

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



kahoot!

à vous de jouer

01

un thème
personnalisé?

FURROW



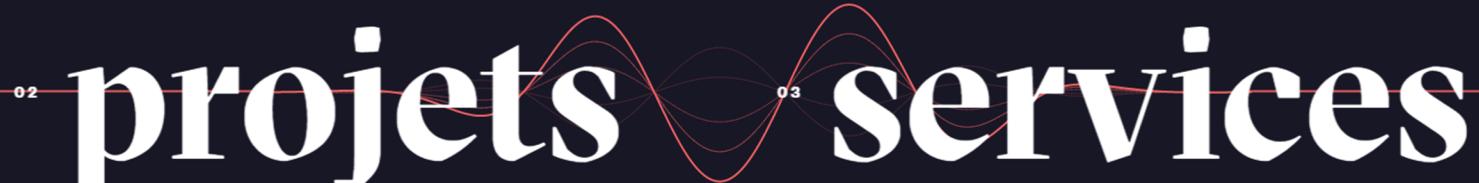
DIG DEEP

furrow.studio

W.
Site of the Day



ce projets services



tb
ig
pa





We are a house
of creativity



tout est
possible avec
WordPress

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

AWWWARDS®

[sites utilisant WordPress](#)



[vitrine officielle de WordPress](#)

02

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



c'est si simple..

pour créer un nouveau thème

il suffit de créer un sous-dossier dans [wp-content/themes/](#) et d'y placer les 2 fichiers magiques :

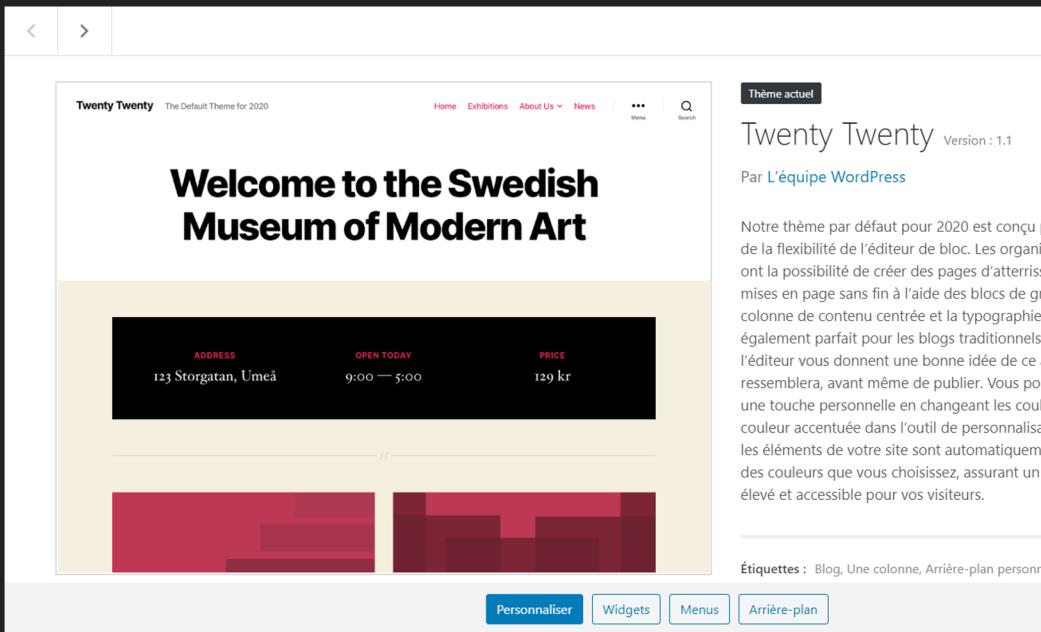
- index.php
- style.css

Filename	Filesize	Filetype
..		
mtl_lumiere		File folder
twentynineteen		File folder
twentyseventeen		File folder
twentytwenty		File folder
index.php	28	PHP File

capture d'écran de votre thème

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

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



feuille de style

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

```
/*
Theme Name: Twenty Twenty
Author: the WordPress team
Author URI: http://wordpress.org/
Description: The 2020 theme for WordPress ...
Version: 1.0
Tags: white, black, minimalist
Text Domain: twentytwenty
*/
```



