

les étapes pour commencer son thème

parce que je suis un peu perdu..

PRÉPARÉ PAR:

Bernard Imbert

Jean-François Leblanc

Adapté par Mathieu Dupont

Pour le cours 420-964-EM

01

créer les gabarits

commençons par la base

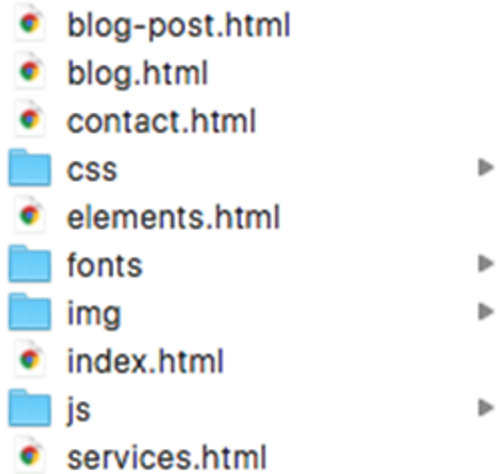
avant même de commencer à organiser notre thème WordPress, il faut envisager la structure de notre site Web. Pour cette raison, nous allons faire notre html et css

- construire les gabarits html que nous utiliserons dans notre site
- faire le css des gabarits
- faire le javascript des gabarits

en bref, faites tout ce que vous avez fait depuis votre cours de Conception Web 1



Quelle serait
la prochaine étape
lorsque nos gabarits
html, css et js
sont terminés?



retour sur « the loop »

La boucle sert à afficher le contenu d'une page, de plusieurs posts (articles) ou d'un autre type de contenu

```
<!-- s'il y a au moins un article à afficher -->
```

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

```
    <!-- tant qu'il y a un nouvel article à afficher, roulez la boucle -->
```

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

```
        <!-- structure html pour afficher le contenu -->
```

```
    <?php endwhile; ?>
```

```
<?php else: ?>
```

```
    <!-- s'il n'y a aucun article à afficher -->
```

```
<?php endif; ?>
```

02

créer les fichiers de base d'un thème

- créer un dossier au nom de votre thème
- créer un fichier `index.php` vide
- créer un fichier `style.css` avec le code nécessaire (mais sans css)
- faire une capture pour votre gabarit nommée `screenshot.png`



il faut maintenant mettre en place notre structure de base

Commençons avec index.php.
prenez le code html que vous
avez créé et copiez-collez le dans
index.php

```
index.php x
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Labs - Design Studio</title>
5   <meta charset="UTF-8">
6   <meta name="description" content="Labs - Design Studio">
7   <meta name="keywords" content="lab, onepage, creative, html">
8   <meta name="viewport" content="width=device-width, initial-scale=1.0">
9   <!-- Favicon -->
10  <link href="img/favicon.ico" rel="shortcut icon"/>
11
12  <!-- Google Fonts -->
13  <link href="https://fonts.googleapis.com/css?family=Oswald:300,400,500,700|Roboto:300,400,700" rel="stylesheet">
14
15  <!-- Stylesheets -->
16  <link rel="stylesheet" href="css/bootstrap.min.css"/>
17  <link rel="stylesheet" href="css/font-awesome.min.css"/>
18  <link rel="stylesheet" href="css/flaticon.css"/>
19  <link rel="stylesheet" href="css/magnific-popup.css"/>
20  <link rel="stylesheet" href="css/owl.carousel.css"/>
21  <link rel="stylesheet" href="css/style.css"/>
22
23
24  <!--[if lt IE 9]>
25    <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
26    <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
27  <[/endif]>
28
29 </head>
30 <body>
31   <!-- Page Preloader -->
32   <div id="preloader">
33     <div class="loader">
34       
35       <h2>Loading.....</h2>
36     </div>
37   </div>
38
39
40   <!-- Header section -->
41   <header class="header-section">
42     <div class="logo">
43       <!-- Logo -->
44     </div>
45     <!-- Navigation -->
46     <div class="responsive"><i class="fa fa-bars"></i></div>
```



