

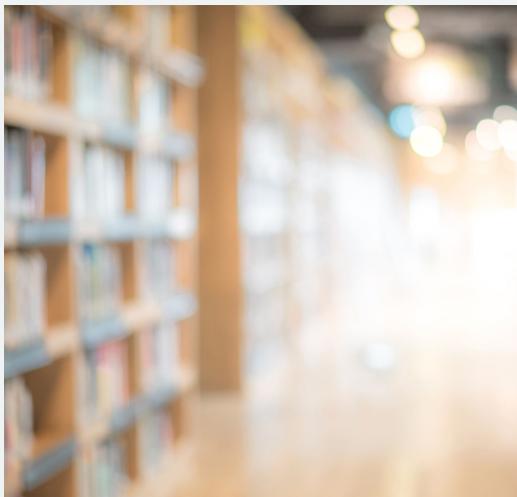


dev.local

local development
and full site editing

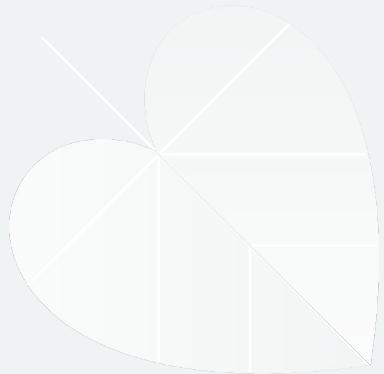
H. Adam Lenz

h.adam.lenz



- Lead Web Developer for IT Digital services at University of North Carolina at Chapel Hill
- Film and Digital media from University of Central Florida
- Wp 1.5
- Das code -> git clone
<https://github.com/hadamlenz/dev.local>

Goals for the workshop



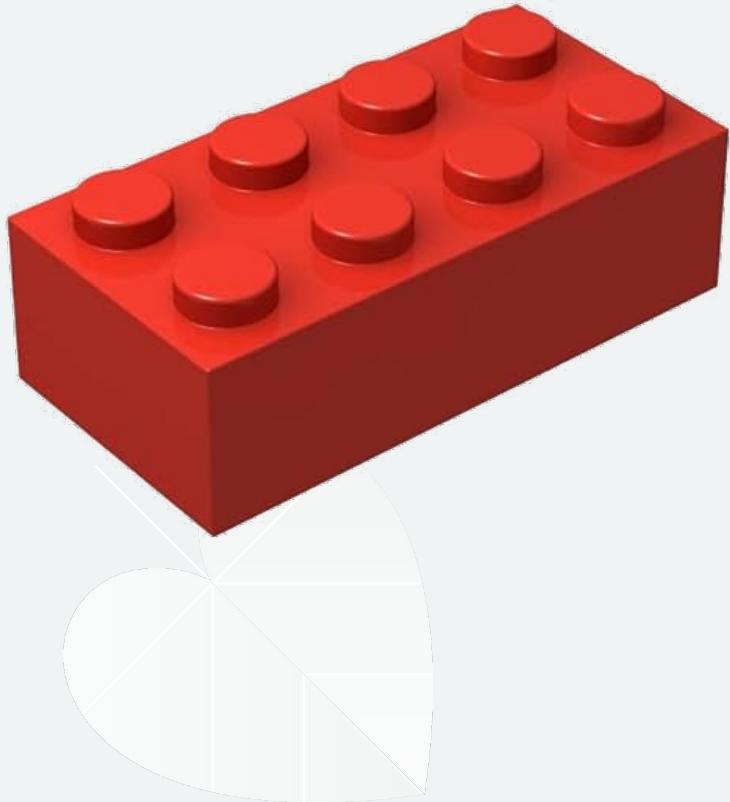
- Separation of concerns
- Why dev local?
- Dev local with what?
- Snacks!
- Let's dev local?
- All while working with full site editing



Thoughts on
full site editing
so far?



The block editor



- Block Editor

Bebo 5.0

Dec 6th, 2018 = 5 ½ years

- Site Editor

Joséphine 5.9

January 25th, 2022 = 2 ½ years

- see issues

My love for WordPress Development

- Presentation lives in the theme
- Functionality live in plugins and core
- Content lives in the database



Separation of concern

a design principle for separating a computer program into distinct sections. Each section addresses a separate concern, a set of information that affects the code of a computer program.

