votre propre thème - *part* 2

menus et autres paramètres

PRÉPARÉ PAR:
Bernard Imbert
Jean-François Leblanc
Adapté par Mathieu Dupont
Pour le cours 420-964-EM

Semaine 6

comment déboguer WordPress?

erreur critique!

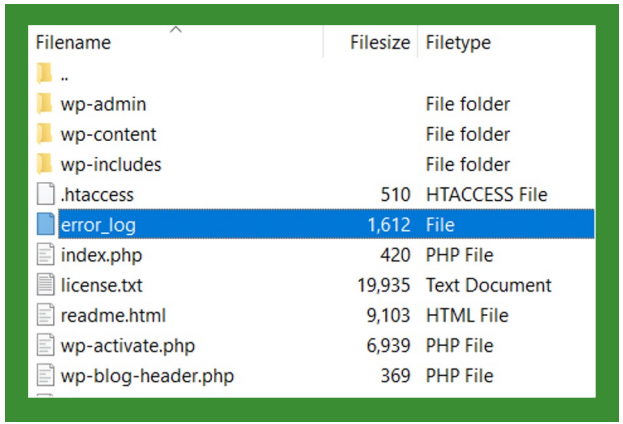
Une erreur critique est survenue sur votre site.

En apprendre plus sur le débogage de WordPress.

on fait quoi?

- → d’abord, on pleure un petit peu 😢
- → ensuite, on se rappelle que WordPress laisse des traces

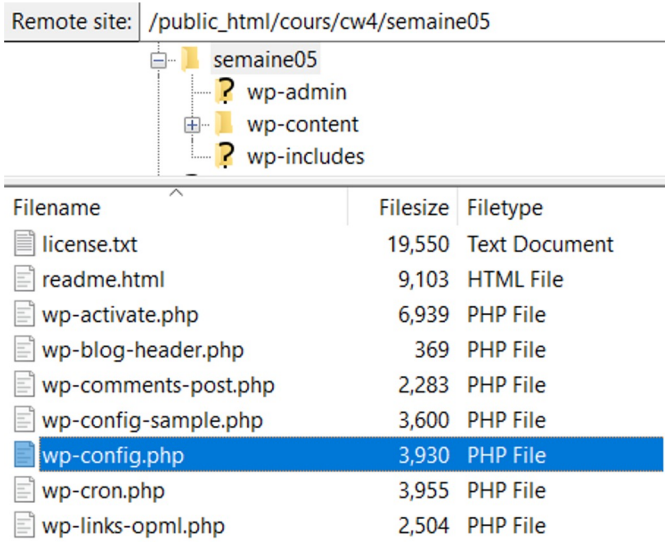
Toutes les erreurs sont stockées
dans un fichier nommé **debug.log**
à la racine de l’installation.



debug.log?

- c’est pas très pratique à consulter
- ça peut contenir des milliers de lignes qui sont là depuis longtemps

WordPress peut nous afficher *live*, dans le navigateur, les erreurs. Il suffit d’activer le mode debug en définissant les propriétés WP_DEBUG ET WP_DEBUG_LOG à true dans le fichier **wp-config.php** à la racine de votre installation.




```
// define('WP_DEBUG', false);

define('WP_DEBUG', true);
define('WP_DEBUG_LOG', true);
```

pendant qu’on développe le site

on ajoute

```
// define('WP_DEBUG', false);

define('WP_DEBUG', true);
```

quand le site va en ligne

on remet

```
define('WP_DEBUG', false);

// define('WP_DEBUG', true);
```

ce n’est pas parfait, mais ça aide!

→ **on voit les erreurs**

Parse error: syntax error, unexpected end of file, expecting elseif (T_ELSEIF) or else (T_ELSE) or endif (T_ENDIF) in **/home/jfleblanc/public_html/cours/cw4/semaine05/wp-content/themes/mtl_lumiere/index.php** on line **34**