N'oubliez pas, pour votre TP02, le contenu du
fichier index.html ne devrait pas être celui de
index.php mais plutôt de front-page.php

intégrer les éléments de base de WordPress

1. contenu de base avec les variables de WordPress
(`bloginfo()`, `the_title()`, `the_content()`, etc.)
2. structurer les inclusions (header, footer, sidebar)
ne pas oubliez l'utilisation de `wp_head()` et `wp_footer()` aux endroits appropriés
3. utiliser « la boucle », aussi appelée « the loop »

1. contenu de base WordPress

```
<head>
  <title><?php bloginfo('name'); ?> <?php wp_title(); ?></title>
  <meta charset="<?php bloginfo('charset'); ?>">
  <meta name="description" content="Meta Description">
  <meta name="keywords" content="lab, onepage, creative, html">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Favicon -->
  <link href="<?php bloginfo('template_url'); ?>/img/favicon.ico" rel="shortcut icon"/>

  <!-- Google Fonts -->
  <link href="https://fonts.googleapis.com/css?family=Oswald:300,400,500,700|Roboto:300,400,700" rel="stylesheet">

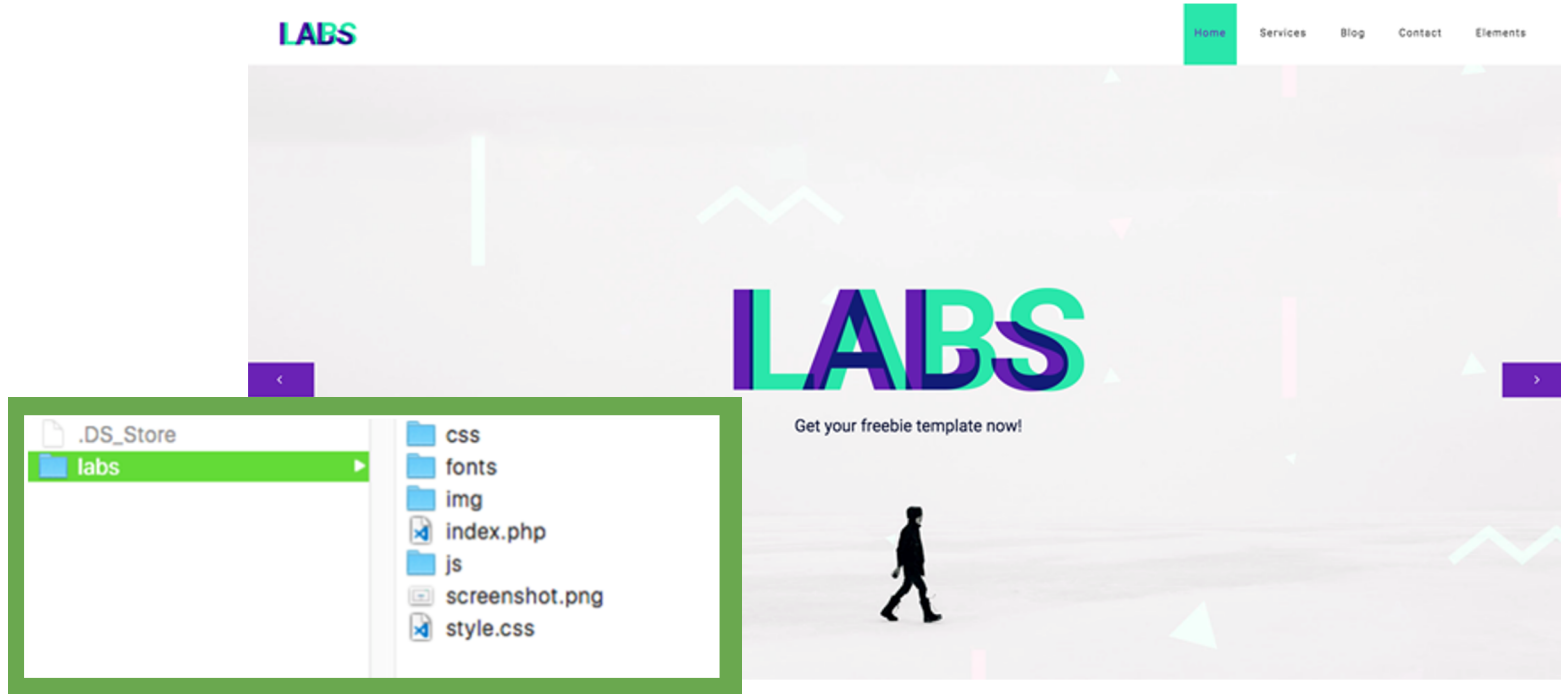
  <!-- Stylesheets -->
  <link rel="stylesheet" href="<?php bloginfo('template_url'); ?>/css/bootstrap.min.css"/>
  <link rel="stylesheet" href="<?php bloginfo('template_url'); ?>/css/font-awesome.min.css"/>
  <link rel="stylesheet" href="<?php bloginfo('template_url'); ?>/css/flaticon.css"/>
  <link rel="stylesheet" href="<?php bloginfo('template_url'); ?>/css/magnific-popup.css"/>
  <link rel="stylesheet" href="<?php bloginfo('template_url'); ?>/css/owl.carousel.css"/>
  <link rel="stylesheet" href="<?php bloginfo('template_url'); ?>/css/style.css"/>

  <!--[if lt IE 9]>
    <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
    <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
  <![endif]>-->

</head>
```

