

SR DISTILLED OIL:

Date :- 3-07-2020

Date

Page

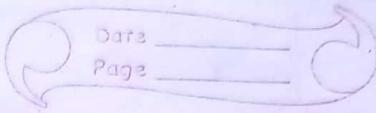
Name :- Gohil Jay K.

Sub :- Word Press

Roll no :- 3048

Class :- BCA Sem 3

Index



NO	Title	Pg.no	Date	sign
*	Unit - 2			
1	what is cms ?	1		
2	Introduction of wordPress	1		
3	features of wordPress	2		
4	Advantage & disadvantage of wordPress	3		
5	Requirement for wordPress	5		
6	Installation of wordPress	6		
7	Explain wordPress Dashboard	13		
8	wordPress Post vs Page	17		
9	wordPress.com vs wordPress.org	20		
10	wordPress setting	22		
11	what is widgets ?	30		
12	what is category ?	36		
13	Explain tag.	38		
Unit - 3				
14	Explain theme in wordPress	40		
15	How to make wordPress theme	42		
16	How to install wordPress theme.	46		
17	Explain wordPress Plugins.	56		
18	How to choose a plugin.	57		

Date _____
Page _____

19 HOW TO INSTALL WORDPRESS PLUGIN. 63

Unit - I

- 20 CONCEPT OF PHP 67
- 21 SYNTAX TO CREATE A CLASS PHP 67
- 22 EXPLAIN OBJECT WITH EXAMPLE 68
- 23 EXPLAIN CONSTRUCTURE FUNCTION 71
- 24 EXAMPLE OF PREDEFINED CONSTRUCTORS 73
- 25 PARAMETERIZED CONSTRUCTORS AND EXAMPLE 74
- 26 ADDITION BY FORM USING CONSTRUCTORS 75
- 27 CONCEPT OF MYSQL IN OOP 81
- 28 CONNECT DATABASE THROUGH TEXTBOX 82
- 29 CONNECTION CREATE EXAMPLE 89
- 30 CREATE THEME FOLDERS IN WORDPRESS (STEPS) 96
- 31 HOW TO CREATE CUSTOM THEME WITH EXAMPLE 106
- 32 ADD A FUNCTION.PHP FILES TO YOUR THEME 110

Q) What is CMS?

Ans: Content Management System.

Content: web data information

Management: insert, update, delete, roles.

System: Handle all system.

Q) What is Word Press?

→ Word Press is an open source content management system (CMS), which allows the user to build dynamic website and blog.

Note:- Word Press is PHP framework.

1) Word Press was initially released on 27th May, 2003 by Matt Mullenweg and Mike Little.

2) Word Press was announced as open source in October 2009.

(1) features of wordPess

- user management - it allows managing user information such as changing the role of the users to (subscriber, contributor, author, editor or administrator), create or delete the users, change the password and user information. The main role of the user manager is Authentication.

- Media Management - it is the tool for managing the media files and folders, in which you can easily upload, organize and manage the media files on your website.

- Theme System : it allows modifying the site view and functionality. it includes images, stylesheet, template files and custom pages.

- Extended with plugins - Several plugins are available which provides custom functions and features according to the users' need.

• Search Engine optimization - It provides several search engine optimization (SEO) tools which makes on site SEO simple.

• Multilingual - It allows translating the entire content into the language preferred by the user.

• Importers - It allows importing data in the form of Post. It imports custom files, comments, Post pages and tags.

* Advantage :-

(1) It is an open source platform, and available for free.

(2) CSS files can be modified according to the design as per user's need.

(3) There are many Plugins and templates available for free. Users can customize the various Plugins as per their need.

(4) It is very easy to edit the content as it uses WYSIWYG editor (What You See is What You Get) is a user interface that allows the user to

directly manipulate the layout of documents without having a layout command.)

5 Media files can be uploaded easily and quickly.

6 It offers several SEO tools which makes on-site SEO simple.

7 Customization is easy according to the user's needs.

8 It allows creating different roles for users for website such as admin, authors, editors and contributors.

* Disadvantages:

1 Using several Plugins can make the website heavy to load and run.

2 PHP Knowledge is required to make modification or changes in the WordPress website.

3

Sometimes software need to be updated to keep the WordPress up-to-date with the current browsers and mobile devices. Updating WordPress version leads to loss of data, so a backup copy of the website is required.

* System Requirements for WordPress

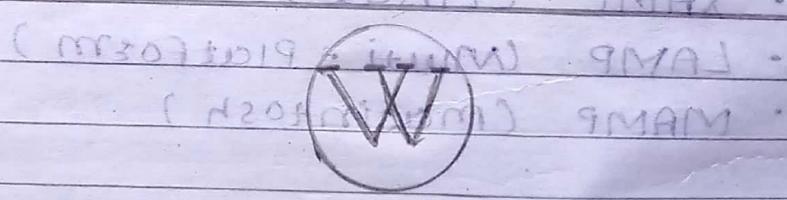
- Database - MySQL 5.0 +
- Web Server -
 - WAMP (Windows)
 - XAMP (Linux)
 - LAMP (Multi-Platform)
 - MAMP (Macintosh)
- Operating System - Cross-Platform
- Browser Support - IE (Internet Explorer 8+), Firefox, Google Chrome, Safari, Opera.
- PHP Compatibility - PHP 5.2 +

*

How to install WordPress?

Step 1 :- Extract the downloaded WordPress folder and upload it into your web server like localhost.

Step 2 :- Open your browser and navigate to your WordPress file path, then you will get the first screen of the WordPress installation as shown in the following screen. In our case, the path is localhost/localhost/wordpress.



English (United States)

Azərbaycan dilini

Bosanski

Català

Cymraeg

Dansk

English (UK)

Continue

→ Select your language for the WordPress and click continue.

STEP 3 :- In this step, you can view information needed for the database before proceeding with WordPress installation.



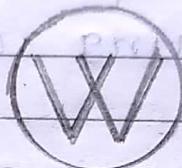
Welcome to WordPress. Before getting started, we need some information on the database.

1. Database name
2. Database username
3. Database Password
4. Database Host
5. Table prefix

Let's go!

Click on Let's go!

Step 4: Here, you have to enter the information about the MySQL database as described in the following screen.



Before you should enter your database connection details. If you're not sure about these, contact your host.

Database name

Username

Password

Database host

Table Prefix

- Database Name : Enter the database name which you have created in MySQL database for WordPress.
- Username : Enter the username of your MySQL database.
- Password : Enter the password which you had set for MySQL database.
- Database host : write the host name, by default it will be localhost.
- Table Prefix - It is used to add prefix in the database tables which helps to run multiple sites on the same database. It takes the default value.

After filling all information, click on submit button.

Step 5: WordPress checks the database setting and gives you the confirmation screen as shown in the following snapshot.



All right, SPARKY! You've made it through this part of the installation. WordPress can now communicate with your database, if you are ready.

Run the install



Click on Run the install.

Step 6 Enter administrative information



Welcome

Information needed

Site title : wordpressdemo

username : adminide

Password : *****

Strong

Your E-mail id : cladmin@gmail.com

Piracy : Allow search engines to

index this site.

Install WordPress

- Site Title - Enter the name of the site which you are going to create in WORDPRESS.

- Username - Enter the username as per your choice while logging in the WORDPRESS.

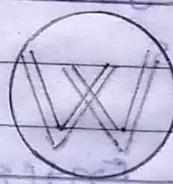
- Password twice - Enter password two times to protect your site

- Your e-mail - Enter your e-mail address which helps to recover the password or any update.

• Privacy - It allows the search engine to index this site after checking the checkbox.

→ After filling all the information, click on the install WordPress button.

Step 7 :- After installation being successful, you will get a screen of the stating success as seen in the following screen.



Success!

Wordpress has been installed.

username : admin

password : Your chosen password

LOG IN

You can view your Username and Password details added in WordPress.

Click on Log in button.

Step 8 :- After clicking on login, you will get a WordPress Admin Panel as depicted in the following screen.

* Explain WordPress Dashboard.

Ans:-

- WordPress Dashboard allows full access to manage a website behind the scenes.
- It is the default page in admin panel which comes right after the login page. It mainly represents every thing that is happening with your blog.
- All the content of dashboard is grouped into widgets. You can be hide / Show these widgets as per your requirement. Their positions can also be changed using drag and drop action.

- Dashboard Menu

The WordPress Dashboard provides navigation menu that contain some menu option such as Posts, media library, Pages, comments, appearance option, Plugins, users, tool and setting on the left side.

- Screen Options

The dashboard contain different types of widgets which can be shown or hidden on some screen, it contains check boxes to show or hide screen option and also allows us to customize section on the admin screen.