The screenshot shows the WordPress theme editor interface for the Twenty Twenty theme. The top navigation bar includes back and forward arrows, a search icon, and a menu icon. The left sidebar displays the theme's name, Twenty Twenty, as the Default Theme for 2020. The main content area features a large heading: "Welcome to the Swedish Museum of Modern Art". Below the heading is a card with address and opening hours information: "ADDRESS 123 Storgatan, Umeå" and "OPEN 9:00". The right side of the editor contains descriptive text and navigation links. A teal box labeled "Version" points to the "Thème actuel" section, which shows "Twenty Twenty Version : 1.1" and credits "Par L'équipe WordPress". Another teal box labeled "Author" points to the author information. A third teal box labeled "Description" points to the main content text, which highlights the theme's flexibility and design features. A fourth teal box labeled "Tags" points to the "Etiquettes" section at the bottom, which lists "Blog, Une colonne, Arrière-plan personnalisé, Couleurs personnalisées,". At the bottom of the editor, there are four buttons: "Personnaliser", "Widgets", "Menus", and "Arrière-plan".

Twenty Twenty The Default Theme for 2020

Theme Name

Version

Thème actuel

Twenty Twenty Version : 1.1

Author

Description

Tags

Personnaliser Widgets Menus Arrière-plan

02

les fonctions php

A photograph of a man with glasses and dark hair, seen from behind, sitting at a desk and working on a computer. He is wearing a light-colored hoodie. Two monitors are on his desk; the left one shows a code editor with purple and white syntax highlighting, and the right one shows a similar screen with blue and green syntax. A keyboard and mouse are in front of him. A small white dog is lying on the desk to his right. The background shows a wooden door and a plain wall.

about time

vous avez créé votre premier thème

Cependant, vous constaterez 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
- C'est ce qu'on appelle dans le jargon WordPress un marqueur de modèle (template tags)
- La fonction bloginfo() permet d'obtenir du contenu dynamiquement grâce à des paramètres

principaux paramètres de bloginfo()

- bloginfo('name');
titre du site
- bloginfo('charset');
jeu de caractère
- wp_title();
titre de la page (précédé du |)
- bloginfo('stylesheet_url');
lien vers la feuille de style (style.css)
- 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

Laboratoire semaine 05

Slogan

Un site utilisant WordPress

En quelques mots, décrivez la raison d'être de ce site.

```
<head>
    <meta charset="<?php bloginfo('charset') ?>">
    <title>
        <?php bloginfo('name') ?> <?php wp_title() ?>
    </title>
    <link rel="stylesheet"
          href="<?php bloginfo('template_url')?>/styles/mes-
          styles.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

- placer un dossier `images` dans le dossier de votre thème
- récupérez le chemin jusqu'au dossier de votre thème et ajoutez-le à l'attribut `src` de l'image

```
  
  
!-- Résultats dans le navigateur en vérifiant la source -->  

```

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

03

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

Syntaxe alternative de PHP

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

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 } ?>
```

« the loop »

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

<!-- tant qu'il y a un nouvel article à afficher, roule la boucle -->
<?php while (have_posts()) : the_post(); ?>
    <!-- structure html pour afficher le 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 options possibles dans « la boucle »

- the_title();
titre
- the_ID();
identifiant unique
- the_excerpt();
résumé
- the_content();
contenu
- the_post_thumbnail();
image à la une
- the_category();
catégorie(s)
- the_tags();
mot(s)-clef(s)
- the_date() ou the_time();
date de publication
- the_permalink();
permalien

référence sur les paramètres de bloginfo()



intégrer le contenu de vos articles (post)

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

<article <?php post_class(); ?> id="post-<?php the_ID(); ?>">
    <h3><?php the_title(); ?></h3>
    <?php the_post_thumbnail('medium'); ?>
    <p class="info">Prix: <?php the_field('prix'); ?></p>
    <p class="info">Lieux: <?php the_field('lieu'); ?> </p>
    <?php the_excerpt(); ?>
    <a href="<?php the_permalink(); ?>" class="more-link">Suite </a>
</article>

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

04

les gabarits WordPress

il y en a une panoplie..

