

**WordPress**

# **votre propre thème**

**on est rendu là**

Semaine 5

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

# un thème personnalisé?

**flexibilité absolue, puisqu'on le code soi-même**

FURROW

=

# DIG DEEP

Site of the Day

[furrow.studio](http://furrow.studio)



# ce projets services

02

< • >

03

14  
15  
16



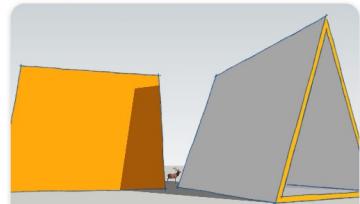


We are a house  
of creativity



# The A-frame Project

**The A-frame Project** est le projet d'une petite famille de 4 qui désire vivre en harmonie avec la nature et construire son chalet au bord de l'eau. Ce site a comme objectif de montrer l'évolution du projet tout au long du processus.



**3 Décembre 2022**



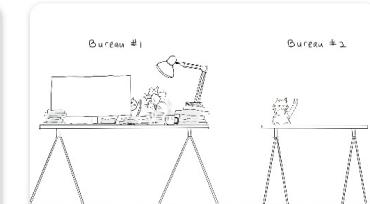
**16 octobre 2022**



**5 septembre 2022**



**21 août 2022**



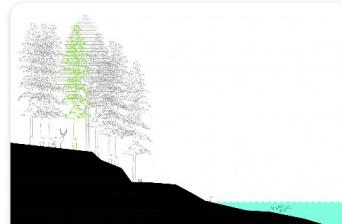
**29 juillet 2022**



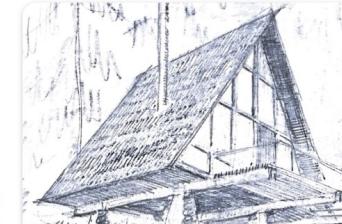
**20 juillet 2022**



**16 juillet 2022**



**9 juillet 2022**



**30 juin 2022**



**27 juin 2022**



**tout est  
possible  
avec  
WordPress**

n'importe quel type de site peut fonctionner avec le moteur de WordPress  
il suffit d'être original

---

**AWWWARDS®**

[sites utilisant WordPress](#)



[vitrine officielle de WordPress](#)

# **les bases**

parce qu'il faut bien commencer  
quelque part

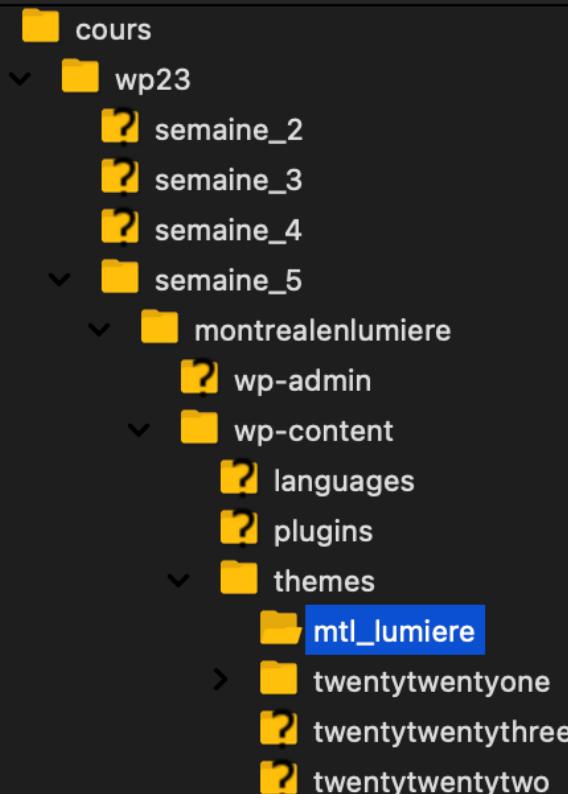
## comment on l'active?

