

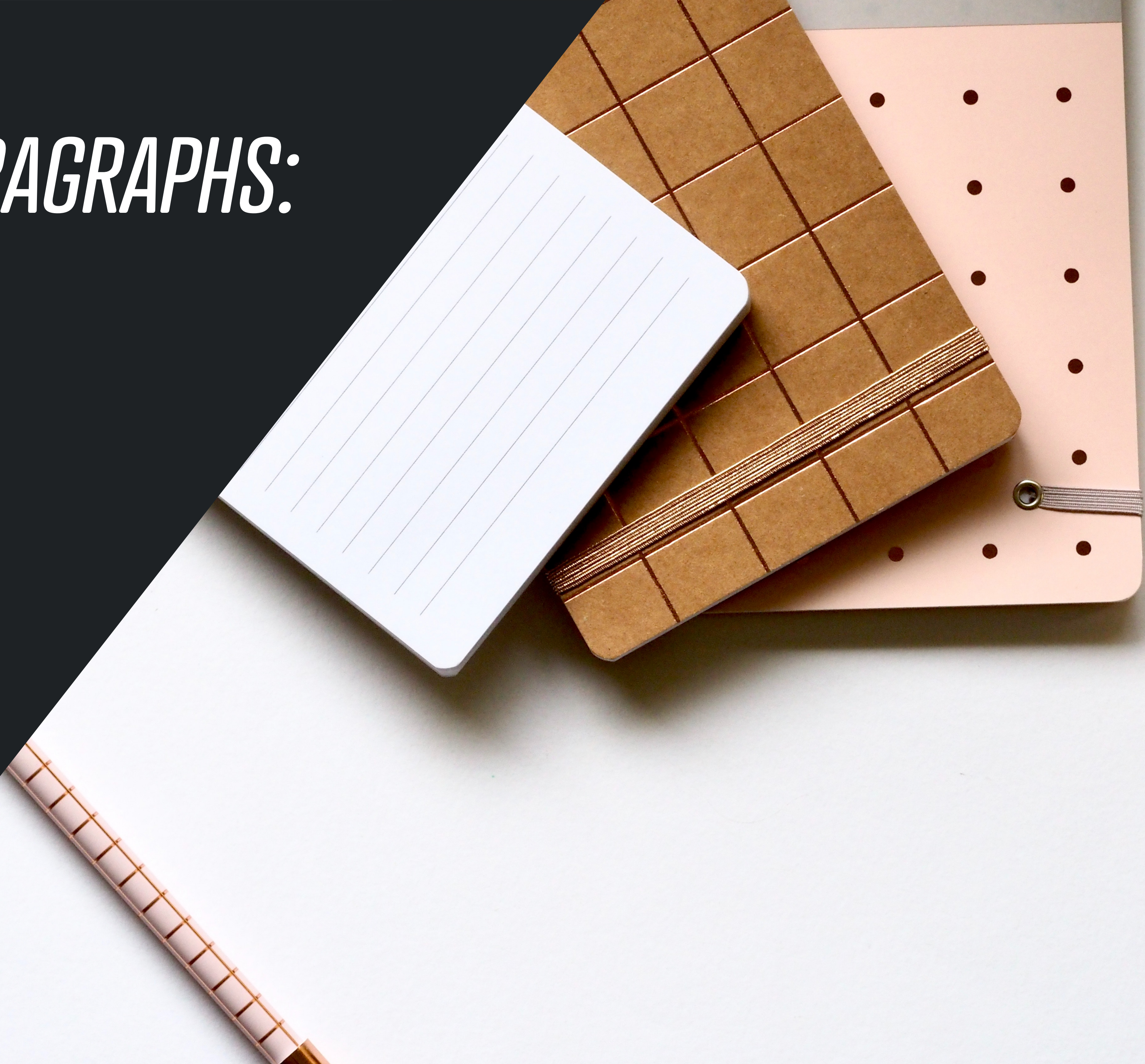
PREPROCESSING PARAGRAPHS:

A BEGINNERS GUIDE

MIDCAMP 2019

FRIDAY: 1:35-2:05PM

ROOM 314A





LARRY WALANGITAN

Senior Developer at Chromatic
[@larrywalangitan](#)





CHROMATIC

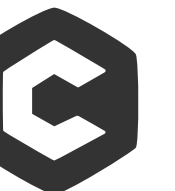
<https://chromatichq.com>

@chromatichq



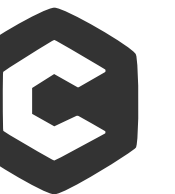
GITHUB REPO:

[HTTPS://GITHUB.COM/WALANGITAN/CHROMATIC_PARAGRAPHS](https://github.com/walangitan/chromatic_paragraphs)



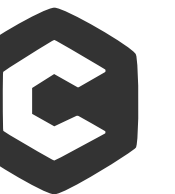
OUTCOMES

- Preprocess paragraphs and structure your preprocessing methods.
- Create custom render arrays and override twig templates.
- Reference nested entities and pull data into paragraph twig templates.
- Debug your preprocessing and twig files without running out of memory.



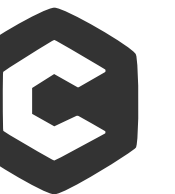
PREREQUISITES

- Familiar with:
 - Drupal 8
 - Twig
 - PHP
 - Paragraphs Module



PARAGRAPHS

- An entity with configurable fields.
- Has different types
- Has view modes







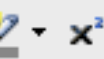






PARAGRAPHS

Paragraph type: *Text*

Title

Welcome

Body

B *I* U ABC    Paragraph Styles Font Family Font Size        

Welcome on our site

Path: p

[Disable rich-text](#)

Text format Full HTML

[More information about text formats](#)

- Lines and paragraphs break automatically.

Remove

Paragraph type: *Quote*

Author

Awesome customer

Quote *

You guys are awesome!

Remove

Paragraph type: *Slideshow*

IMAGES

[Show row weights](#)

FILE INFORMATION

OPERATIONS

 Selection_272.png (10.86 KB)

Remove

Add a new file

Browse... No file selected. Upload

Files must be less than **2 MB**.
Allowed file types: **png gif jpg jpeg**.