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 section, WordPress tentera par défaut de trouver le gabarit le mieux adapté selon un arbre de décision.



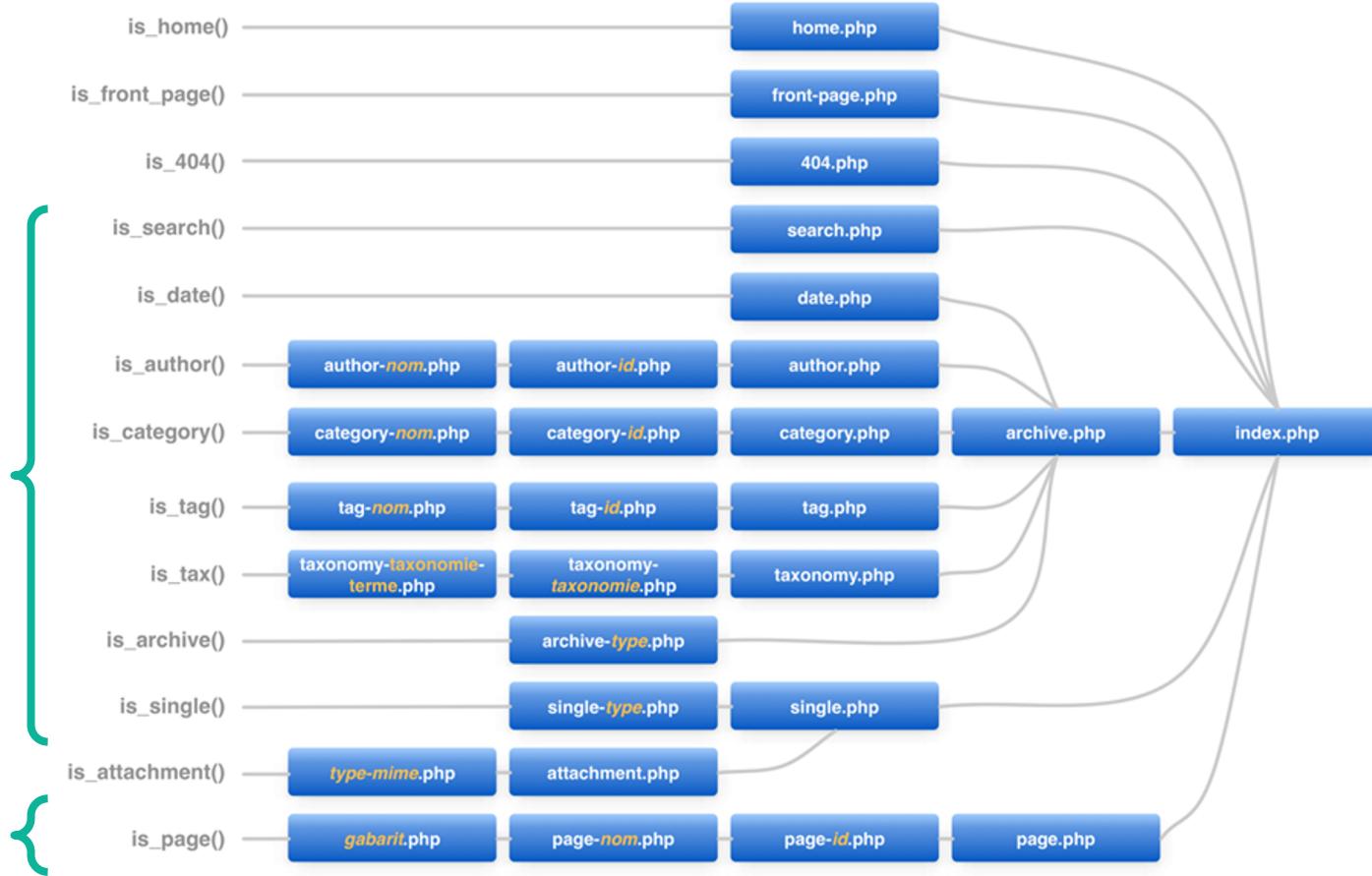
un quoi?