- Welcome

It includes the customize your site button which allows customizing your WordPress theme. The center column provides some of the useful links such as creating a blog Post, creating a page and view the front end of your website.

-> Last column contains links to widgets, menus, setting related to comments and also a link to the

first steps with WordPress. Page in the WordPress codex, website. Next column contain links to widgets, menus, setting related to comments and also a link to writing, editing, publishing, reviewing, printing.

AMP →

Quick Draft :

The Quick Draft is a mini post editor which allows creating, saving and publishing a post from admin dashboard. It includes the title for the draft, some notes about the draft and save it as a draft.

→

WordPress News :

The WordPress news widget display the latest news regarding the software etc. from the official WordPress blog.

→

Activity :

The Activity widget includes latest comments on your blog, recent post and recently published posts. It allows you to approve, disapprove, reply, edit, or delete a comment.

It also allows you to move a comment to spam.

- At a Glance :

This section gives an overview of your blog's Posts, numbers of published Posts and Pages, and numbers of Comments. When you click on these links, you will be taken to the respective screen. It displays the current version of running WordPress along with the currently running theme on the site.

IMP

Que are WordPress Post vs Pages

Post and Page appears to be same on the dashboard as both are used to add content on your site. People get confused in their usage.

- Pages are static. They don't change frequently. Some examples of pages are contact us, about us, introduction, etc.
- Posts are what you make an update on your post. They need to change very frequently as they need to keep your site fresh and attractive for your users.

Post

↳ →

Post use the site content which are published on a site with an exact date and time. They can be categorized in a systematic way on the basis of category and tags.

→

If you use a blog post, then you'll write posts to update your new blog. They are listed in a reverse chronological order on a site.

→

They generally consist of informational topics related to education, news, regular activities, feminism or any other topic. The content frequently changes for the end user as new posts are updated regularly. As the post gets older, a user has to dig inside to find it.

→

Posts have the commenting feature, which can be turned on or off.

It allows discussion on your blog if you have turned on the commenting.

- > As Posts are Published timely, they make up the RSS content of your blog. Your users will be notified about your most recent post via RSS feeds.

Pages

Pages are static and not changed frequently such as contact us, about us, privacy policy, etc. They don't have date and time published, however, database stores the published date or time of the page.

- > They are not included in RSS feeds as they don't have date and time published on the page.
- > They are neither listed by date nor can be categorized by the category and tags. They can make a hierarchy by nesting pages under each other making one of them the parent page.
- > They are not social, hence they don't have the commenting option.

*

WordPess.com vs WordPess.org

Both the site are part of WordPess
only difference is about "who is hosting
your website".

WordPess.com

WordPess.com is the site where
you don't have to do anything. All
the developing work will be handled
itself. You don't have to purchase
software, manage web servers or pay
for hosting.

These are some limitations with
this site. For example, by default
your domain name includes
WordPess.com, you can't modify
your site's coding, upload any
themes or Plugins. Your site will
be free for the lifetime, but you
have to pay to upgrade it for
more specifications.

- Upgrades also allows you to use a custom domain like you can use anything as the domain. Further upgradation also enhances appearance of your site, upload videos or photos and many more.
- Your site will be secured and spam free as it will be chose under WordPress security. You only have to sign in and choose your blog name and focus on your content.
- This site preferred by everyone but mostly bloggers, photographers, artist etc.
- [WORDPRESS.ORG](http://wordpress.org)
- [WORDPRESS.ORG](http://www.wordpress.org) is the site where you have to manage your own blog or website. You can find free WordPress software and install on your web server.
- It gives full control and freedom over your site including the WordPress software. Most of the WordPress showcase site are made on self hosting.

→ with full control if user places full responsibility of the site upon the user. From technical decisions to security issues etc and everything has to be managed by the user itself.

* WordPress Settings

* list out available setting in WordPress.

In this chapter, we will study about General setting in WordPress. WordPress general setting is used to set the basic configuration setting for your site. In the setting administration screen, it is a default setting screen.

→ Following are the steps to access the general settings.

→ Step 1 - click on setting → General option in WordPress.

Dashboard

Home

update (3)

 Post media Pages comments Appearance Plugins (3) Users Tools Settings General Writing Reading Discussion Media Permalinks Plugins Themes Translations Advanced

Dashboard

Welcome to WordPress!

We've assembled some links to get you started.

Get Started

Customize Your Site

or change your theme completely.

 1 page

→ Step 2 : The General setting page is displayed as shown in the following snapshot

-> Following are the details of the fields on general setting page.

- Site Title :- It displays the name of the site in the template header.
- Tagline :- Displays a short sentence about your site.
- WordPress Address (URL) :- It is the URL of WordPress directory where your core application files are present.
- Site Address (URL) :- Enter the site URL which you want your site to display on the browser.

E-mail Address - Enter your e-mail address which helps to recover your password or any update.

Membership - Anyone can register an account on your site after you check this checkbox.

New user Default Role - The default role is set for the newly registered user or member.

Timezone - Sets the time zone based on the particular city.

Date Format - Sets the date format as you need to display on the site.

Time Format - Sets the time format as you need to display on the site.

Week Starts On - Select the week day which you prefer to start for WordPress calendar. By default it is set as Monday.

Site Language - Sets the language for the WordPress dashboard.

→ Step 3 : After filling all the information about general settings, click on Save Changes button. It saves all your general setting to information.

*

The writing setting + screenshot

Using Gutenberg Block Editor

The writing setting controls the writing experience and provides option for customizing WordPress site. These setting control the features in the adding and editing Post, Pages, and post types, as well as the optional functions like Remote Publishing, Post via e-mail, and Update services.

→ Following are the steps to access the writing setting.

Step (1) - To change writing setting, go to Setting → writing option

Writing Setting

Formatting

Convert emoticons like :-)

and :-P to graphics on display

WordPress should correct invalidly nested XHTML automatically.

Default Post category Uncategorized ▾

Default Post Format Standarded ▾

Post via e-mail

To post to WordPress by e-mail you must set up a secret e-mail account with POP3 access. Any mail received at this address (random string) will be processed based on its content.

Mail server email.example.com • Port 110

Login name login@example.com

Password password

Default Mail category Uncategorized ▾

update services

Save changes

Following are the details of the fields on the page

- **Formatting :-** This field defines two sub option for better user experience
 - The first option convert emoticons to like :) and :-P to geographics on display will turn text-based emoticons into geographic-based emoticons.
 - The second option WordPress should correct invalidly nested XHTML automatically corrects the invalid XHTML placed within the posts or pages
- **Default Post category -** It is a category to be applied to a post and you can leave it as uncategorized.
- **Default Post format -** It is used by themes to select post format to be applied to a post or create different styles for different types of posts.

Post via e-mail - this option uses e-mail address to create posts and publishes post on your blog through e-mail. To use this, you'll need to set up a secret e-mail account with a POP3 access, and any mail received at this address will be posted.

- Mail Server - it allows sending the e-mails that you send to WordPress and stores them in the database. For this you need to have a POP3 compatible mail server and it will have URL address such as mail.example.com, which you should enter here.
- Login name - To create posts, WordPress will need its own e-mail account. The login name will use this e-mail address and should be kept as a secret as spammers will post links redirecting to their own websites.
- Password - set password for the above e-mail address.
- Default Mail category - it allows selecting custom category for all the posts that

use Published via Post by e-mail feature.

• Update services - when you publish a new post, WordPress will automatically notify the site update services in the box. See the update services on the Codex for the long list of possible services.

→ Step (3) - After filling all the above information, click on save changes button to see your information.

*

What is widgets?

Widgets are small blocks that perform specific function. These give design and structure control to the WordPress theme. Some specific features of a widget are -

- They help you add content and feature.
- They can be easily dragged and dropped in widgets area.

- They vary from theme to theme.
They are not same for every theme.

* WordPress Posts - What is Posts ?

Posts allows you to write a blog and post it on your site. They are listed in reverse chronological order in front page of your blog.

Home

Update post
→ media

Pages

Comments

Appearance

All Post

Add new

Categories

tags

Plugins (1)

Uses

How to adding new post

Go to Post → Add new

Title - Enter the title your post.

Content - Here you have to write Content of your post.

In visual mode, you can write in normal format.

In text mode, you can write in HTML format.

Preview - This button allows you to have a look on your post before publishing it.

* How to Publish Post :-

- After filling above two fields click on Publish button. It means you are ready to publish your site.