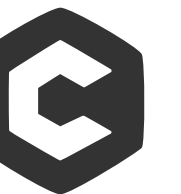
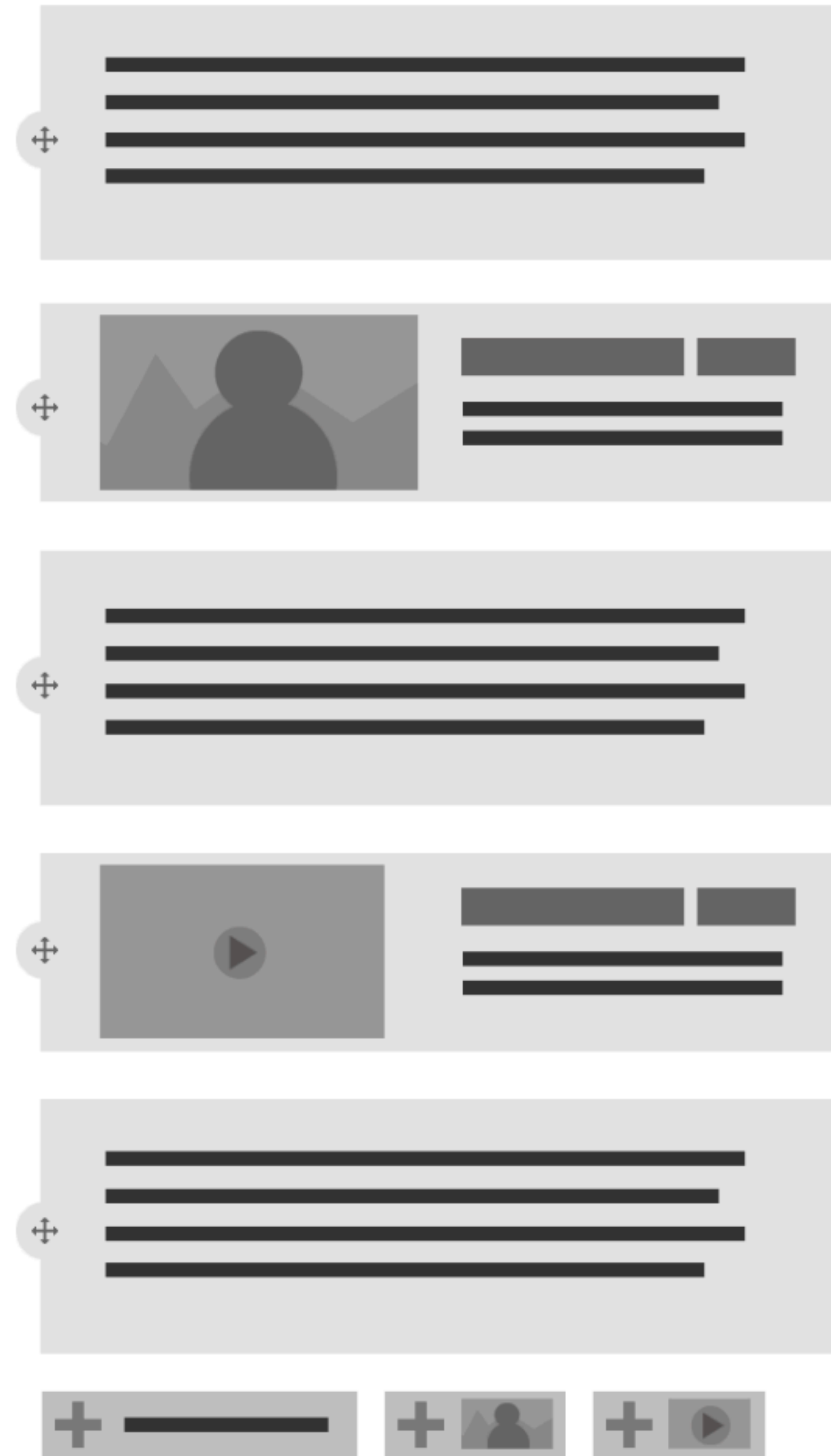
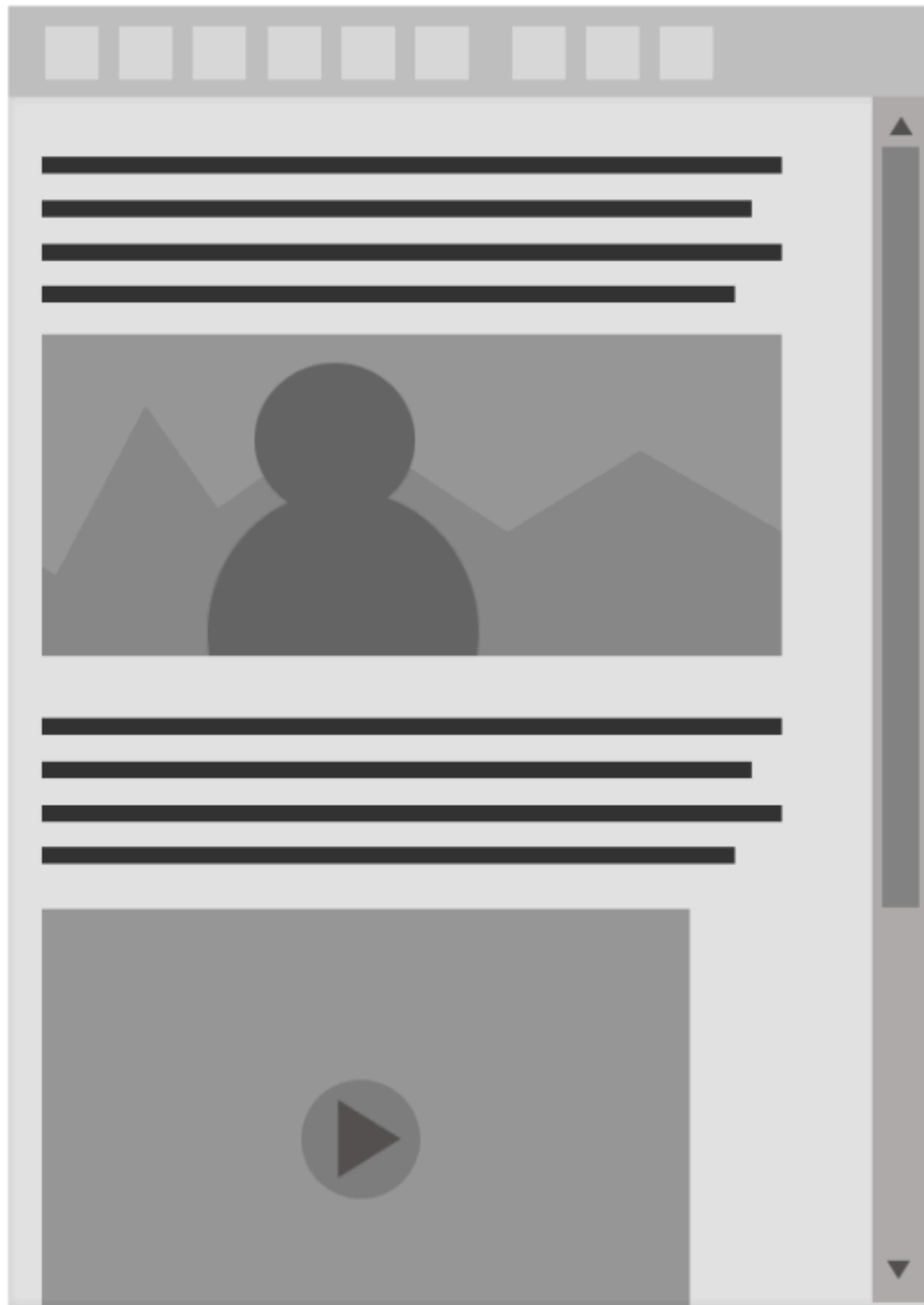
Remove

