WordPress

The most powerful blogging and website content management system

By: Hany Sayed

Agenda

What is CMS ?

Advantage of CMS ?

Example of CMS ?

What is wordpress ?

Why using wordpress ?

Installation and configuration

How it work ?

Theme

Hooks

Conditional Tag

Plug-in

Database

WordPress

The most powerful blogging and website content management system

What is CMS ?

Content management system:

is software application or set of related programs that are used to create and manage digital content

What is the advantage of CMS ?

- SEO-friendly URLs
- Integrated and online help,
- including discussion boards,
- Group-based permission systems,
- Full template support and customizable templates,
- Easy wizard-based install and versioning procedures,
- Admin panel with multiple language support

Example of CMS

- Joomla
- Wordpress
- Drupal
- Backdrop CMS
- Pulse CMS

What is WordPress ?

WordPress is an online, open source website creation tool written in PHP. it's probably the easiest and most powerful blogging and website content management system (or CMS) in existence today.

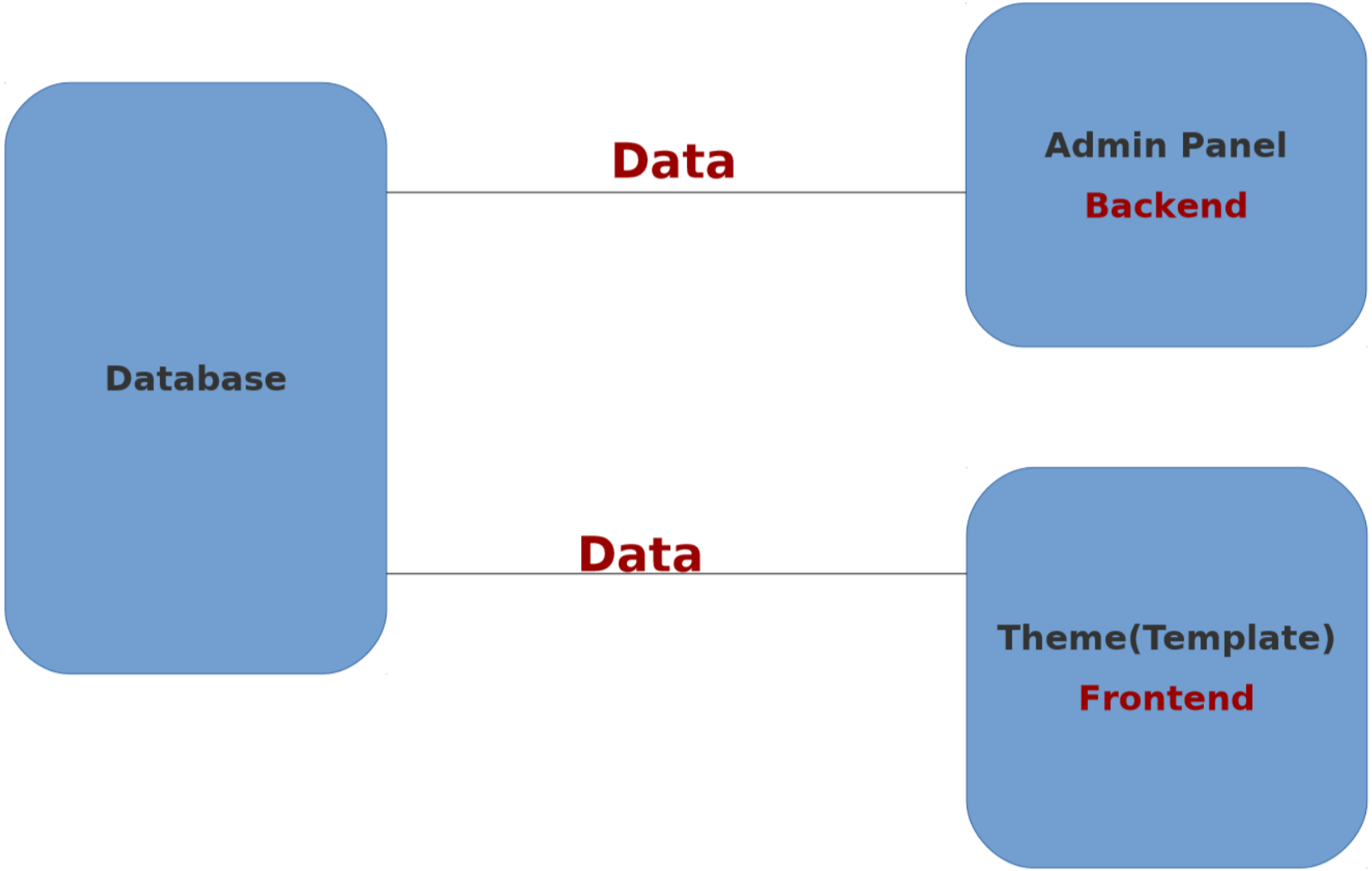
Why using wordpress ?