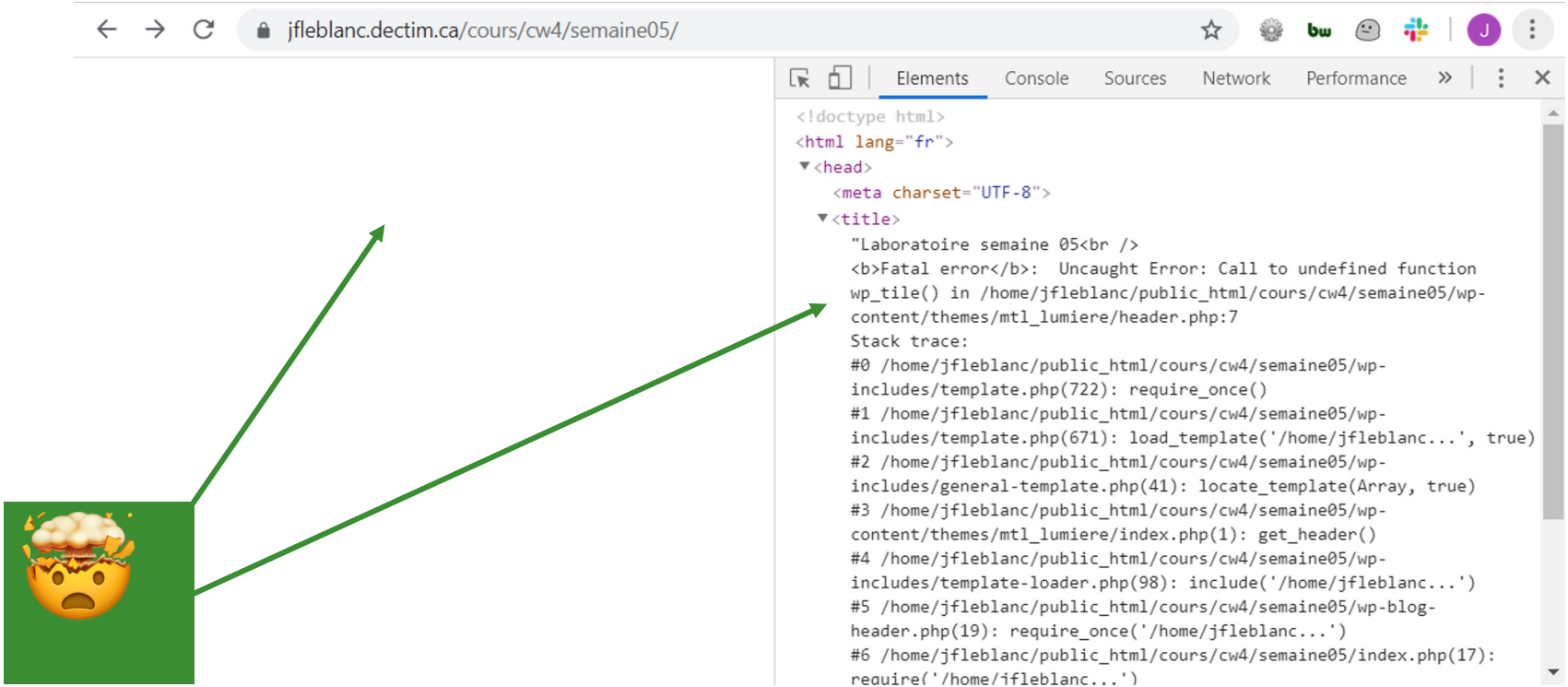
→ **on peut corriger**

```
<div id="principal">
    <?php if( have_posts()) : ?>
        <?php while (have_posts()) : the_post(); ?>
            <article <?php post_class(); ?> id="article<?php the_ID(); ?>">
                ...
            </article>
        <?php endwhile; ?>

</div>
```

endif manquant

parfois, on doit chercher un peu



remarque #1

- une seule erreur est affichée à la fois
- dès que le PHP frappe une erreur, il arrête tout

PHP est un préprocesseur, c'est à dire qu'il va interpréter le code avant de le rendre, comme Sass. Dès qu'il frappe une erreur, il arrête son interprétation et donc, n'affiche plus rien par la suite.