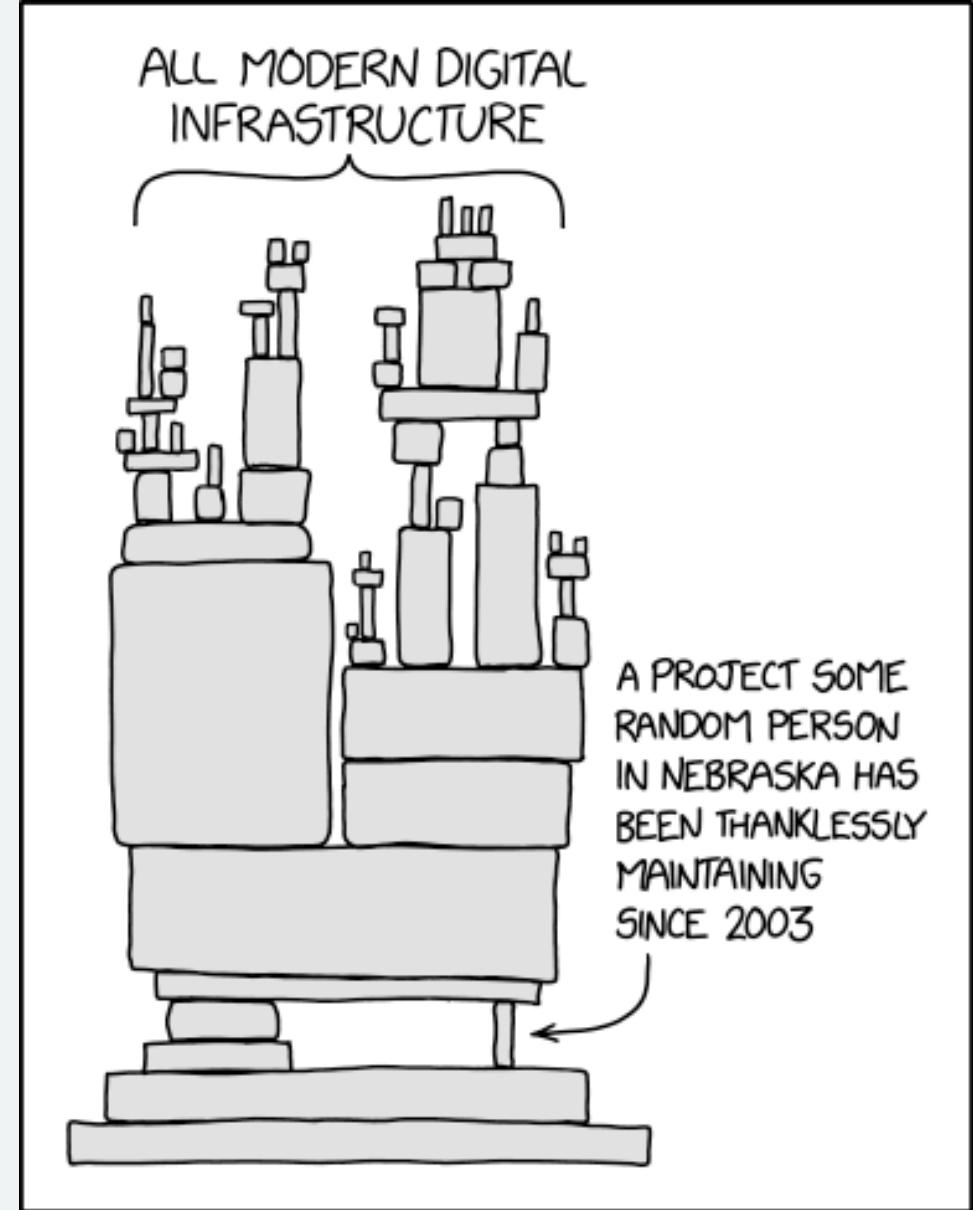
the classic editor

- Styles -> enqueueued styles in the theme
- Structure -> defined in the theme template files and rendered using template hierarchy
- Functionality -> could be in the theme, should be in the plugins or core



The Pitfalls of not separating concerns

- When all the components are mashed together it is much easier to break something on accident
- Creates institutional knowledge in the code base due to dependency



Styles with the block editor

01

Styles start at
theme.json - a file!

02

Global styles - in
db

03

Templates and
part styles - in files
but edits are in
the db

04

Local styles
applied to
individual parts -
can be stored in
the db

05

Enqueued styles -
these are files

An aerial photograph of a dense forest of coniferous trees, showing a repeating pattern of green foliage.

10000 foot view



From high up it looks like the
block editor doesn't
separate the concern very
well



Blocks are very
portable

Structure of a block



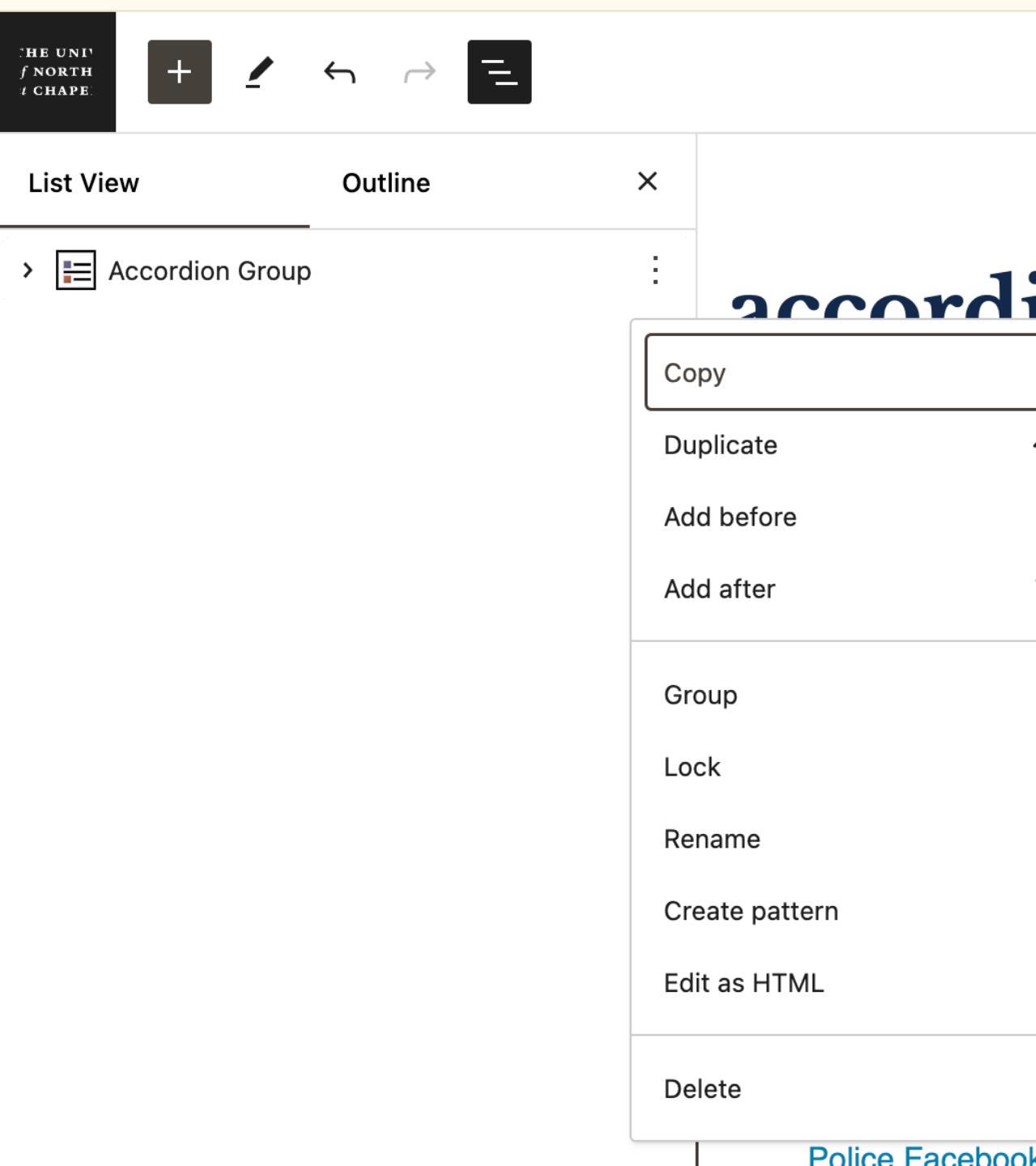
```
<!-- namespace:block name {json} /-->
```

