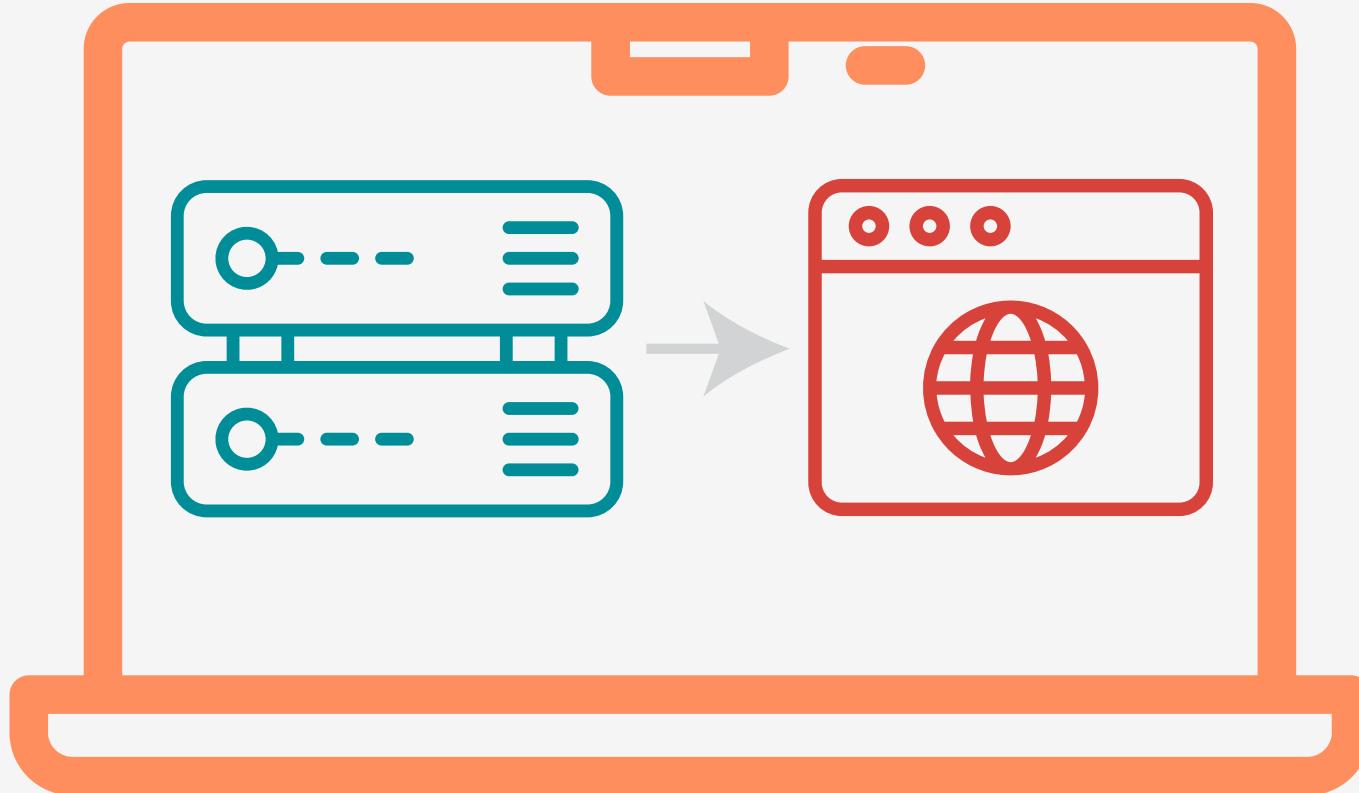
TAke a look, y'all: IMG\_4346.jpeg

10:15 PM · May 26, 2015 · Twitter for iPhone

---

**86.8K** Retweets    **4,939** Quote Tweets    **121K** Likes

# Local server



# Remote server



# Remote server

Creating basic websites – Quarto website · andrewheiss.github.io/quarto-websites\_2025-10/materials/basic-websites/

## Quarto websites

Home  
Setup  
Materials  
Intro to Quarto  
**Creating basic websites**  
Advanced website features  
Publishing  
Customization and branding  
Interactivity  
Colophon

Materials > Creating basic websites

# Creating basic websites

## Slides

[View slides in full screen](#)



On this page

- Slides
- Exercises

[View source](#)  
[Report an issue](#)

