

# WordPress pour les développeurs PHP

Intervenant : Jean-Frédéric VINCENT



The background of the slide features a close-up photograph of several slices of orange stacked on a rustic, dark wooden surface. A semi-transparent purple rectangular overlay covers the middle portion of the image, serving as a background for the text.

Module 4

# Le fichier functions.php

# Objectif du module

- Mettre en place le fichier functions.php
- Comprendre le système de crochets (hooks)
- Activer les images mises en avant (post thumbnails) pour les articles (posts)
- Activer les emplacements menus dans le panel admin de WordPress
- Positionner le menu dans le template (header.php)

# Les fichiers du thème orange

Nous allons ajouter et créer **1 nouveau fichier** à notre thème : orange

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

- header.php
- footer.php

- page.php
- page-contact.php
- single.php (*post article*)



- functions.php



Comment fonctionne le fichier **functions.php** ?

- Chaque page PHP du thème a automatiquement accès à ce fichier
- Il permet de définir des constantes PHP.
- Il permet d'activer ou désactiver des sections entières du Back Office
- ( Image à la une, Menu , Widget, Custom Post et autres ...)
- Chaque fonction est reliée à un système de "crochets" (hooks en anglais)

## 04 Le fichier functions.php

# functions.php



functions.php

Voici un exemple du fichier **functions.php** du thème par défaut : twenty nineteen

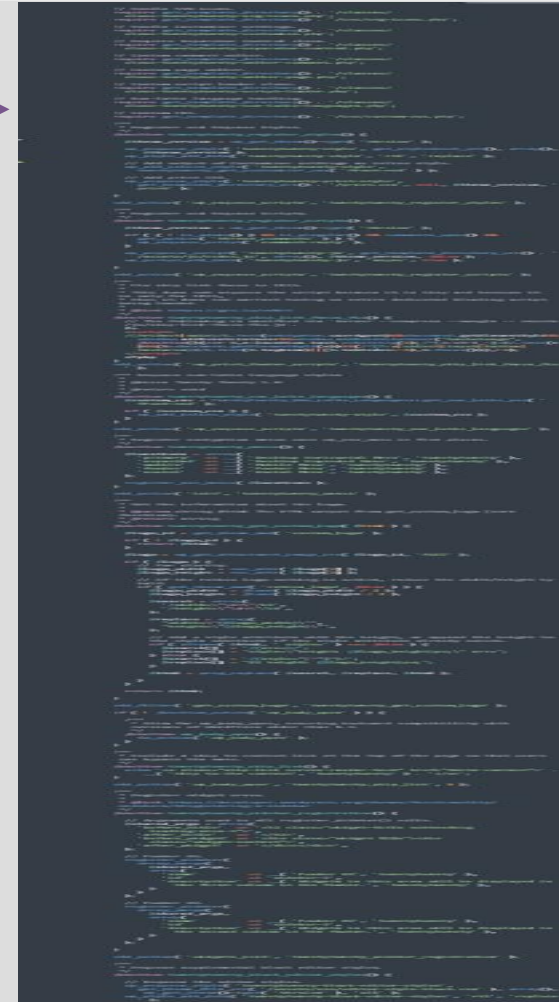
### Inconvénients

- Code monolithique
- Obligation de mettre beaucoup de commentaires

### Avantages

- + • Concentre toutes les modifications du thème

⚠ A noter : Ne pas oublier le "s" au nom du fichier **functions.php**





# Les différentes parties de functions.php



functions.php

- 1 Définir des constantes
- 2 Activer les images mise en avant
- 3 Activer les menus
- 4 Mettre un filtre pour les classes CSS su menu



# Définir les constantes



functions.php

## ETAPE 1 : Définition de constantes PHP utiles pour notre thème

Nous allons les réutiliser dans chaque Template

<?php

functions.php

```
// -----  
// 1 - Les Constantes  
// -----  
define( 'THEMEROOT', get_stylesheet_directory_uri() );  
define( 'IMG', THEMEROOT. '/img' );  
define( 'JS', THEMEROOT. '/js' );  
define( 'CSS', THEMEROOT. '/css' );  
// -----
```



