Integrating a main menu

• just following the procedure and menu can be in wp to use.

First • Create the menu + order it
• Integrate the menu (where it is needed) = using PHP snippet
• Use CSS to display it the way you want
Creating the menu .. some steps need to be done on wordpress side.
• Dashboard Appearance > Customize > Menus
• Click button Create New menu
• Supply a name + click Next
• Click button Add item
Adding links to the menu
• Click on a page from the right panel (to adds a link to the menu)
• Selecting a page from the right menu = to set different options
Integrating the menu in a template
• Integrate it to a page (very often in header.php)
• Use following PHP snippet:
For more information about WordPress menus:

Development of a dynamic web site Lesson plan
Editing themes files
• It is possible to open the themes files directly from the folders using your favourite editor
BUT preferable to use WordPress tools
• WordPress is a complex ecosystem = using WP tools prevent potential errors
Themes code editor
• Dashboard Appearance > Theme Editor = to visualize and edit theme's files
(select them from the right panel.)
• Bottom left of the tool = drop-down menu => access to different functions documentation
• Make sure to hit the Update file button before leaving the page.
Themes custom CSS editor
• Dashboard Appearance > Customize > Additional CSS
• Main panel = show your home page + the CSS editor's panel (left side)
• CSS added automatically shows on the preview (no need to refresh)
• To visualize other pages of the site to style them, simply use the links network
Customization advices
Use the browser's inspector
To identify the specific elements and style you want to edit
Use of existing classes and IDs
Ensures elements are coherent throughout the site
BUT many says it is preferable to use custom stylesheet
Using the Additional CSS tool = adds CSS without modifying the existing ones
Use of custom stylesheet
Best way to edit a WordPress theme
Make sure custom CSS file is declared after the default ones

The function.php file

This allows what can be possible and how simple could be using function in WordPress
• functions.php = already created + mandatory
• WordPress functionalities can be activated + personal custom functions can be coded in it
• hooks system is used to launch proper function at key moments = IMPORTANT concept
function.php purposes

• Several things can be accomplished using this file
BECAUSE many WordPress functionalities cannot be activated from the dashboard
• Previously, we have used this file to activate the feature image + automatic title tag generation
• Among others, the following actions can be performed using function.php file :
• Activate feature image function;
• declare menus and widgets zones ;
• declare CSS stylesheets and JS scripts files;
• declare new types of publications and taxonomies ;
• declare custom functions;
• manage AJAX requests;
• reformat URLs;
• customize certain plug-ins settings;
• customize administration's interface;
• create API Rest paths;
The hooks system
• The hook system allows you to use a specific features at various moments
• It makes it possible to interact with the theme and the extensions
SO their default behaviours may be modified (without touching their native codes)
The hooks : 2 different types are seen during session . What is done to write their name . Simply put them in bracket to visible (Actions / Filters)

Actions : what sort of trigger will user implement on active page ...Actions define key moments when our own custom functions are triggered allowing
additional features.

Filters : this will allow how the trigger be showed up during action time. Filters are used to retrieve values at various given moments so they can be modified or
updated

Example of the use of hooks in function.php file

• functions.php file will mainly use hooks such as in the following example.
Explanation :
adding_img()
Custom function to drag img on admin sitr
add_action()
Function add_action() calls the hook. The key moment is admin_menu. which means that
the function is called when WordPress gets ready to display the admin menu.
So, in resume, the function tells WordPress to launch the custom function which add ing during launch of admin dashboard

Loading external scripts and stylesheets

• USUALLY we link our scripts and stylesheets in the header + being careful to place the links to custom files last
• It cannot be done this way with WordPress BECAUSE theme/extensions declare their own scripts and stylesheets
• WordPress automatically loads and manages all these files in order using the function wp_head()
(placed in header.php)
THEREFORE scripts + stylesheets have to be declared in functions.php
Declaring scripts and stylesheets in functions.php
Done using add_action('wp_enqueue_scripts', 'your_function');

```
<?php
function myproject_register_assets() { // Custom function name
wp_enqueue_script( // To declare jQuery CDN
'jquery',
'https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js',
false,
'3.4.1',
```

```
true
);

wp_enqueue_script( // To declare external JS
'myproject',
get_template_directory_uri() . '/js/script.js',
array( 'jquery' ),
'1.0',
true
);

wp_enqueue_style( // To declare a stylesheet at the root
'myproject',
get_stylesheet_uri(),
array(),
'1.0'
);

wp_enqueue_style( // To declare another CSS file
'myproject',
get_template_directory_uri() . '/css/main.css',
array(),
'1.0'
);
}
add_action( 'myproject_enqueue_scripts', 'myproject_register_assets' );
```

First parameter is js file . Will be called on using template

Second will push others to use on tht tine.

wp_enqueue_script() : Allows to declare a JS file in the page
wp_enqueue_style(): Allows to declare a CSS file
Parameters to add to the function :
First parameter (the handle):
Name given to your file (e.g.: your project's name).
Second parameter (file address):
Using get_template_directory_uri() supplies the path to the theme. To declare a CSS file in the theme's root, the use get_stylesheet_uri() alone.
Array():
The array is used to have WordPress scripts and stylesheets declared first.
Boolean:
A boolean is used last in scripts declaration. If true, the script is placed at the end of the page, if false the script is placed at the beginning of the page.
jQuery :
In order to use the latest version of jQuery, since WordPress does it another way for older Explorer browsers, we have declared another version.