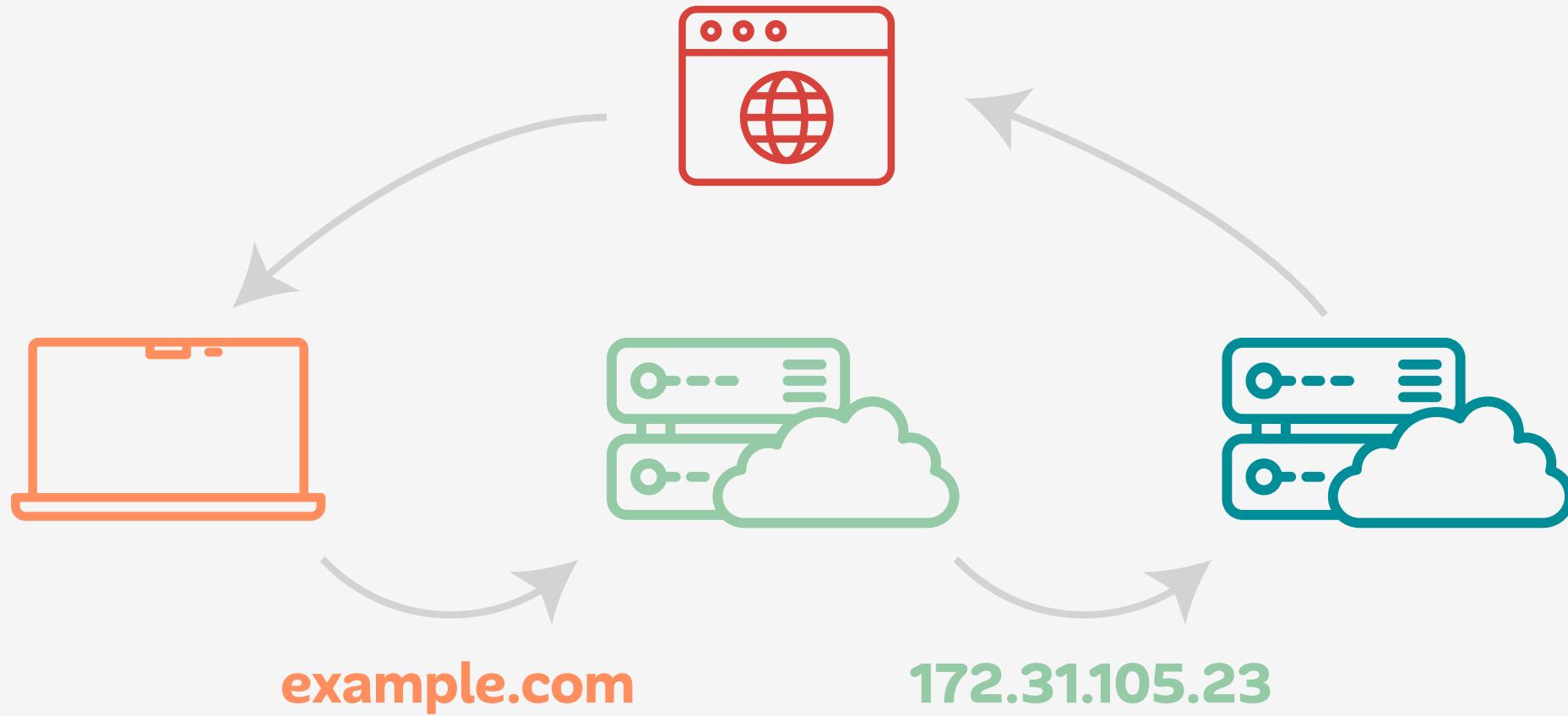
# Domain names

All computers on the internet have IP addresses

`http://8.8.8.8`

Domain names map IP addresses to memorable,  
human-readable names

# Domain names



# Costs

## Server space

Can be free!

Typically \$5+/month

## Domain name

Not free!