une fois les fonctions de base intégrées, il faut s'assurer d'avoir les bons fichiers dans le dossier de gabarits

1. contenu de base WordPress



2. structurer les inclusions

Établir les zones répétitives de votre gabarit et en faire des fichiers d'inclusion.

- header
- footer
- sidebar
- publicité
- etc.

Celles-ci sont à titre de référence. Elles peuvent changer d'une interface à l'autre, mais **header** et **footer** seront pratiquement toujours là.

2. structurer les inclusions header.php

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <title><?php bloginfo('name') ?> <?php wp_title(); ?></title>
    <meta charset="<?php bloginfo('charset') ?>">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <!-- Google Fonts -->
    <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,400" rel="stylesheet">

    <!-- Stylesheets -->
    <link rel="stylesheet" href="<?php bloginfo('template_url') ?>/css/bootstrap.min.css">
    <link rel="stylesheet" href="<?php bloginfo('template_url') ?>/css/font-awesome.min.css">
    <link rel="stylesheet" href="<?php bloginfo('template_url') ?>/css/flaticon.css">
    <link rel="stylesheet" href="<?php bloginfo('template_url') ?>/css/animate.css">
    <link rel="stylesheet" href="<?php bloginfo('template_url') ?>/css/owl.carousel.css">
    <link rel="stylesheet" href="<?php bloginfo('template_url') ?>/css/styles.css">

</head>
<body>
    <div id="preloader">
        <div class="loader"></div>
    </div>

    <header class="header-section sp-pad">
        <a href="index.html" class="logo">
            
        </a>
        <nav class="main-menu">
            <ul>
                ...
            </ul>
        </nav>
    </header>
```

2. structurer les inclusions

footer.php

```
<footer class="footer-section spad">
```

```
...
```

```
</footer>
```

```
<script src="<?php bloginfo('template_url') ?>/js/jquery-3.2.1.min.js"></script>
```

```
<script src="<?php bloginfo('template_url') ?>/js/bootstrap.min.js"></script>
```

```
<script src="<?php bloginfo('template_url') ?>/js/owl.carousel.min.js"></script>
```

```
<script src="<?php bloginfo('template_url') ?>/js/mixitup.min.js"></script>
```

```
<script src="<?php bloginfo('template_url') ?>/js/circle-progress.min.js"></script>
```