or

```
<!-- namespace:block name {json} -->
```

HTML Content including nested blocks

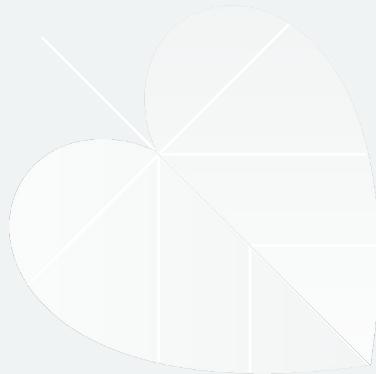
```
<!-- / namespace:block name -->
```



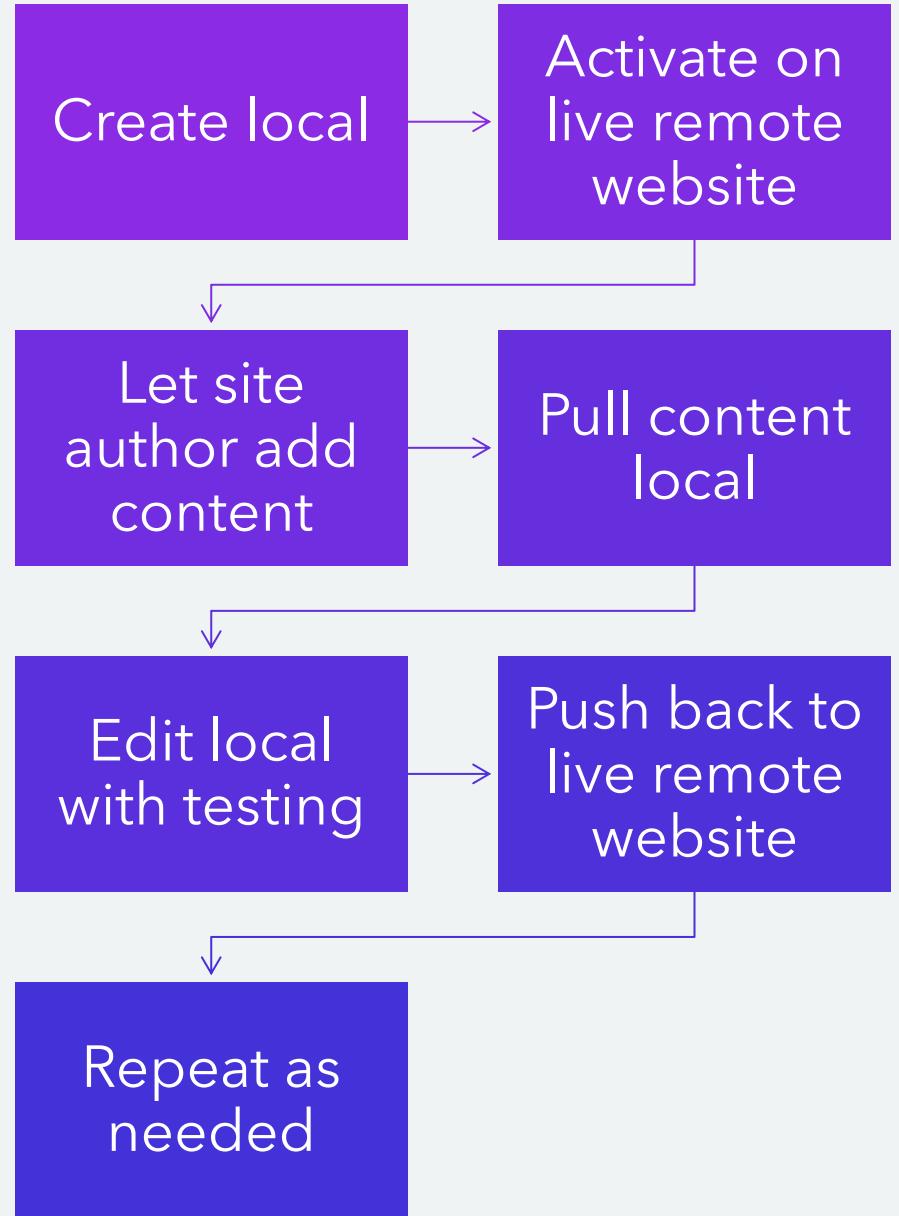
You can copy a
block and paste it
into a file

Look in the
wp_posts table

- wp_block
- wp_global_styles
- wp_navigation
- wp_template



My classic editor workflow for developing a theme or plugin



Why though?



don't break the live site!



Why local dev?

- Reduced development time
- No internet connection needed



Testing via IEEE

Software testing is a crucial activity in the software development life cycle that aims to evaluate and improve the quality of software products. Thorough testing is essential to ensure software systems function correctly, are secure, meet stakeholders' needs, and ultimately provide value to end users.