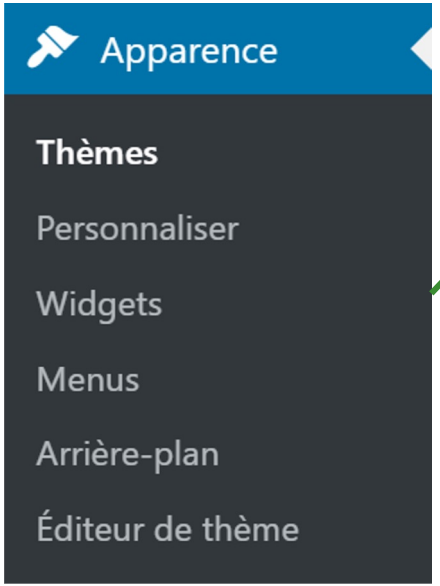
remarque #2

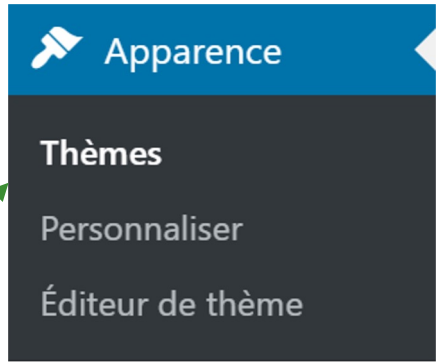
- quand vous avez une page blanche, vérifiez si l'admin est toujours accessible
- si oui, changez de thème
- si vous avez encore un pépin avec le nouveau thème, désactivez les extensions une à une

les fonctions WordPress

elles sont rendues où, mes
images mises en avant?

certaines options ne sont affichées que lorsque les thèmes les supportent





Pour que les menus soient disponibles, le thème doit déclarer qu'il les supporte.

C'est possible en utilisant le fichier **functions.php**.

comprendre le fichier functions.php

S’il est présent à la racine du thème, ce fichier PHP sera automatiquement chargé et exécuté à chaque requête.

Il permet :

- → de spécifier les fonctionnalités que supportera notre thème (images mises en avant, menus, etc.)
- → d’ajouter des filtres et des actions modifiant dynamiquement le rendu des pages

mise en garde

afin d'éviter un possible crash du PHP, il est fortement recommandé de **ne pas mettre la balise de fermeture dans le fichier functions.php**

```php
<?php
    // insérez le code PHP ici
    // Ne mettez pas la balise de fermeture
```

en fait, on ne devrait jamais mettre la balise de fermeture **?>** dans les fichiers qui ne contiennent que du PHP

fonctionnalités supportées

la fonction add_theme_support() permet de spécifier quelles fonctionnalités sont supportées par votre thème

- pour autoriser l'utilisation des **images mises en avant** et déterminer leur taille par défaut :

```
add_theme_support('post-thumbnails');
set_post_thumbnail_size(800, 480);
```

- pour que WordPress crée des copies des images à des tailles données :

```
add_image_size('vignette', 220, 180);
add_image_size('xxl', 2200, 1800);
```

functions.php

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

```php
add_image_size('vignette', 220, 180);
```