```
<script src="<?php bloginfo('template_url') ?>/js/main.js"></script>
```

```
</body>
```

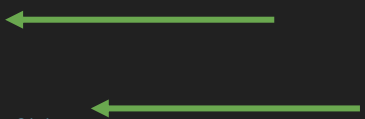
```
</html>
```

2. structurer les inclusions

N'oubliez pas d'intégrer les fonctions importantes de WordPress

header.php


```
<link rel="stylesheet" href="<?php bloginfo('template_url') ?>css/styles.css">
...
<?php wp_head(); ?>
</head>
<body <?php body_class(); ?>>
<div id="preloader">
    <div class="loader"></div>
</div>
```

A diagram with two green arrows. The first arrow points from the right to the line containing the <?php wp_head(); ?> code. The second arrow points from the right to the line containing the <?php body_class(); ?>> code.

footer.php

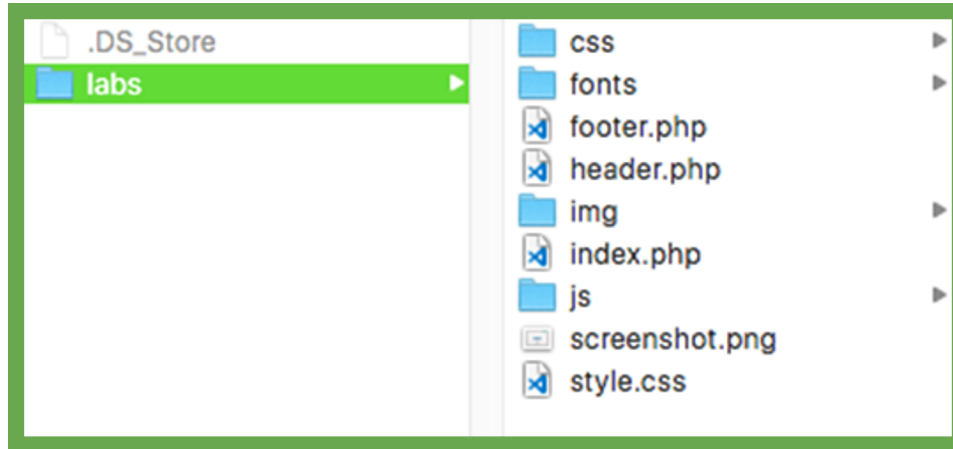
```
<script src="<?php bloginfo('template_url')
?>/js/main.js"></script>

<?php wp_footer() ?>
</body>
</html>
```

A diagram with a single green arrow pointing from the right to the line containing the <?php wp_footer() ?> code.

2. structurer les inclusions

notre thème commence à prendre forme



prochaine étape: on intègre « the loop »

3. « the loop »

identifier le contenu que nous voulons dynamiser

```
<div class="cards">
  <!-- single item -->
  <div class="card_portfolio">
    <a href="project1.html" class="card_link">
      <div class="card_media">
        
      </div>
      <h2>Nom du projet</h2>
      <p>Lorem ipsum sit dolor amet.</p>
      <a href="#lien">En savoir plus</a>
    </a>
  </div>
  <!-- single item -->
  <div class="card_portfolio">
    <a href="project1.html" class="card_link">
      <div class="card_media">
        
      </div>
      <h2>Nom du projet</h2>
      <p>Lorem ipsum sit dolor amet.</p>
      <a href="#lien">En savoir plus</a>
    </a>
  </div>
  <!-- single item -->
  <div class="card_portfolio">
    <a href="project1.html" class="card_link">
      <div class="card_media">
        
      </div>
      <h2>Nom du projet</h2>
      <p>Lorem ipsum sit dolor amet.</p>
      <a href="#lien">En savoir plus</a>
    </a>
  </div>
</div>
```

3. « the loop »

remplacer ce contenu à l'aide de la boucle

```
<div class="cards">
  <?php if (have_posts()) : ?>
    <?php while (have_posts()) : the_post(); ?>