A close-up photograph of a person's hand pointing their index finger towards a computer monitor. The monitor displays a block of Python code. The code appears to be a script for a 3D modeling application, specifically Blender, based on the use of bpy and context variables. The script handles various operations like mirroring objects and selecting them. The background is dark, making the bright screen stand out.

```
    mirror_mod = modifier_obj
    # Set mirror object to mirror
    mirror_mod.mirror_object = mirror_object
    if operation == "MIRROR_X":
        mirror_mod.use_x = True
        mirror_mod.use_y = False
        mirror_mod.use_z = False
    elif operation == "MIRROR_Y":
        mirror_mod.use_x = False
        mirror_mod.use_y = True
        mirror_mod.use_z = False
    elif operation == "MIRROR_Z":
        mirror_mod.use_x = False
        mirror_mod.use_y = False
        mirror_mod.use_z = True

    # Selection at the end - add
    one.select = 1
    other.select = 1
    context.scene.objects.active = eval("Selected" + str(modifier))
    mirror.select = 0
    bpy.context.selected_objects.append(mirror)
    data.objects[one.name].select = 1
    print("please select exactly one object")

- OPERATOR CLASSES ----

@types.Operator:
def execute(self, context):
    # X mirror to the selected
    context.object.mirror_mirror_x = self.mirror_X
    context.object.mirror_mirror_y = self.mirror_Y
    context.object.mirror_mirror_z = self.mirror_Z

    # context:
    # context.active_object is not set
```

Testing via IEEE

- Risk Migration
- Confidence in the software
- Adherence to standards
- Optimization
- Cost/Time savings



```
mirror_mod = modifier_obj
# Set mirror object to mirror
mirror_mod.mirror_object = ob
if operation == "MIRROR_X":
    mirror_mod.use_x = True
    mirror_mod.use_y = False
    mirror_mod.use_z = False
elif operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
elif operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

# Selection at the end - add
if ob.select == 1:
    mirror_ob.select = 1
else:
    context.scene.objects.active = eval("ob")
    ("Selected" + str(modifier))
    mirror_ob.select = 0
    bpy.context.selected_objects.append(mirror)
    data.objects[one.name].select = 1
    print("please select exactly one object")

- OPERATOR CLASSES ----

@types.Operator:
def execute(self, context):
    # X mirror to the selected
    # object.mirror_mirror_x"
    mirror X"
    if context:
        if context.active_object is not None:
```



Why do I tho?

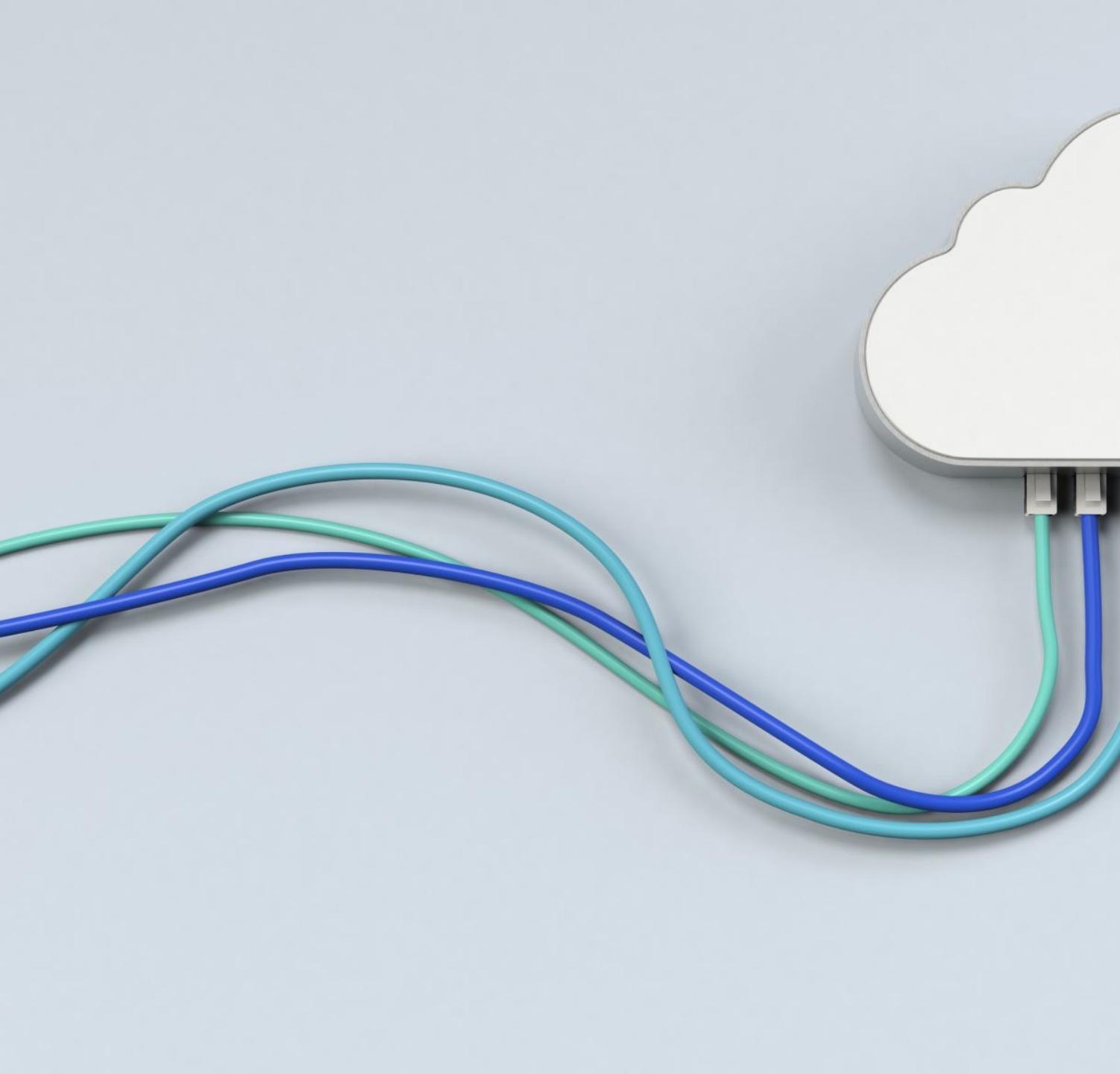
- I can make terrible mistakes
- I can try unsafe things
- I can break the site and put it back together
- I can click buttons when I don't know what they do
- I can ensure that the code and db are going to work, and that other issue is server related
- Nobody sees any of this, users are not met with...

WordPress White screen of Death (WWSoD)



Why not tho?