\$10-20+/year

iwantmyname		
Domains	Transfer	Plugins
Affiliate	About	Support
andrewheiss.christmas	😊 \$39.00 1st yr	
andrewheiss.compare	😊 \$78.01 1st yr	
andrewheiss.club.tw	😊 \$30.99 1st yr	
andrewheiss.contact	😊 \$26.99 1st yr	
andrewheiss.cooking	😊 \$64.03 1st yr	
andrewheiss.co.at	😊 \$14.99 1st yr	
andrewheiss.courses	😊 \$72.25 1st yr	
andrewheiss.co.gg	😊 \$0.00 1st yr	
andrewheiss.co.je	😊 \$0.00 1st yr	
andrewheiss.co.nl	😊 \$10.99 1st yr	
andrewheiss.co.no	😊 \$29.99 1st yr	
andrewheiss.day	😊 \$29.99 1st yr	

# Quarto hosting options

# Deployment services

Quarto has built-in support for several different hosting services for publishing documents, presentations, dashboards, and websites

The screenshot shows the Quarto documentation interface. At the top, there's a navigation bar with links for Overview, Get Started, Guide, Extensions, Reference, Gallery, Blog, and Help. To the right of the navigation are social sharing icons (Twitter, GitHub, LinkedIn, etc.) and a search bar. A "supported by" logo for Posit is also present.

The main content area is titled "Publishing Basics". It has a sidebar on the left with a navigation tree:

- Guide
  - Authoring
  - Computations
  - Tools
  - Documents
  - Presentations
  - Dashboards
  - Websites
  - Books
  - Manuscripts
  - Interactivity
  - Publishing
    - Publishing Basics**
    - Quarto Pub
    - GitHub Pages
    - Posit Connect
    - Posit Cloud
    - Netlify
    - Confluence
    - Hugging Face Spaces
    - Other Services
    - Publishing with CI
- Projects
- Advanced

The main content starts with an "Overview" section, followed by a "Getting Started" section. The "Getting Started" section contains a table comparing three publishing services:

Destination	Description
<a href="#">Quarto Pub</a>	Publishing service for Quarto documents, websites, and books. Use Quarto Pub when you want a free, easy to use service for publicly available content.
<a href="#">GitHub Pages</a>	Publish content based on source code managed within a GitHub repository. Use GitHub Pages when the source code for your document or site is hosted on GitHub.
<a href="#">Posit Connect</a>	Publishing platform for secure sharing of data products within an organization. Use Posit Connect when you want to publish

On the right side of the content area, there's a "On this page" sidebar with links to "Overview" and "Getting Started". Below that are "Edit this page" and "Report an issue" buttons.

<https://quarto.org/docs/publishing/>

## Easy

- **Quarto Pub**: Free!; everything's public; no custom domains
- **Posit Connect**: \$\$; public and private; custom domains

## Medium

- **GitHub Pages**: Free!; everything's public; custom domains
- **Netlify**: Free and paid; public and private; custom domains

## Advanced

- **DIY**: Set up a server yourself and do whatever you want

# What I use

- **Quarto Pub:** [example](#) + [example](#)
- **GitHub Pages:** [example](#) + [example](#) + this very course website!
- **DIY:** [example](#) + [example](#)

# **What we'll do together**

- Quarto Pub
- GitHub Pages

# Deploying with Quarto Pub

# Quarto Pub

<https://quartopub.com/>

- **What:** Free hosting service for all types of Quarto documents, run by Posit
- **When to use:** When you want to publish something on the internet quickly and you don't really care about (1) privacy or (2) the domain name
- **Difficulty level:** The easiest!

[Complete documentation](#)


 sort by: **Newest** ▾

**pinr resources**

**pinr resources**

AUTHOR Andrew Heiss  
Because this is better than an ephemeral Zoom chat...

**pinr resources**  
Last updated 3 days ago

**ARNOVA 2025 proposal**

AUTHORS Suparna Chaudhry, Andrew Heiss  
AFFILIATIONS Lewis and Clark College, Georgia State University  
PUBLISHED May 29, 2025

**Title**  
(15-word limit)  
Civil Society Crackdowns and the Spatial Redistribution of Foreign Aid

**Track**  
Global Civil Society, Social Movements, and Grassroots Associations

**What is your research question or thesis statement?**  
Existing research explores the effect of civil society restrictions on overall

**ARNOVA 2025 proposal**  
Last updated 2 weeks ago

**Escaping inline R chunks**

**New Quarto-style inline chunks**

Quarto uses this syntax for inline chunks, with the language name wrapped in curly braces

```
The value of n is {{r} round(pi, 2)}
```

t that renders to this ↓

```
The value of n is 3.14
```

To show the inline code verbatim, either wrap the {{r}} in double braces:

```
‘{{{{r}} round(pi, 2)}}’ → {{r} round(pi, 2)}
```

