Web Components Are Easier Than You Think

(They're Not Even Half as Scary as a Horde of Cyborg Zombies)

by John Rhea

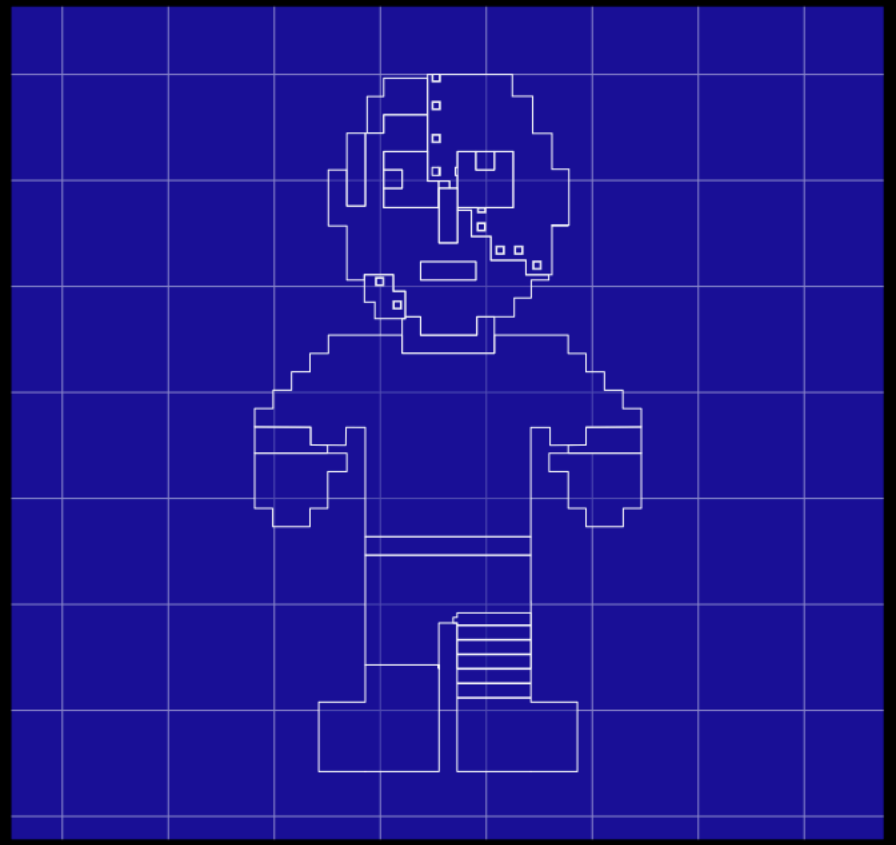
What is a Web Component?