..on y arrive

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-clef	/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

articles



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.

méthodes

is_front_page() vs is_home() ?

- `is_front_page()` : est vrai pour la page d'accueil définie dans les réglages qu'il s'agisse d'une page spécifique ou de la liste des derniers articles
- `is_home()` : est vrai pour la page affichant la liste des derniers articles que ce soit la page d'accueil ou non. (ex.: page blogue en mode site Web)

un gabarit pour les pages : page.php

- si on veut une mise en page différente pour les pages versus les articles. Il ne suffit que de créer un fichier `page.php` et d'ajuster le design à nos besoins
- si WordPress est configuré pour afficher une page spécifique comme page d'accueil (plutôt que la liste d'articles), le gabarit `front-page.php` sera assigné à la page en question
- pour créer un gabarit spécifique à une seule page :
 - `page-nom.php` (si le site utilise les liens optimisés pour le SEO)
 - `page-id.php` (si le site utilise les liens du type ?p=12)

un gabarit pour plusieurs pages

- on peut créer des gabarits de page pouvant être utilisés par plusieurs pages (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: Mon gabarit */  
?>
```



Attributs de page

Modèle :

- Modèle par défaut
- Modèle par défaut
- Mon gabarit
- (Aucun parent)

Le fichier devrait utiliser la nomenclature suivante : template-[nom_du_template].php. Il peut être à la racine ou dans un sous-dossier.



page-nom.php vs template-[nom_gabarit].php?

- `page-nom.php` est intéressant pour un site qui change peu. Un gros désavantage est que lorsque le nom de la page change, il faut changer le gabarit aussi
- `template-[nom_gabarit].php` est intéressant car il s'adapte peu importe le contenu des pages. Comme il est configuré dans une page, on n'a rien à changer au niveau du gabarit pour qu'il continue de fonctionner normalement

la meilleure pratique?

**un gabarit qu'on peut
choisir dans les réglages
d'une page**

À propos

Classique

À propos du festival

MONTRÉAL EN LUMIÈRE, c'est un des plus grands festivals d'hiver au monde, une folie qui enregistre annuellement 1 300 000 visites de festivaliers venus vivre pleinement l'hiver montréalais par le biais d'une programmation inusitée, alliant arts de la scène, gastronomie et activités familiales extérieures gratuites, sans oublier une nuit complète de découvertes des plus originales !

Document Bloc

État et visibilité

Permalien

Image mise en avant

Définir l'image mise en avant

Discussion

Attributs de page

Modèle : Modèle par défaut

Page parente : (Aucun parent)

Ordre 0

05

les fichiers en inclusions

copier/coller on aime ça..

A close-up photograph of a young girl with light brown hair and bangs, wearing round-rimmed glasses. She is holding her head in both hands, looking directly at the camera with a distressed expression. The background is a solid purple color.

..mais là, ça va faire!

- 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
- formulaire de recherche : searchform.php
- commentaires : comments.php
- etc.

appeler les fichiers en inclusion

- get_header();
header.php
- get_footer();
footer.php
- get_sidebar();
sidebar.php
- get_search_form();
searchform.php
- comments_template();
comments.php
- get_template_part('composante');
composante.php



utilisera le fichier
composante.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

```
<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>  
    ©2021, Montréal en lumière - Exercice dans le cours Programmati...  
</footer>  
<?php wp_footer(); ?>  
</body>  
</html>
```



elles sont là!

le fichier sidebar.php

- ce fichier contiendra le contenu qui se situera dans la colonne latérale
- 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 n'y a aucun code obligatoire qui doit apparaître dans ce fichier

sidebar.php

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

index.php révisé avec les inclusions

```
<?php get_header(); ?>
<div id="banniere">
(...)
<div id="principal">
    <?php if (have_posts()) : ?>
        <?php while (have_posts()) : the_post(); ?>
            <article>[...]</article>
        <?php endwhile; ?>
    <?php else : ?>
        <p>Oups! Il n'y a rien!</p>
    <?php endif; ?>
</div>
<?php get_sidebar(); ?>
<?php get_footer(); ?>
```



la fin

exercice s05

créer son thème personnalisé

➔ instructions sur le site du cours dans semaine05 ➔