*Pensez commentaires !*

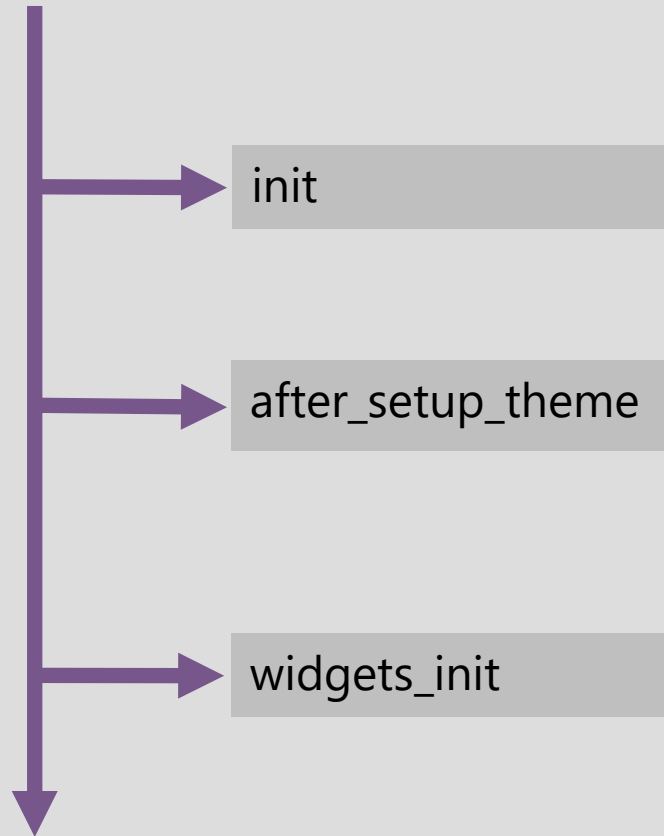
1





# La Time line du chargement

Voici La **Time line** du chargement de WordPress



Chaque étape du chargement de WordPress est définie par un nom :

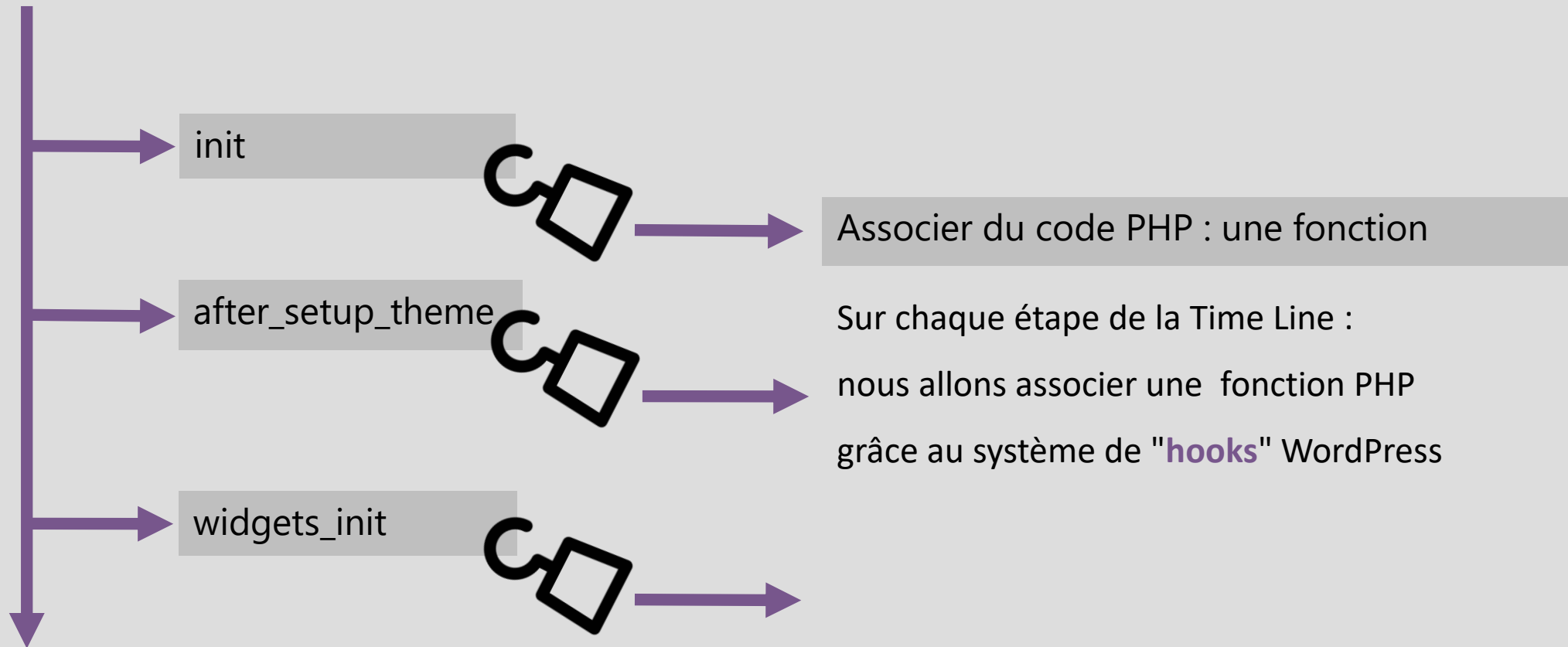
- Initialisation
- Après le chargement du thème
- Au moment de l'initialisation des widgets
- ...