- lorsqu'on crée un thème personnalisé, on ne peut pas l'installer directement par l'interface administrateur de WordPress
- il faut le téléverser au bon endroit à l'aide d'un client FTP (FileZilla)
- vous pourrez ensuite activer votre thème à partir de l'interface administrateur

# pour créer un nouveau thème

il suffit de créer un sous-dossier dans **wp-content/themes/** et d'y placer les 2 fichiers magiques :

- **index.php**
- **style.css**



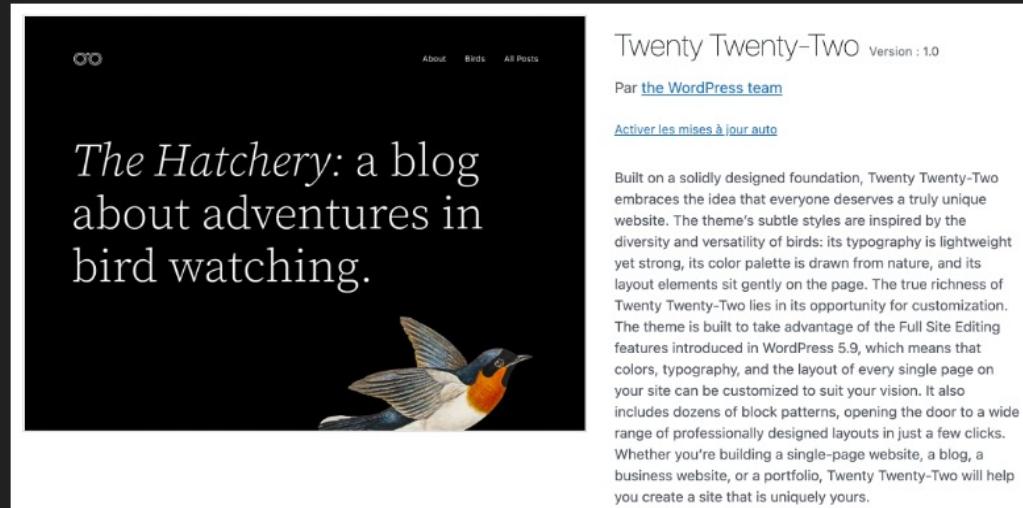
Nom de fichier ▲

- ..
- index.php
- style.css

# capture d'écran de votre thème

bien qu'optionnelle, il est fortement recommandé d'en produire une :

- index.php
- style.css
- **screenshot.png**



# style.css

Afin que votre thème s'affiche dans Apparence > Thèmes, vous devez inscrire les informations suivantes dans l'en-tête du fichier **style.css** tout en haut :

Seul le **Theme Name** est obligatoire.

Ce fichier ne devrait contenir que ces informations (**pas de CSS**).

```
/*
Theme Name: Twenty Twenty-Two
Author: the WordPress team
Author URI: http://wordpress.org/
Description: Built on a solidly designed...
Version: 1.0
Text Domain: twentytwentytwo
Tags: one-column, custom-colors, ...
*/
```



Version

Theme Name



About Birds All Posts

*The Hatchery: a blog  
about adventures in  
bird watching.*

Description



Tags

Author

Twenty Twenty-Two Version : 1.0

Par [the WordPress team](#)

[Activer les mises à jour auto](#)

Built on a solidly designed foundation, Twenty Twenty-Two embraces the idea that everyone deserves a truly unique website. The theme's subtle styles are inspired by the diversity and versatility of birds: its typography is lightweight yet strong, its color palette is drawn from nature, and its layout elements sit gently on the page. The true richness of Twenty Twenty-Two lies in its opportunity for customization. The theme is built to take advantage of the Full Site Editing features introduced in WordPress 5.9, which means that colors, typography, and the layout of every single page on your site can be customized to suit your vision. It also includes dozens of block patterns, opening the door to a wide range of professionally designed layouts in just a few clicks. Whether you're building a single-page website, a blog, a business website, or a portfolio, Twenty Twenty-Two will help you create a site that is uniquely yours.