[gabarit].php

taille prédéfinie

```php
the_post_thumbnail('medium');
```

taille par défaut

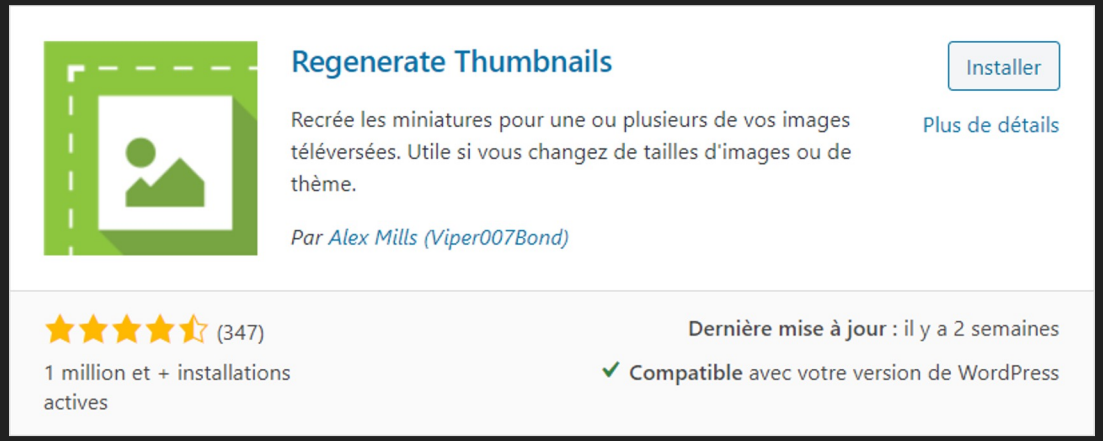
```php
the_post_thumbnail();
```

taille personnalisée

```php
the_post_thumbnail('vignette');
```

<extension-regenerate-thumbnails>

attention, ce n’est pas du vrai code

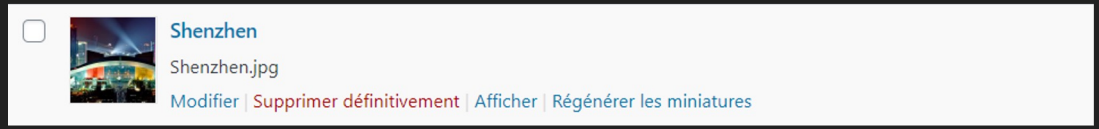


Lorsque vous modifiez les tailles de vos images. Cette petite extension est très intéressante pour vous assurer que WordPress régénère les images au bon format.

regenerate thumbnails

- toutes d'un coup

- une à la fois



comme toute bonne chose à une fin

Une fois terminé, vous pouvez désinstaller l'extension **Regenerate Thumbnails**.

À l'avenir, toutes les images que vous téléversez seront dans les bonnes dimensions que vous aurez spécifiées.

Bonnes pratiques
Toujours effacer les extensions et les images qui ne servent plus. Vos sauvegardes en seront plus légères et vous aurez plus d'espace libre sur votre serveur.

</extension-regenerate-thumbnails>

on revient aux choses sérieuses

retour à functions.php

Il est aussi possible de spécifier le nombre maximum de mots pour l'affichage du texte résumé (the_excerpt).

dans **functions.php** :

```php
function new_excerpt_length($length) {
    return 20; // Nombre maximum de mots
}
add_filter('excerpt_length', 'new_excerpt_length');
```

ajouter un menu

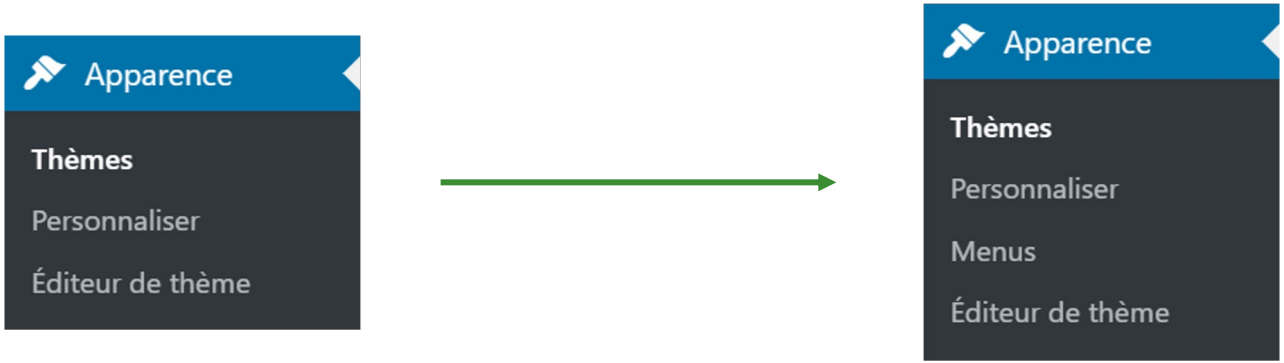
c’est quoi le plat du jour?

enregistrer des emplacements pour les menus

- Lorsque vous utilisez l'interface WordPress pour ajouter un menu , vous indiquez au logiciel l'endroit où le menu devrait s'afficher (ex.: sidebar, footer_1, menu_principal, etc.).
- Pour pouvoir permettre cela dans votre thème, vous devez enregistrer des emplacements prévus à cet effet et ajouter le code dans votre thème aux endroits désirés afin d'afficher le contenu de ces nouveaux emplacements.

enregistrer des emplacements pour les menus

Par défaut, WordPress n’active pas de menu dans un thème personnalisé. La première étape consiste donc à activer les options de menu de WordPress dans functions.php



1. l'indiquer à WordPress

→ enregistrer **un** emplacement de menu :

