Twitter: @lighthouse_labs | Instagram: @lighthouselabs

<WordPress Workshop>

<configuration, structure & theming/>

LIGHTHOUSE LABS

<Prerequisites/>

MAMP Stack

Up until now, your stack was either a RAILS or a MERN one. For WordPress, Apache, MySQL, PHP are needed to make it work.

WordPress source

The project is declined in two versions. The .com version that is free and hosted by Automattic with limitations, and the .org version with the 'complete' experience.

The _s theme

Called the thousand hour head start, this theme helps us to build a look and a structure with all the basics already included.

<Intro/>

<What is WordPress ? />

<History/>

WordPress is a free and open-source CMS based on PHP & MySQL that was originally a fork of *b2/cafelog* in 2003. One it's original founders, Matt Mullenweg is still present as an advisor to shape its future.

<Usage/>

The usage of WordPress as a CMS on a project has been rising since it’s original release. As of the end of 2018, it is used on **32%** of the web.

This market share is cumulating from the blog only version (WordPress.com) and the self hosted version (WordPress.org)

<Milestones/>

- 2003 : Release of WordPress 0.7
- 2004 : Added support for plugins
- 2005 : Static pages & basic theme support
- 2013 : Initial REST API support
- 2018 : Block Editor (Gutenberg)

<Intro/>

<What does WordPress look like for a user ? />

LIGHTHOUSE LABS

<On the frontend/>

Probable that you already know since almost a third of the websites are built using WordPress.

The Obama Foundation, Toyota Motors Brasil, Angry Birds, thisisFINLAND, Canada.com, etc

<On the backend/>

Also known as the Dashboard for WP developers, this is where posts, pages, site & theme configuration are done for the website.

It can be WYSIWYG or coded.

<Intro/>

<I need to learn a new language... again ?! />

LIGHTHOUSE LABS

<HTML/>

As the building blocks of any website, it’s important to understand HTML and the properties of its elements. Since the pages are programmatically generated, you don’t actually need to type a lot of it.

<CSS/>

Since a lot of websites share a lot of basic behaviours, differentiation will be mostly done through styling.

CSS specialization and optimisation are interesting traits for frontend in agencies using WordPress.

<JavaScript/>

Adding interaction to a project is essential, and of course, JavaScript in the main tool to use.

It is important to know that by default jQuery is installed and activated in compatibility mode.

<PHP/>

Created in 1994, this language is the backbone of the WordPress CMS since it’s inception. This is one of the main complaints of a couple of programmers about the ecosystem.

As a server rendered language, themes, plugins and filters are all built using WordPress functions that are built in PHP.

<PHP />

<Let’s learn the basics />

LIGHTHOUSE LABS

<PHP integration/>

Since it was made to be integrated with HTML and generate HTML, the inclusion of PHP in a file always starts with `<?` or `<?php` and always closes with `?>`.

Example : `<?php echo "Hello World"; ?>`

<PHP variables/>

Basic variables are referenced using the $ sign to declare the value like `$bob = 4`.

Constants are defined differently since they use the define function like this `define('PI',3.1415926535)`.

<PHP functions/>

Declaring a function in PHP is the same as mosts languages :

```
function randomFunction(parameter, ..){
    instructions;
    return value;
}
```

<PHP conditions/>

Like functions, conditions and operators are like most languages. As of PHP7, the spaceship operator is valid (<=>)

```
if(condition){ … } else { … }
```

```
if(condition): … else: … endif;
```

<PHP loops/>

There is less looping options in PHP than in JavaScript, but the main ones are there.

```
for($i = 0; $i < 10; $i++){ … }
while(condition){ … }
do{ … } while(condition);
foreach($array as $value){ … }
```

<PHP output/>

Since the code is rendered on the server, output is limited compared to languages like JavaScript.

The main way to show content is to use

```
<?php echo “Something to show”; ?>
```

<PHP + WP Functions/>

Since WordPress generates a lot of functions for us to use in templates and plugins, they will be written in PHP and it's important to use proper PHP syntax.

In a theme environment, `the_title();` will show the title of the current page or post.

<WP Frontend/>

<Let’s show some content />

LIGHTHOUSE LABS

<Themes/>

Up until very recently, it was near impossible to show WordPress content without a theme.

They act like a canvas to show the user generated content and are built using PHP, HTML & CSS.

<Child Themes/>

Themes are interesting, but since it's more exciting to add functionality and tailor a theme to our needs, people started to modify basic themes.

The update process would revert those modifications, and the way to protect against this is the concept of a child theme.

<Builder Themes/>

With the rise of accessibility in website creation, a lot of designers started to make WordPress websites without knowing how to code.

These specific themes are built so it is possible to build a structure and add functionality without knowing PHP.

<Custom Themes/>

When a generic theme or a builder theme is not enough, it is possible to build a custom theme to tailor exactly to the client needs.

Most agencies will make custom themes as they are faster and more easily maintainable than builder themes.

<Headless Frontend/>

With newer technology stacks getting more and more popular, it is more frequent to see a WordPress installation without a proper theme.

The content is outputted through a REST API and is usable in a variety of languages and stacks.

<WP Backend/>

<Let’s organize our content />

LIGHTHOUSE LABS

<Post type/>

In the original release, it was only possible to generate posts since *b2* was a blogging platform.

This type is by default ordered by date, does not support hierarchy and supports categories.

<Page Post type/>

As the popularity of WordPress rose, people wanted more than a blogging platform, and thus, the page post type was born.

Pages are not ordered in any way, do not support categories, but will support hierarchy and multiple templates.

<Custom Post type/>

Most websites can be organized using pages and posts, categories and hierarchy.

However, it is possible to add custom post types to better organize your content. A CPT would be useful if it was needed to separate portfolio items from blog posts.

<WP Backend/>

<Let’s add extra functionality />

LIGHTHOUSE LABS

<Repository plugins/>

Out of the box, WordPress has a lot of functionality. Pretty soon in its development cycle, plugin support was added so that developers could customize the experience.

To support its ecosystem, WordPress validates plugins and add them to a repository for easy use.

<Custom plugins/>

Like with themes, sometimes what is proposed is not enough. It is then possible to create custom plugins to add specific functionality and concepts to your project.

They can range from modifying the look of the website to adding new ways to organize data.

<Headless WP/>

<Or how to use the WP Rest API />

LIGHTHOUSE LABS

<WP & JSON/>

As of WordPress 4.4, it is now possible to access the entire content (posts, pages, users, src) through the REST API.

The data is then available in a JSON format and can be used in a variety of cases.

<React/>

Since it can be considered as the View part of a MVC pattern, it's practical to feed a React project using data coming from a WordPress backend.

All the business logic of users, posts and pages are managed through WordPress, and React manages how we see it.

<Gatsby.js/>

A more specific examples of using React & WordPress, this static site generator takes data from a source, manages it internally using GraphQL and generate static pages and optimize images.

Since it’s static, it is very interesting for SEO conscious projects.

<Anything that reads JSON really />

React & Gatsby.js are not the only solutions to create apps and websites using a headless WordPress installation.

You could use Python, Ruby, C#, JavaScript, or your preferred programming language to read, parse and render the content from WordPress.

</ WordPress Workshop>

<Thanks ! />

LIGHTHOUSE LABS