      <!-- single item -->
      <div class="card_portfolio">
        <a href="project1.html" class="card_link">
          <div class="card_media">
            <?php the_post_thumbnail('vignette'); ?>
          </div>
          <h2><?php the_title(); ?></h2>
          <p><?php the_excerpt(); ?></p>
          <a href="<?php the_permalink(); ?>">Suite</a>
        </a>
      </div>

    <?php endwhile; ?>
  <?php endif; ?>
</div>
```

03

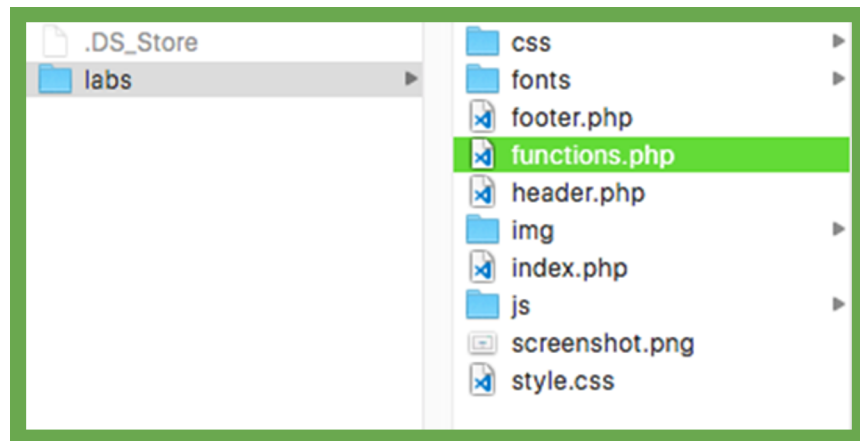
les fonctions de WordPress

**activons les différentes options
requises pour notre thème**

les fonctions de WordPress

1. créer le fichier de base
2. activer nos options de base (image à la une, longueur du résumé)
3. créer nos menus
4. créer nos widgets

1. créer le fichier de base le fichier functions.php



2. activer les options de bases

- image à la une
- taille d'image
- etc.

Assurez-vous d'activer seulement les options qui vous seront utiles. Il ne faut pas polluer notre interface d'administration avec plein de matériel inutile.

```
<?php
```

```
// support image à la une
```

```
add_theme_support('post-thumbnails');
```

```
set_post_thumbnail_size(800, 480, true);
```

```
// ajouter un nouveau format d'image
```

```
add_image_size('vignette', 220, 100, true);
```

```
// etc.
```

3. créer les menus

n'oubliez pas les étapes de création d'un menu

→ prévenir WordPress

functions.php

→ déterminer les emplacements des menus

gabarits PHP

→ créer la structure des menus

interface d'administration

encore une fois, assurez-vous de ne créer que les zones dont vous aurez réellement besoin

3. créer les menus prévenir WordPress dans functions.php

```
// enregistrer les zones de menu  
function wp_enregistrer_menu() {  
    register_nav_menus(array(  
        'menu_principal' => 'Menu principal',  
        'menu_footer' => 'Menu footer',  
    ));  
}  
add_action('init', 'wp_enregistrer_menu');
```


3. créer les menus déterminer les emplacements des menus dans les gabarits

```
<header>
  <nav class="main-menu">
    <ul>
      <li><a href="index.html">Home</a></li>
      <li><a href="projets.html">Portfolio</a></li>
      <li><a href="blog.html">Blog</a></li>
      <li><a href="contact.html">Contact</a></li>
    </ul>
  </nav>
</header>
```