Add New post

hello

Publish



Add media

Save draft

Preview

B I abc ≡ iE GG - ≡ ≡ ≡ ≡ o. o.

status draft + Edit

You can use comment, post -
→ sophes, quotes, hyphens/
dashes and other typical
symbols in the Post like.

publish immediately Edit

Move to trash

Publish

Categories

All categories most used

blogs

clothing

Uncategorized

Look at the above snapshot, these
use some other option conceivable in
Publish section.

Preview Post / Preview - let you see a
Preview of your Post before Publishing
it.

Permalink - It stands for Permanent link. It contains a user-friendly Post name derived from Post title. This can be edited using "Edit" button.

Save Draft - A Post can be saved here.

Status - let you change the status of your Post to Pending and Draft.

Visibility - Change visibility to Public, Password Protected and Private.

Publish - Change the Post Publish date and time.

* How to Edit Post.

To edit post, go to Posts > All Post.

→ Look at the above Snapshot, this is the first list of all the created posts. Being your mouse above a Post, following options will appear.

- → LOOK at the above snapshot, there are options to edit. One is Edit and other one is Quick Edit.
- on clicking Edit, following page will appear.
- → LOOK at the above snapshot, here you can edit title or content of your post.
- on clicking Quick Edit, following page will appear,

disgarme

→ disgarme

- → Look at the above snapshot, here you can edit title, date and slug of your post. And if want, you can also change category for your post.

* How to deleting a Post:

To delete a Post, go to Post > All Posts

disgarme

→ Look at all the above snapshot, Trash option will let you delete the respective post.

* HOW TO PEEVIEW POST

Peeview option helps to see your post before publishing it to the public. It is better to have a look on your post that how it looks before publishing to the world.

* What is Category?

Categories allow a user to divide its content into different section.

Different topics publishing in a single website can be divided into different categories. It tells the reader what a post is about and they can easily find their content from a lot.

Name - Name of category which has to be unique.

Slug - It is used in the URL to display your category. For example, if your category is Clothing and slug is women, then your URL will be like shopping.com/shop/women.

Parent - Here, you can add sub-category to a category or you can choose none.

This is optional.

Description: Descriptions are optional. You can add a brief description to your category.

After filling all the above entries, click on Add New category button.

On the right side of the page your new category will be added.

* How to Delete Category

You can also delete categories from the above snapshot. By deleting a category, its post will not be deleted instead they will be transferred to by default category (if that post will not be listed under any other category).

→ Precise remember you can't delete a by default category before deleting it assign by default status to some other category.

* What is tag?

With the help of tags, similar posts can be grouped together. Hence, it makes easier for the users to search a particular post. It tells the user what a post is about. Tags are similar to categories but still different.

→ Position of the tag may differ depending on the theme. Sometimes they may appear at the bottom, top, or in the sidebar.

How to add tag

To add a tag, go to Posts > tags

Tags

Add new tag

Name

Slug

Description

Add tag

Look at the above Snapshot, here you can see how to fill these fields.

Name: Name of tag which must be unique. Here it is shopping destinations.

Slug: It describes a Post in URL. It is generally in lowercase and contain only letters, numbers and hyphens. Here we are using shopping.

Description: It is a brief intro about the tag. When you hover (bring pointer over the tag) the tag, this description is displayed.

→ After filling all the above information click on Add New Tag button.

* Explain theme in WordPress.

- with the help of WordPress themes You can design the layout and appearance of your website in the front-end.
- When WordPress is installed, it has a pre-installed theme which is very simple and unattractive. But no one wants a website to be that much simple. To make it more attractive one install and apply different themes from WordPress.
- Themes provide a style (including font styling), colors, page layout, widget position, etc. You can change the themes of your site without changing the content of the site.

Themes provides a unique look to your site.

→ TYPES OF WORDPRESS THEME

Themebase thousands of themes available on the wordpress. Some are free and some are paid. You can choose a theme according to your choice.

Free :

Wordpress provides many good looking themes absolutely free in the wordpress theme directory. If you are using your site for a small audience or yourself, go for free one. Because there are some chances that same theme is being used by another site.

Premium :

Premium themes may cost you starting from \$1,000. Price varies depending upon the design and requirement of your site.

One advantage in premium themes is that they provide a reliable code base and support, it gives you a more enhanced security.

*

How to make a WordPress theme?

Building a WordPress themes is somewhat technical work. To build a theme you should have a little knowledge about the following development languages.

- HTML

- CSS

- JavaScript

- PHP

- Database

You should know how to use a web server.

*

Selecting WordPress Theme

When selecting a WordPress theme should be selected very wisely, so that it perfectly fits on your site. You can select any one either paid or free. It should be complemented with your site.

- > Themes affect a website in a lot of ways. Heavy themes with 100K+ files may slow down your website and Google will degrade its ranking. If a theme is not coded properly, it becomes difficult to add any plugin or change that theme.
- > Theme Selection is very important as everything for your website depends upon theme.
- > Given below are some of the points which may help the users to select a theme for their website.

Look for simplicity.

Your website should look good, but for that you shouldn't compromise with your site's functionality. Theme styling should not be complicated. Complexity will make things difficult for your users. All you need is a good theme so don't go for an attractive one compromising your users.

Responsive

Your theme should be compatible with all type of screen size. A large number of traffic generates from devices other than laptops and PCs. Google also gives better ranking to the responsive sites. Lot of theme is by default responsive but still there are some theme which are not responsive. So look for a theme which is responsive to make your website accessible by all device users.

Because Support

make sure the theme you select is supported by all the browser selecting it, test it on different browser for mobiles as well as laptop like chrome, internet explorer, and firefox.

Plugin Support

Plugins are very important for a WordPress site. They enhance the functionality of your site, make sure that your selected theme must support all the popular plugins.

Multiingual SUPPORT

Your theme should support multilingual to support different language for users.

HELP OPTION

Only some developers provide support option with free theme, not all. So, if you messed up with your site then you may have to solve it on your own. So either than facing unauthorised problems, you should better choose a theme which has a good documentation and support help option.

SEO Friendliness

To increase your site performance on search engines, chose a theme which is SEO friendly. Many theme developers will let you know whether their theme is SEO friendly or not.

Ratings and Reviews

Before downloading a theme, have a look on its rating and reviews. A theme has both good and bad reviews, but if there are more negative reviews then read it carefully.

and consider it once again.

* How to install WordPress Themes?

For theme installation, first you need to select a theme either free or a premium one. There are a lot of sites from where you can download a theme. Theme will be downloaded in zip format. The next process is installation process. Installation has some steps to be followed.

These are two methods from which you can install theme.

- Installing WordPress using admin theme search.
- Installing WordPress using upload method.

* Installing WordPress using admin theme search.

To install free theme from word press.org directory, login into your account and click **Appearance > Themes**

① my site + NEW

② Dashboard

Home

Update

Dashboard

I start

Post

Get started

media

start

Pages

Customize your site

Comment

start

Appearance

Theme

start

Plugins

Customize

start

users

widgets

start

tools

menus

start

Setting

Headers

start

Install Plugins

Look at the above snapshot, click on Theme option. Following page will appear in front of you.

Look at the above snapshot, click on Theme option, Following Page will appear in front of you.

Dashboded Theme Add new

Posts

media

Pages

Comments

Appearance

↓
Theme

category

widgets

menu's

Headers

Install Plugins

editors

Theme 1

Theme 2

Install

Install

Theme 3

Theme 4

Install

Install

click on add new button to Add new theme

My site 0 + New

Dashboard Add Themes Upload Theme

Post

media

Pages

Comments

Apearance

Theme

Customise

Widgets

menus

Headers

Install Plugin

Editors

⑯ featured

Popular Latest

favourites

featured filter

✓ Installed.

Look at the above snapshot, here you will seen an option for featured filters.

Appearance

Theme

Customize

Widgets

Menus

Headers

Plugins

Editor

Plugins

uses

Tools.

(15) Features popular letters + features all

APPLY FILTERS

Layout

Features subject

Grid layout Accessibility Bing

One column Buddypress E-commerce

LOOK at the above snapshot, here you can filter your search for themes by applying your choice for Layout, features and subject.

Dashbaord

POST

media

PAGES

comments

Theme

Customize

Widgets

menus

headers

Install Plugins

uses

⑯ Feature Popular *Feature Filter

APPLY filter ↴

clear

LOOK at the above snap shot , we have selected 4 filters Criterias , now click on button APPLY filters to apply the filters.

PAGES

Apearance

⑯ Feature Popular latest

Theme

coustomise

filtering by Headers widgets

widgets

menus

Install Plugins

BLOG Edit

USES

Theme

Details &

Preview

Tick [Install]

Look at the above snapshot, select a theme and take your mouse above it. click on Install button.