Free open source script Easy to Use and Learn Extendable by Using Themes and Plugins Search Engine Friendly Easy To Manage WordPress Can Handle Different Media Types

Installation and configuration

- Download script from *https://wordpress.org/download/*
- Then extract compressed file and move folder into your local web server (apache) Document Root access script from your browser *http://localhost/wordpress*
- After configuration wp-config.php file will be created under wordpress folder

How it work ?


Database
Data
Admin Panel
Backend
Data
Theme(Template)
Frontend

How it work ? Cont.

Admin Panel

Access admin from your_wordpress_site_url?wp-admin.php

1- CRUD operation on posts
2- CRUD operation on categories
3- CRUD operation on users
4- CRUD operation on comments
5- manage themes
6- manage plug-ins
7- manage menus

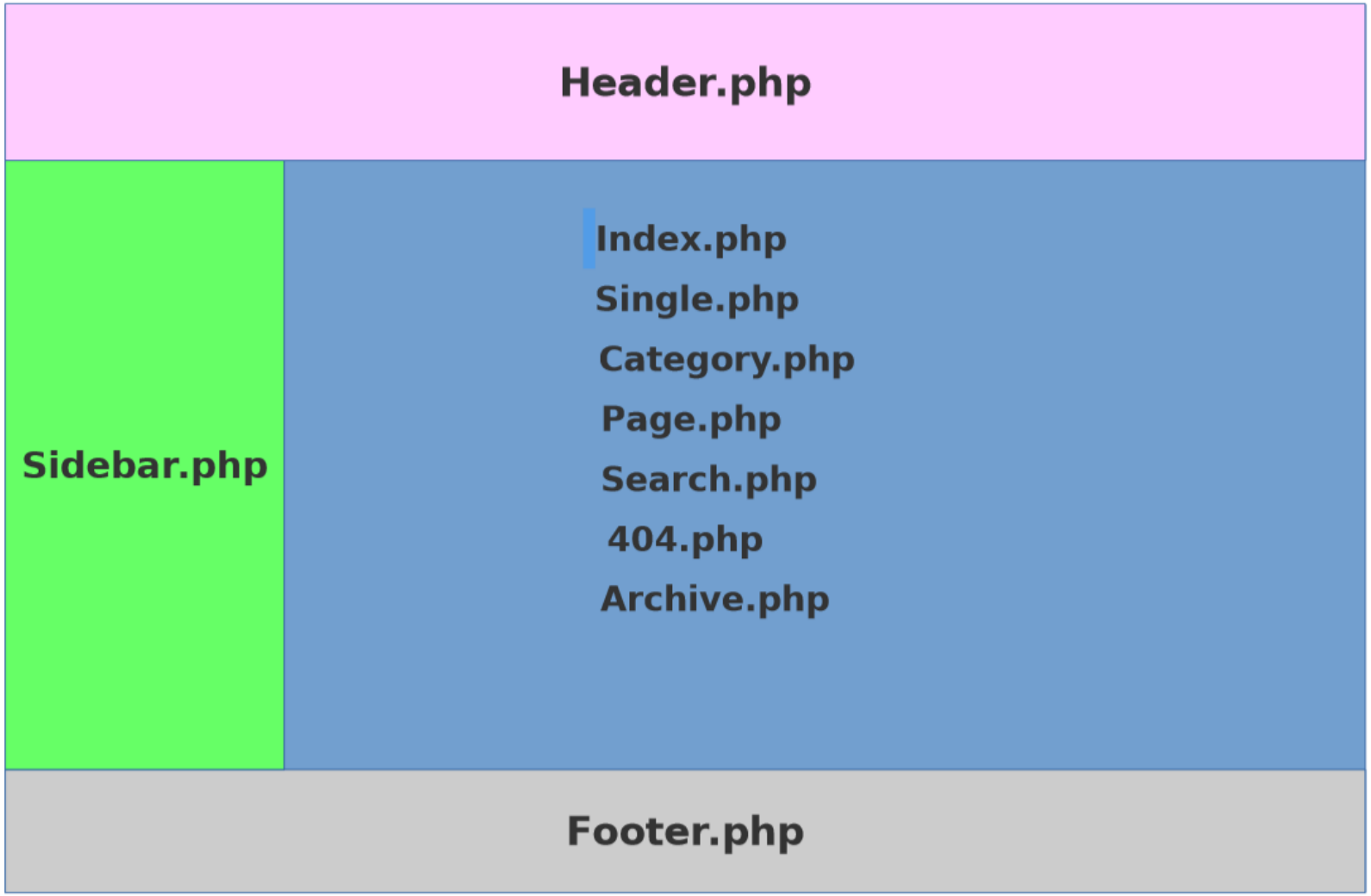
Admin side is for users who have admin role and other roles

