

ЛАБОРАТОРНА РОБОТА №10

Тема: Препроцесор (шаблонізатор) Pug, препроцесор SCSS/SASS.
Автоматизація процесів верстки.

Мета: навчитися використовувати препроцесор Pug для спрощення та автоматизації роботи з HTML-шаблонами, а також освоїти основи препроцесорів SCSS/SASS для організації та спрощення написання CSS-коду. Ознайомитися з інструментами автоматизації процесів верстки, (наприклад, Gulp або Webpack) для автоматизованої обробки Pug та SCSS/SASS файлів, оптимізації робочого процесу і підвищення ефективності розробки веб-інтерфейсів.

Вимоги до звіту:

1. Для автоматизації та компіляції Pug, SCSS/SASS використати Gulp, Vite, Webpack.
2. Дотримуватись SASS Guidelines.
3. Результати завдань лабораторної роботи повинні бути завантажені до особистого репозиторію студента.

Самостійна робота:

- [Gulp Documentation](#)
- [Vite Documentation](#)
- [Webpack Documentation](#)
- [SASS Documentation](#)
- [SASS Guidelines Documentation](#)
- [Pug Documentation](#)

Практична частина

Завдання:

1. Налаштувати середовище для роботи з Pug та SASS.
2. Використовуючи код HTML лабораторної роботи роботу 8 та 9, переписати з використанням Pug.
 - a. Створити базовий шаблон;
 - b. Використати Template Inheritance;
 - c. Розбити сторінку на блоки (logo.pug, navigation.pug, header.pug, footer.pug і т.д.);
 - d. Використати змінні (Variables) та Interpolation;
 - e. Використати цикли та умови;
 - f. Використати Mixins з передачею даних;

3. Використовуючи CSS код лабораторної роботи роботу 8 та 9, переписати з використанням SASS або SCSS.
4. Використати вкладеності та основні можливості препроцесора SASS:
 - a. Селектор батька «&»;
 - b. Наслідування іменування;
 - c. Використати змінні (Variables);
 - d. Використати Mixins;
 - e. Використати функції та оператори;
 - f. Розбити код на логічні модулі;

Контрольні запитання:

1. Що таке препроцесор (шаблонізатор) Pug, які його переваги у створенні HTML-шаблонів?
2. Які основні синтаксичні відмінності між Pug та стандартним HTML?
3. Що таке препроцесор SASS, чим він відрізняється від звичайного CSS?
4. Які основні переваги використання препроцесора SASS?
5. Як створюються змінні в SCSS/SASS, і як їх можна використовувати для оптимізації стилів?
6. Що таке міксини в SCSS/SASS, і як їх застосовувати для повторного використання коду?
7. Як працює вкладеність у SCSS/SASS, і які її переваги для організації коду?
8. Які інструменти автоматизації процесів верстки використовуються для роботи з Pug та SCSS/SASS?
9. Як налаштувати Gulp або Webpack для автоматичної обробки файлів Pug та SCSS/SASS?
10. Які переваги надає автоматизація процесів верстки при роботі над великими проектами?