# Le système de hooks



functions.php

Voici La **Time line** du chargement de WordPress avec les crochets ou **hooks** en anglais

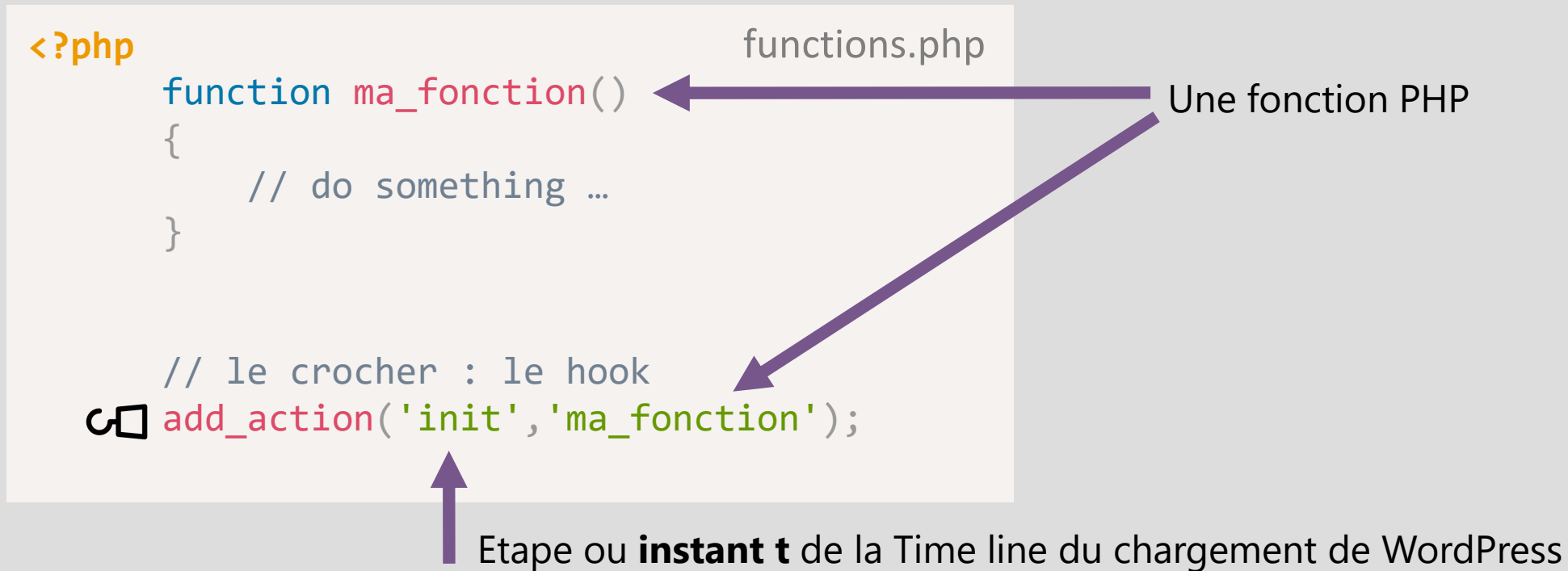


# Définir les constantes

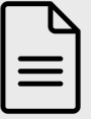


functions.php

Mise en place d'une fonction avec un crochet ou **hook** en anglais



# Activer les images mise en avant



functions.php

**Etape 2** : Activation des "images mises en avant" (**post-thumbnails**) pour Article (**post**)

```