Étiquettes : Une colonne, Couleurs personnalisées, custom-menu, Logo personnalisé, Éditeur de style, Images mises en avant, Modification complète du

Activer

Supprimer

# le code PHP comment ça marche?

# **les balises d'ouverture et de fermeture**



**<?php**



**?>**

**le code PHP est  
imbriqué à travers le  
code HTML**

**toujours entre une  
balise d'ouverture  
`<?php` et une balise de  
fermeture `?>`**

```
<?php  
$nb_pommes = 3;  
$nb_oranges = 2;  
$nb_fruits = $nb_pommes + $nb_oranges;  
?>
```

```
<!DOCTYPE html>  
<html>  
<body>  
  
<h1>J'ai <?php echo $nb_fruits; ?> fruits!</h1>  
  
</body>  
</html>
```

résultat

**J'ai 5 fruits!**

# Syntaxe alternative de PHP

Avant d'aller plus loin, vous devez comprendre la syntaxe alternative de PHP.

Ces 4 bouts de code donnent le même résultat :

## PHP

```
<?php  
if (true) :  
    echo "<p>Test</p>";  
endif;  
?>
```

```
<?php  
if (true) {  
    echo "<p>Test</p>";  
}  
?>
```

## PHP + HTML

```
<?php if (true) : ?>  
<p>Test</p>  
<?php endif; ?>
```

```
<?php if (true) { ?>  
<p>Test</p>  
<?php } ?>
```

# Syntaxe alternative de PHP

Avant d'aller plus loin, vous devez comprendre la syntaxe alternative de PHP.

Ces 4 bouts de code donnent le même résultat :

## PHP

```
<?php  
    if (true) :  
        echo "<p>Test</p>";  
    endif;  
?>
```

code  
PHP

```
<?php  
    if (true) {  
        echo "<p>Test</p>";  
    }  
?>
```

## PHP + HTML

```
<?php if (true) : ?>  
<p>Test</p>
```

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

```
<?php if (true) { ?>  
<p>Test</p>
```

```
<?php } ?>
```

# Syntaxe alternative de PHP

## PHP

```
<?php  
    if (true) :  
        echo "<p>Test</p>";  
    endif;  
?  
  
<?php  
    if (true) {  
        echo "<p>Test</p>";  
    }  
?  
?
```

2 différentes syntaxes des blocs

Avant d'aller plus loin, vous devez comprendre la syntaxe alternative de PHP.

Ces 4 bouts de code donnent le même résultat :

## PHP + HTML

```
<?php if (true) : ?>  
    <p>Test</p>  
<?php endif; ?>
```

```
<?php if (true) { ?>  
    <p>Test</p>  
<?php } ?>
```

# **les fonctions PHP**

# Vous avez créé votre premier thème.

Cependant, vous constatez qu'aucun contenu ne s'affiche. Que devrait contenir le fichier index.php?

- le fichier index.php contient un mélange de code PHP et de code HTML. C'est de cette façon que sont construits les gabarits WordPress
- les fonctions PHP propres à WordPress seront donc utilisées pour afficher le contenu entré dans le système et le HTML pour la structure de la page
- il faut notamment utiliser ces fonctions pour indiquer correctement où trouver la feuille de style, les images, etc.

## la fonction bloginfo()

- bloginfo() est une fonction PHP spécifique à WordPress.
- La fonction bloginfo() permet d'obtenir du contenu dynamiquement grâce à des paramètres.

# principaux paramètres de bloginfo()

- bloginfo ( '**charset**' ) ;  
jeu de caractère
- bloginfo ( '**name**' ) ;  
titre du **site**
- **wp\_title** () ;  
titre de la page
- bloginfo ( '**template\_url**' ) ;  
lien vers le dossier du thème actif
- bloginfo ( '**url**' ) ;  
lien absolu du site

[référence sur les paramètres de bloginfo\(\)](#)



## Réglages généraux

Titre du site Web

Montréal en lumière

```
<head>

<meta charset="<?php bloginfo('charset'); ?>">

<title><?php bloginfo('name') ?> <?php wp_title(); ?></title>

<link rel="stylesheet" href="<?php bloginfo('template_url'); ?>/style/style.css"/>

</head>
```

façon optimale d'inclure  
notre feuille de style

# ajouter des images à notre thème?

Il faut d'abord faire la distinction entre images dynamiques et statiques :

