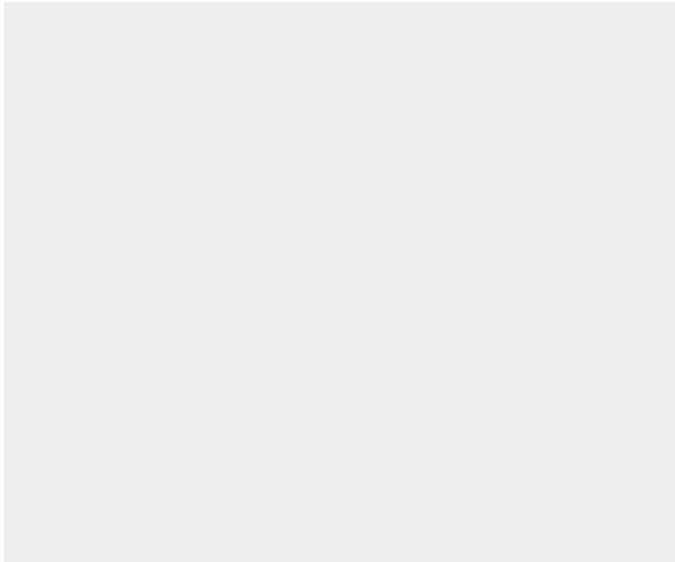


# Hi  rarchie des Templates

Selon la page appel  e, WordPress utilisera un des templates existants. Mais lequel aura la priorit   ?

## Du global au sp  cifique (et vice-versa)



L'image la plus **importante** de ce tutoriel :



Ce sch  ma d  finit le **chemin** que WordPress emprunte pour choisir le template    utiliser. D  s que l'un d'eux correspond, il est utilis   et le chemin s'arr  te l  .

Par exemple, l'internaute veut afficher les archives d'Avril 2010 : `blog.fr/2010/04`.

- WordPress rentre dans la boucle au niveau de `Archive Page` car il s'agit d'une page **d'archive**.
- Ensuite il bifurque vers `Date Archive` car il s'agit d'une archive **dat  e**.
- Ici on a   crit `/2010/04`. Il s'agit donc d'une archive **mensuelle**.
- WordPress tombe alors sur `date.php` et l'utilisera s'il existe.
- Si `date.php` n'existe pas, WordPress utilisera le template `archive.php`.
- Si ce dernier n'existe pas non plus, il utilisera `index.php`.

Etant donn   que `index.php` est le dernier recours, il est **indispensable**.

Beaucoup de gens confondent `home.php` et `index.php` parce qu'un th  me peut se passe du premier mais pas du second. En effet, il est pratique d'utiliser `index.php` comme page d'accueil. Comme   a, si un template n'existe pas, on affiche la page d'accueil. Mais il faut comprendre la diff  rence : `home.php` est le template de la page d'accueil, `index.php` est le template par **d  faut**.

## Globaliser

Si vous voyez que les templates `date.php`, `category.php` et `tag.php` sont **identiques**, vous pouvez les supprimer et cr  er    la place un fichier `archive.php`. De ce fait, vous gagnerez du temps    la maintenance si vous souhaitez modifier une partie du code.

Vous pouvez aller plus loin en utilisant **uniquement** `index.php` avec quelques conditions PHP dans le code, mais nous verrons   a plus loin.

Attention, lorsque je dis que les templates sont **identiques**, cela ne veut pas dire qu'ils afficheront la *m  me* chose sur le site, c'est juste qu'ils utilisent le m  me **code**. Leur *structure* est donc identique. Par exemple, `date.php` peut avoir un header, la liste des billets, et un footer. `category.php` peut avoir **exactement** le m  me code, la m  me structure. C'est WordPress qui sera assez intelligent pour afficher des billets *diff  rents* dans les deux cas (selon la date dans un cas, selon la cat  gorie choisie dans l'autre).

## Sp  cifier

Dans l'autre sens, il est possible pour un m  me template de cr  er une version plus **sp  cifique**. Par exemple, vous voulez une page diff  rente pour la cat  gorie « Hardware ». Au lieu d'utiliser le template `archive.php` ou `category.php`, nous pouvons cr  er un template sp  cifique : `category-hardware.php`.

Cette m  thode de templates tr  s sp  cifique s'av  rera tr  s utile pour cr  er des **templates de Page** (i.e. de Pages WordPress). Nous verrons   a plus loin aussi.

LE TUTORIEL	
Introduction et Installation	1
Structure: contenu + th��me + plugins	2
Anatomie d'un th��me	3
Hi��rarchie des Templates	4
Les includes	5
La boucle	6
Single.php : le template d��di��	7
Templates de Page	8
Boucle modifi��e, boucles multiples	9
Custom Post Type, Custom Taxonomy	10
Conditional Tags	11
Les Widgets WordPress	12
Post Formats = Tumblr	13

Le Guide WordPress est un site **gratuit** destin   aux graphistes et int  grateurs qui souhaitent cr  er leur propre th  me WordPress de A    Z.