Paragraph type

Slideshow

Add another paragraph

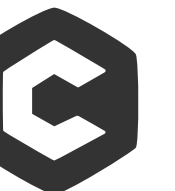




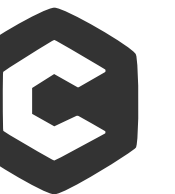


PREPROCESSING

- Data transformation
- Separation of Concerns
 - Logic lives outside of the template
- Customization - *Render Arrays*



A PREPROCESSING RECIPE



01

Each *paragraph type* is defined in **hook_theme.**

02

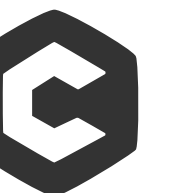
All preprocessing handled by a **Class.**

03

Each *paragraph type* has a corresponding **method.**

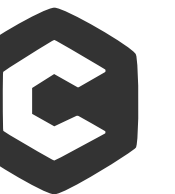
04

Each *paragraph type* has a corresponding **twig template.**



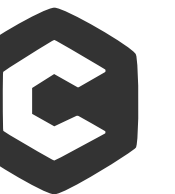
HOOK_THEME: SETUP

- *Base hook*
- *Variables*
- *Path*



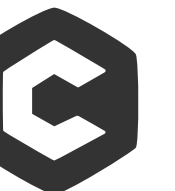
HOOK_THEME: SETUP

```
'paragraph__foo' => [  
  'base hook' => 'paragraph',  
  'variables' => [  
    'header' => NULL,  
    'text' => NULL,  
  ],  
  'path' => $path . '/templates/paragraph',  
],
```



HOOK_THEME: BASE HOOK

- **Base hooks** - defined in another module's *hook_theme*
- **paragraphs.module** has a paragraph base hook
`'base hook' => 'paragraph',`
- Allows us to override and extend the base hook with our own custom variables.