- You are creating a server, like super comp sci
hackerman
- Git merge conflicts
- Testing at scale



The parts needed to make a good local

- The server which creates the local host
- A way to route traffic to the local host
- The code -> WordPress core, the themes and the plugins
- The database
- The media files used *
- A way to add SSL
- Error logs

```
error_log( var_export( $stuff, true ) )
```

A php function
to send this
variable to the
php error logs

Doesn't mess
up the view like
var_dump

```
[29-Jul-2024 03:09:13 UTC] hello world  
[29-Jul-2024 03:22:46 UTC] hello world  
[29-Jul-2024 03:22:49 UTC] hello world  
[29-Jul-2024 03:23:47 UTC] hello world  
[29-Jul-2024 03:25:48 UTC] hello world  
[29-Jul-2024 03:27:49 UTC] hello world  
[29-Jul-2024 03:29:50 UTC] hello world  
[29-Jul-2024 03:31:51 UTC] hello world  
[29-Jul-2024 12:01:01 UTC] hello world  
[29-Jul-2024 12:01:02 UTC] hello world  
[29-Jul-2024 12:02:01 UTC] hello world  
[29-Jul-2024 12:03:01 UTC] hello world  
[29-Jul-2024 12:05:02 UTC] hello world
```

VS Code Log Viewer Extension and Local by Flywheel

You can see the logs right in vs code

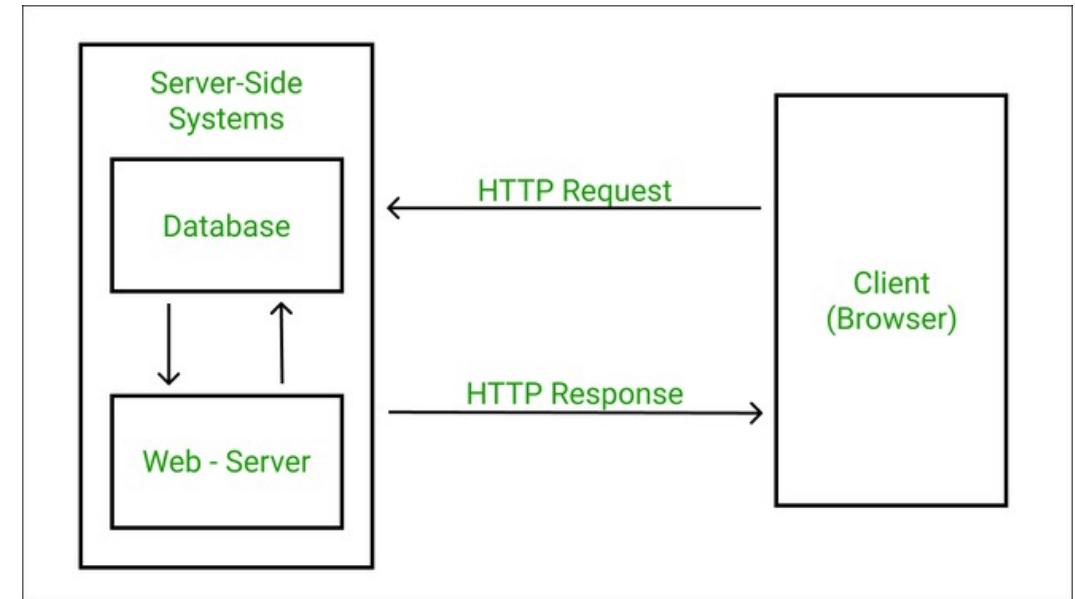
You can see what time it is in Greenwich, London