// ... functions.php
// -----
// 2 - Activer les images mises en avant dans Article (Post)
// En Anglais : the post thumbnail
// -----
function my_theme_setup()
{
    // active les images mises en avant dans Article (Post)
    add_theme_support('post-thumbnails');
}

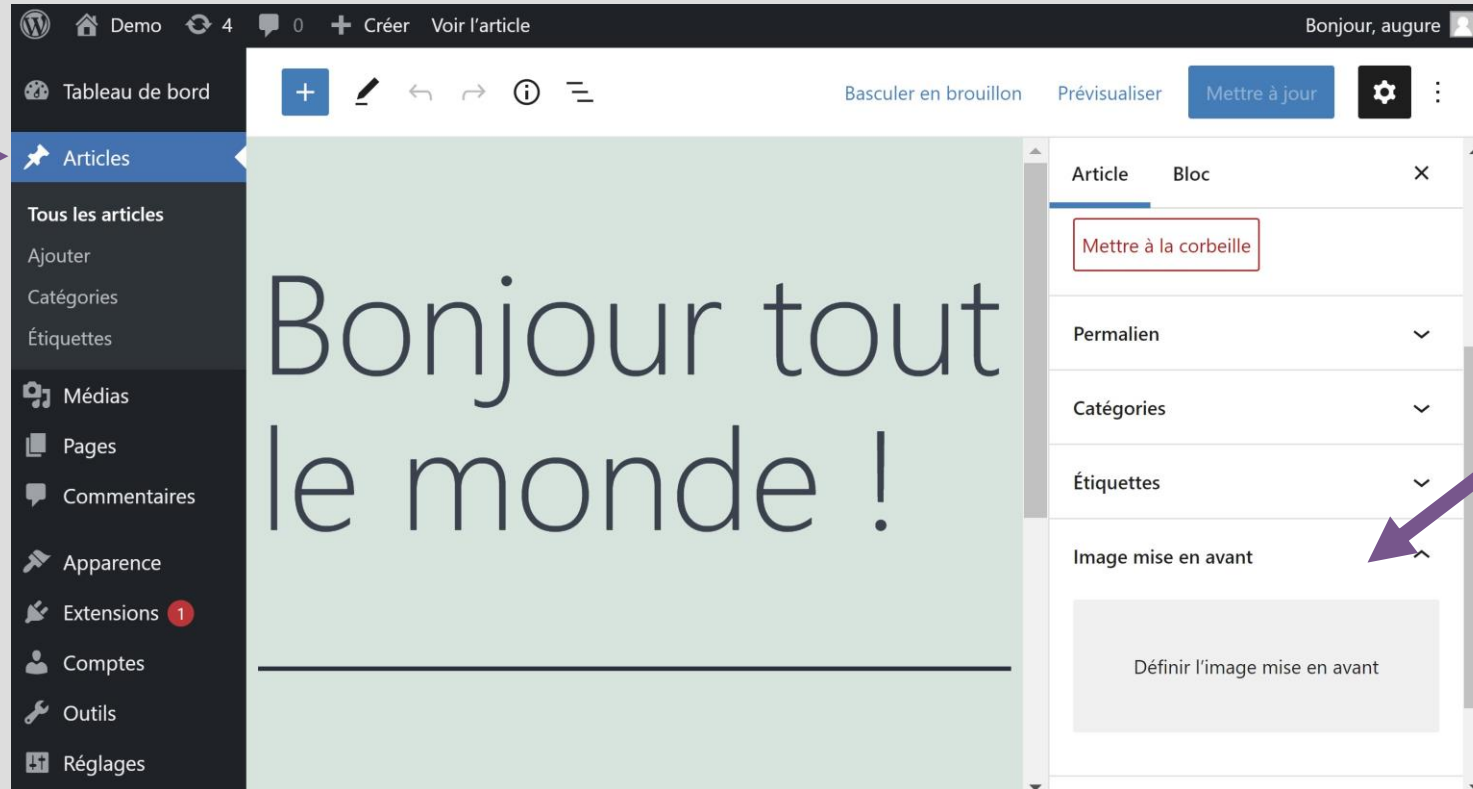
// le crochet : le hook
add_action('after_setup_theme', 'my_theme_setup');
// -----
```

