

WordPress è un CMS (Content Management System) - **gratuito!**

Con WordPress si creano siti **dinamici (database MySQL)**,

WP utilizza i **Tem**i, ovvero sono un insieme di file (layout, immagini e fogli di stile) che si occupano di mettere insieme – generare la componente grafica e la dimensione «comportamentale» del sito. (*front end*)

Per tutti i dettagli aggiuntivi entrano in gioco i **Plugin**, che sono dei veri e propri “programmi” scaricabili ed installabili, non autonomi, utilizzati per interagire ed estendere le funzionalità base di WordPress.

BACHECA DI WP

The screenshot shows the WordPress dashboard interface. At the top, there's a navigation bar with the WordPress logo, 'I miei siti', 'WebDocesi', and 'Nuovo' buttons. The user is logged in as 'Ciao, admin'. The main content area is titled 'Bacheca' and features a welcome message: 'Benvenuto su WordPress! Abbiamo raggruppato alcuni link per aiutarti ad iniziare:'. Below this, there are three columns of quick actions: 'Come iniziare' with a 'Personalizza il tuo sito' button; 'Passi successivi' with links to 'Modificare la pagina iniziale', 'Aggiungere altre pagine', and 'Visualizza il tuo sito'; and 'Altre azioni' with links to 'Gestione widget o menu', 'Attiva o disattiva i commenti', and 'Maggiori informazioni su come iniziare'. The dashboard is divided into several widget areas: 'In sintesi' showing 9 articles and 12 pages; 'News and Updates' with a message about themes; 'Attività' listing recent publications; 'Bozza rapida' for creating a new post; and 'Novità di WordPress' with news about Meetup Torino and WordCamp Milano. A large dashed box on the right indicates a widget area for dragging and dropping widgets.

TEMI DI WP

1 miei siti WebDiocesi + Nuovo Eventi

Ciao, admin

Auto

Temi 2

Cerca tra i temi installati...

WP-Starter

A WordPress Child Theme

Sticky

Attivo: WP-Starter-webdiocesi [Personalizza](#)

WP-Forge

Tiled Gallery

[Personalizza](#)

Grazie per aver creato con WordPress.

Versione 4.6.1

TEMA CHILD

Un child theme è un tema che eredita tutte le caratteristiche del tema “padre”, tranne quelle che noi vogliamo personalizzare.

La procedura per creare un child theme:

Apriamo la cartella wp-content/themes della nostra installazione di WP: qui si trovano tutti i temi che abbiamo installato.

Creiamo una nuova cartella, assegnandole il nome “nostro_tema_child”.

Creiamo all’interno della cartella un file di testo vuoto, che deve chiamarsi obbligatoriamente style.css

Apriamo adesso il file style.css e copiamo al suo interno il seguente testo

```
/*
```

```
Theme Name: nostro_tema_child
```

```
Theme URI: http://www.indirizzo\_nostro\_sito/
```

```
Description: Personalizzazione del nostro_tema
```

```
Author: Mario Rossi
```

```
Author URI: http://www.indirizzo\_nostro\_sito/pagina\_autore/
```

```
Template: nostro_tema
```

```
Version: 1.0
```

```
*/
```

Per impostazione predefinita, tutti i file creati (con l'eccezione di functions.php) sostituiranno i file equivalenti del tema genitore:

quindi per ereditare il css del padre aggiungiamo al nostro style.css:

```
@import url("../twentyeleven/style.css");
```

Lo stesso principio vale anche per tutti gli altri file che compongono un tema WordPress (header.php, index.php, footer.php, ecc).

Se ad esempio vogliamo modificare il file header.php dobbiamo semplicemente creare un file con lo stesso nome nella directory del child theme e personalizzarlo come vogliamo.

Il file function.php di un child theme non viene sostituito ma caricato in aggiunta al function.php del parent theme.

Se aggiungiamo una funzione al file function.php del parent theme questa sarà eliminata al primo aggiornamento del tema. Se invece la scriviamo nel file function.php del child theme non scomparirà.

Homepage & Personalizzazione

The image shows a screenshot of the WebDiocesi website's customization interface. On the left, a sidebar titled 'Stai personalizzando WebDiocesi' contains a list of settings: Tema attivo (WP-Starter-webdiocesi), Header Settings, Background Image Settings, Main Menu Settings, Color Settings, Front Page Settings, Content Settings, Main Sidebar Settings, Footer Sidebar Settings, Footer Settings, Foundation Buttons, Menu, and Widget. The main content area displays the website's header with the logo and tagline 'WebDiocesi la tua diocesi sul web', a navigation menu, and a large banner image of people's hands on grass with the text 'Noi siamo pronti, voi?'. Below the banner, there is a 'Follow Us' section with social media icons and a 'Focus' section with a small image.