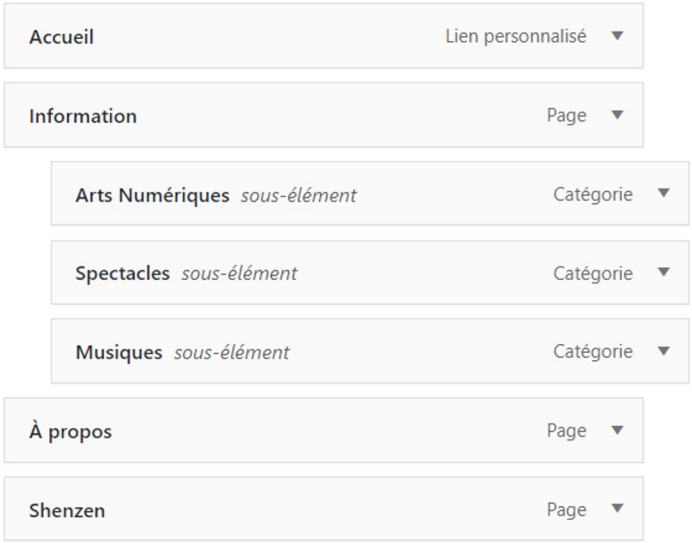
```php
register_nav_menu('menu_principal', 'Menu principal');
```

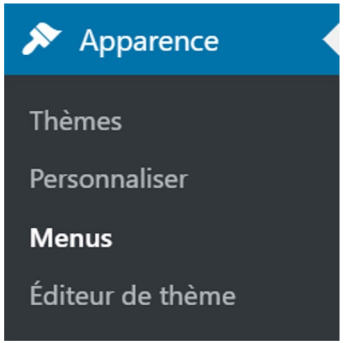
→ enregistrer **plusieurs** emplacements de menus :

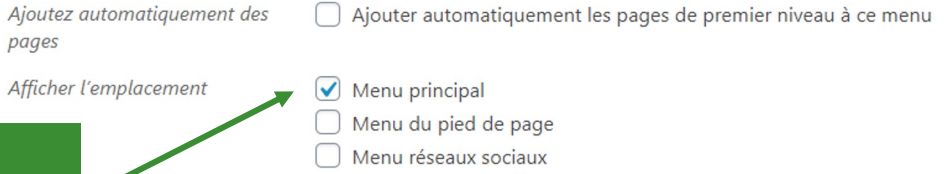
```php
register_nav_menus(array(
    'menu_principal' => 'Menu principal',
    'menu_footer' => 'Menu du pied de page',
    'menu_sociaux' => 'Menu réseaux sociaux',
));
```

2. création et assignation du menu

il faut ensuite aller créer notre menu dans l'interface d'administration







n'oubliez pas d'assigner votre menu à un emplacement

3. sélection d’un emplacement

Le menu est maintenant disponible, mais rien ne s’affiche dans notre page.

Ne vous inquiétez pas, c’est normal! Nous devons maintenant lui dire où nous voulons que le menu s’affiche. Nous appelons ça une zone d’emplacement.

Pour déterminer une zone dans votre thème, vous devrez mettre le code suivant à l’endroit désiré dans votre gabarit :

```php
<?php wp_nav_menu(array(
    'theme_location' => 'menu_principal')
); ?>
```

🚀Houston! on a un menu

la structure par défaut du menu affiché sera :

```
<div class="menu-nom-menu-container">
    <ul id="menu-nom-menu" class="menu">
        <li id="menu-item-id-item"class="menu-item">
            <a href="lien">Texte</a>
        </li>
    </ul>
</div>
```

codex: référence wp_nav_menu()

Voici les différentes classes (CSS) qui sont générées par WordPress :

- **.menu-nom-menu-container**: sur le conteneur du menu (ou <nav>)
- **.menu**: sur le <ul> contenant le menu
- **.sub-menu**: sur le <ul> contenant les pages enfants
- **.menu-item**: sur chaque <li> du menu
- **.current-menu-item**: sur le <li> de la page courante
- **.current-menu-parent**: sur le <li> parent de la page courante (lorsque sous-nav)
- **.current-menu-ancestor**: sur les <li> ancêtres de la page courante (lorsque sous-sous-nav)
- **.menu-item-home**: sur les <li> de la page d'accueil

codex: référence wp_nav_menu()

Lors de l’intégration de code HTML/CSS dans un thème, prenez tout de suite en considération le nom des classes générées par WordPress.

Vous sauverez énormément de temps.

De toute façon, c’est une belle nomenclature de classes de menu, pourquoi ne pas toujours l’utiliser?


la fin