Themes

Themes(Templates) frontend ,
By default you can find three parent themes in wp-content/themes folder .
You can extends one of them and create child theme or create your own theme from scratch

Themes Cont.

Build your theme



Themes Cont.

Build your theme
Create new folder in wp-content/themes and create these files
style.css
header.php
footer.php
sidebar.php
index.php
single.php
category.php
search.php
page.php
archive.php
404.php
comments.php
functions.php

Themes Cont.

style.css

files used for declare and configure your theme by these lines

```
 /*
Theme Name: First Custom Theme
Author: ITI Theme
URI: http://iti.gov.eg
Description: Custom Theme
Version: 1.0
Author URI: http://example.com
 */
```

Style.css files used also for styling

Themes Cont.

header.php

file used for put header html tags which will be included in each page,
do not forget to call **wp_head()** wordpress build in function before close </head> tag
Do not forget to set path for included css files and javascript files correctly by using **bloginfo('url')** function before any path **wp_head()** function used for fire wp_head wordpress **hook**,
What are wordpress hooks ?

Themes Cont.

footer.php

file used for put footer html tags which will be included in each page,
do not forget to call **wp_footer()** wordpress build in function
before close </body> tag
wp_footer() function used for fire **wp_footer** wordpress **hook**,

Themes Cont.

sidebar.php

file used for put sidebar html tags which will be included in each page

Themes Cont.

index.php

file act as home page and it consist of

```
<?php
	// wordpress built in function
	get_header(); // will call and include header.php file
?>
 <?php
	// between header and footer we need to get posts from database and display posts in our theme
?>
<?php
	// wordpress built in function
	get_sidebar(); // will call and include sidebar.php file
	get_footer(); // will call and include footer.php file
 ?>
```

Themes Cont.

Retrieve Posts

To retrieve posts from database there are several ways First, create instance from WP_QUERY class and pass args in constructor WP_QUERY class reference https://codex.wordpress.org/Class_Reference/WP_Query

```
$query=new WP_QUERY("post_type=post&showposts=5&orderby=last");
$query = new WP_Query( array( 'cat' => '-12,-34,-56' ) );
$query = new WP_Query( array( 'category__not_in' => array( 2, 6 ) ) );
$query = new WP_Query( array( 'category_name' => 'staff' ) );
$query = new WP_Query( array( 'author' => 123 ) );
```

Themes Cont.

Retrieve Posts Cont.

```
 $query=new WP_QUERY("post_type=post&showposts=5&orderby=last");
while ($query->have_posts()):
        $query->the_post();
        // to print post title
        the_title();
        // to print post creation time
        the_time('time_formate');
        // to print post author name
        the_author();
        // to print post categories
        the_category(',');
        // to display post image
        the_post_thumbnail('slider');
        // to print post content
        the_content();
        // to print post url
        the_permalink()
endwhile;
```

To create pagination linkes

```
 <?php the_posts_pagination(); ?>
```

Themes Cont.