2

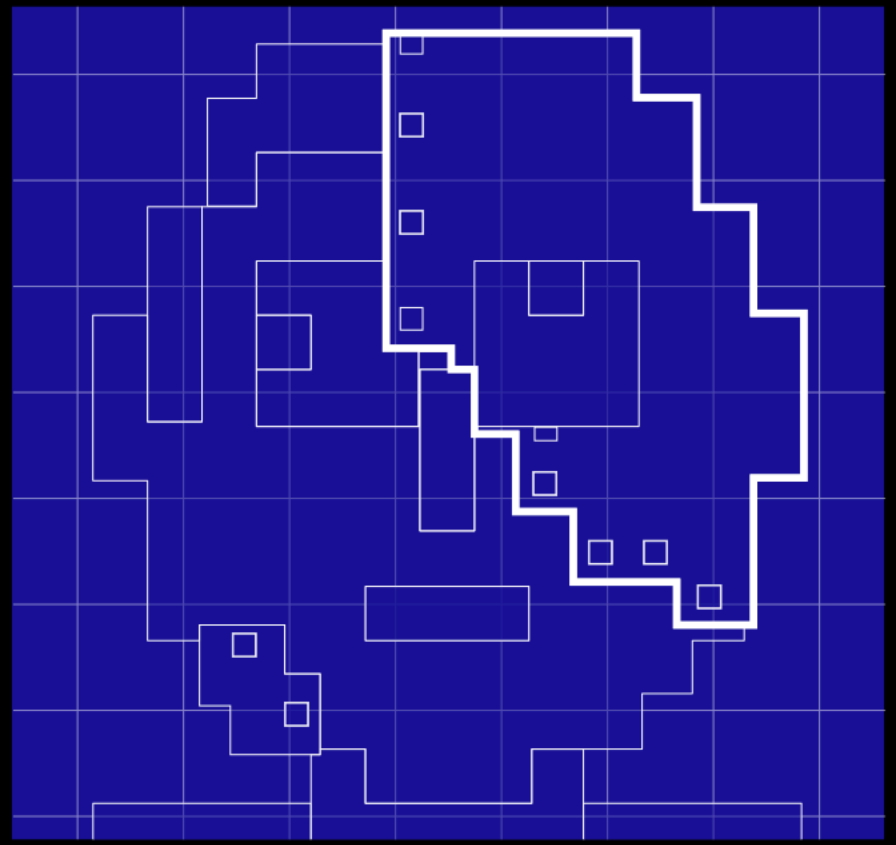
Flavors

Autonomous Custom Elements

Customized Built-in Elements



HTML



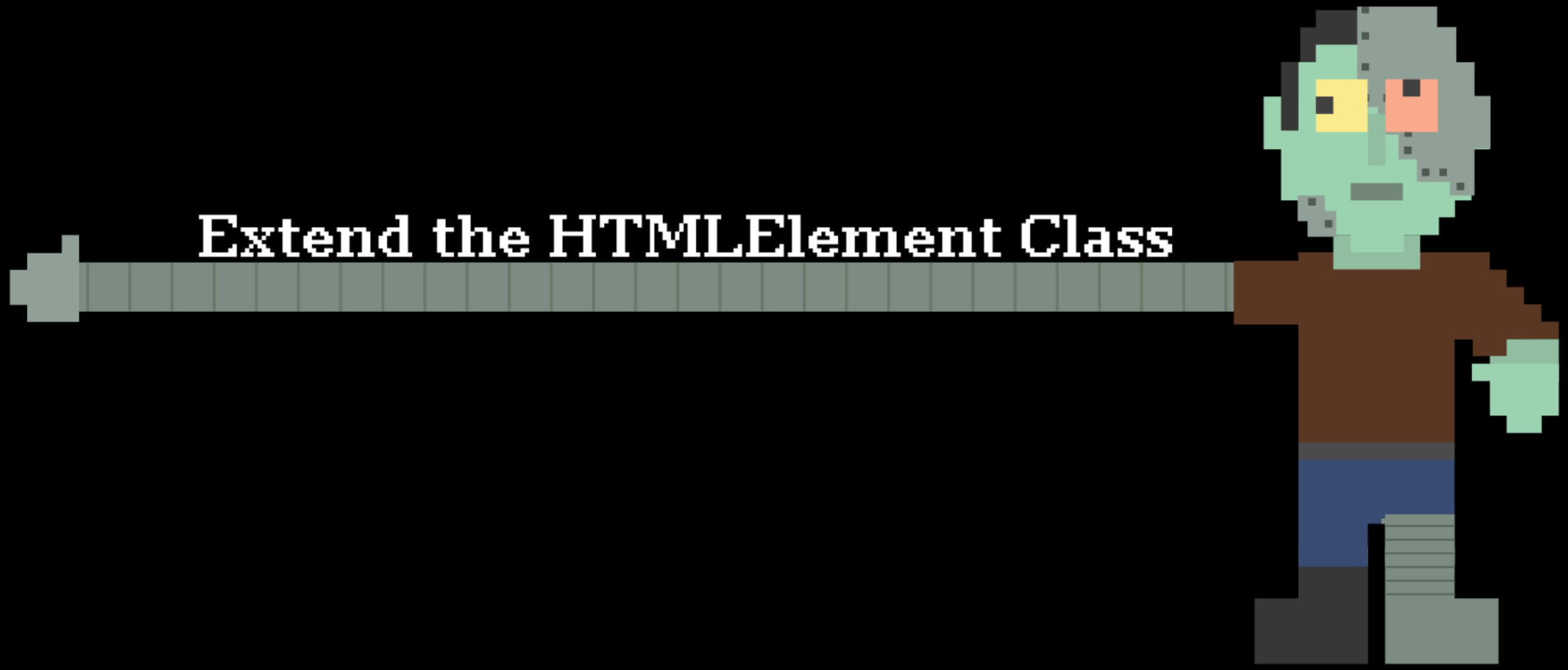
<template>

<slot>

Encapsulated CSS

JavaScript

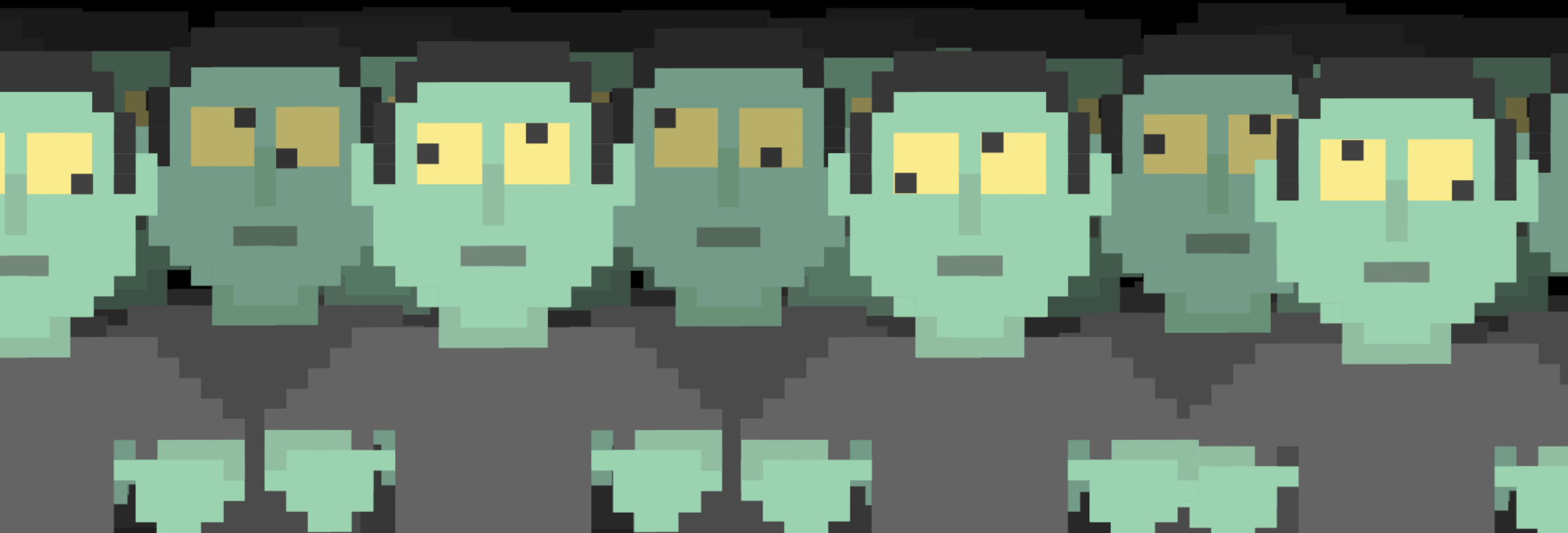
Extend the HTMLElement Class



JavaScript

Night of the Living Code

a.k.a. Failing Publicly



WordPress & Web Components

1. Use it in the editor
2. Output from a Theme
3. Output from a Block

Prerequisite 1

Add Your HTML Template to the Footer

```
<?php function diy_ezwebcomp_footer() { ?>
  <!-- print/echo template code. -->
<?php }
add_action( 'wp_footer', 'diy_ezwebcomp_footer');
```

Enqueue Any Unencapsulated Styles

```
wp_enqueue_style( 'ezwebcomp_style',
  get_template_directory_uri() .
  '/ezwebcomp.css', '', '1.0',
  'screen' );
```

Prerequisite 2

Prerequisite 3

```
wp_enqueue_script( 'ezwebcomp_js',
  get_template_directory_uri() .
  '/js/ezwebcomp.js', '', '1.0',
  true );
```