HOOK_THEME: VARIABLES

- ***variables*** - an array of custom variables and their default values.

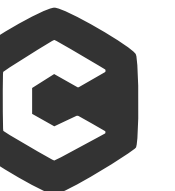
```
'variables' => [  
  'header' => NULL,  
  'text' => NULL,  
],
```



HOOK_THEME: PATH

- *path* - where a paragraph's custom template lives

```
'path' => $path . '/templates/paragraph',
```



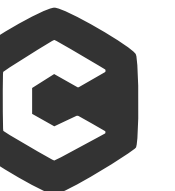
PREPROCESS CLASS

- `src/paragraphPreprocessor.php`

```
<?php
namespace Drupal\chromatic_paragraphs;

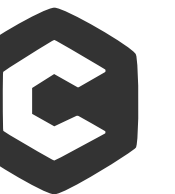
use Drupal\file\Entity\File;

/**
 * Custom Preprocessor methods for Chromatic Paragraphs.
 */
class ParagraphsPreprocess {
```



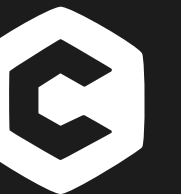
PREPROCESS METHOD

- ***preprocessFoo***
 - Retrieves data
 - Returns variables by reference for the render array defined in ***hook_theme***




```
/**
 * Custom Preprocessor methods for Chromatic Paragraphs.
 */
class ParagraphsPreprocess {

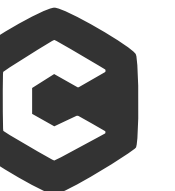
    /**
     * Implements hook_preprocess_paragraph__foo().
     */
    public function preprocessFoo(array &$variables) {
        $content = $variables['content'];
        $variables['header'] = $content['field_header'][0]['#markup'];
        // {{ header }}
        $variables['text'] = $content['field_body'][0]['#text'];
        // {{ text }}
    }
}
```



PARAGRAPH TEMPLATE

- paragraph—foo.html.twig

```
<div class="foo">
  <h2 class="foo-header">{{ header }}</h2>
  <div class="foo-text">
    <p>{{ text }}</p>
  </div>
</div>
```

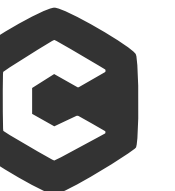


SERVICES.YML

- Create a service:

`chromatic_paragraphs.paragraphs_preprocess`

- Calls **ParagraphsPreprocess** class



chromatic_paragraphs.services.yml ×

```
services:
```

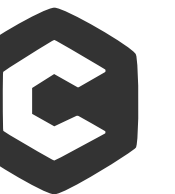
```
  chromatic_paragraphs.paragraphs_preprocess:
```

```
    class: Drupal\chromatic_paragraphs\ParagraphsPreprocess
```



TEMPLATE_PREPROCESS_PARAGRAPH__FOO

- This function prepares variables for our paragraph__foo template.
- Use the new **service** to call our *preprocessing method*.