Single.php

file used to display post details to access single.php your_wordpress_url?p=post_id

```
the_post();
// to print post title
the_title();
// to print post creation time
the_time('time_formate');
// to print post author name
the_author();
// to print post categories
the_category(',');
// to display post image
the_post_thumbnail('slider');
// to print post content
the_content();
// to display comment template
 comments_template();
```

Themes Cont.

category.php

file used to display post under certain category to access category.php
your_wordpress_url?cat=cat_id

To display categroy name single_cat_title()

```
while (have_posts()):
        the_post();
        // to print post title
        the_title();
        // to print post creation time
        the_time('time_formate');
        // to print post author name
        the_author();
        // to print post categories
        the_category(',');
        // to display post image
        the_post_thumbnail('slider');
        // to print post content
        the_content();
endwhile;
```

Themes Cont.

search.php

search.php file used to display search result for certain keyword. to access search.php , your_wordpress_url?s=keyword

To display keyword the_search_query()

```
while (have_posts()):
        the_post();
        // to print post title
        the_title();
        // to print post creation time
        the_time('time_formate');
        // to print post author name
        the_author();
        // to print post categories
        the_category(',');
        // to display post image
        the_post_thumbnail('slider');
        // to print post content
        the_content();
endwhile;
```

Themes Cont.

404.php

file used to display message to user when request page which not found

Themes Cont.

functions.php

file used to hold core functions in the theme. I can use it to set default actions before the page render such as,
Register menus,
Create image thumbnails ,
Add filter into the content ,
Do some Actions Hooks

Hooks

Hooks are simply a standardized way of ‘‘hooking’’ into WordPress.
Using hooks you can execute functions at specific times in the WordPress process, allowing you to alter how WordPress functions and the expected output.
Hooks are the primary way plugins interact with your content in WordPress

Two types of hooks can be used
Actions and **Filters**

Hooks | Action Hooks

Popular Action Hooks
publish_ post : Triggered when a new post is published.
create_category : Triggered when a new category is created
switch_theme : Triggered when you switch themes.
admin_head : Triggered in the <head> section of the admin dashboard.
wp_head : Triggered in the <head> section of your theme.
wp_footer : Triggered in the footer section of your theme usually directly before the </body> tag.
init : Triggered after WordPress has finished loading, but before any headers are sent. Good place to intercept $_GET and $_ POST HTML requests
admin_init : Same as init but only runs on admin dashboard pages.
user_register : Triggered when a new user is created.
comment_ post : Triggered when a new comment is created

Hooks | Action Hooks

Example

```
<?php
      function custom_css() {
?>
 <style type="text/css">
      a {
            font-size: 14px;
            color: #000000;
      }
</style>
<?php
       } add_action('wp_head', 'custom_css');
?>
```

Hooks | Filter Hooks

Filter hooks are used to modify WordPress content before saving it to the database or displaying it to the screen. For example,a Filter hook is available for the content of the post or page. This means you can alter that content after **it is retrieved from the database, but before it is displayed in your browser.**

```
<?php
	add_filter( $tag, $function_to_add, $priority, $accepted_args );
?>
```

Hooks | Filter Hooks

Popular Filter Hooks

the_content : Applied to the content of the post or page before displaying.
the_content_rss : Applied to the content of post or page for RSS inclusion.
the_title : Applied to the post or page title before displaying.
comment_text : Applied to the comment text before displaying.
wp_title : Applied to the page <title> before displaying.
get_categories : Applied to the category list generated by get_categories .
the_ permalink : Applied to the permalink URL.

Hooks | Filter Hooks

Example

```
<?php
    function custom_css($title)
        {
            return '<h1>'.$title.'</h1>‘
         }
    add_filter(’the_title’, ‘custom_css’);
?>
```

Conditional Tag

Functions return True | False
is_single();
is_home();
is_front_page();
is_category();
is_archive();
is_search();
is_404();
is_page();

Plug-in

1- Create folder with “your_plugin_name” under wp-content/plugins
2- Create your_plugin_name.php file and write these lines

```
/*
        Plugin Name: Open Source Plugin
        Plugin URI: http://iti.com/
        Description: this is first plugin
        Author: iti
        Author URI: http://iti.com/
        Text Domain: ITI PHP CMS COURSE
        Version: 1.0

*/
```

3- Activate your plugin admin panel / plugins then select your plugin and active it
4- Write some line in

```
// use action hook to call your plugin function ..
add_action('admin_menu', 'first_plugin');
function first_plugin() {
// add your plugin to dashboard
add_dashboard_page("title", "menue title", 'manage_options', __FILE__, 'your_func');
}
```

Database

To access wordpress database directly , use

```
<?php

        Function do_operations(){
                global $wpdb; $result=$wpdb->get_results('sql statement');
         }
?>
```

Thank You

E-mail Address :
Hanysayed.a@gmail.com