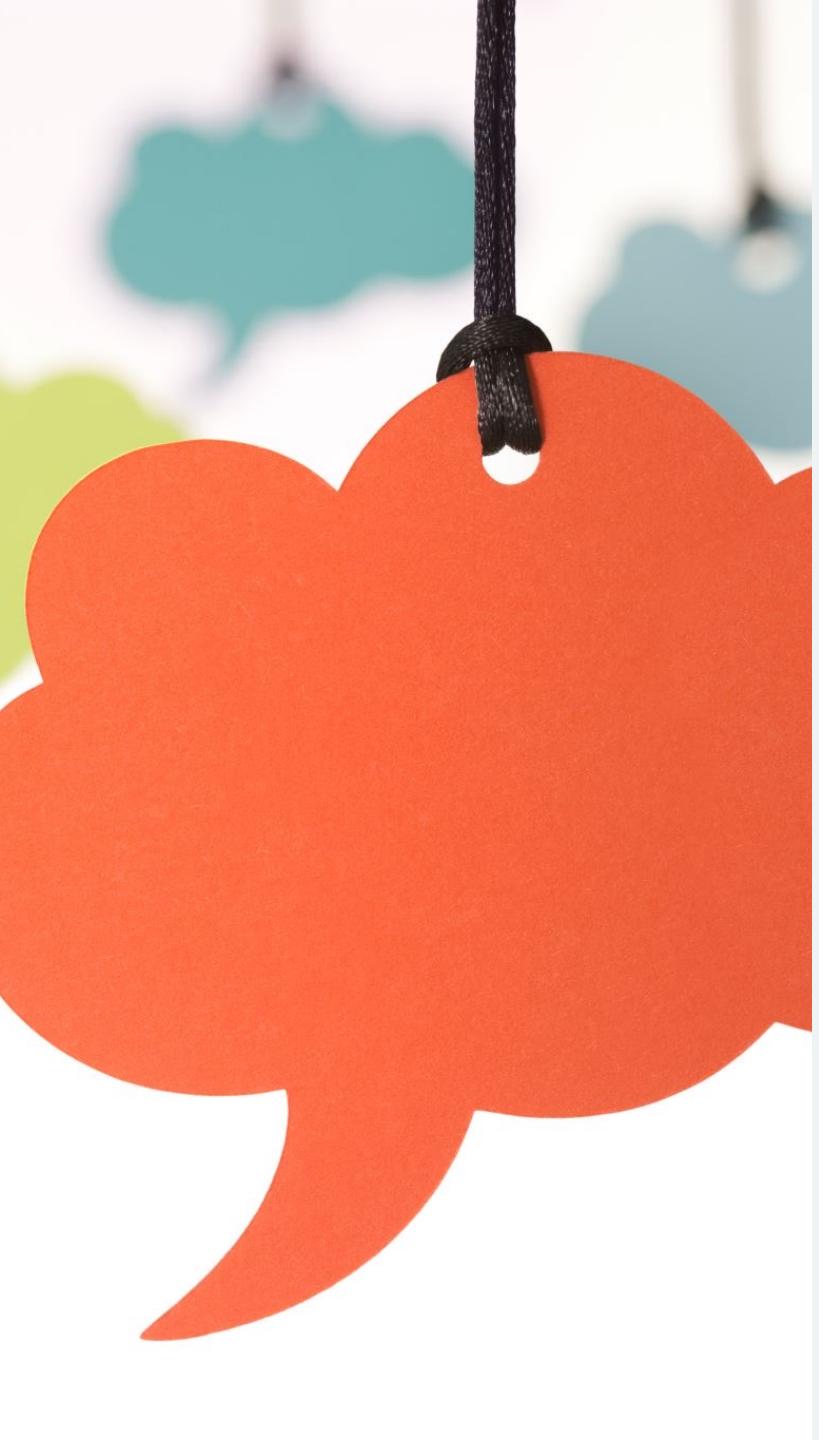
The server



- web server (apache, nginx) a user requests a page, this sends back the markup
- Server-side code processing (php, php-fpm) when a page is requested, does processing to inform the creation of markup including getting content from the database
- A database (mysql, mariadb) - holds content

An http request



A large, solid orange speech bubble hangs from a black string tied in a knot at the top. The background behind the bubble is white, and there are blurred green and blue shapes visible in the distance.

We are going to argue
about which one is better



Local Dev Applications

- WAMP,
- MAMP,
- XXAMP,
- Local by Flywheel or Pantheon
- What did I forget?

Local by Flywheel

- Works really well with migrate db and WP Engine
- <https://localwp.com/help-docs/getting-started/how-to-import-a-wordpress-site-into-local/>
- <https://deliciousbrains.com/wp-migrate-db-pro/doc/importing-wordpress-local-development-environment.>

WP Migrate

- I used it a lot when I worked on smaller sites
- Export a site with migrate db import with local by flywheel
- We cannot use it in our multi-sites because they are too big
- These types of limitations pushed me to explore more

LocalDev by Pantheon

- At the time of writing this seems dead
- <https://pantheon.io/localdev>
- rip



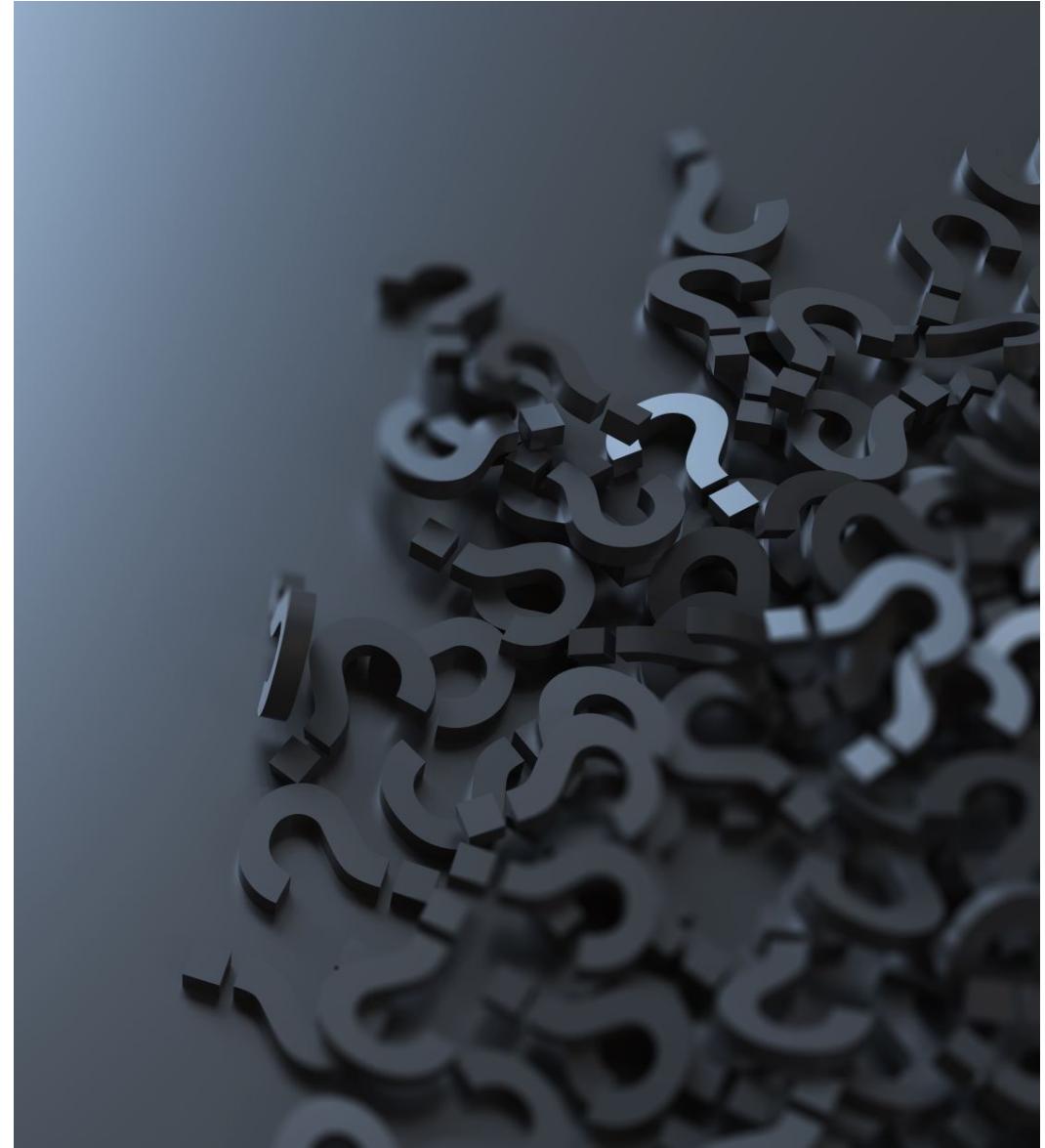
These are always
going to be the
easiest to work with

A bit more in the weeds

- Docker (compose)
- Vagrant
- Container applications
- More?