Enqueue JavaScript

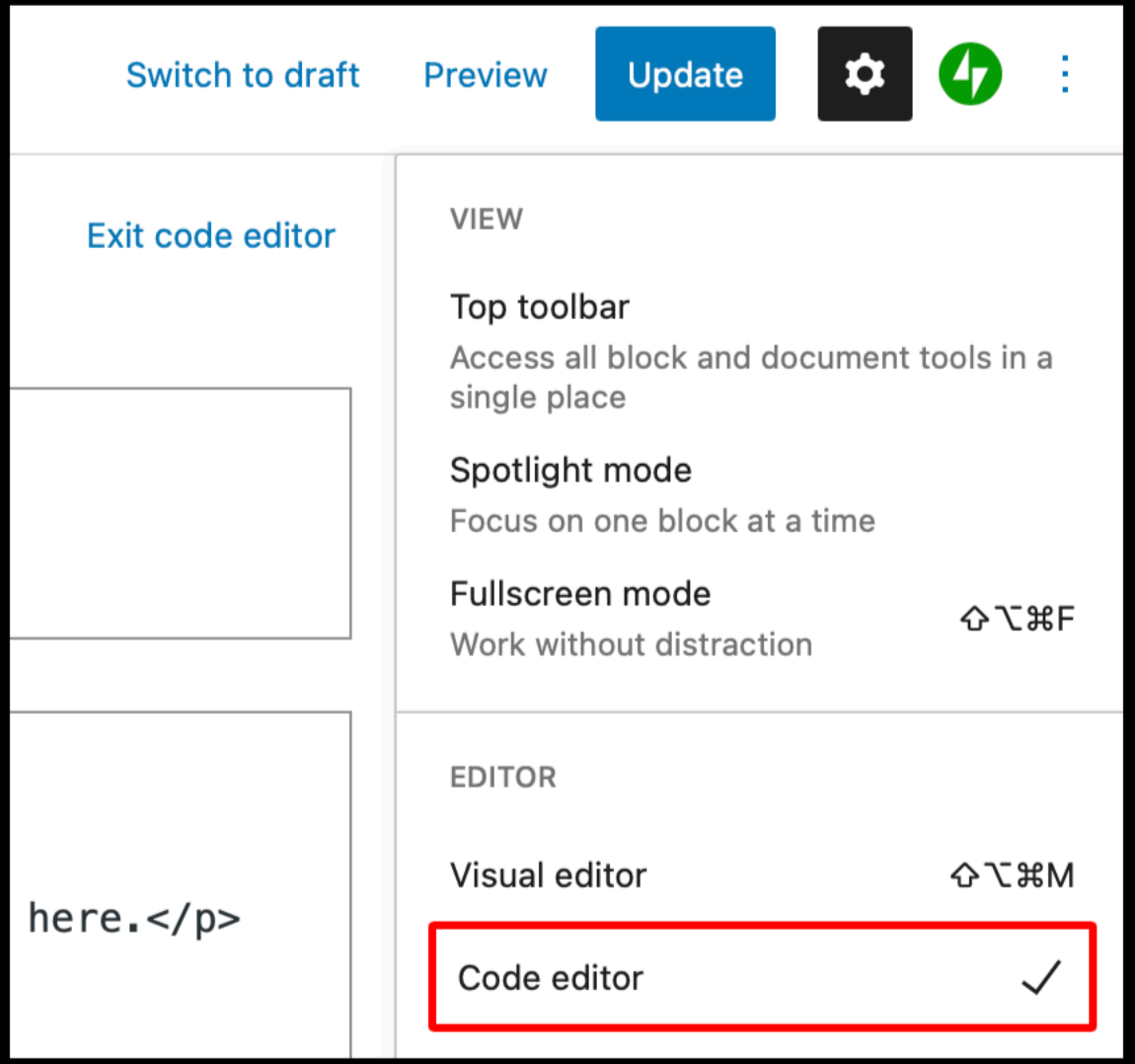
1 Use it in the editor

Don't Filter Out Your Web Component

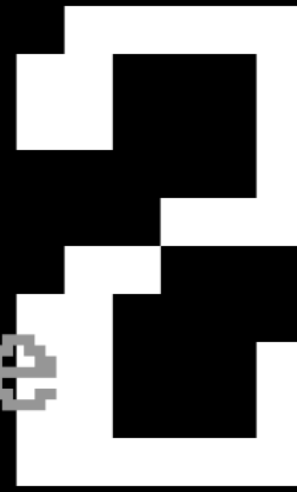
Don't Filter Out Your Web Component

```
function add_webcomp_to_kses($the_allowed_tags) {
  $the_allowed_tags['zombie-profile'] = array();
  foreach ($the_allowed_tags as &$tag) {
    $tag['slot'] = true;
    $tag['part'] = true;
  }
  return $the_allowed_tags;
}
add_filter('wp_kses_allowed_html',
  'add_webcomp_to_kses');
```