Pages

Comments

Appearance

Theme

Comments

Widgets

menus

Headers

Install Plugins

Edition

Plugins

Uses

Tools

Theme

Details &

Preview

Frame work

Mikva

[Activate]

Look at the above snapshot, select a theme and take your mouse above & click on Activate button, And you have successfully activated your WORDPRESS theme.

* **Install WORDPRESS using UPLOAD method:**

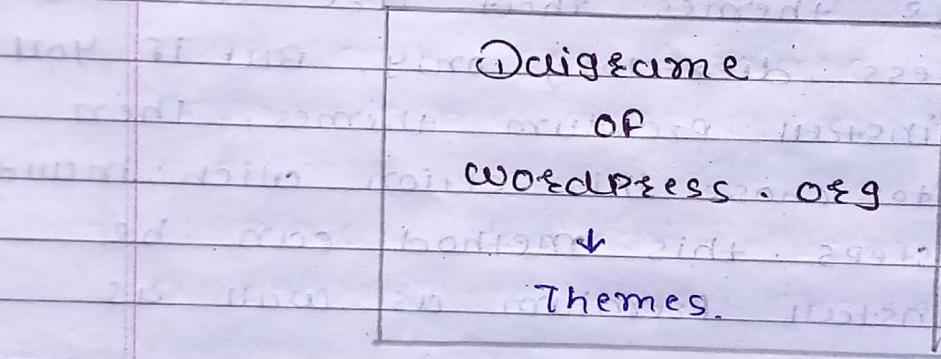
The above method was only to install free themes that are available in WORDPRESS directory only. But if you want to install premium themes, then You have to choose this option which includes following steps. This method can be used to install premium as well as free themes.

* **Downloading theme**

WORDPRESS site itself has a large collection on themes. All the themes on the WORDPRESS site is fully examined and tested. If you are not satisfied with these themes you can go to other sites to download a theme. Make sure you select trusted site to download a theme as anyone can create a theme for WORDPRESS.

- Themes are downloaded in the ZIP format.

Look at the above snapshot, this is the site. Here click on themes. You will be directed to the following page.

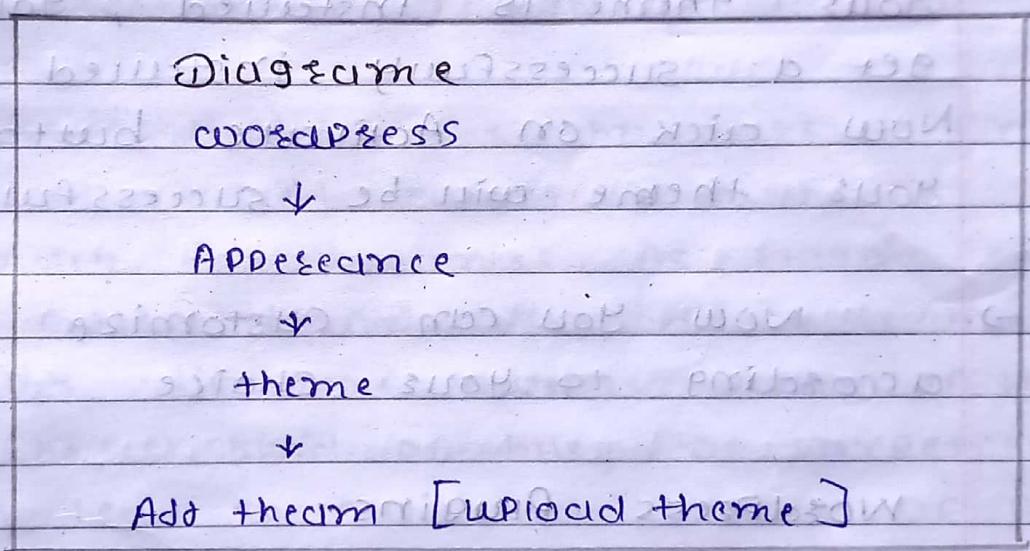


- LOOK at the above snapshot, on this page you'll have a lot of options to select themes.
 - Select a theme and then click on Download button.
 - Theme will be downloaded in zip format in your.

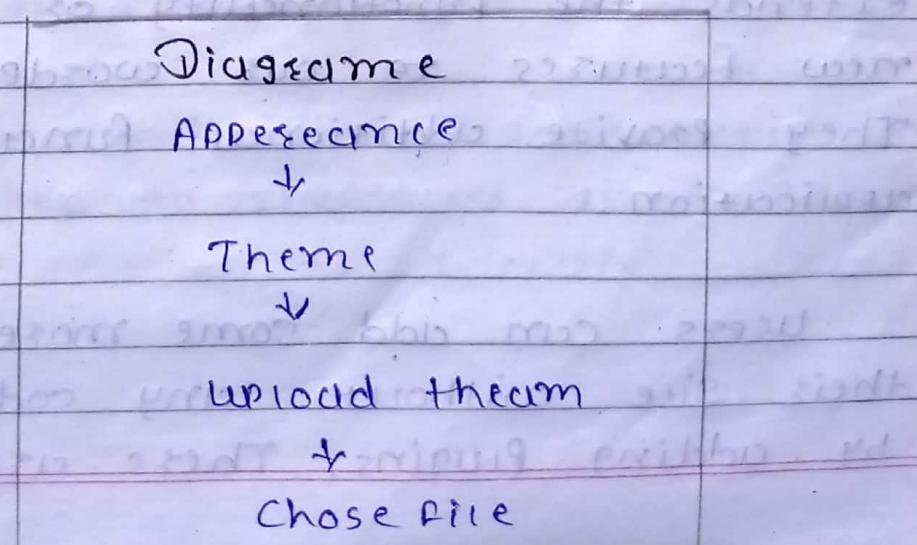
* Installing theme.

To install the downloaded theme, you have to login to your WordPress and click on Appearance > Theme.

Now click on Add new button, following screen will appear in front of you.



- Look at the above snapshot, click on **Upload theme** button.



- Look at the above snapshot, choose the file that you installed and click Install Now.
- Look at the above snapshot, once your theme is installed, you will get a successfully installed message. Now click on Activate button, and your theme will be successfully installed.
- Now you can customize it according to your choice.

* WordPress Plugins:

WordPress Plugins are programs written in PHP Scripting language that extends the functionality or add some new features to our WordPress site. They provide additional function to an application.

- Users can add some more function in their site without any coding skills by adding Plugins. There are thousands

of freely and commercially Plugins available. For each and every function there is a plugin available in the market.

You can also uninstall a plugin from your site.

* How to choose a plugin:

It is difficult for the users to find out the best suited plugin for them, especially for beginners. As plugins are open source hence every plugin is maintained by different owners. Some of them may be a kind of busy or inexperienced which lead to outdated version of plugin. According to your need you may have to install more than one plugin, but that's not a problem. You can install more than one plugin.

Here we'll show a list which may help you to choose a right plugin.



Make a list of your requirements.

First of all, you need to know what you want your site. Make a list of all the features you want to implement in your site. Remember to make decision making easier. It's not necessary that you'll find all of it in one plugin. Feel free to install more than one plugin.

Search Plugins:

It is recommended to search plugins through Google search rather than WordPress directory. WordPress.org search is not that much helpful.

Diagram:

Look at the above snapshot, this is WordPress directory search. It shows the search result by relevance which shows a lot of option. Use relevant word in your search otherwise you may not get the desired result.

Enter a search, for example we entered MCP as our search. It displays all MCP Plugins from the directory.

Select a plugin to see its description and information.

Compare them.

Choose 4 or 5 plugins and compare them by reading their description, description contains information about that plugin, why it is used, what are its functions, supported versions, etc.

Look at the snapshot, at the right hand side you can see what version it supports, when it was last updated, how many people have installed it and its ratings.

Reviews

60 reviews

Average Rating

5 stars



4 stars

3 stars

4.2 out of 5 stars

2 stars

1 star

LOOK at the above snapshot, this is the reviews of users with their settings. When a user rates a plugin, he is asked to give a review. To read this review, click on the rating star and the reviews with that star will be displayed.

SUPPORT OVERVIEW

SUPPORT

[View Support Forum](#)

SUPPORT gives an overview of numbers of SUPPORT threads opened for last 2 months. It also tells how many SUPPORT threads have been resolved. To view the details of SUPPORT threads, click on the SUPPORT tab.

Compatibility

Compatibility display the latest version of WordPress with version of plugin.