These are more
configurable to be
exactly what you need
but take more work to
get running



Some other useful tools

- WSL on windows to get a nix system
- Nodejs/npm
- Git
- Composer
- Migrate DB

WordPress playground tools

- @wordpress/env - the docker one
- @wp-now/wp-now - wp as a binary
- VS Code extension - in vs code, uses the binary





WP-Now

- A NodeJS app with an AMP stack in it
- Temporary by design
- Super fast
- Works on wsl and mac
- No SSL
- Blueprints allow a lot



@wordpress/env

- Its basically just a docker-compose built into WordPress
- Requires docker, git
- No SSL support yet
- Programable with node
- Has logs *



My Docker Setup is a lot like a local host app

- Docker-compose
- Traefik to Handle routing to all the containers and SSL
- Centralized database, mySQL backups, phpmyadmin
- whoami.test
- Separate compose.json for each WordPress, includes PHP and nginx

Local Certificate CA

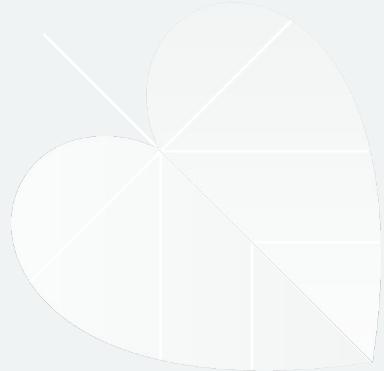
- You make one certificate authority and trust it
- All the certs you make from now on with that CA are trusted
- <https://deliciousbrains.com/ssl-certificate-authority-for-local-https-development/>



TLDs

- .test - intended for use in the testing of software
- .local and localhost - isn't going to be ever be a tld on the internet.
- localhost is automatically found by chrome, without a DNS or /etc/hosts
- Handled pretty well with dnsmasq

Docker



- A program that allows for the deployment of containers as applications that work in different environments in isolation
- It works on a lot of systems
- You build each functional unit in a container and them connect them all - Separate concerns

A close-up photograph of a screen or display panel showing a dense grid of small, glowing circular pixels. The colors transition from dark blue at the bottom to bright white and yellow-green at the top, creating a gradient effect. The dots are arranged in a regular, staggered pattern.

Windows Subsystem for linux

ick

If you wish to make a local copy of your remote site

- Invent the universe... get a local dev running
- Get the files need, WordPress themes plugins, maybe media
- Get the database
- Import all that into your local dev

Built-in tools for exporting

- Tools > Export
- Site editor > Styles > Menu > Export
- Wp-cli



3rd party tools for exporting and importing

- Migrate-db
- Wp import/export all





Get the remote database (and more)

- Wp-migrate-db export
- Wp cli

```
wp db export - > database.sql
```
- Phpmyadmin -> export database
- SQL Client -> export
- Pantheon export

Import the remote database

- Wp-migrate-db import
- Drag and drop on local by flywheel
- Wp cli

```
wp db import database.sql
```
- Phpmyadmin -> import database
- SQL Client -> export





Common pitfalls with importing the db

- Prefixes can be different, especially if you are pulling from a multisite
- Multisite settings that are only set on the whole network are not available when making a single site, like users
- Importing a table like wp_options when there is already a table called wp_options makes an error. Use `DROP TABLE IF EXISTS` table,table...

Setup your local

You should aim to put all the assets in the same place locally as the remote

- WordPress Are you using web_docroot? Set it up like that
- Themes, plugins, uploads all go in wp-content
- Local wp-config.php pointing to local db