Or wrap the expression in an extra set of backticks:

```
‘`{{{{r}} round(pi, 2)}}`’ → `{{r} round(pi, 2)}`
```

**Escaping inline R chunks**  
Last updated 4 weeks ago

**MMs and AMCs**

AUTHOR Andrew Heiss PUBLISHED August 11, 2025

```
library(tidyverse)
library(logit)
library(marginaleffects)
library(loglogit)

yogurt <- logit::yogurt |>
  mutate(feat = feat == 1) |>
  mutate(price_fct = factor(price))
```

**OLS**

**Model**  
It generates negative predicted probabilities but people still love OLS for this kind of stuff, and it's arguably okay-ish ↴

```
model_ols <- lm(choice ~ price_fct + feat + brand, data = yogurt)
```

**MMs and AMCs**  
Last updated last month

**Fuzzy `rdrobust()` stuff**

AUTHOR Andrew Heiss PUBLISHED August 28, 2025

So I think the reviewer is misunderstanding what rdrobust does. It's a nonparametric approach, so it doesn't follow a standard 2SLS pattern. Your equations 1 and 2 are parametric, though, which maybe is throwing the reviewer off.

I didn't go into the details for nonparametric RDD in my class because they're complicated and, as a result, most people don't put the equations in their papers (from what I've seen).

Here's a grossly oversimplified version of the official original version (see p. 2306 at [https://rdpkgs.github.io/references/CaliendoCattaneoTitunik\\_2014\\_ECMA.pdf](https://rdpkgs.github.io/references/CaliendoCattaneoTitunik_2014_ECMA.pdf)), the original paper for nonparametric RDD and the paper everyone cites).

Basically, the fuzzy RDD LATE is the Wald ratio of two discontinuities:

**Fuzzy `rdrobust()` stuff**  
Last updated last month

**Bayesian {marginaleffects}**

AUTHOR Andrew Heiss PUBLISHED August 7, 2025

```
library(tidyverse)
library(marginaleffects)
library(brms)
library(parameters)
library(bayesrules)
library(scales)
library(tidybayes)
```

# Nicer colors and basic theme  
clsrs <- mmaColors::mma.colors("ustwo")  
theme\_set(theme\_bw())

# Get data from `bayesrules` and clean it up a bit  
data(climbers\_sub, package = "bayesrules")  
  
climbers <- climbers\_sub |>  
select(-

**Bayesian {marginaleffects}**  
Last updated 2 months ago

**Session 13 questions**

AUTHOR Andrew Heiss PUBLISHED July 29, 2025

**Sidenotes**  
You don't need to include {magrittr} and {viridis} in your `p_load()` function. {magrittr} gets invisibly loaded by {dplyr}, and {dplyr} gets loaded when you use {tidyverse}, so it's already there. {viridis} gets invisibly loaded by {ggplot2}, and {ggplot2} gets loaded when you use {tidyverse}, so it's also already there.

**Removing legends**  
I notice that to remove guides we have used 3 types of codes e.g.,

A dashboard that is a dashboard



**A dashboard that's not a dashboard**

**Row**

```
library(tidyverse)
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point()
```



# How to publish

1. Go to [quartopub.com](https://quartopub.com) and create a free account.  
Make sure you're logged in.
2. In the terminal, run this:

```
Terminal
```

```
quarto publish
```

3. Select Quarto Pub and press enter. Answer all the other questions.
4. Wait for the site to render and upload.
5. Done!

# Other details

- To update the site later, run `quarto publish` in the terminal again.
- Manage other site settings at your [quartopub.com](https://quartopub.com) dashboard.

# Your turn

Publish your playground website to Quarto Pub.

1. Go to [quartopub.com](https://quartopub.com) and create a free account.  
Make sure you're logged in.
2. In the terminal, run this:

Terminal

```
quarto publish
```

3. Select Quarto Pub and press enter. Answer all the other questions.
4. Wait for the site to render and upload.

07 : 00

# Deploying with GitHub Pages

# GitHub Pages

- **What:** Free hosting service for repositories hosted on GitHub
- **When to use:** When you want a (free) longer term hosting solution (with a custom domain if you want) and you're already using GitHub for version control
- **Difficulty level:** Medium

[Complete documentation](#)

# Three ways; three difficulties

1. **Easy:** Render your site to `docs/` and include that folder in your git repository.
2. **Medium:** Use the `quarto publish` terminal command to render the site and push it to a `gh-pages` branch.
3. **Hard:** Use GitHub Actions to build an on-the-fly virtual computer that remotely rebuilds and publishes your site every time you commit to your repository.