```
<header>
  <?php wp_nav_menu(array(
    'theme_location' => 'menu_principale',
    'container' => 'nav',
  )); ?>
</header>
```

3. créer les menus

créer la structure des menus

Nom du menu

Menu 1

Enregistrer le menu

Structure du menu

Glissez chaque élément pour les placer dans l'ordre que vous préférez. Cliquez sur la flèche à droite de l'élément pour afficher d'autres options de configuration.

Page 1

Page ▼

Page 2

Page ▼

Page 3

Page ▼

Page 4

Page ▼

Catégorie 1

Catégorie ▼

Catégorie 2

Catégorie ▼

Réglages du menu

Ajoutez automatiquement des pages

☐ Ajouter automatiquement les pages de premier niveau à ce menu

Afficher l'emplacement

☒ Menu principal

☐ Menu réseaux sociaux

Supprimer le menu

Enregistrer le menu

4. créer les widgets

n'oubliez pas les étapes de création d'un ~~menu~~ widget

→ prévenir WordPress

functions.php

→ déterminer les emplacements des ~~menus~~ widgets

gabarits PHP

→ ajouter ou créer les widgets

interface d'administration

les étapes pour ajouter de nouveaux widgets sont identiques aux menus, à l'exception d'une syntaxe légèrement différente

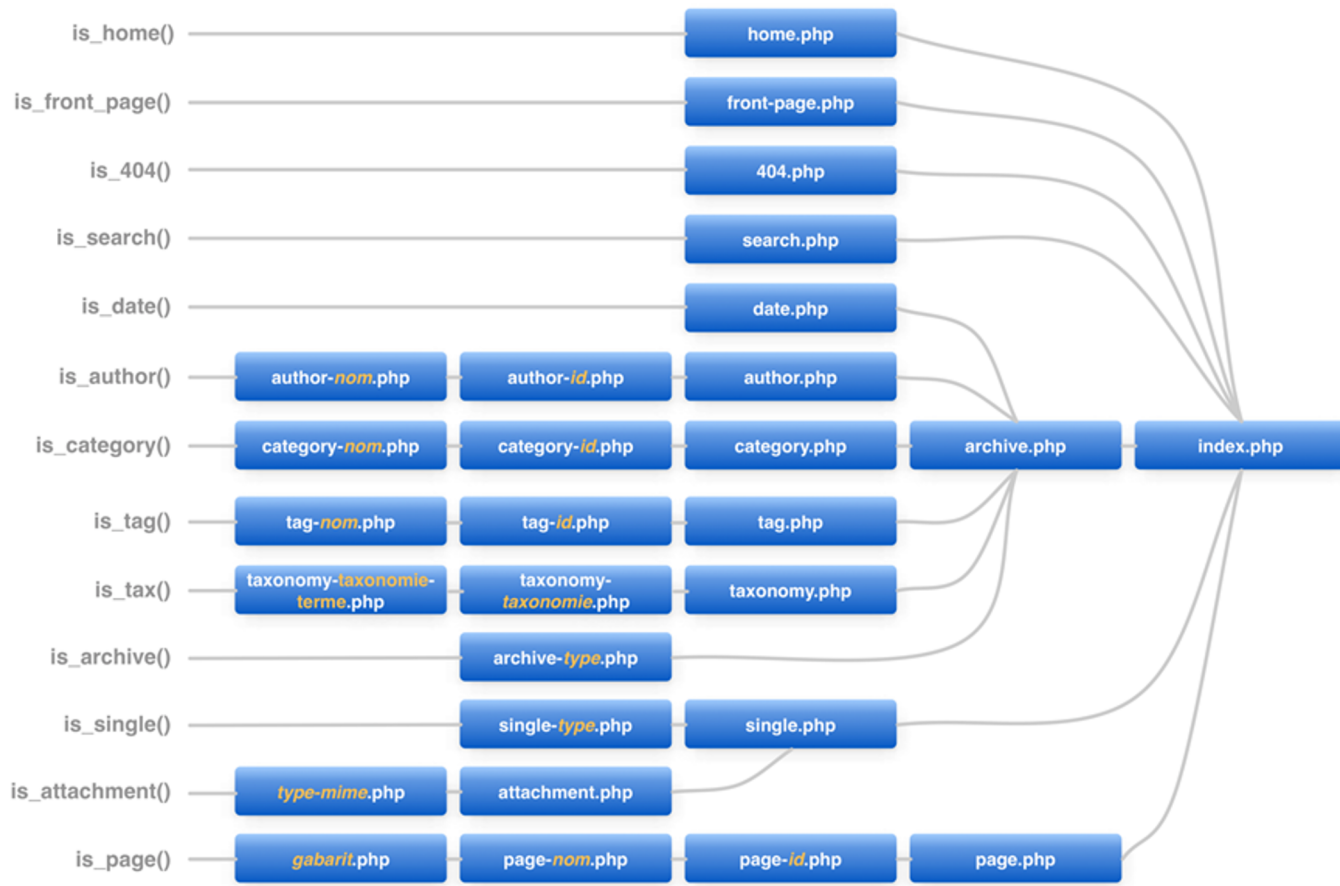
comme pour les menus, ne créez que ce dont vous avez réellement besoin

04

les gabarits

**notre première page est pas mal...
il faut maintenant faire les autres!**

articles



pages

les gabarits

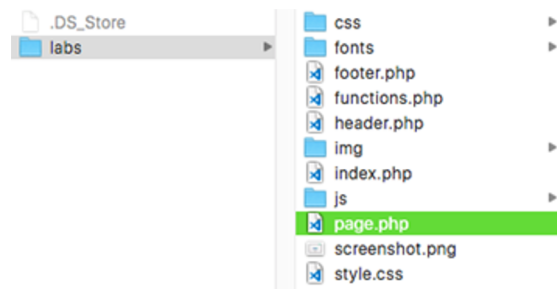
Il faut maintenant faire le bon choix des gabarits. Tel que vu dans nos cours et pratiqué avec l'exercice hiérarchie, faites le bon choix en fonction de votre projet.

- créer le fichier .php de votre gabarit
- copier le contenu de votre index.php dans ce nouveau gabarit
- utilisez le fichier .html correspondant que vous aurez créé et utilisez-le pour apporter les modifications à ce nouveau gabarit
- changer ou créer des boucles afin de remplacer tout votre contenu statique par celui provenant de votre base de données

les gabarits

exemple: page.php

une fois le contenu de votre index.php copié dans page.php, identifiez le contenu à modifier avec votre gabarit .html que vous aurez préalablement créé et apportez-le dans page.php afin d'y inclure vos boucles pour afficher le bon contenu