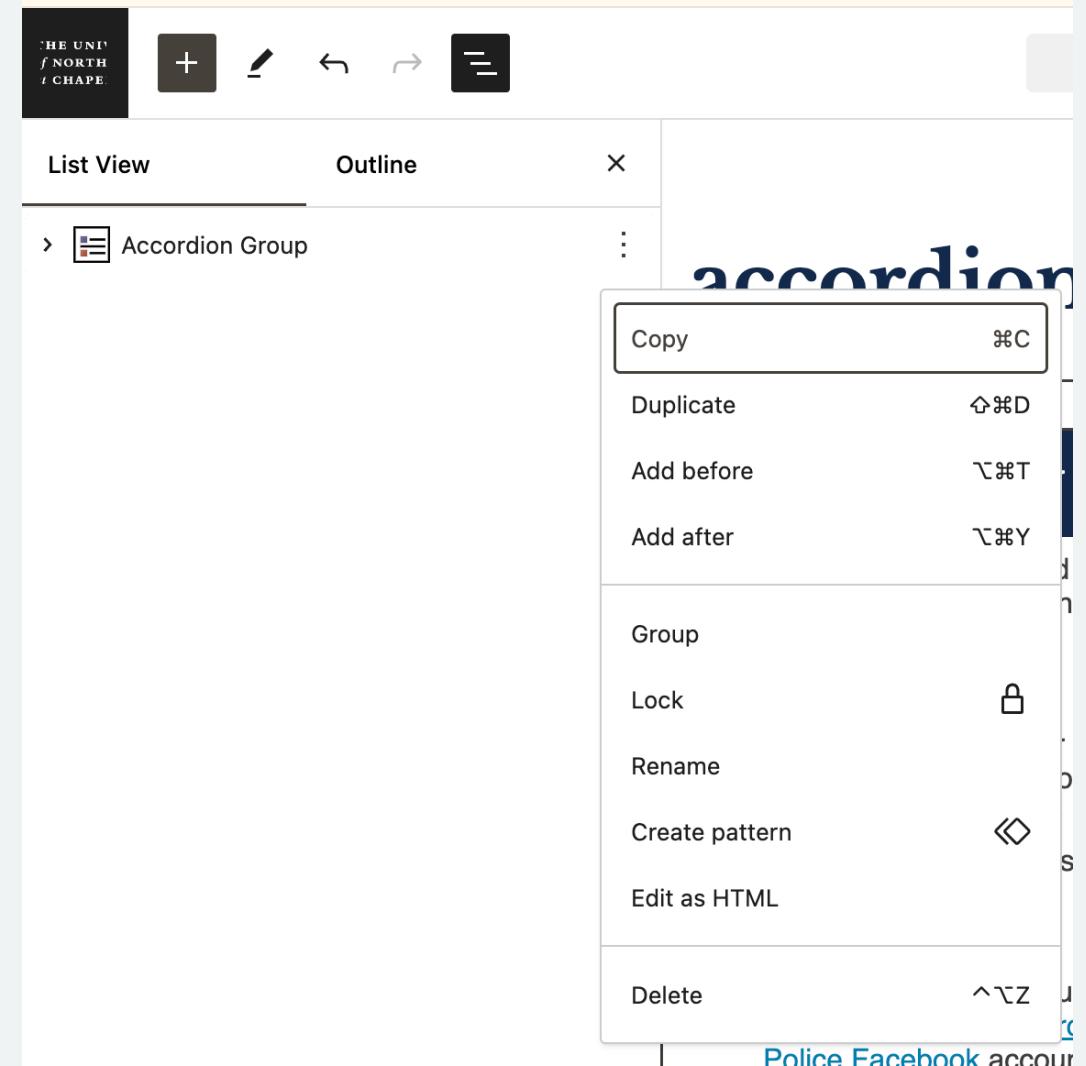
Deployment back to the live



- I would highly recommend against pushing the database from your local back up to your live site
- Is the change something that you want all your theme users to see?
 - yes - make it part of the theme, push a theme update*
 - no - deploy just the change you made*
- Grab the parts

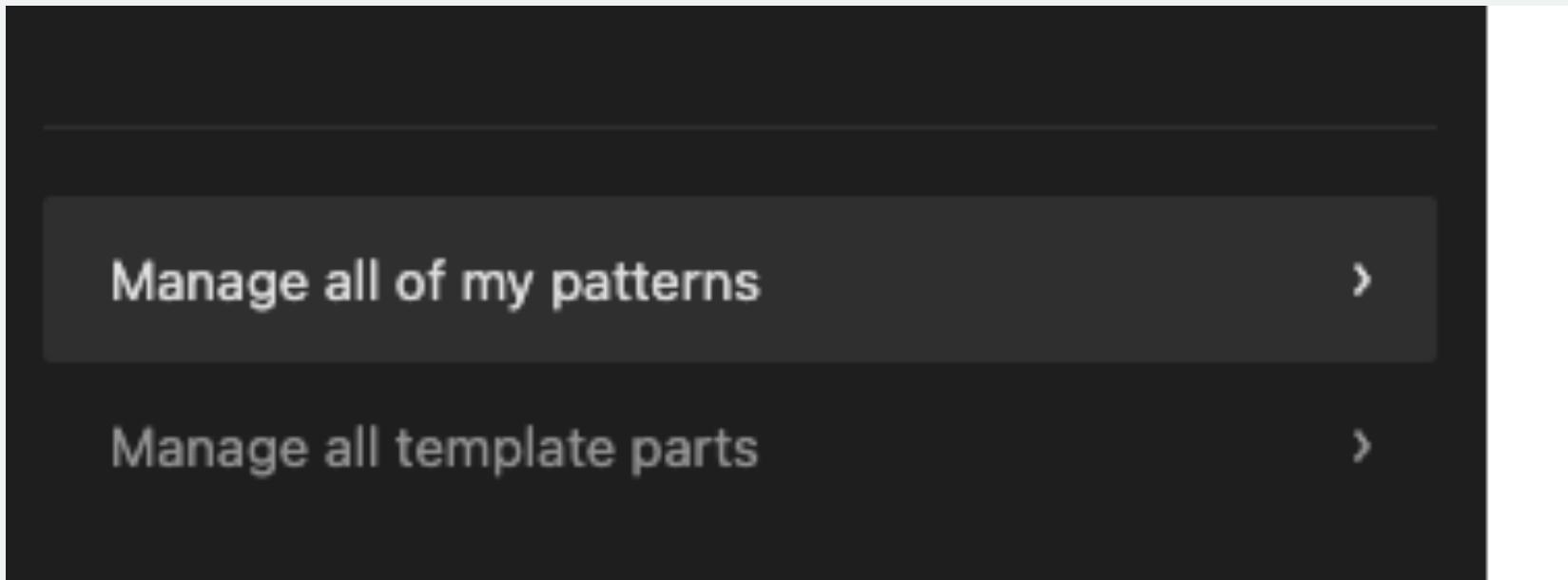
Moving a customized block

- Copy
- Paste
- profit



Moving a pattern

- This used to be a bit easier
- `/wp-admin/edit.php?post_type=wp_block`



Moving the site styles

- Not impossible, but prone to failure.
- You can export a theme.json that represents what you've made, but it doesn't import

What I'd like to see

- Easy drag and drop migrations

More env

- <https://github.com/woocommerce/woocommerce-admin/blob/main/docker/wc-admin-wp-env/README.md>
- wp-env run composer composer install
- <https://wordpress.org/support/topic/problem-with-wp-env/>
- https://dev.to/cryptus_neoxys/setting-up-nodejs-with-nvm-on-wsl-2-3828

resources

- https://en.wikipedia.org/wiki/Separation_of_concerns
- https://en.wikipedia.org/wiki/Separation_of_content_and_presentation
- https://www.explainxkcd.com/wiki/index.php/2347:_Dependency
- <https://groups.google.com/g/comp.lang.c++/c/rYCO5yn4IXw/m/oITtSkZOtoUJ>
- <https://alistapart.com/article/separationdilemma/>
- <https://github.com/WordPress/playground-tools?tab=readme-ov-file>
- <https://www.computer.org/resources/importance-of-software-testing>