We're going to do Option 1!

# Some git knowledge required



General version control



Remote version control  
hosting + collaboration

## Helpful resources:

- [GitHub's Hello World tutorial](#)
- [Happy Git with R](#)

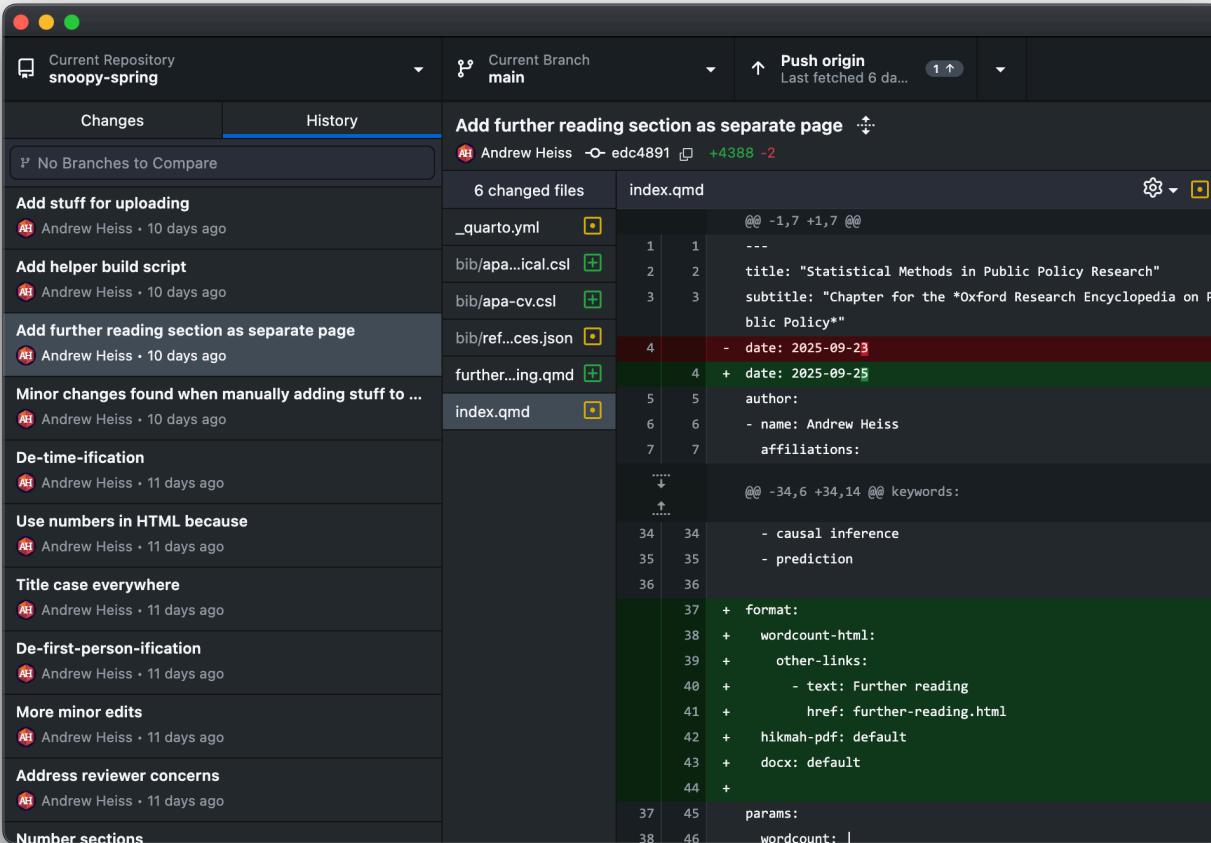
# Interacting with git

GitHub Desktop

RStudio

Positron / VS Code

Terminal



The screenshot shows the GitHub Desktop application interface. At the top, there are dropdown menus for 'Current Repository' (set to 'snoopy-spring'), 'Current Branch' (set to 'main'), and 'Push origin' (status: 'Last fetched 6 da...'). Below the header, there are tabs for 'Changes' and 'History', with 'Changes' being active. The main pane displays a list of commits from 'Andrew Heiss' over the last 11 days. One commit, titled 'Add further reading section as separate page', is expanded to show a diff view. The diff view compares two versions of the file 'index.qmd'. The changes include a modification to the date field from '2025-09-23' to '2025-09-25', and additions for 'keywords', 'format', 'wordcount-html', 'other-links', and 'params'. The commit message for this specific change is 'Add further reading section as separate page'.

```
diff --git a/index.qmd b/index.qmd
--- a/index.qmd
+++ b/index.qmd
@@ -1,7 +1,7 @@
 1 1 ---
 2 2 title: "Statistical Methods in Public Policy Research"
 3 3 subtitle: "Chapter for the *Oxford Research Encyclopedia on Pu
 4 4 blic Policy*"
- date: 2025-09-23
+ date: 2025-09-25
 5 5 author:
 6 6 - name: Andrew Heiss
 7 7 affiliations:
@@ -34,6 +34,14 @@
 34 34 - causal inference
 35 35 - prediction