Compatibility

WordPress 4.6.1

+ Plugin 1.0.4

= Not enough data

Look at the above Snapshot, WordPress latest version is 4.6.1 and plugin current version is 1.0.4

Below the People's vote for the given combination of WordPress and plugin is displayed. If more number of people say it works, then the

number of people saying it is broken then this combination will work. Here no one has voted for this.

Screen shots

Plugin screenshots shows a preview of how the plugin will look like from front-end and back-end. Screenshots can be more helpful than plugin description.

FAQ

Frequently Asked Question help in a lot of way. Before start using a plugin in your website, go through all these questions. It will help you in the smoother use of plugin.

Testing Speed

A site's speed depends upon the plugin installed in it. Check your site's speed before and after installing a plugin. If speed slows down, then better watch for a new plugin.

* How to install WordPress Plugins

Installing Plugins will provide extra functionality to your site. To install a plugin, you just need to put the plugin file into your WordPress file. After installation, you may activate it or deactivate it.

These are three methods from which you can install plugins.

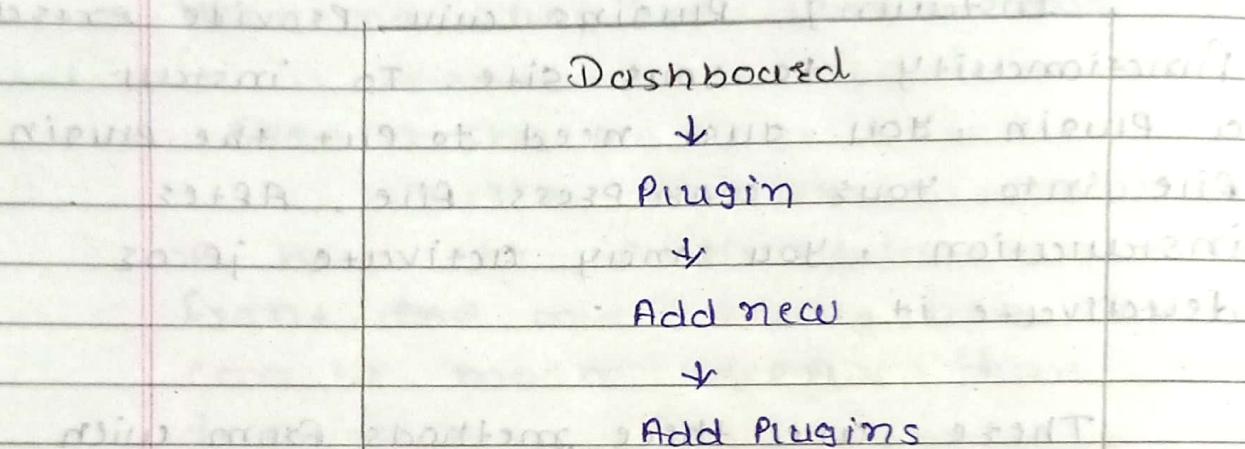
- Installing plugin using search
- Installing plugin using uploading
- Installing plugin manually using FTP

Here, we'll discuss all three methods.

Installing Plugin using Search

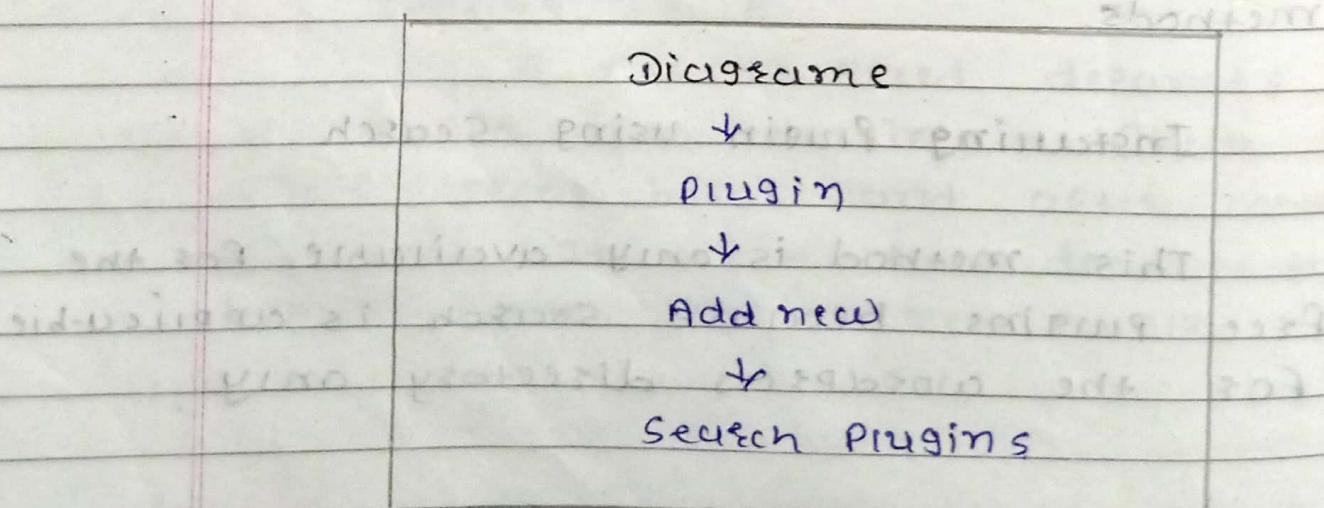
This method is only available for the free plugins because search is applicable for the WordPress directory only.

Login to your CloudPress and go to Plugins > Add New.



Look at the above snapshot, this page will be displayed. Here in the search option, you can give a plugin name the function name for which you want that plugin.

→ For example, we are searching for form plugin.



Look at the above snapshot, result related our search is displayed with a list of matching Plugins. we can select a plugin matching our criteria.

Once you have selected a plugin, click on Install Now button. Your plugin will be installed but to activate it you have to click on Activate button.

* Installing Plugin using uploading:

From this method Paid Plugin
~~file~~ can also be installed.

→ First of all, download a plugin file from the source which will be in zip format. Now goto Plugins > Add New Page.

Dashboard



Plugin



Add New



Upload plugin

(Select plugin zip file)

→

Look at the above snapshot, click on UPLOAD Plugin, it will bring you to the following page.

Plugin

Add new

UPLOAD PLUGIN

Choose file

Install now

→

Look at the above snapshot where you have to choose the downloaded plugin file and click on Install now.

Installing plugin from uploaded file:
bbpress.2.5.10.zip

Unpacking the package...

Installing the plugin...

Plugin install successfully.

Activate Plugin | Return to plugin install

→ Look at the above snapshot,
Plugin has successfully installed showing
a message. To activate this Plugin
you need to activate it.

* OOP in PHP (Object oriented programming)

* Syntax to execute class PHP

```
<?php  
class myclass  
{  
    // Class properties and method  
    go here.  
}  
?>
```

* Important note:

In PHP, to see the contents of
the class, use var_dump(). The var-
dump() function is used to display
the structured information about one
one or more variables.

* Object :

A class defines an individual instance of the data structure. we define a class once and then make many object that belong to it. objects are also known as an instance.

- An object is something that can perform a set of related activities.

Syntax

```
<?php
```

```
class myclass {
```

// class properties and methods
go here.

```
}
```

```
$obj = new myclass;
```

```
var_dump ($obj);
```

```
?>
```

① exce

```
<?php
```

```
class bca 3
```

```
{
```

```
private $a = "Hello BCA 3";
```

Public function shows()

```
{
```

```
echo $this->a;
```

```
}
```

```
}
```

```
$nm = new bca3(); // Create object
```

```
$nm->shows();
```

~~② var_dump(\$nm); // Show the structured information (type and value)~~

```
??
```

(2)

```
<?php
```

```
class bca3 {
```

```
{
```

```
public $name; // Set name
```

```
public $no;
```

```
function set-name($name) {
```

```
$this->name = $name;
```

```
}
```

```
function get_name () {  
    return $this->name;  
}
```

```
function set_no ($no) {  
    $this->no = $no;  
}
```

```
function get_no () {  
    return $this->no;  
}
```

```
$nm = new bca3();  
$nm->set_name ("KSC");  
$nm->set_no ("001");
```

```
echo "Name is - ". $nm->get_name();  
echo "<br>";  
echo "No is - ". $nm->get_no();
```

?>

* Constructor Function :

Constructor Functions are special type of functions which are called automatically whenever an object is created. So we take full advantage of this behaviour by initializing many things through constructor function.

PHP provides a special function called `construct()` to define a constructor. You can pass as many arguments you like into the constructor function.