Time line



# Activer les images mise en avant

**Vérification** : Activation des images mise en avant pour Article dans le panel admin de WordPress



Une fois le fichier **functions.php** modifié :  
l'onglet Image mise en avant apparaît.

2

# Afficher l'images mise en avant



single.php

Afficher l'image "mise en avant" (**post-thumbnail**) sur la page de l'article **single.php**

```
<?php the_post_thumbnail('full', ['class'=>'d-block w-100']); ?>
```

*Choix de la taille de l'image*

```
// 150 x 150 "cropped" ou recarer  
the_post_thumbnail( 'thumbnail' );  
  
// 300 x ... (hauteur en fonction)  
the_post_thumbnail( 'medium' );  
  
// 768 x ... (hauteur en fonction)  
the_post_thumbnail( 'medium_large' );  
  
// 1024 x ... (hauteur en fonction)  
the_post_thumbnail( 'large' );  
  
// taille originale  
the_post_thumbnail( 'full' );
```

*Génère ce code html*

```

```

*A noter : Le "echo" est compris dans la fonction*

2

# Afficher les images mise en avant



single.php

Afficher l'image "mise en avant" (**post-thumbnail**) sur la page de l'article **single.php**

```
<?php
    get_header();
    while(have_posts()):
        the_post();
?>
<h1><?php the_title() ?></h1>
<p><?php the_content() ?></p>

<?php the_post_thumbnail('full', ['class'=>'d-block w-100']); ?>

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

single.php

2



# Demonstration

Création du fichier functions.php

# Les menus

**ETAPE 3** : Pour mettre en place le ou les menus, nous devons le faire en **2 actions** :

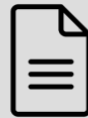
Je déclare

A

```
<?php functions.php
//...
register_nav_menus($tab);
//...
```

### A. Déclarer le ou les menus

- Son identifiant
- Le nom qui apparaîtra dans le panel Admin de WP



functions.php

J'affiche

B

```
<?php header.php
//...
wp_nav_menu($tab2);
//...
```

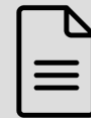
### B. Afficher le ou les emplacements où **afficher** le menu :



header.php



footer.php



...

J'affiche

B