 36 36
 37 + format:
 38 + wordcount-html:
 39 + other-links:
 40 + - text: Further reading
 41 + href: further-reading.html
 42 + hikmah-pdf: default
 43 + docx: default
 44 +
 37 45 params:
 38 46 wordcount: |
```

# Basic workflow

1. Make changes
2. Commit changes to lock them in
3. Push commits to remote server

*Advanced stuff you don't have to worry about:* branches, collaboration, pull requests, etc.

# .gitignore

Use `.gitignore` to hide things from git.



## General rules

**Commit things that make outputs; don't commit outputs themselves.**

**Don't commit user-specific things (or passwords!)**

### .gitignore

```
1 .Rproj.user
2 .Rhistory
3 .RData
4 .Ruserdata
5
6 data/private-data.csv
7 output/*.png
8 all_my_passwords.txt
```

# How to publish

## Part 1: Get your project code on GitHub

1. Go to [github.com](https://github.com) and create a free account.
2. Add a `.gitignore` file to your website project with this:

```
.gitignore
1 .Rproj.user
2 .Rhistory
3 .RData
4 .Ruserdata
5
6 /_site/
```

3. Initialize a git repository for your website project.
4. Commit the files and push to GitHub.

# How to publish

## Part 2: Tell Quarto to use GitHub Pages

5. Change the `output-dir` option in `_quarto.yml` to `docs`.

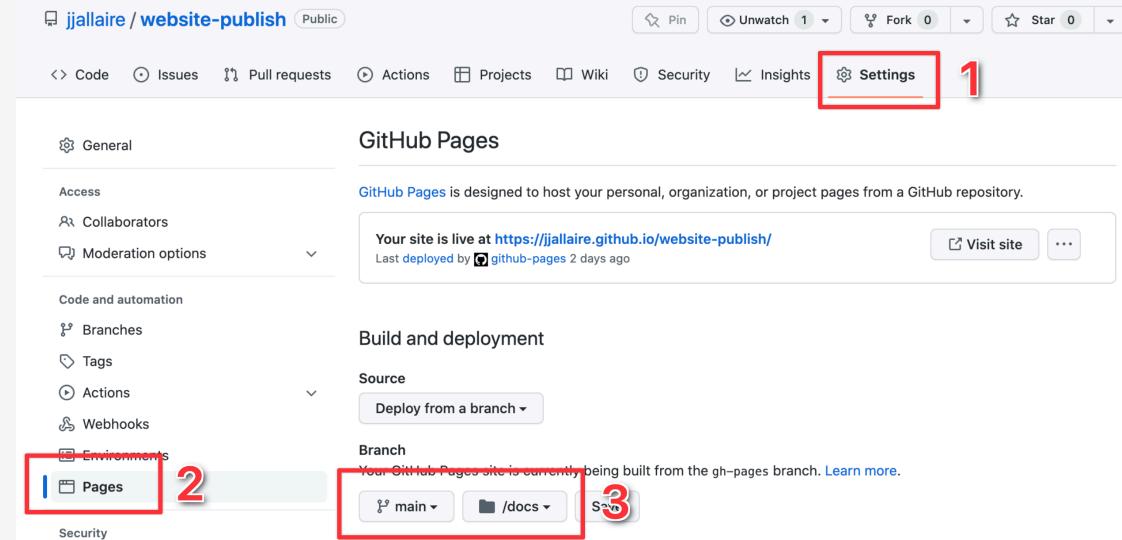
```
_quarto.yml
project:
  type: website
  output-dir: docs
```

6. Add an empty file named `.nojekyll` to the root of your website project (this tells GitHub to not run its own static site generator).
7. Render your site.
8. Commit the newly created `docs/` folder and `.nojekyll` file and push to GitHub.

