Create a custom loop in WordPress

When researching how to use Gutenberg Ramp to partially allow Gutenberg on a WordPress site, I decided that I wanted to use a custom post type for the test so I wouldn't screw up the existing content for my blog.

There is a problem with this idea. Custom post types don't appear in the default loop of a WordPress type so they wouldn't appear along with regular posts in the blog homepage.

Most times, this is what we want. We want to use the custom post types on their own specialized loops. However, in this case, it is not what I want so I have to figure out a way to integrate posts and essays (the custom post type) in the same loop.

The code below is a modified version of the experiment I ran on my development syste. It is not complete, just enough to show the steps I took.

```
<?php
// WP_Query arguments
$args = array(
  'post_type' => array( 'essa'post_type'),
); // 1

// The Query
$myQuery = new WP_Query( $args ); // 2
  if ( $myQuery->have_posts() ) : // 3
    if ( is_home() && ! is_front_page() ) : // 4
?>
      <header>
        <h1 class="page-title screen-reader-text"><?php single_post_title(
      </header>
    <?php
    endif;

    // Start the Loop
```

```
    while ( $myQuery->have_posts() ) : $myQuery->the_post(); // 5

      get_template_part( 'template-parts/content', get_post_type() ); 'ter

    endwhile; ?>

  else :

    get_template_part( 'template-parts/content', 'none' ); // 7

  endif;
  ?>
```

The custom loop code does the following:

1. Sets up an array of arguments for the custom loop. In this case we only want an array of `post_type` to hold the values we want to use in the loop: posts and essays in the singular form
2. Creates the new query using the `$args` array
3. Checks if the query has posts (is not empty)
4. Checks if this is the home page (blog index) and not the front page (static home defined in admin)
5. Loop through the posts
6. Display their content using the corresponding template part if there is one
7. If the loop is empty (the check on step 3 returned false) display the none content part to the user