- Following example will create one constructor for books class and it will initialize price and title for the book at the time of object creation.

function construct (\$price1, \$price2) {

\$this → title = \$price1;

\$this → price = \$price2;

}

Example

Date
Page 72

③

```
<?php
```

```
class A {
```

```
}
```

```
echo "this is demo of method of  
class A";
```

```
}
```

```
function A() {
```

```
echo "this is user define constructor  
of class A";
```

```
}
```

```
}
```

```
$data = new A();
```

```
$data -> checkA();
```

```
?>
```

Note: Here we have called checkA() method but we didn't call A() method because it automatically called when object is initialized.

→ Note: PHP constructors, if a class name and function name will be similar in that case function is known as

Constructors is special type of method because its name is similar to class name. Constructors automatically calls when object will be initializing.

Java 2013

Example of Predefined constructor.

<?php

(class) constructor predefined

class A {

function A() { self => echo

echo "this is use define constructor
demo"; }

}

function __construct() {

echo "this is demo of predefined
constructor of PHP";

}

}

\$date = new A();

?>

* Parameterized constructor example

<?PHP

```
class emp {  
    public $name;  
    public $no;
```

Function -- constructor (\$a, \$b)

{

```
$this -> name = $a;
```

```
$this -> no = $b;
```

echo "this is parameterized constructor
demo";

}

Function Show()

{

```
echo "<bed>";
```

```
echo $this -> name;
```

```
echo "<bed>";
```

```
echo $this -> no;
```

}

\$date = new emp("KSC", "123"); // assign value
in parameter.
\$date -> show(); // get or display value
which is assign.

* Addition by form using Constructors

<?php

class addition {

public \$a;

public \$b;

function __construct(\$x, \$y) {

\$this->a = \$x;

\$this->b = \$y;

}

function add() {

\$add = \$this->a + \$this->b;

echo "Sum of two textbox is :=". \$add;

}

\$data = new addition(\$_POST["a"], \$_POST["b"]);

\$data->add();

?>

<form method="post">

enter your first number <input type="text"

name="a"/>

enter your second number <input type
"text" name = "b" />

<input type = "submit" value = "+" />

</form>

* addition and subtraction

<?php

class addition {

 Public \$a ;

 Public \$b ;

 function __construct (\$x, \$y) {

 \$this -> a = \$x ;

 \$this -> b = \$y ;

 function add() {

 \$add = \$this -> a + \$this -> b ;

 echo "sum of two textbox is : ". \$add

}

 function sub() {

 \$sub = \$this -> a - \$this -> b ;

 echo "subtraction of two textbox is : ". \$sub

}

}

```
$data = new additions($_POST["a"], $_POST["b"]);
```

```
if (isset($_POST["add"])) {  
    $data->add();  
}  
else {  
    $data->sub();  
}  
??>
```

```
<form method="post">
```

```
    enter your first number <input type="text"  
        name="a" /> <br/>
```

```
    enter your second number <input type="text"  
        name="b" /> <br/>
```

```
    <input type="submit" value="+" />
```

```
</form>
```

* Addition, subtraction, multiplication and division example

<?php

class addition

{

Public \$a;

Public \$b;

function __construct(\$x, \$y)

{

\$this->a = \$x;

\$this->b = \$y;

}

function add()

{

\$add = \$this->a + \$this->b;

echo "sum of two textbox is = " . \$add . "
";

}

function sub()

{

\$sub = \$this->a - \$this->b;

echo "subtraction of two textbox is = " . \$sub . "
";

}

function multi()

{

\$multi = \$this->a * \$this->b;

echo "multiplication of two textbox
is = ". \$multi . "
";

}

Function division()

{

\$division = \$this->a / \$this->b;

echo "division of two textbox is = ".

\$division . "
";

}

}

\$data = new addition (\$_POST["a"], \$_POST["b"]);

if (isset(\$_POST["add"]))

{

\$data->add();

}

else if (isset(\$_POST["sub"]))

{

\$data->sub();

}

else if (isset(\$_POST["multi"]))

{

\$data->multi();

}

```
else if (isset($_POST["division"]))  
{  
    $data1 = division();  
}  
else if (isset($_POST["multi"]))  
{  
    $data1 = multiplication();  
}  
else if (isset($_POST["add"]))  
{  
    $data1 = addition();  
}  
else if (isset($_POST["sub"]))  
{  
    $data1 = subtraction();  
}  
else if (isset($_POST["division"]))  
{  
    $data1 = division();  
}  
  
<form method="post">  
    enter your first number <input type="text"  
        name="a"> <br>  
    enter your second number <input type="text"  
        name="b"> <br>  
    <input type="submit" name="add" value="+">  
    <input type="submit" name="sub" value="-">  
    <input type="submit" name="multi" value="*"/>  
    <input type="submit" name="division" value="/">  
  
</form>
```

OUT-PUT

multiplication of two textbox is = 200

enter your first number

enter your second number

| | | | |
|----------------------------------|----------------------------------|----------------------------------|----------------------------------|
| <input type="button" value="+"/> | <input type="button" value="-"/> | <input type="button" value="*"/> | <input type="button" value="/"/> |
|----------------------------------|----------------------------------|----------------------------------|----------------------------------|

* Most IMP

→ programme to connect mysql in OOP.

< ?PHP

```
class dbclass {
    var $host = "localhost";
    var $user = "root";
    var $pass = "";
    var $db = "ooptest";
```

Public function getconnection()

```
$con = mysqli_connect($this->host, $this->
    user, $this->pass, $this->db);
```

```
if ($con)
{
    echo "connection create";
}
else
{
    echo "not connected";
}

$cons = new dbclass();
$cons->getconnection();

?>
```

→ out-put

connection create

* Connect database - through text box

```
<!DOCTYPE html>
<html>
<head>
    <title></title>
</head>
<body>
```

```
<div style="background-color: silver;  
border: 2px solid maroon; width: 440px;  
height: auto; margin-left: 230px;">
```

```
<h1> Enter data for connection </h1>
```

```
<form method="post">
```

```
<div>
```

```
  <input type="text" name="txtHost"  
    placeholder="Enter host name" />
```

```
</div>
```

```
<div>
```

```
  <input type="text" name="txtUserName"  
    placeholder="Enter user name" />
```

```
</div>
```

```
<div>
```

```
  <input type="text" name="txtPass"  
    placeholder="Enter Pass name" />
```

```
</div>
```

```
<div>
```

```
  <input type="text" name="txtDB"  
    placeholder="Enter database name" />
```

```
</div>
```

```
<div>
```

```
  <input type="submit" name="sub"  
    value="connect" />
```

```
</div>
```

```
</form>
```

```
</div>
</body>
</html>
```

```
<?php
```

```
class dbclass
{
```

```
    public $host;
```

```
    public $user;
```

```
    public $pass;
```

```
    public $db;
```

```
    public function getConnection ($h, $u, $p,
```

```
{
```

```
        $this->host = $h;
```

```
        $this->user = $u;
```

```
        $this->pass = $p;
```

```
        $this->db = $d;
```

```
    $con = mysqli_connect ($this->host, $this->
    user, $this->pass, $this->db);
```

```
if ($con)
```

```
{
```

```
    echo "connection created";
```

```
}
```

```
else
```

```
{  
    echo "not connected";  
}  
}  
}  
$con = new dbclass();  
if (isset($_POST["sub"]))  
{  
    $con->getconnection($_POST["txthost"],  
    $_POST["txtuname"], $_POST["txtpass"],  
    $this->dbfix . $_POST["txtdb"]);  
}  
else  
{  
    echo "check again";  
}  
?>
```

* new example

```
<?PHP  
define ('HOST', 'localhost');  
define ('USER', 'root');  
define ('PASS', "");  
define ('DB', 'ooptest');
```

Class DB

```
{  
    function __construct()  
    {  
        $con = mysql_connect('HOST', 'USER',  
        'PASS') or die ('connection Error!'.  
        mysql_error());  
        mysql_select_db('DB', $con) or die  
        ('DB connection Error!'.mysql_error());  
    }  
}
```

Class USES

```
{  
    Public function __construct()  
    {  
        $db = new DB;  
    }  
  
    Public function register($fn, $name, $username, $email, $pass)  
    {  
        $pass = md5($pass);  
        $checkuse = mysql_query("select id from  
        users where email = '$email'");  
        $result = mysql_num_rows($checkuse);  
    }  
}
```

```
if ($result == 0)
{
    $register = mysqli_query ("insert into users
    (tgn_date, name, usename, email, password)
    values ('$tgn_date', '$name', '$usename',
    '$email', '$pass')") or die (mysqli_error());
    return $register;
}
else
{
    if ($tgn == "http://localhost/test")
        return false;
}
```

```
Public function login ($email, $pass) {
    $pass = md5 ($pass);
    $check = mysqli_query ("select * from
    users where email = '$email' and password
    = '$pass'");
    $data = mysqli_fetch_array ($check);
    $result = mysqli_num_rows ($check);
    if ($result == 1)
    {
        $SESSION ['login'] = true;
        $_SESSION ['id'] = $data ['id'];
    }
}
```

```
return true;
```