Don't Filter Out Your Web Component



Output from a Theme



Theme Output

Switch to draft Preview Update

Post Block

A. C. Shanty

What's a pirate's favorite letter?
Rrrrrrrr!
Aye! But his first love be the C

Status & visibility

Visibility Public

Publish July 21, 2021 8:13 pm

Stick to the top of the blog

Author

johnrhea

Move to trash

30 Revisions

Permalink

Categories

Tags

Featured image

Replace Image

Remove featured image

Custom Fields

Name	Value
Age	41
Infection Date	November 7, 2026
Interests	- sea shanties - reality tv - brains - more brains

Delete Update

Add New Custom Field:

Name	Value
— Select —	

Enter new

Add Custom Field

Custom fields can be used to add extra metadata to a post that you can use in your theme.

Theme Output

```
<zombie-profile>
  <img src="<?php echo wp_get_attachment_image_url(
        get_post_thumbnail_id() ); ?>" slot="profile-image">
  <span slot="zombie-name"><?php echo get_the_title(); ?></span>
  <span slot="z-age">
        <?php echo get_post_meta(the_ID(), 'Age', true); ?></span>
  <span slot="idate">
        <?php echo get_post_meta(the_ID(), 'InfectD', true); ?></span>
  <ul slot="z-interests">
        <?php echo get_post_meta(the_ID(), 'Interests', true); ?></ul>
  <span slot="statement"><?php echo get_the_content(); ?></span>
</zombie-profile>
```

Output from a Block

Block Output - PHP

```
function ez_webcomp_register_block() {
  wp_register_script('ez-webcomp', plugins_url('block.js', __FILE__),
    array('wp-blocks', 'wp-element', 'wp-editor'),
    filemtime(plugin_dir_path(__FILE__) . 'block.js')
  );
  wp_register_style(
    'ez-webcomp', plugins_url('ezwebcomp-style.css', __FILE__),
    array(), filemtime(plugin_dir_path(__FILE__) . 'ezwebcomp-style.css')
  );
  register_block_type('ez-webcomp/zombie-profile', array(
    'editor_style' => 'ez-webcomp', 'editor_script' => 'ez-webcomp'
  ));
}
add_action('init', 'ez_webcomp_register_block');
```

Block Output - JS - Attributes

```
attributes: {
        name: {
          type: 'string',
          source: 'text',
          selector: '.zname', },
        mediaID: {
          type: 'number', },
        mediaURL: {
          type: 'string',
          source: 'attribute',
          selector: 'img',
          attribute: 'src', },
        age: {
          type: 'string',
          source: 'text',
          selector: '.age', },
        infectdate: {
          type: 'date',
          source: 'text',
          selector: '.infection-date' },
        interests: {
          type: 'array',
          source: 'children',
          selector: 'ul', },
        statement: {
          type: 'array',
          source: 'children',
          selector: '.statement',
        },
    },
```

Block Output - JS - Save Function

```
save: function (props) {
    var attributes = props.attributes;
    return el(
      'zombie-profile',
      {},
      el('img', {
        src: attributes.mediaURL,
        slot: 'profile-image'
      }),
      el(RichText.Content, {
        tagName: 'span',
        slot: 'zombie-name',
        className: 'zname',
        value: attributes.name,
      }),
      el(RichText.Content, {
        tagName: 'span',
        slot: 'z-age',
        className: 'age',
        value: attributes.age,
      }),
      el(RichText.Content, {
        tagName: 'span',
        slot: 'idate',
        className: 'infection-date',
        value: attributes.infectdate,
      }),
```

Block Output - JS - Save Function - 2

```
        el(RichText.Content, {
          tagName: 'ul',
          slot: 'z-interests',
          value: attributes.interests,
        }),
        el(RichText.Content, {
          tagName: 'span',
          slot: 'statement',
          className: 'statement',
          value: attributes.statement,
      })
    );
},
```