



Fundamental notions of programming (NFP)

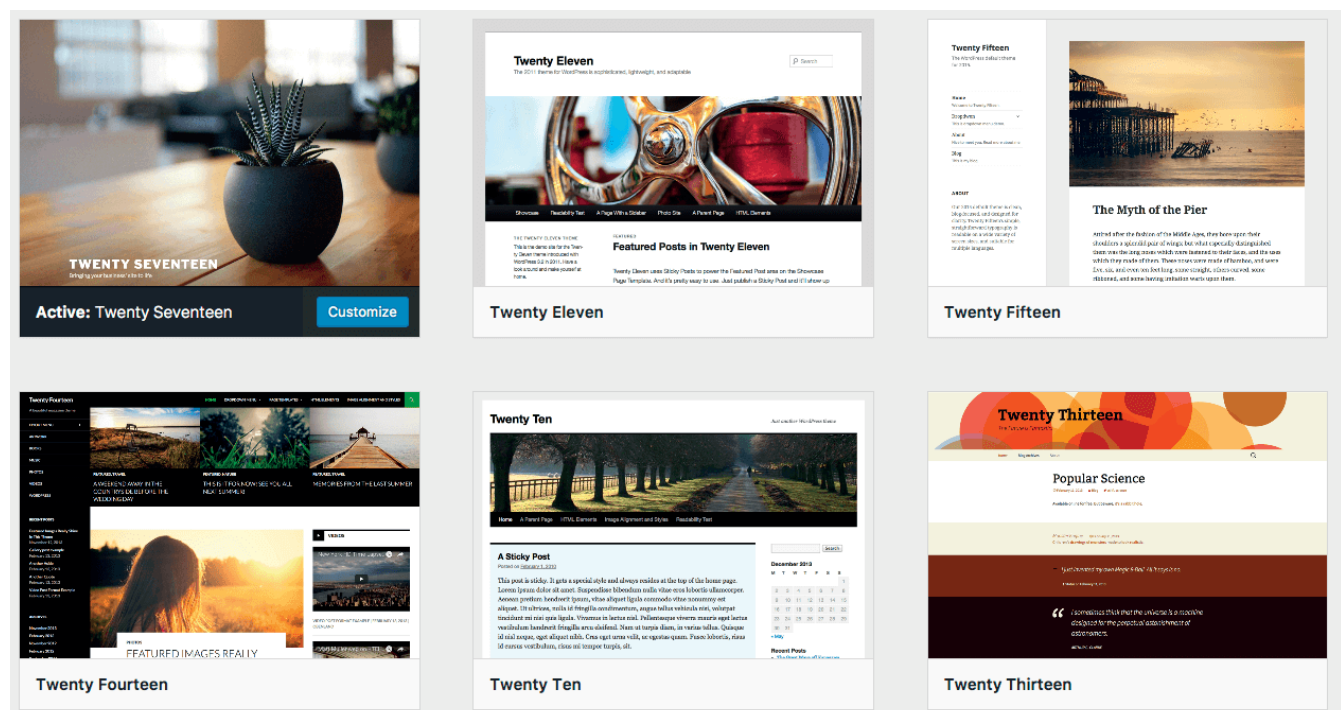
Class 13

Developing WordPress themes

Knowing HTML, CSS, and JavaScript make it possible for you to create excellent beautiful websites. You most probably heard about WordPress, you know what it is but you're not sure how to use it, or even why it might be needed. One thing is sure, this platform being so popular, one day or another, you will have to use it. A client may ask to use a specific theme, to adapt an existing theme or even to create a new them from scratch.

As priorly mentioned, even though WordPress was originally built as a blogging platform, it is now also considered a content management system (CMS). Any website that will need to be regularly updated can greatly benefit from a CMS. For instance, in the case of a blog, new posts will need to be added, a restaurant website might want to be able to update the menus, and an e-commerce could want to be able to update prices, add items, change delivery options, etc.

WordPress themes consist into files organized to work together in order to create the design and functionality of a WordPress site. Themes differs from each other, offering different choices of styles, layouts and functionalities. But, for various reason, you may want to develop your own WordPress themes. This can be for your own personal use, for a client's project or to submit to the WordPress theme Directory.



Reasons you may want to create your own WordPress theme :

- To benefit from a unique look.
- To take advantage of templates, tags, and the *WordPress Loop* to generate different results and looks.
- To provide alternative templates for specific site features (e.g. : pages categories, search result pages, etc).
- To be able to quickly switch between two site layouts easily.

A custom WordPress theme has many benefits

- It separates the presentation styles and template files from the system files (so the site upgrades without enormous changes to the visual presentation).
- It allows customization of the site's functionality specific to a given theme.
- It allows quick changes to the visual aspect and layout.
- It makes it possible for a site owner to manage it without having to learn CSS, HTML, and PHP.

Reasons to build your own WordPress theme

- It's an opportunity to learn more about CSS, HTML, and PHP.
- It's an opportunity to put to practice your expertise with CSS, HTML, and PHP.
- It's creative and challenging.
- It's fun (most of the time).
- Feeling of accomplishment for giving something back to the WordPress Community (Well, you will be able to brag about it).