Le pagine

- non vengono associate a categorie e tag, ma possono essere organizzate gerarchicamente, associandole ad una pagina “genitore”.
- non sono mostrate secondo una base cronologica come avviene per gli articoli, ma secondo una nostra precisa scelta (ordinamento)
- non vengono richiamate in un ciclo nella home per mostrare i contenuti, come avviene per gli articoli
- non sono associate a nessuna categoria e tag
- non devono necessariamente avere tutte lo stesso layout, è quindi possibile differenziare come visualizzare una pagina
- non sono usate per mostrare contenuti freschi, per questo scopo ci sono gli articoli.

“TEMA” E “TEMPLATE”

Il template è il layout che verrà applicato al singolo elemento.

Su WordPress esiste un template diverso per ogni tipologia di contenuto, uno che si occuperà di visualizzare i post, uno per le pagine, uno per le categorie, uno per i risultati di ricerca e così via.

Il tema invece, è l'insieme di tutti questi template.

I files principali che compongono un tema sono due, index.php e style.css

Gli altri template rispettano una precisa gerarchia:

http://codex.wordpress.org/images/1/18/Template_Hierarchy.png

GERARCHIA DEI TEMPLATE

HOMEPAGE

- 1 front-page.php
- 2 home.php
- 3 index.php

Post

- 1 single-[post-type].php
- 2 single.php
- 3 index.php

Page

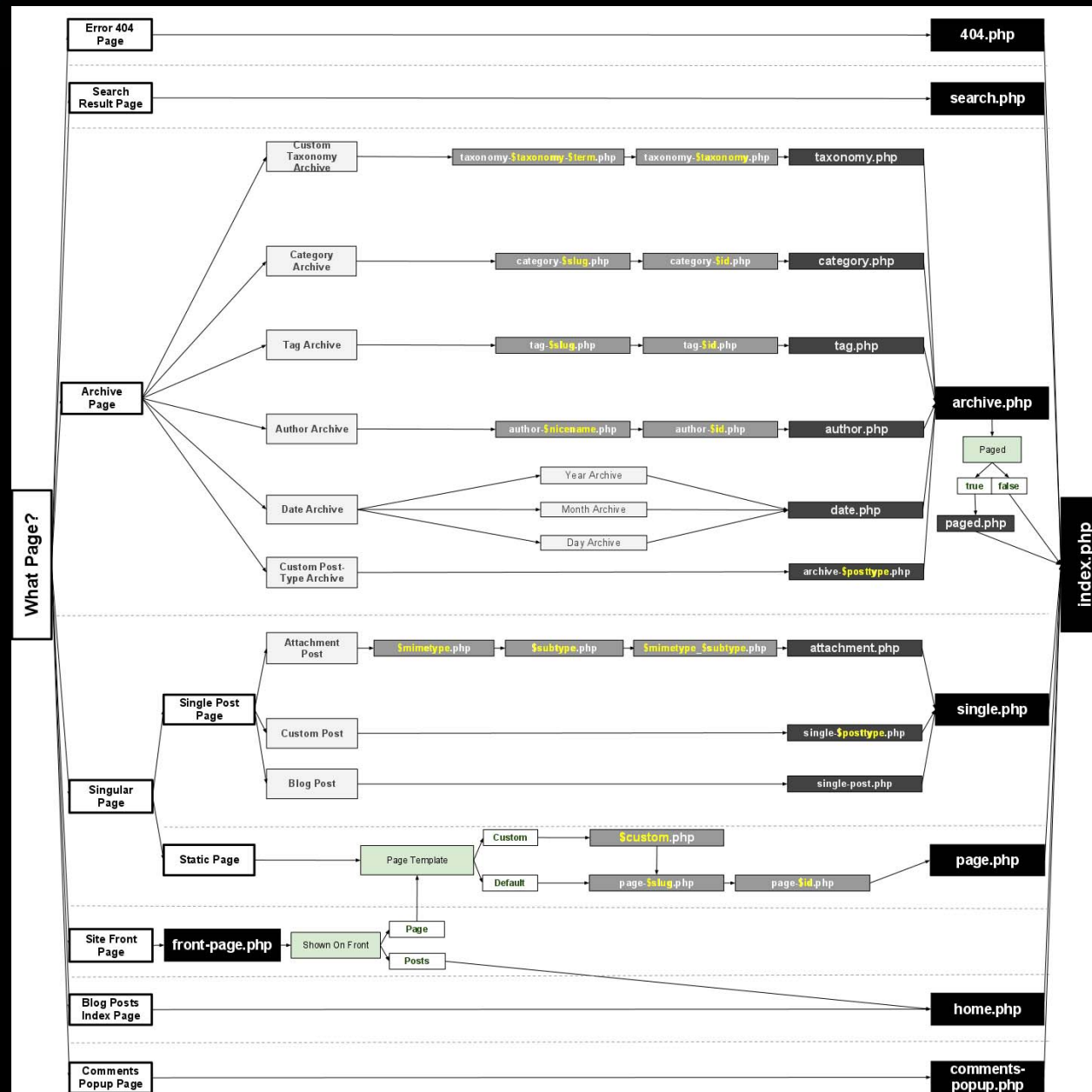
- 1 [custom-template].php
- 2 page-[slug].php
- 3 page-[id].php
- 4 page.php
- 5 index.php