```
<?php footer.php
//...
wp_nav_menu($tab2);
//...
```

3

# Activer les menus



functions.php

Activer les emplacements du menu pour le panel admin de WordPress

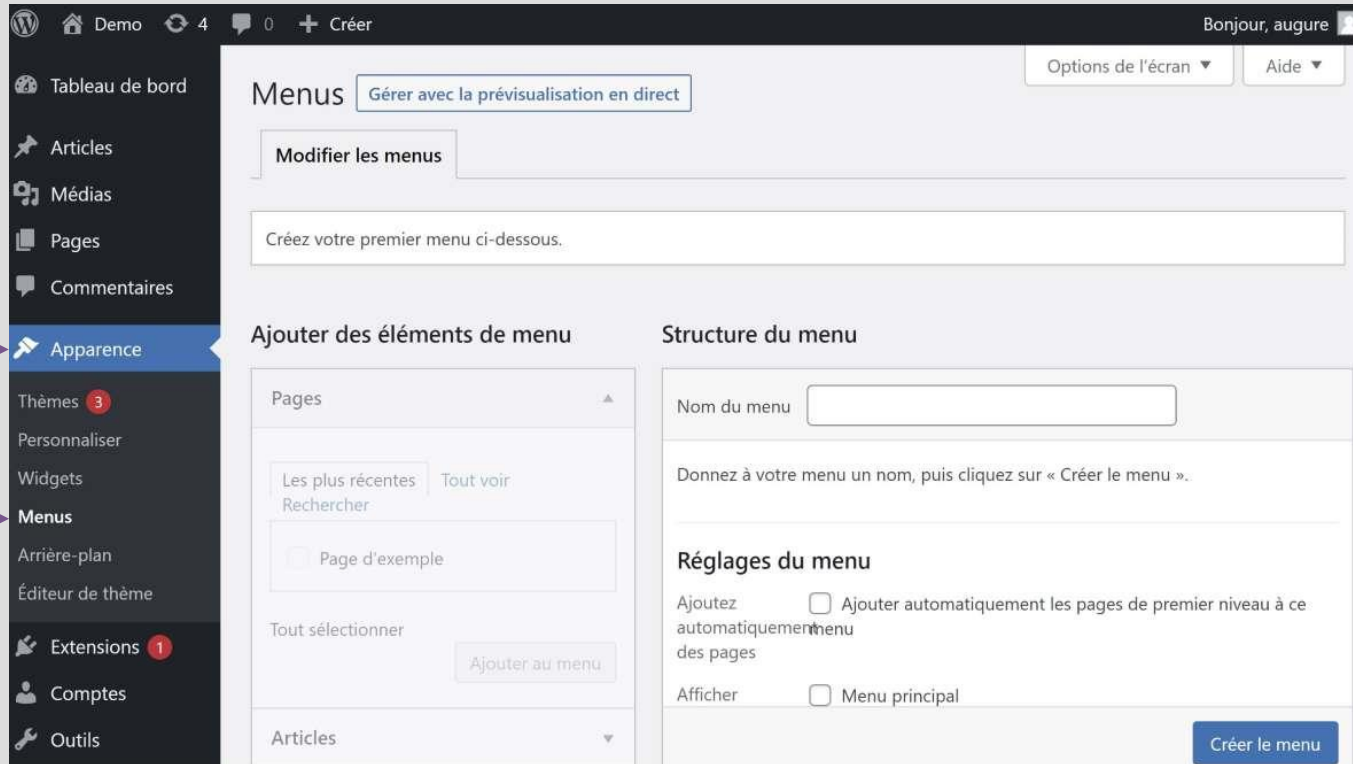
```
<?php                                                                    functions.php
// -----
// 2 & 3 - Activer les images mise en avant ET les menus
// -----
function my_theme_setup(){
    add_theme_support('post-thumbnails'); // img mise en avant
    ➔ register_nav_menus([
        // ['identifiant' => 'Nom dans le Back Office Wordpress' ]
        'menu_haut' => 'Menu Haut de Page',
        'menu_bas'  => 'Menu Bas de Page'
    ]);
}
add_action('after_setup_theme', 'my_theme_setup'); // le crochet
// -----
```

3

## 04 Le fichier functions.php

# Activer les menus

**Vérification** : Activer les emplacements du menu pour le panel admin de WordPress



Une fois le fichier **functions.php** modifié :  
l'onglet **Menus** mise en avant apparaît.

- 1) Créer des menus
- 2) Associer un menu à un emplacement

# 3



**Vérification** : Activer les emplacements du menu pour le panel admin de WordPress

```
<!-- [...] -->                                     header.php
<body>
  <nav>
    <?php
      $menu["theme_location"] = "menu_haut";
      $menu["menu_class"]     = "navbar-nav ml-auto";
      $menu["container"]     = false; // pas de div
      wp_nav_menu($menu);
    ?>
  </nav>
```

Identifiant défini dans functions.php

Attribut : **class=** de <ul>

Pas de balise <div>  
autour de la balise <ul>

```
<div> <ul>...</ul> </div>
```

3

# Activated : les liens activés



functions.php

```
//----- functions.php
// 4 . Ajouter les class css du menu
//-----

function my_filter_a($attributes){ // <ul><li><a class="nav-link" ...>
    $attributes["class"] = "nav-link";
    return $attributes;
}
add_filter('nav_menu_link_attributes','my_filter_a');

function my_filter_li($classes){ // <ul><li class="my-li active">
    $classes[] = "nav-item";
    // si cette classe : 'current-menu-item' est présente j'ajoute active
    if (in_array('current-menu-item',$classes)){
        $classes[] = "active";
    }
    return $classes;
}
add_filter( 'nav_menu_css_class','my_filter_li');
```

4