- **images dynamiques** : images qui peuvent être ajoutées par les éditeurs du site Web via l'interface administrateur
- **images statiques** : images qui sont ajoutées par le développeur du site et qui ne peuvent être modifiées que par lui (non accessibles via le CMS)

# ajouter des images statiques

- placez un dossier **images** dans le dossier de votre thème
- à l'aide de la fonction **bloginfo()**, récupérez le chemin jusqu'au dossier de votre thème et ajoutez-le à l'attribut **src** de l'image :

```
  
  
<!-- Résultat dans le navigateur : --&gt;<br/><img src='https://usager.dectim.net/cours/wp/semaine_5/montrealenlumiere/wp-content/themes/ml_lumiere/images/logo.jpg'>
```

- Ceci s'applique seulement aux images statiques. Nous verrons plus tard le fonctionnement pour les images dynamiques.

# récupération du contenu

on l'a entré dans le CMS,  
faudrait bien qu'ça serve!

# comment récupérer et intégrer les contenus de WordPress?

- il faut utiliser une séquence de lignes de code appelée « **la boucle** » ou « ***the loop*** »
- par défaut, la boucle affiche les 10 derniers articles. Cela peut-être modifié dans la section **Réglages > lecture**

# « the loop »

```
<!-- s'il y a au moins un article à afficher -->
<?php if ( have_posts() ) : ?>

    <!-- tant qu'il reste un article à afficher, on y avance -> the_post(); -->
    <?php while ( have_posts() ) : the_post(); ?>

        <!-- C'est ici le cœur de la boucle, où on affiche du contenu! -->

    <?php endwhile; ?> <!-- fin du while -->

<?php else : ?> <!-- s'il n'y a aucun article à afficher -->

    <p>Aucun article disponible!</p>

<?php endif; ?> <!-- fin du if -->
```

# plusieurs fonctions utiles dans « la boucle »

- `the_title();`  
**titre**
- `the_ID();`  
**identifiant unique**
- `the_excerpt();`  
**résumé**
- `the_content();`  
**contenu**
- `the_post_thumbnail();`  
**image à la une**
- `post_class();`  
**classes de l'article (ou du post)**
- `the_category();`  
**catégorie(s)**
- `the_tags();`  
**mot(s)-clef(s)**
- `the_date() ou the_time();`  
**date de publication**
- `the_permalink();`  
**permalien**



# intégrer le contenu de vos articles (posts)

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

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

<article <?php post_class(); ?> id="article<?php the_ID(); ?>">
    <h3><?php the_title(); ?></h3>
    <?php the_post_thumbnail('medium'); ?>
    <?php the_excerpt(); ?>
    <a class="more-link" href="<?php the_permalink(); ?>">Suite...</a>
</article>

<?php endwhile; ?>

<?php endif; ?>
```

# résultat

```
<article class="post-23 post type-post status-publish format-standard hentry category-arts-numeriques category-spectacles" id="article23">
    <h3>Le design de Shenzhen en 24 affiches</h3>
    
    <a class="more-link" href="http://mdupont.dectim.net/cours/wp23/semaine_5/montrealenlumiere/le-design-de-shenzhen-en-24-affiches/">Suite...</a>
</article>

<article class="post-26 post type-post status-publish format-standard hentry category-arts-numeriques category-spectacles" id="article26">
    <h3>Intensional Particle</h3>
    
    <a class="more-link" href="http://mdupont.dectim.net/cours/wp23/semaine_5/montrealenlumiere/intensional-particle/">Suite...</a>
</article>

<article class="post-28 post type-post status-publish format-standard hentry category-arts-numeriques category-spectacles" id="article28">
    <h3>Résonances boréales</h3>
    
    <a class="more-link" href="http://mdupont.dectim.net/cours/wp23/semaine_5/montrealenlumiere/resonances-boreales/">Suite...</a>
</article>

<article class="post-41 post type-post status-publish format-standard hentry category-musiques category-spectacles" id="article41">
    <h3>Betty Bonifassi</h3>
    
    <a class="more-link" href="http://mdupont.dectim.net/cours/wp23/semaine_5/montrealenlumiere/betty-bonifassi/">Suite...</a>
</article>

