Integrating a main menu  
• A few steps are required:  
• 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  
• 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  
• 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 (Actions / Filters)  
Actions :  
Actions define key moments when our own custom functions are triggered allowing   
additional features.  
Filters :  
Filters are used to retrieve values at various given moments so they can be modified or   
updated