# How to publish

## Part 3: Tell GitHub Pages to serve your site

9. From your GitHub repository, go to Settings > Pages and configure the repository to publish from the `docs` directory of your `main` branch.



10. Wait for GitHub to build and deploy the site.

# Other details

- To update the site later, render the site, commit to GitHub, and wait for the site to rebuild.  
Track the progress under the “Actions” section of your GitHub repository.
- Add a custom domain at Settings > Pages

# Your turn, part 1

## Get your project code on GitHub

1. Go to [github.com](https://github.com) and create a free account.
2. Add a `.gitignore` file to your website project with this:

```
.gitignore
1 .Rproj.user
2 .Rhistory
3 .RData
4 .Ruserdata
5
6 /_site/
```

3. Initialize a git repository for your website project.
4. Commit the files and push to GitHub.

10:00

# Your turn, part 2

## Tell Quarto to use GitHub Pages

5. Change the `output-dir` option in `_quarto.yml` to `docs`.

```
_quarto.yml
project:
  type: website
  output-dir: docs
```

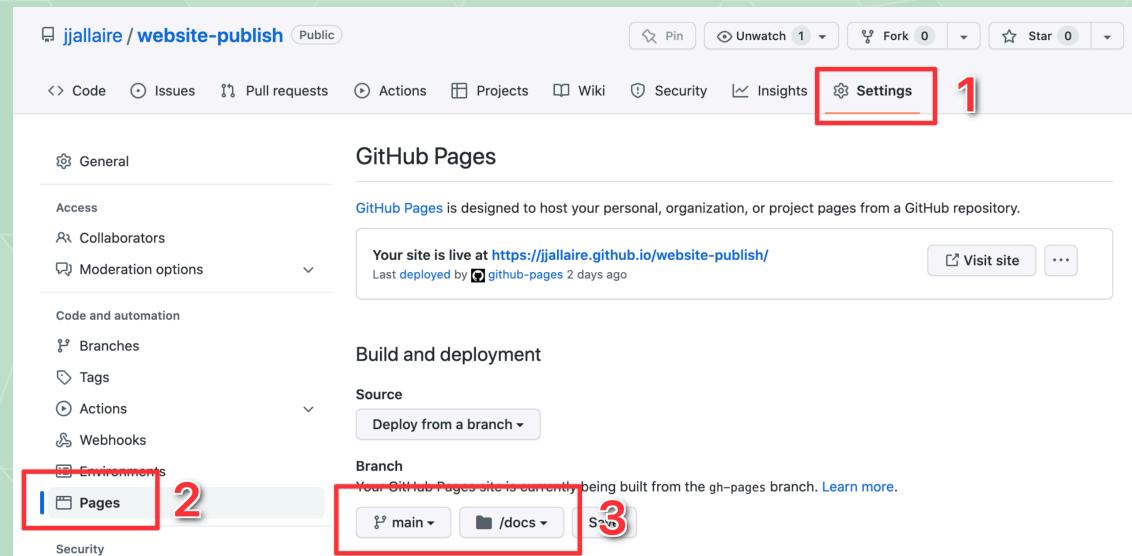
6. Add an empty file named `.nojekyll` to the root of your website project (this tells GitHub to not run its own static site generator).
7. Render your site.
8. Commit the newly created `docs/` folder and `.nojekyll` file and push to GitHub.

10:00

# Your turn, part 3

## Tell GitHub Pages to serve your site

- From your GitHub repository, go to Settings > Pages and configure the repository to publish from the `docs` directory of your `main` branch.



- Wait for GitHub to build and deploy the site.

10:00



# Course outline

-  ~~Intro to Quarto~~
-  ~~Creating basic websites~~
-  ~~Advanced website features~~
-  ~~Publishing~~
- Customization and branding
- Interactivity