Category

- 1 category-[slug].php
- 2 category-[id].php
- 3 category.php
- 4 archive.php
- 6 index.php

Tag Page

- 1 tag-[slug].php
- 2 tag-[id].php
- 3 tag.php
- 4 archive.php
- 5 index.php



FILES FONDAMENTALI PER IL FUNZIONAMENTO DEL TEMA

La cartella in cui WordPress va a memorizzare i temi è wp-content/themes/

1. index.php , il file più importante, senza di esso il nostro tema non funzionerebbe, nè verrebbe riconosciuto da WordPress.
2. header.php , il file dove caricare la parte alta del tema, oltre che il richiamo ai fogli di stile, file javascript, tags di apertura etc.
3. single.php , il template che si occuperà di visualizzare i post.
4. page.php , il template che si occuperà di visualizzare le pagine.
5. comments.php , il template che si occuperà di visualizzare i commenti ma, può anche non essere aggiunto.
6. archive.php , il template che si occuperà di visualizzare le categorie.
7. search.php , il template che si occuperà di visualizzare i risultati di ricerca.
8. sidebar.php , la barra laterale del nostro tema.
9. 404.php , pagina di errore, nel caso la pagina richiesta non esista.
10. footer.php , files dove inserire la parte finale del nostro tema, oltre ai tags di chiusura.

TEMPLATE PERSONALIZZATO

E' indispensabile un collegamento ftp alle cartelle del sito WP

E' possibile assegnare per ogni pagina, un template diverso, ovvero una struttura della pagina diversa:

Per es.

- copio il file page.php del tema e lo rinomino in page-nome-template.php
- poi apro il file ed inserisco il nome del template, sostituendo il vecchio nome del template con questo nuovo:

```
<?php
/*
Template Name: nome nuovo template
*/
?>
```

Ora per ogni pagina verrà mostrata la possibilità di scegliere anche il nuovo template

Al momento il nuovo template è uguale al template originale della pagina, ma adesso possiamo creare e nostre modifiche al codice php e HTML del file appena creato.

TEMPLATE PERSONALIZZATO PER UNA CATEGORIA DI ARTICOLI

- copiamo il file archive.php e rinominiamolo category-7.php

Vogliamo, per es. personalizzarlo con l'inserimento di una sidebar e quindi

- includiamo nel template di category-7.php, dopo il richiamo al loop, o comunque nel punto desiderato, il richiamo alla nostra nuova sidebar:

```
<?php if( ! dynamic_sidebar( 'sinistrappuntamenti' ) ) :?>  
    <?php endif; ?>
```

Dentro la nuova sidebar, potremo richiamare widgets, ma per attivarla dobbiamo prima registrare la nuova sidebar con la funzione register_sidebar(), all'interno di function.php:

```
register_sidebars(1, array(  
    'name' => 'colonna sinistra appuntamenti',  
    'id' => 'sinistrappuntamenti',  
    'before_widget' => '<div id="%1$s" class="widget %2$s">',  
    'after_widget' => '</div><div class="clear"></div>',  
    'before_title' => '<h3 class="widgettitle">',  
    'after_title' => '</h3>' )  
);
```

Template personalizzato: aggiungere una sidebar- area widget al nostro tema.

- creiamo un nuovo file sidebar-right.php
 - includiamolo nel template che ci interessa, per es. *nome nuovo template.php*, dopo il richiamo al loop, con la funzione `get_sidebar()` :

per richiamarla nel punto desiderato, quindi in *nome nuovo template.php*, aggiungiamo

```
<?php if( ! dynamic_sidebar( 'sidebar-right' ) ) :?>
<?php endif; ?>
```

- all'interno del nuovo file, sidebar-right.php, scriviamo, per es.:

```
<div id="sidebar-right-area">
    <h2>Cerca nel sito</h2>
    <?php get_search_form(); ?>
</div>
<div class="clear"></div>
```

- Registriamo la nostra nuova area widget nel file function.php:

```
if (function_exists('register_sidebar')) {
    register_sidebar(array(
        'name' => 'sidebar-right',
        'id' => 'sidebar-right-area',
        'description' => 'La nostra widget area.',
        'before_widget' => '<div class="widget-box">',
        'after_widget' => '</div>',
        'before_title' => '<h2>',
        'after_title' => '</h2>'
    ));
}
```

Utility e guide utili per wordpress

<http://codex.wordpress.org/> Guida Ufficiale WP

<http://www.html.it/guide/guida-wordpress/> Ottima guida in italiano

<http://www.robertoiacono.it/> Ottima guida in italiano

<http://www.w3schools.com/cssref/default.asp> CSS