



#### Learning Objectives:

- ▶ Set up an interior page template on WordPress using a predefined template design
- ▶ Import dynamic content using WordPress functions
- ▶ Adjust theme templates for parent and children pages
- ▶ Create a menu of children page links for a parent page
- ▶ Order menu items using a custom sequence
- ▶ Set up dynamic navigation menus that are controlled from within the WordPress Admin

References are to the slides from Week 4

#### Task 1: Set up an interior page template on WordPress using a predefined template design

- Start your website and login to the admin dashboard
- Add two new pages: About Us, Privacy Policy and add some dummy text to each.
- View each new page in the browser. ( may look similar to slide 9)
- Open **interior\_page.html** in VisualStudio Code - available from the starterCode zip on myElearning from Week 3 and copy the block of code that begins just after the header element until the div that begins with:

```
<div class="page-section page-section--beige">
```

- Edit **page.php** in your themes folder: remove the output lines from within the loop we've been working on and add the copied code from interior\_page.html (see slide 11)
- Refresh and observe the new design (similar to slide 12)

#### Task 2: Import dynamic content using WordPress functions

- Use a dynamic title in **page.php**, by replacing the headline with a call to **the\_title()** (slide 14)
- Remove the generic content from the div with the class 'generic-content' and replace with a call to **the\_content()** (slide 16)
- Set up a dynamic path to the background image used on the interior page (see slide 18)
- Set up a dynamic unique title for the interior page in the browser tab by editing the **functions.php** file (see slide 20).
- Refresh and verify that all of the changes have resulted in dynamic content for the About Us and Privacy Policy pages
- Open the **header.php** file and fix the blank link to the About Us page by removing the # and inserting code that echos the url of the particular page. (see slide 24)

Fictional University

About Us Programs Events Campuses Blog Login Sign Up

- Fix the link on the logo so that when clicked, it lets users get back to the homepage or root url. (slide 25).
- Fix the navigation link and logo in the **footer.php** file as well (slide 26).

#### Task 3: Adjust theme templates for parent and children pages

- Add two new child pages to the About Us page -> Our History, Our Goals via the dashboard (see slide 29).
- Add a new child page, Cookie Policy, to the Privacy Policy page.
- Edit **page.php** to make the title of the pages in the breadcrumb area (yellow box) dynamic (see slides 31-32).
- Write code that shows the breadcrumb area only if we are on a child page, otherwise don't show it. (see slides 34-36). You should have something looking like slide 37.
- Write code to fix the blue box in the breadcrumb area to reflect the parent page title (see slides 38-40). Verify that the links work in both parent and child pages for what you've set up so far

#### Task 4: Create a menu of children page links for a parent page

- In **page.php**, delete the hardcoded links for the right-hand menu area on the page (see slide 45).
- Output all of the pages present in the site in the menu but delete the page title label from being printed (see slides 46-49)
- Fill in the numerical value of the postID for the following pages in the table. Compare your values with your seat mate's values.

Page	PostID	Page	PostID
About Us		Privacy Policy	
Our History		Cookie Policy	
Our Goals		Sample Page	

- Write code that uses an associative array in order to only output the child pages of the Privacy Policy page using the id from the table above (see slide 50). Check that the code works for Privacy Policy. Does it work for the About Us page?
- Replace the postID from the step above with **get\_the\_id()** (see slide 53). Does it now work for the About Us page? How about the Our History page?
- Insert an if/else block that checks whether we are on a child or parent page so that the code works for both parent and child pages. (see slide 55).
- Fix the blue header block on the menu so that the title and the href link value are dynamic (slides 58-59).
- Tweak your code to cater for those pages that are not parent or child pages. No menu should be shown on these types of interior pages. (see slides 60-63).

#### Task 5: Order menu items using a custom sequence

- Write code to order the child pages in the menu in the specific order we created them (slide 64-66).

#### Task 6: Set up dynamic navigation menus that are controlled from within the WordPress Admin

- Register a new menu in WordPress via the admin dashboard (slides 71-73).
- Create a new menu by following the instructions on slide 74.
- Edit the header.php file according to slide 75. You should see something like slide 76.