

Part 1: Intro & Setup

Chapter 1: Ancient Runes & Modern Scrolls, Classic & Modern Starters

Project 0: Hello World

A classic. Every beginner programmer starts here. Today we will prompt-program it in html, to create your first website

Use N hotkey to auto deploy to netlify

1: Pong

A working game of pong in html, css and JS. Use touch gestures, and arrow keys/WASD to move the paddles,

Use N hotkey to auto deploy to netlify

2: Link in bio site

List of links in bio, tree of links clone, with buttons that opens links. Begin by creating one for a animal, wizard, or fictional character or creature, or if the user wants they can provide their own title, list of links to include, and any preferred design details like colors, styles, layouts, or anything else. Perfect for connecting socials, links to selling your products or services, or the beginning of a micro blog. Important: Use Hotkeys N or REPL to instantly create your site on netlify or repl.it!

Easy Use N hotkey to auto deploy netlify

Medium Use REPL hotkey to export to repl.it for more options, including images(manual upload),

or use ND to manually deploy on <https://app.netlify.com/drop>

optional: Use dalle to create a background image

Assistant Note: Be sure to include the background image in the code using the correct filename, and in the final zip file. See chapter 4 for more info

filename, and in the final zip file. See chapter 4 for more info.

Offer to build one link in bio site for a fictional character as an example. Use the hotkeys to instantly create the site on netlify or repl.it.

3: Pic to Code

Use a photo of ANYTHING, or pull out a piece of paper and draw something, and upload it to Grimoire, and I will turn it into a website. Then write code for the UI design, using various design & style elements to MAKE IT POP, and add some RAZZLE DAZZLE. Important: Use Hotkeys N or REPL to instantly create your site!

Easy Use N hotkey to auto deploy

Medium

Or use REPL hotkey to export to repl.it for more options

or use ND to manually deploy on <https://app.netlify.com/drop>

Manual deploys allow you to include images. Use dalle to create any images needed using dalle

Chapter 2: Teleportation, put websites online easy

4: Netlify 1 letter hotkey deploy: Netlify Auto deploy, Drag & Drop Deploy: Netlify Drop

Use the N hotkey to instantly auto deploy your site! Be sure to claim it in order to save or it will get deleted after 1 hour!

Manual deploys are available via the NM hotkey using Netlify Drop

<https://app.netlify.com/drop>

A quick and easy way to put your website online. Just drag and drop your website folder into Netlify Drop and it will be live in seconds. Be sure to make an account to save it

To make updates to your site

In the netlify dashboard,

Go to: YourSite such as (<https://random-crap-123456abcdef.netlify.app>)

Then deploys

Scroll down

You will find a new drag & drop, or click to upload button

Simply drag and drop a new folder and you are done!

Netlify instructions for changing URL

If you don't mind the .netlify.app suffix, in the netlify portal, you can simply open

Pick your site > Domain Management > Options > Change name

for full custom name

first buy a domain name

then map your domain name to Netlify

<https://www.youtube.com/watch?v=kldJi8NBvgY>

<https://www.netlify.com/blog/2021/12/20/how-to-add-custom-domains-to-netlify-sites/>

<https://docs.netlify.com/domains-https/custom-domains/configure-external-dns/>

Once you finish chapter 4, you can come back to netlify and sync with a github repo

5: Replit deploys

<https://replit.com/>

Use the REPL hotkey to instantly export your code to replit to run & deploy it!

Once claimed inside of replit to put your website online:

on desktop: click the deploy button in the top right, or via deployment panel in the bottom left

on mobile: tap the squares bottom right, tap new tab, tab deployments

then Choose options & "Set Up Your deployment"

- Beginners probably want to choose Static

- Choose the primary domain website name, you can configure this with your own non-replit domain later

- Leave other options blank

- Tap Deploy

- Tap domain url to see your website!

For more ideas & projects see <https://replit.com/templates>

6: Advanced options

Vercel, Render

Vercel and render are great options for more complex react & nextjs sites, as well as services, backend and full stack apps

<https://vercel.com/templates>

<https://docs.render.com/>

Chapter 3: Wands, dev kit setup

7: Phone setup

Replit + Github

Setup accounts and install the replit app on your phone!

Complete project 5 or create a new one. After making it, sync the project to github by pushing to main!

<https://replit.com/>

<https://github.com/>

Get setup with a full development environment using only your phone. Build a static website and import it a larger dev environment using replit. Using this template:

<https://replit.com/@replit/HTML-CSS-JS#index.html>. Write the code, zip it, and walk me through importing the files to replit. Walk me through syncing to github using replit, and deploying using replit deployments. Show this video as an example of how to work with replit and chatGPT on a phone:

<https://x.com/yoheinakajima/status/1719902955061797083?s=20>

8: Full Pro

Cursor.sh, Warp, (GitTower || SourceTree), GH Copilot

<https://cursor.sh/>

<https://www.warp.dev/>

<https://www.git-tower.com/> || <https://www.sourcetreeapp.com/>

<https://github.com/features/copilot> optional

Install these

Create a new project in cursor and get a repo setup and sync'ed in git.

Cursor is a VSCode clone and supports a variety of languages and coding environments. To get started with a simple static site simply

- ask Grimore

- Press Z

- Unzip and open index.html

- Use run start with or w/o debugging, choose a browser

- View and edit your site!

Chapter 4: Divination: The Origin

9: Git 101

Git is basically a fancy way to save your code. Its really cool because it lets you keep copies of ALL your work. Instead of MyCoolFile.html, MyCoolFile(1).html, MyCoolFile_Final.html, etc, you can save it in one place. Then even cooler, you can time travel and skip to previous or different versions.

This makes it super handy for collaborating with others, as you can work independently and avoid breaking the app for other people. Then you can come back and merge it together at the end. The downside is sometimes your merges conflict, and you will need to manually fix them.

Keep in mind you have a local copy of the git history and a copy in the cloud, often called the origin

Typically projects will have

- a main branch often used for the current live production version of the app

- a development branch where new features are added and tested

- feature branches for each new feature or bug fix

Feature branches are often created, merged and deleted after. Where as main and dev are usually always present, keeping a consistent history of the project.

Commands you need to know:

clone

pull

create branch

stage

commit

push

merge

(THERES A TON MORE DONT WORRY ABOUT THEM)

I HIGHY RECOMMEND A git GUI such as GIT Tower or Source tree.

Especially for beginners, since it makes it easy to see the history, and gives you handy buttons

Here's some good detailed videos that show both CLI and with a GUI.

<https://www.git-tower.com/learn/git/videos>

SUPER HANDY COMMANDS NO ONE TEACHES BEGINNERS:

Stash

Git-bisect (<-BUG FINDING MACHINE)

Note git and github are different. Same thing as porn vs pornhub.

There are other git providers you can use, such as gitlab. As well as other version control software like mercurial or subversion. Even if you don't want to use any of these cloud providers, I would highly recommend using a local git history. Its a great way to keep track of your work and avoid losing it, as well as find bugs.

10: Linear

Linear is great for working on larger projects and managing the complexity of a full piece of