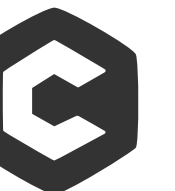

```
/**
 * Chromatic Paragraphs Preprocess Class.
 */
define('CHROMATIC_PARAGRAPHS_PREPROCESS_CLASS', 'chromatic_paragraphs.paragraphs_preprocess');

/**
 * Implements hook_preprocess_paragraph__foo().
 */
function template_preprocess_paragraph__foo(&$variables) {
  \Drupal::service(CHROMATIC_PARAGRAPHS_PREPROCESS_CLASS)->preprocessFoo($variables);
}
```



NESTED ENTITIES

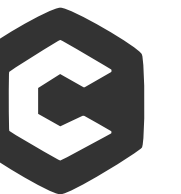
- Retrieving data from entity references.
- Steps:
 1. Load the referenced entity
 2. Return values




```
// Return an entity reference
$nested_entity = !empty($content['field_another_node'][0]['#node']) ? $variables['content']['field_another_node'][0]['#node'] : NULL;
if ($nested_entity) {
  // Return the field_header value of the entity reference.
  $variables['header'] = $nested_entity->field_header->value;
  // {{ header }}
}
```



PUTTING IT ALL TOGETHER



01

Each *paragraph type* is defined in **hook_theme.**

02

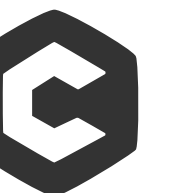
All preprocessing handled by a **Class.**

03

Each *paragraph type* has a corresponding **method.**

04

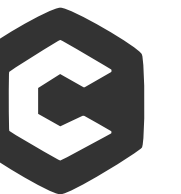
Each *paragraph type* has a corresponding **twig template.**



USE CASE



COMPONENT LIBRARIES



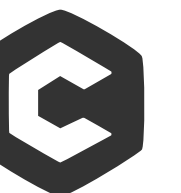
BLOCKQUOTE

Since quoted content can itself contain citations, `cite` elements used for quote attribution should be wrapped in a `footer` element at the end of the blockquote. This ensures that only `cite` elements that make reference to source material are styled as such. Also, note that `cite` is used for the source material's title, but not the author. If you're just quoting an author without the specific work, just use the `footer` element without a `cite`.

“ When you are wrestling for possession of a sword, the man with the handle always wins.

Snow Crash, by Neil Stephenson

```
<blockquote cite='https://en.wikipedia.org/wiki/Snow_Crash'>
  <p>When you are wrestling for possession of a sword, the man with the handle always wins.</p>
  <footer>
    <cite><a href="https://en.wikipedia.org/wiki/Snow_Crash">Snow Crash</a></cite>, by Neil Stephenson
  </footer>
</blockquote>
```



AUDIO PLAYER `</>`



[Download this episode](#)

```
<div class="audiofield">
  <div class="mediaelementaudio_frame">
    <audio class="mejs__player" controls="">
      <source src="http://feeds.soundcloud.com/stream/377215811-thetalkshow-211-the-last-jedi-holiday-spectacular.mp3" />
      Your browser does not support the audio element.
    </audio>
  </div>
  <label for="mediaelement_player_item_x5JYQIkyfhIpKa4g"><a href="http://feeds.soundcloud.com/stream/377215811-thetalkshow" />
</div>
```

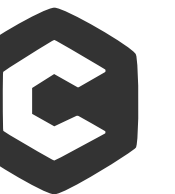


WHEN IT ALL GOES WRONG



DEBUGGING TOOLS

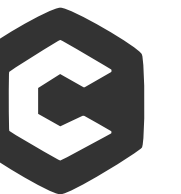
- Module: Twig Vardumper
- Symfony Component: var_dumper




```
{# Credit: MortonDK #}  
{# Returns all available values for a template #}  
<ol>  
  {% for key, value in _context %}  
    <li>{{ key }}</li>  
    {% if loop.index == 2 %}  
      <pre>{{ dump(value) }}</pre>  
    {% endif %}  
  {% endfor %}  
</ol>
```



QUESTIONS & ANSWERS



GET INVOLVED!





CONTRIBUTION DAY - SATURDAY

- ***You don't have to know code to give back!***
- New Contributor training 10am-12pm with **AmyJune Hine** of Kanopi Studios
- 10am to 4pm



SESSION REVIEW: MID.CAMP/215