{

else

{

return false;

}

```
Public Function fname ($id)
```

{

```
$result = mysqli_query ("select * from  
users where id = '$id'");
```

```
$row = mysqli_fetch_array ($result);
```

```
echo $row ['name'];
```

}

```
Public Function session()
```

{

```
if (isset ($_SESSION ['login']))
```

{

```
return $_SESSION ['login'];
```

}

connection create Pro

<?php

define ('host', 'localhost');

define ('username', 'root');

define ('password', ''');

define ('db', 'looptest');

class DB

{

function __construct()

{

\$con = mysqli_connect(host, username,
password, db) or die ("Error". mysqli-
error (\$con));

if (\$con) {

{

echo "connected successfully";

// \$this -> mycon = \$con;

// return \$this -> mycon;

}

else

{

echo "Error";

}

}

```
// $dbcon = new DB();
```

```
?>
```

```
; ('root', 'root') or die  
(2) insert.php('inserted') or die  
; ('', '') or die  
<?php ; ('test900', 'db') or die
```

```
include_once 'connect.php';
```

```
class usesinsert
```

```
{
```

```
public function insert()
```

```
{ $dbcon = new DB(); }
```

```
$dbcon = new DB();
```

```
$inset = "insert into users values  
(NULL, 'exam', 'exam', 'exam@gmail', 112345)"
```

```
$res = mysqli_query($dbcon, $mycon, $inset)
```

```
if ($res)
```

```
{
```

```
echo "data inserted";
```

```
}
```

```
else
```

```
{
```

```
echo "error";
```

```
}
```

{
}

```
$user = new userinsert();  
$user->insert();
```

?>

③ getdata.php

<?php

```
require_once 'connect.php';
```

```
class getdata
```

{

```
public function getdatas()
```

{

```
$dbconn = new DBC();
```

```
$sql = "select * from users";
```

```
$res = mysqli_query($dbconn->mycon, $sql);
```

```
while ($row = mysqli_fetch_array($res)) {
```

{

```
echo "<table>";
```

```
echo "<tr>";
```

```
echo "<td>";
```

```
echo $row[1];
```

```
echo "<td>";  
echo "<td>";  
echo $row[3];  
echo "</td>";  
echo "<td>";  
echo "<td>";  
}  
}  
}
```

```
$data = new Getdata();  
$data -> getdata();  
?>
```

(3) getdata.php

```
<?php
```

```
require_once 'connect.php';
```

```
?>
```

```
<html>
```

```
<head>
```

```
<title> getdata </title>
```

```
</head>
```


<td><div href="delete.php?id=1">
<?php echo \$row[0]; ?>> Delete
</td>

Date _____
Page 94

<td>

<?php echo \$row[2]; ?>

<td>

<td>

<?php echo \$row[3]; ?>

<td>

<?php echo \$row[4]; ?>

<td>

<td>

<td>

<?php

}

}

\$data = new getdata();

\$data->getdates();

?>

<table>

<body>

<html>

4. delete.php

<?php

require_once 'define-cup2.php';

```
$dbconn = new DBC();  
  
if (isset($_GET["id"]))  
{  
    $id = $_GET["id"];  
    $q = " delete from users where id=$id ";  
    $result = mysqli_query ($dbconn->mycon, $q);  
    if ($result)  
    {  
        header ('location : getdetel.php');  
    }  
    else  
    {  
        echo "error";  
    }  
}  
?>
```

*

Step 1 : Create a folder to hold your theme files.

If we are going to be building themes, we need to know where the files that make up a WordPress theme live in a WordPress installation. This is pretty easy, we know that a WordPress installation typically has a root directory named WORDPRESS. Here is what our root directory looks like in PHP Storm.

Folders

- wp-admin
- wp-content
- wp-includes

The folder that we are most interested in right now is the wp-content folder. Within the wp-content folder is a folder named themes. Do you know what this folder is for? Yep, that's right it is the folder that holds one or more themes that you would like to use with your WordPress website.

In this themes folder we find three additional folders of twenty fifteen, twenty sixteen, and twenty seventeen. These folders contain the three themes that WordPress ships with by default. Notice below that there is also a folder name custom theme. Go ahead and create that folder as well. In your installation as this is where we will be creating our WordPress theme from scratch.

-> WORDPRESS FOLDER STRUCTURE

- ▼ WORDPRESS C:\xampp\htdocdev\WORDPRESS\WORDPRESS
 - ▶ wp-admin
 - ▼ wp-content
 - ▶ plugins
 - ▼ themes
 - customtheme
 - ▶ twenty fifteen
 - ▶ twenty seventeen
 - ▶ twenty sixteen

Custom theme

(1) style.css

/*

Theme Name : Eduinfo for school

Version : 1.0

Author : BCA 3 Team

Author URI : https://wordpress.org/

Description : we have created this theme
for school purpose to manage school
activity with each ui

Tags : Reach ui , school theme , three
column , two column , custom design

*/

Step 2 : Create style.css and index.php
in your custom theme folder.

We now know where WordPress theme
files are in the file system. We
also have created a new folder named
eduinfotheme in our themes folder.
We are now going to create two
empty files in this directory.

→ one is called index.php and the other is called style.css

File Edit View Navigate code Refactor Tools
WordPess > wp-content > themes > customtheme > index.php

| Project | index.php | style.css |
|--------------|--------------------|-----------|
| ► wordPess | index.php | style.css |
| ► wp-admin | | |
| ► wp-content | | |
| ► Plugins | | |
| ► theme | | |
| | custom theme | |
| index.php | ► twenty fifteen | |
| style.css | ► twenty seventeen | |
| | ► twenty sixteen | |

Let us now populate these files with the bare minimum we need to get a new theme going to in wordPess.

style.css

wordPess actually reads the comments that you place in the style.css file. This is where you specify specific information about the theme you are building.

The style.css is a stylesheet (css) file required for every WordPress theme. It controls the presentation (visual design and layout) of the website pages.

In our snippet here we simply assign a theme name, the Author & the Author's URI and the version number of our theme

/*

theme name: eduinfo

Author: BCA 3ITeam

Author's URI: https://vegibit.com

Version: 1.0

*/

index.php

In this file we just want to output something to the screen to prove that our custom theme is working.

Step 3 :- Activate your theme from the WordPress Dashboard.

At this point we can visit our WordPress Dashboard and navigate to Appearance → Theme and behold, we see the new theme.

We can click "theme details" to drill down on our custom theme and find that the information that we had entered into the style.css file has worked. We can see the theme has a name of customtheme, with version 1.0 by the author vegibit, and a link to the URI we had provided. Very cool!

Step 4 :- Add code to output the post title and post text.

We've take the liberty to populate a couple of example posts in the database so we can work with that data during this tutorial. Right now, our theme just outputs custom theme! to the page when we visit our site no matter how many posts are in the database. Let us now

move on to fetching some data from the database, and outputting it to the page. Specifically, we want to fetch the post title and post content of all posts and view them on the homepage. Let's give that a shot now.

First let's see what we have for Posts in the WordPress Dashboard

| Dashboard | Posts | Add New |
|--|-------------------------|--|
|  Posts | All (2) Published (2) | |
| All Posts | | <input type="checkbox"/> Title Author categories tags |
| Add New | | <input type="checkbox"/> PHP Tutorial blog post action PHP |
| Categories | | <input type="checkbox"/> WordPress Tutorial blog post |
| tags | | <input type="checkbox"/> Title Author categories Tags |
|  Media | | |
|  Pages | | |
|  Comments | | |

Leveraging the WordPress Loop

Now we can talk a little bit about the WordPress loop. The WordPress loop is basically the engine that makes WordPress run. It is

View this loop, that theme developers check for Posts and display them on the page as needed. If it does the format is follows. If the database has Posts, let's loop over them while there are still posts, otherwise we will let the user know there are no posts. It looks like this in PHP code.

```
<?php  
if ( have_posts() ):  
    while (have_posts()):  
        the_post(); ?>  
    <?php endwhile;  
else:  
    echo '<p>There are no posts! </p>'  
end if;  
?>
```

Notice that the loop makes use of two functions in its most basic form. Those are `have_posts()` and `the_post()`. The `have_posts()` function

does only one thing. It tells you if there are any posts in the database to loop over. This function will return.

either true or false and then is it. If it returns true then these are no posts to loop over. This function if it returns false, then there are no posts to loop over. The other function the `post()` does not return anything. Its job is to get WordPress ready to output posts. Specifically, if receives the next post, sets up the post, sets the in the loop `PROPERTY` to true. So far, our page will still not output any information about our blog posts, but we can update that now in our `index.php` file.

