

WordPress : créer ses propres tailles d'images

Déclarer de nouvelles tailles d'images

Fonction `add_image_size()`

Pour créer vos propres tailles d'images en PHP pour WordPress, vous pouvez utiliser la fonction `add_image_size()`.

Voici comment l'utiliser :

1. Ouvrez le fichier **functions.php** de votre thème actif.
2. Ajoutez le code pour définir votre propre taille d'image personnalisée
3. Enregistrez le fichier **functions.php** et rafraîchissez votre site WordPress

```
<?php
```

```
// fonction pour définir une taille d'image personnalisée
function theme_custom_image_sizes() {
    // ajoutez ici vos tailles d'images personnalisées
    add_image_size( 'nom_de_la_taille', largeur, hauteur, recadrage );
}
```

```
// action pour ajouter vos tailles d'images personnalisées
add_action( 'after_setup_theme', 'theme_custom_image_sizes' );
```

Dans le code ci-dessus, vous devez remplacer '**nom_de_la_taille**' par un nom unique pour votre taille d'image.

Ensuite, spécifiez la largeur et la hauteur souhaitées pour la taille d'image personnalisée.

Vous pouvez également définir la valeur de recadrage (**cropping**) pour le redimensionnement de l'image.

Les options disponibles pour le recadrage sont **false** (par défaut, sans recadrage), **true** (recadrage forcé) ou un tableau spécifiant les points d'ancrage du recadrage.

Fonction `set_post_thumbnail_size()`

L'extension `set_post_thumbnail_size()` permet de **définir une taille spécifique par défaut pour les images mises en avant (featured images)**.

Elle permet de contrôler la taille par défaut de l'image mise en avant lorsqu'elle est affichée dans différents contextes du thème.

```
<?php
```

```
// définir la taille par défaut de l'image mise en avant
set_post_thumbnail_size( largeur, hauteur, recadrage );
```

L'argument **recadrage** est facultatif et détermine si l'image doit être recadrée pour s'adapter exactement à la taille spécifiée.

Les options pour l'argument **recadrage** sont **true** (pour recadrer) ou **false** (pour conserver les proportions de l'image).

Après avoir défini la taille par défaut de l'image mise en avant à l'aide de **set_post_thumbnail_size()**, WordPress utilisera cette taille lorsqu'il affichera l'image mise en avant sans spécifier de taille personnalisée.

Cela garantit une cohérence visuelle en définissant une taille par défaut uniforme pour les images mises en avant dans votre thème.

Remarque : **set_post_thumbnail_size()** ne redimensionne pas rétroactivement les images déjà téléchargées. Cela n'affecte que les futures images mises en avant. Si vous souhaitez appliquer la nouvelle taille aux images existantes, vous pouvez utiliser un plugin de régénération des miniatures, comme **Regenerate Thumbnails**.

Utiliser les tailles d'images

Maintenant, vous pouvez utiliser votre taille d'image personnalisée dans vos modèles ou vos fichiers de thème en utilisant la fonction **the_post_thumbnail()** avec le paramètre **'nom_de_la_taille'**.

Par exemple :

```
<?php

// afficher l'image avec votre taille d'image personnalisée
the_post_thumbnail( 'nom_de_la_taille' );
```

Lorsque vous appelez **the_post_thumbnail()** avec le nom de votre taille d'image personnalisée, **WordPress redimensionnera automatiquement les images pour correspondre à la taille spécifiée.**

Attention ! Pour pouvoir utiliser les images mises en avant, il vous faut activer cette fonctionnalité via la fonction **add_theme_support()**.

```
<?php

// activer l'utilisation des images mises en avant depuis le fichier
functions.php du thème actif
add_theme_support( 'post-thumbnails' );
```

Remarque : inutile de rajouter une **balise HTML ** elle est automatiquement générée par **the_post_thumbnail()** si une image à la une est renseignée.

Exemple de code généré :

```

```

Les attributs **srcset** et **sizes** dans la **balise HTML ** sont utilisés pour **spécifier des versions d'images alternatives et adaptatives, ce qui permet au navigateur de choisir la meilleure image à afficher en fonction de la résolution et de la taille de l'appareil utilisé pour visualiser la page.**

L'attribut srcset : il permet de spécifier une liste de chemins vers différentes versions d'une même image, chacune ayant une résolution différente. Le navigateur peut alors choisir la meilleure option en fonction de la résolution de l'écran.

L'attribut sizes : il permet de fournir des informations supplémentaires sur la taille d'affichage de l'image. Cela aide le navigateur à décider quelle version de l'image doit être chargée. L'attribut **sizes** utilise des **médias queries** pour définir la largeur de l'image en fonction de la largeur de la fenêtre d'affichage ou de la taille de l'élément HTML contenant l'image.

Les instructions **srcset** et **sizes** sont compatibles avec plus de 95 % des navigateurs web du marché (cf. le site [CanIUse](#)).

Pour résumer, **srcset** et **sizes** permettent d'éviter de charger une image de 2 000 pixels de largeur sur un mobile qui ne dispose que de 768 pixels.

Remarque : si vous ajoutez de nouvelles tailles d'images après avoir déjà téléchargé des images, vous devrez régénérer les miniatures existantes à l'aide d'un plugin tel que "[Regenerate Thumbnails](#)" pour que les nouvelles tailles d'images s'appliquent rétroactivement.