

import a template in a Wordpress theme

Creation of the theme

create the basic theme files

- style.css
- screenshot.png
- functions.php
- Copy the `index.html` file from your HTML template and all its dependencies
- rename `index.html` to `index.php` to create our basic template
- In `index.php` Modify the relative links like css or images link by using this functions :

```
get_template_directory_uri    //-> theme parent
get_stylesheet_directory_uri  //-> theme enfant
```

example :

```

```

Don't forget to add the '/' after the function call

Create includes

Separate header and footer code in include files :

```
header.php
footer.php
```

- We replace in `index.php` the code of the header by `<?php get_header (); ?>` and the code of the footer by `<?php get_footer(); ?>`
- Use the wordpress tag templates to include scripts in the head and just before closing BODY tag in the new includes `header.php` and `footer.php`

```
<?php wp_head(); ?>    //-> Just before the closing head
<?php wp_footer(); ?>  //-> Just before the closing body
```

- Transfer loading scripts and styles into a function

- Use the wordpress methods in `functions.php`
- https://developer.wordpress.org/reference/functions/wp_enqueue_style/
- https://codex.wordpress.org/Plugin_API/Action_Reference/wp_enqueue_scripts

```
wp_register_script()
wp_enqueue_script()
wp_enqueue_style()
```

Add a nav menu

```
register_nav_menu() -> declaration of the in the functions.php
wp_nav_menu()      -> display the menu code in the template
```

- https://codex.wordpress.org/Function_Reference/register_nav_menu/
- https://developer.wordpress.org/reference/functions/wp_nav_menu/

Add widget areas in the footer (sidebar)

```
`register_sidebar()` //-> declaration of the widget area in functions.php
`dynamic_sidebar()` //-> usage in the template
```

Do not confuse with `get_sidebar ()` which gets the `sidebar.php` inclusion file

Home page template and index.php template for the loop

Created a `front-page.php` template for the homepage after defining a page and changing the display preferences of Wordpress: `settings -> read -> static homepage` by specifying the 'home' page created for the occasion.

```
index.php      -> nouveau contenu avec la boucle (voir ci-dessous)
front-page.php -> contenu actuel de index.php
header.php
footer.php
```

In `index.php` we modified the content to use the Wordpress loop:

```
<?php if (have_posts()) : ?>
    <!-- If there is posts to display -->

    <?php while (have_posts()) : the_post(); ?>
        <!-- As long as he finds articles, he will execute the following
instructions for each article -->
```

```

        <div class="post">
            <h3 class="post-title">
                <a href="<?php the_permalink(); ?>"><?php the_title(); ?></a>
            </h3>
            <p class="post-info">
                Posted on <?php the_date(); ?> in <?php the_category(' ', ' '); ?>
by <?php the_author(); ?>.
            </p>
            <div class="post-content">
                <?php if(is_single() || is_page() ) {
                    the_content();
                } else {
                    the_excerpt();
                }
            <?>
            </div>
        </div>
    <?php endwhile; ?>
    <!-- end of the while loop -->
    <?php else : ?>
    <!-- If no post has been found... -->
    <p class="nothing">
        Nothing to display
    </p>
    <?php endif; ?>

```

- List of the template tags:
- https://codex.wordpress.org/Template_Tags

Templates for the pages, and the posts

You can use `index.php` to manage all views, but you can also create templates for each type of post (detail page).

Creating templates for specific views:

```

single.php
page.php

```

Once these files are created, you can customize the display of each template.

Templates pour les catégories et les tags

```

archive.php

```

This template allows to modify the display of the lists of articles (posts)

Post thumbnails

To make our theme compatible with the images in the front page, you have to declare the support of the function in the `functions.php` file:

- https://developer.wordpress.org/reference/functions/add_theme_support/

```
add_theme_support( 'post-thumbnails', array( 'post' ) );  
// the second parameter defines in which type of post the function is  
activated, here only the 'post' (not the pages)
```

- Then to display the featured images in the template (here in `single.php`):

```
<?php  
    //Displays the post thumbnail  
    if ( has_post_thumbnail() ) {  
        the_post_thumbnail();  
    }  
?>
```

- To create a specific image size for your theme :

in `functions.php` :

```
add_image_size( 'post-wide-thumbnail', 400, 140, true );  
//(image slug, width, height, cropped);
```

- The thumbnails must then be regenerated with the help of the 'regenerate thumbnails' extension (to be installed) to recreate all the thumbnails with the new image sizes.
- We specify in the template the slug of the image size that we want to use (here in `single.php`):

```
the_post_thumbnail('post-wide-thumbnail');
```

Page Template

- You may want to display the `sidebar.php` in a template, we add it to the `page.php` file:

```
<?php get_header(); ?>  
<section>  
    <div class="wrapper">  
        <div class="container">  
            <article class="page-content">
```

```

        <?php if (have_posts()) : ?>

            <?php while (have_posts()) : the_post(); ?>

                <div class="post">
                    <h3 class="post-title">
                        <a href="<?php the_permalink(); ?>"><?php
the_title(); ?></a>
                    </h3>

                    <div class="post-content">
                        <?php
                            the_content();
                        ?>
                    </div>
                </div>
            <?php endwhile; ?>

        <?php endif; ?>
    </article>
    <aside class="sidebar">

        <!-- Here is the inclusion of the sidebar: -->
        <?php get_sidebar(); ?>

    </aside>
</div>

</div>
</section>
<?php get_footer(); ?>

```

Then you have to create a new widget area in `functions.php` and call it in our `sidebar.php` to display the widgets.

Creating a modified loop in a template

It is thanks to the parameters of the `query_posts` function that we can modify the main Wordpress loop.

Take the example of the `category.php` template. If we want this template to display a paged list of up to 20 items, this is what we will call the Wordpress loop:

```

<?php get_header(); ?>

<div class="content">
    <?php query_posts('posts_per_page=20'); ?>
    <!-- We modify the default Wordpress loop parameter -->
    <?php get_template_part('loop'); ?>

```

```
<!-- We call the loop include : loop.php -->
</div>

<?php wp_reset_postdata(); ?>
<!-- Lets restaure the default parameters for future loop calls -->

<?php get_sidebar(); ?>
<?php get_footer(); ?>
```

The `WP_Query` function allows you to create multiple loops but is more complex to set because it creates objects.

According to the official documentation, we can perfectly use the function `query_posts ()` provided to reset it with the `wp_reset_postdata ()` function.