Step 5: Add a link to each post.

What about linking to each individual post so that we can view a post on its own either than as part of just the homepage? we can do

that quite easily, again with a specific function that WordPress provides. For this task, we can make use of the `permalink()` function. We can update our code like so:

```
<?php  
if(have_posts()): while(have_posts()): the_post();?  
    <h2><a href="php the_permalink()?"&gt;<br/        <?php the_title();?></a></h2>  
        <?php the_content();?>  
    <?php endwhile;  
  
else:  
    echo '<p>There are no posts!</p>';  
endif;  
?>
```

Now, we can click on each individual post title, and navigate to a page that has just that one post, very cool!

* How to custom theme with example

(1) style.css

/*

Theme Name : Eduinfo For school

Version : 1.0

Author : BCA 3 Team

Author URI : https://wordpress.org/

Description : We have create this theme
for school purpose to manage school
activities with each vi

Tags : Reach vii School theme, three
column, two column, custom design.

*/

body {

background-color : silver;

}

.91 {

border : 2px solid red;

background-color : yellow;

height : auto;

}

```
.clear {  
    clear: both;  
}
```

```
nav ul li {  
    display: flex;  
    float: right;  
}
```

```
nav ul li i {  
    list-style: none;  
    line-height: 30px;  
}
```

```
nav ul li a {  
    display: block;  
    text-decoration: none;  
    padding: 5px 20px 0px 0px;  
}
```

```
.main {  
    background-color: maroon;  
    border: 2px solid black;  
    height: 100%;  
    width: 100%;  
}
```

```
.div1 {  
    width: 30%; float: left;  
    height: 700px;  
    border: 3px solid green;  
}  
{
```

```
.div2 {  
    width: 69%;  
    height: 700px;  
    float: right;  
    border: 3px solid green;  
}
```

```
footer {  
    width: 100%;  
    border: 3px solid red;  
    text-align: center;  
}
```

(2) index.php

```
<?php header('Content-type: text/html');  
get_header(); ?>
```

```
<div class="main">  
  <div class="div1">  
    </div>  
  <div class="div2">  
    </div>  
  </div>  
<?php  
get_footer();
```

(3) headers.php

```
<html>
<head> <!-- Content -->
<title> </title> <!-- Body -->
<link rel="stylesheet" type="text/css"
      href="cup-content/themes/eduinfocustom/
style.css" />
<?php cup_head(); ?>
</head>
<body>
<header class="gj">
  <h2 style="float: left;">Mylog</h2>
<nav>
  <ul>
    <li><a href="#">Home </a></li>
    <li><a href="#">contactus </a></li>
  </ul>
</header>

```

```
<li><a href="#"> About us </a></li>
<li><a href="#"> Services </a></li>
</ul>
</nav>
<div class="header">
</div>
</header>
```

4. footer.php

```
<footer>
<h2> This is Footer Area </h2>
</footer>
<?php wp_footer(); ?>
</body>
</html>
```

* Add a functions.php file to your theme.

At this point, we have four files in our custom theme. Those are index.php, style.css, header.php, and footer.php. Probably the next most important file we need to have is the functions.php file.

The function.php file in WordPress does many things for your theme. It is the file where you place code to

modify the default behavior of WordPress. You can almost think of functions.php as a poem of a plugin for WordPress with a few key points to remember:

- Does not require unique header text
- Stored in the folder that holds your theme file
- Executes only when in the currently activated theme's directory.
- Applies only to the current theme
- Can call PHP functions, WordPress functions and custom function points

One thing we need badly in our theme is some better styling! Let's create a function in our functions.php file to include the style.css file into our theme. Here is how we can achieve that goal.

```
<?php  
function custom_theme_assets() {  
    wp_enqueue_style('style', get_stylesheet_uri());  
}  
}
```

add_action('wp_enqueue_scripts',
'custom_theme_assets');

-> This piece of code will include, or
make reactive, the stylesheet of our
custom theme. Now you might be
wondering why we are using a custom
function, when it seems like we could
just as easily manually link to
the stylesheet ourselves in the header.
PHP file. Well, this comes down to
doing a little more work up front
for a bigger return on your effort
lates. As themes get more complex
and more assets are added, you
will be happy to have this one
function that can handle all the
heavy lifting for you.

Now it's time to make things
look a little more pretty. First let's
add a wrapping `<div>` with a class
of `container`. The opening `<div>` will be
in `header.php`, while the closing `</div>`
will be in `footer.php`. We'll also
strip the post output in `index.php`.

with an <article> tag that has a class of post. This will give us classes to target in our style.css file so that we can set page width among other things. We'll also add some better styling to style.css in this step.

Explain the loop function in WordPress

(1) the_posts()

(2) have_posts()

(1) have_posts() :

Determines whether current WordPress query has posts to loop over.

Return

(bool) True if posts are available, false if end of the loop.

This function checks whether there are more posts available in the main WP Query object to loop over. It calls have_posts() method on the global \$wp_query object.

* General tags available in WordPress

1) get_header()

Syntax

```
<?php get_header($name); ?>
```

→ Parameters

\$name

(String) (optional) calls for header-name

Default: None

2) get_footer()

Syntax

```
<?php get_footer($name); ?>
```

Parameters

\$name

(String) (optional) calls for footer-

name.php

Default: None

3) get_sidebar (string \$name = null)

Parameters

\$name

(string) (optional) The name of the specialised sidebar

Syntax

```
<?PHP get_sidebar ('nice-bar'); ?>
```

4) get_search_form (bool \$echo = true)

6) wp_title();

wp_title (string \$sep = '&equo;', bool
\$display = true, string \$separation = '')

Syntax

```
<title><?PHP wp_title(); ?></title>
```

7) single_post_title()

Syntax

```
<?PHP single_post_title ($prefix, $display); ?>
```

8) comments - template();

Comments - template (string \$file = '/comments.php', bool \$separate_comments = false)

Imp

9) add-theme-support();

add-theme-support (string \$feature)

10) wp-footer();

Syntax : <?php wp_footer(); ?>

11) get-template-directory-vari();

Example

```

```

12) body-class()

body-class (string/array \$class = '')

Syntax

```
<body <?php body-class(); ?>>
```

13) wp-head()

<?php wp_head(); ?>

* Category tags

1) category-description()

Syntax

<?php echo category_description(\$category_id); ?>

2) single_cat_title()

Syntax

<?php single_cat_title(); ?>

3) the_category()

Syntax

<?php the_category('& separator', \$post_id); ?>

* Link tags :-

1) the_permalink();

Syntax :- <?PHP the_permalink(); ?>

As link with text +

<a href="<?PHP the_permalink(); ?>">

Permalink

2) get_permalink();

get_permalink (int \$post, \$post,
bool \$leavename = false)

✓ 3) home_url();

Syntax : <?PHP home_url (\$path, \$scheme); ?>

(4) get_home_url()

Syntax :

<?PHP echo get_home_url(); ?>

5) site_url()

Syntax

<?PHP site_url(\$path, \$scheme); ?>

Syntax - 2

<?PHP echo site_url(); ?>

6) get_site_url():

get_site_url(int \$blog_id = null, string
\$path = '', string \$scheme = null)

* Author tags function

1) the_author():

<?PHP the_author(); ?>

2) get_the_author():

<?PHP \$author = get_the_author(); ?>

3) the_author_link():

<?PHP the_author_link(); ?>

4) get_the_author_link():

<?PHP get_the_author_link(); ?>

5) the_author_meta():

<?php the_author_meta(\$field, \$use_id);?

6) the_author_posts():

<?php the_author_posts();?>

AMP

what are the type of hooks in WordPress
and mention their function?

There are two types of hooks

(1) Action hooks

(2) Filter hooks

How action hooks are used to create
WordPress theme or plugin with shortcode
without changing the original files.

Action hooks allow you to insert
additional code from an outside
resource, whereas filter hooks will
only allow you to add content
or text at the end of the post.

(2) what are the steps that you have to follow for WordPress plugin development?

- Create a unique name
- Create a the plugin's folder
- Create a sub folder of PHP files, translations, and assets.
- Create the main plugin file and fill and fill in the header information.
- Create activation and de-activation functions
- Create an uninstall script
- Create a readme.txt file
- To detect path to plugin file use proper constants and functions.