```
<header class="header-section">
    ...
</header>

<main>
    <!-- hero début -->
    <section class="hero">
        <h2>Lorem ipsum</h2>
    </section>
    <!-- hero 2 fin -->

    <!-- google map -->
    <section class="map">
        <h2>Nous trouver</h2>
    </section>
    <!-- google map fin -->

    <section class="contact">
        <h2>Nous contacter</h2>
    </section>
</main>

<footer>
    ...
</footer>
```

05

les *custom post type*

il faut maintenant créer le contenu
propre à notre projet

intégrer les éléments de base de WordPress

Suivre les étapes apprises lors du cours 07 afin de créer vos propres types de contenu personnalisé

1. créer votre nouveau type à l'aide de generatewp.com
2. ajouter le code généré à votre fichier `functions.php`
3. intégrer votre contenu à ce nouveau type à l'aide de l'interface administrateur



les custom post type

deux options s'offrent à vous

1. créer la boucle avec le bon `post_type` afin de l'afficher dans un de vos gabarits

```
<?php
    query_posts(array(
        'post_type' => 'nom-du-type-de-contenu',
        'post_status' => 'publish',
        'showposts' => 10
    ));
?>
<?php while (have_posts()) : the_post(); ?>
    <h2><?php the_title(); ?></h2>
    <?php the_content(); ?>
<?php endwhile ?>
```

2. créer un gabarit spécifique pour ce type de contenu
`single-nom-de-votre-type.php`

on a fait le tour!

Vous avez maintenant créé votre propre thème avec tous les gabarits nécessaires. C'est le temps de téléverser vos fichiers sur votre serveur et de tester le tout.

Vous avez maintenant tous les outils en main afin de réaliser le TP.

cette semaine

Exercice + TP02