<article class="post-38 post type-post status-publish format-standard hentry category-musiques category-spectacles" id="article38">
    <h3>Radio Radio</h3>
    
    <a class="more-link" href="http://mdupont.dectim.net/cours/wp23/semaine_5/montrealenlumiere/radio-radio/">Suite...</a>
</article>

<article class="post-35 post type-post status-publish format-standard hentry category-musiques category-spectacles" id="article35">
    <h3>Mouse on the Keys</h3>
    
    <a class="more-link" href="http://mdupont.dectim.net/cours/wp23/semaine_5/montrealenlumiere/mouse-on-the-keys/">Suite...</a>
</article>
```

# **les gabarits WordPress**

**Les gabarits servent à présenter le contenu d'une page différemment selon le type de contenu (article, page, etc.).**

# c'est quoi un gabarit WordPress?

Le contenu de notre site Web risque d'avoir des mises en page différentes d'une section à l'autre. Les gabarits sont là pour ça.

Ainsi, une **page** pourra être présentée différemment d'un **article** par exemple.

Pour chaque type de contenu, WordPress tentera par défaut de trouver le gabarit le mieux adapté selon un arbre de décisions.



For oEmbeds: embed-{post-type}-{post\_format}.php → embed-{post-type}.php → embed.php → wp-includes/theme-compat/embed.php

■ Primary Template ■ Secondary Template ■ Variable Template ■ Page Type

WordPress vérifie l'existence des gabarits de gauche à droite dans le tableau de la page précédente.

Si un gabarit n'existe pas dans le dossier de thème, il passe au suivant et ainsi de suite.

Vous n'êtes pas obligé de tous les créer car il y aura toujours le fichier **index.php** qui est utilisé par défaut au bout de la chaîne. Ce dernier est donc obligatoire.

# les principaux gabarits

TYPE DE PAGE / SECTION	URL (PAR DÉFAUT)	GABARIT UTILISÉ
accueil	/	home.php
archives (année, mois ou jour)	/2020 /2020/01 /2020/01/22	date.php
catégorie	/category/geek	category.php
mot-clé	/tag/iphone	tag.php
résultat de recherche	?s=cadeau	search.php
post	/2011/12/25/le-dec-tim	single.php
page	/a-propos	page.php
404	/page-qui-nexiste-pas	404.php

## index.php

- **gabarit par défaut**
- utilisé lorsqu'aucun gabarit plus spécifique n'existe dans le thème pour la page à afficher

## single.php

- gabarit pour les **articles**

## page.php

- pour une mise en page différente des **pages** versus des articles

# **front-page.php vs home.php**

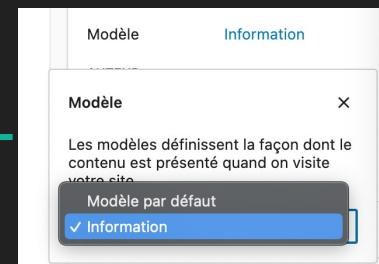
- `front-page.php` : s'il existe, il est **assigné à la page d'accueil**, qu'il s'agisse d'une page statique ou de la liste des derniers articles
- `home.php` : s'il existe, il est **assigné à la page affichant la liste des derniers articles**, que ce soit la page d'accueil ou non (sauf si c'est la page d'accueil **ET** que `front-page.php` existe)

# template-[nom].php

Un gabarit spécifique à une seule page (ou plusieurs)

- On peut créer un gabarit de page pouvant être utilisé par une ou plusieurs pages spécifiques (sélection manuelle).
- Pour se faire, il suffit de créer un fichier PHP dans le thème et d'y mettre le commentaire suivant au début :