Starter themes

Starter themes are standalone solutions designed to be modified. It is basically a toolkit, making it as easy as possible to get a design up and running without having to build the basic setup required to make a WordPress theme functional. For instance, after developing a HTML prototype, you could use a starter theme to transform it into a WordPress theme.

One of the most famous starter themes is *Underscores* (from Automatic's development team). It sets basic coding standards as a baseline so developers can start creating secure themes.

Underscores includes many features such as:

- Custom template tags for optimal theme performance.
- Built-in mobile navigation (through a JavaScript file).
- Sample CSS layouts.
- Plenty of documentation comments to guide your way.

Child themes

A child theme makes it possible to change specific aspects of a site's appearance while preserving the theme's design and functionality. To understand how it works, it is first important to understand the relationship between parent and child themes.

Parent theme

A parent theme is a set of required WordPress template files and assets making a theme to work. A child theme inherits the design and style of the parent theme as well as its functions, but it can be customized. Different aspects of the parent theme can be modified in the child theme and those are kept separate from the parent theme's files.

Child themes:

- make modifications portable and replicable;
- keep customization separate from parent theme files;
- allow parent themes to be updated without destroying the modifications;
- put to profits the effort and testing of the parent theme;
- reduce development time;
- consist in a great way to learn about theme development.

Important features and functions

The WordPress Loop

The WordPress loop is PHP code that is used in WordPress themes to display a list of posts in a web page. It is made of functions ran by default to display posts.

Template tags are used to format the output, to customize how each post inside the loop is displayed. Several template tags work only inside the WordPress loop and can be used to format, arrange, and publish post data. The loop is one of the most important aspects of the WordPress code and at the core of most queries.

When WordPress documentation specifies a tag must be placed within the Loop, it means that the tag will be repeated for each post. For instance, the following information are repeated by default for each post :

Title (the_title())
Time (the_time())
Categories (the_category()).

Example of the most simple index page :

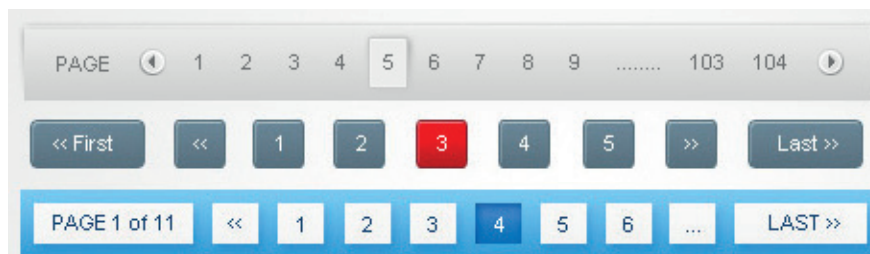
```
<?php
get_header();

if (have_posts()) :
    while (have_posts()) :
        the_post();
        the_content();
    endwhile;
endif;

get_sidebar();
get_footer();
?>
```

The Pagination

WordPress offers built-in navigational functionality. Simple links or numbered pagination can be used to indicate the previous page or the next page. WordPress can also split a single post, or a list of posts, into multiple pages for *paged* navigation. Number of posts to be listed on each page can be set on the **Reading screen** (*wp-admin > Settings > Reading*).



Wordpress security aspects

WordPress security is a very important subject. Malwares, spam and phishing are everywhere and many pirates are using security breaches in websites to do their wrongs. Google is regularly blacklisting websites for such reasons (over 10,000 websites daily for malware and around 50,000 weekly for phishing).

Anyone serious about their websites will need to pay attention to the WordPress security best practices in order to protect your website against hackers and malware.

Here are a few actions that can be taken to increase your WordPress website security:

- Change the Default «admin» username
- Disable File Editing
- Disable PHP File Execution
- Limit Login Attempts
- Add Two Factor Authentication
- Change WordPress Database Prefix
- Change WP-Admin URL
- Password Protect WP-Admin and Login (use your email as login)
- Automatically log out Idle Users
- Add Security Questions to WordPress Login
- Scanning WordPress for Malware and Vulnerabilities

Security Plugin

Some additional plugins can be install such as Sucuri Scanner (probably the most popular). It audits and monitors the website, keeping track on everything that happens and it includes file integrity monitoring, failed login attempts, malware scanning, etc.

Upon activation, go to the Sucuri menu in the WordPress admin. You will be asked to do is Generate a free API key enabling audit logging, integrity checking, email alerts, and other important features.

References / sources :

<https://www.taniarascia.com/developing-a-wordpress-theme-from-scratch/>
https://codex.wordpress.org/Theme_Development
<https://www.dreamhost.com/wordpress/guide-to-developing-a-wp-theme/>
<https://torquemag.io/2017/08/beginners-guide-to-creating-a-theme-underscores/>
<https://developer.wordpress.org/themes/advanced-topics/child-themes/>
<https://www.wpbeginner.com/glossary/loop/>
https://codex.wordpress.org/The_Loop
<https://codex.wordpress.org/Pagination>
<https://www.wpbeginner.com/wordpress-security/>