# Лабораторна робота №4 Динамічні веб – сторінки.

**Мета**: опанувати навички ділення HTML-сторінок на окремі блоки та генерація динамічних HTML сторінок з використанням мови програмування PHP.

**Теоретичні відомості**

Веб-сторінки можуть складатись із статичних текстових файлів, що зберігаються у файловій системі веб-сервера ([статичні веб-сторінки](http://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%87%D0%BD%D0%B0_%D0%B2%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0&action=edit&redlink=1)), або веб- сервер може створювати сторінки за запитом браузера ([динамічні веб-](http://uk.wikipedia.org/wiki/%D0%94%D0%B8%D0%BD%D0%B0%D0%BC%D1%96%D1%87%D0%BD%D0%B0_%D0%B2%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0) [сторінки](http://uk.wikipedia.org/wiki/%D0%94%D0%B8%D0%BD%D0%B0%D0%BC%D1%96%D1%87%D0%BD%D0%B0_%D0%B2%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0)).

Динамічна веб-сторінка ([англ.](http://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) dynamic web page) — [веб-сторінка](http://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0), вміст якої може змінюватись. Частини динамічних сторінок зберігаються як окремі файли (статичні частини, php-скрипти) на диску веб-сервера. За запитом клієнта сторінка збирається з набору цих файлів (рис. 4.1) – технологія SSI (Server-Side Includes - включення на стороні сервера).



Рисунок 4.1 – Принцип роботи SSI

В PHP існують ряд функції емуляції SSI: include, include\_once, require, require\_once.

Різниця між функціями include і require полягає в типі генеруються помилок підключення. Якщо підключений файл відсутній, то include видасть помилку рівня «Warning» (тобто код продовжить виконаються). А require – видасть помилку рівня «Fatal error» (тобто код завершує виконаються). Щоб помилки взагалі не виводяться, можна поставити перед ім'ям функції знак

«собаки» @ - вона гасить всі помилки, які сталися при виконанні коду. Так робити не треба, тому як @ дуже повільний оператор (краще використовувати функцію для керування відображенням помилок – error\_reporting).

Для запобігання помилок множинного включення файли підключають через include\_once або require\_once. Різниця між якими аналогічна попередньої парі функцій.

Поділ на файли найчастіше відповідає структурним елементам сторінок. Як правило, структура сторінки сайту має чітку організацію. В якості прикладу розглянемо структуру наведену на рис. 4.2 яка складається з:

1. шапка (header), до якої можуть входити логотип, назва, слоган, телефон, меню навігації;
2. основна частина (content) – меню навігації по тематичних розділах, основна інформація, зображення, банери;
3. нижня частина (footer) – копірайт, адреси, телефони, лічильники та банери, меню.



Рисунок 4.2 Структура оформлення сторінок сайту

Звичайно можливо різні варіанти та комбінації представленої структури.

Аналогічно діленню сторінок сайту на логічні елементи HTML-код сторінок може бути розбитий на відповідні блоки коду, збережений у вигляді php-скриптів і розкладені за призначенням в деревовидну структуру папок на веб-сервері. Приклад організації папок на сервері наведено на рис. 4.3.

 

Рисунок 4.3 – Структура папок і модель «складання» сторінки сайту з окремих файлів (php-скриптів).

**Завдання**

1. Обрати довільний шаблон сайту за визначеною з викладачем тематикою.
2. Провести аналіз структурних елементів шаблону, виділити його загальні та характерні блоки, наповнити блоки необхідним змістом.
3. У кореневій веб-директорії створити паки: css, js, images, modules (рис. 4.3).
4. Розділити сторінки сайту на структурні елементи виконуючи наступні кроки:
	* скопіювати стилі шаблону до папки css;
	* скопіювати скрипти до папку js;
	* скопіювати всі необхідні зображення до папку images;
	* вибрати головну сторінку сайту та перенести її до папки

«modules/main» з ім'ям index.php (якщо необхідно, виправити відносні шляхи до файлів стилю, зображень і т.д.);

* + відкрити файл modules / main / index.php і послідовно вирізати його структурні елементи (шапку та нижню частину, верхне меню і т.д.), зберігаючи їх в окремих файлах папки includes та підключаючи їх використовую функції include або require.
	+ вміст основної частину сторінки залишаємо без змін.
	+ якщо, тест роботи сторінки пройшов вдало, то копіюємо отриманий index.php в інші модулі і змінюємо їх вміст..
1. Змінити файл index.php (він є основною точкою входу) так, щоб він виконував пошук прийнятого за методом GET параметру «mod» (за замовчуванням mod=main) та виконував підключення файлу index.php з відповідної папки modules.
2. Змінити посилання верхнього меню для виклику сторінок через одну точку входу (наприклад: index.php?mod=main – посилання на головну сторінку).
3. У кореневій папці веб-сервера створити директорію «lab4» та скопіювати до неї отримані файли..
4. Протестувати результати виконаної роботи.
5. Оформити звіт про виконання лабораторної роботи.



**Контрольні питання**

1. Яка різниця між функціями include та require?
2. Коли може виникнути помилка множинного підключення файлів?
3. Який скрипт як правило використовують у якості основної точкою входу?
4. Чому доцільно використовувати одну точку входу?