```
<?php  
/* Template Name: Information */  
?>
```



Le nom du fichier doit utiliser la nomenclature suivante : **template-[nom].php**



# template-[nom].php

The screenshot shows a WordPress page editor interface. At the top, there are tabs: "Basculer en brouillon", "Aperçu", "Mettre à jour" (highlighted in blue), a gear icon, and a more options icon. Below the tabs, there's a table with two columns: "Page" and "Bloc". The "Page" column contains sections for "Résumé", "Visibilité", "Publier", and "Adresse web". The "Bloc" column contains "Modèle" and "Information". A large teal arrow points from the bottom right towards the "Modèle" section. A modal window titled "Modèle" is open, explaining that models define how content is presented. It lists "Modèle par défaut" and "✓ Information" (which is highlighted with a blue background).

Page	Bloc
Résumé	
Visibilité	Public
Publier	18 février 2018 21 h 14
Adresse web	<a href="http://mdupont.dectim.net/cours/wp23/semaine_5/montrealenlumiere/information/">mdupont.dectim.net /cours/wp23/semaine_5/montrealenlumiere/information/</a>
Modèle	Information

Modèle

Les modèles définissent la façon dont le contenu est présenté quand on visite votre site.

Modèle par défaut

✓ Information

# **les fichiers en inclusion**

copier-coller on aime ça, mais...

- Certaines parties du code étant généralement les mêmes à travers les différentes pages d'un site, il est préférable de regrouper ces parties dans des fichiers séparés que l'on va inclure dans le fichier principal d'affichage des contenus.
- Cela évitera d'avoir à faire la même modification dans plusieurs fichiers et limitera les possibilités d'erreurs ou d'oubli.
- Lorsqu'on utilise une fonction d'inclusion, c'est comme si on indiquait de copier le code du fichier à inclure à l'endroit où la fonction est appelée dans le fichier principal.

# une structure réutilisable et modulaire

header.php

index.php

sidebar.php

footer.php

# les fichiers en inclusion

- en-tête : header.php
- pied de page : footer.php
- zone secondaire : sidebar.php
- etc.

# appeler les fichiers en inclusion

```
→ get_header();  
    header.php
```

```
→ get_footer();  
    footer.php
```

```
→ get_sidebar();  
    sidebar.php
```

```
→ get_template_part('exemple');  
    exemple.php
```



utilisera le fichier  
exemple.php à la base  
de notre thème

# le fichier header.php

- Ce fichier contiendra le contenu qui débutera chaque page.
- Cette page doit contenir au minimum les en-têtes html nécessaires jusqu'à l'ouverture du <body>.
- Remarquez l'ajout des fonctions `wp_head()` et `body_class()`.
- `wp_head()` est un hook qui permettra à WordPress de venir incruster du code pour ses propres besoins (javascript, css, etc.). C'est primordial de le mettre.
- `body_class()` ajoute des classes utiles.

# header.php

```
<!DOCTYPE html>  
<html lang="fr">    ←  
    <head>  
        [...contenu...]  
        <?php wp_head(); ?>  
    </head>  
  
<body <?php body_class(); ?>>  
    <header>  
        [...contenu...]  
    </header>
```



bien que les balises <html>  
et <body> soient ouvertes ici,  
elles ne seront fermées que  
dans footer.php

## le fichier footer.php

- Ce fichier contiendra le contenu qui terminera chaque page.
- Cette page doit contenir au minimum les balises de fermeture du html, soit </body> et </html>.
- Il faut aussi utiliser la fonction wp\_footer() avant la fermeture de la balise body.

# footer.php

```
<footer>  
    ©2023, Montréal en lumière  
</footer>  
  
<?php wp_footer(); ?>  
  
</body>  
</html>
```



elles sont là!

## le fichier sidebar.php

- Ce fichier contiendra le contenu qui se situera dans une colonne latérale ou une région quelconque.
- Le sidebar sert généralement à l'affichage de widgets (bloc de texte, liste de catégories, formulaire de recherche, un calendrier, etc.), de menus ou de contenu secondaire.
- Il contient habituellement un bloc `<aside>`.

# sidebar.php

```
<!-- Zone secondaire de contenu -->  
<aside>  
    <h3>Les à-côtés</h3>  
    [ . . . ]  
</aside>
```

# index.php révisé avec les inclusions

```
<?php get_header(); ?>

<!-- Bannière -->
<div id="banniere"> [...] </div>

<!-- Zone principale de contenu -->
<div id="contenu">
    <div id="principal">

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

                <article> [...] </article>

                <?php endwhile; ?>
            <?php endif; ?>

    </div>
    <?php get_sidebar(); ?>
</div>

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



la fin