

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Комплексна робота №2
з дисципліни
«Технології програмування користувацьких
інтерфейсів (Front-end)»

Виконав:
студент групи ІО-24
Довгань М. С.

Перевірив:
Ковальчук О. М.

Київ - 2024

Тема: Використання препроцесорів для CSS.

Тематика сторінок: LESS, SASS.

Посилання на репозиторій GitHub:

https://github.com/maksdovhan/front-end_complexwork

Налаштування збірки стилів: для першого, другого та четвертого макету я використовував препроцесор SASS (для першого та другого з розширенням .scss, для четвертого .sass) для інших двох - третього та п'ятого макету я застосував LESS. Увесь вихідний код файлів .scss (.sass) та .less я писав у Visual Studio Code, а перетворив у фінальні збірки .css за допомогою playground на офіційних сайтах даних препроцесорів (<https://sass-lang.com> та <https://lesscss.org>), що для мене виявилось найоптимальнішим варіантом.

Опис: загалом, якщо казати за труднощі, які виникли при використанні даних препроцесорів - SASS (.scss та .sass) і LESS, то їх майже не було. Хоч це й було моє перше знайомство із ними, але я швидко впорався з їх розбором, синтаксисом та можливостями. Можливо, деякими труднощами було розібратися з LESS, хоч даний препроцесор мені виявився дуже схожим на SCSS, але по ньому в інтернеті викладено набагато менше інформації, у порівнянні з SASS. І, наскільки я зрозумів, SASS використовується частіше, маючи більшу популярність серед розробників та розширене ком'юніті.

Висновки: дана робота мені сподобалась та познайомила з препроцесорами CSS (до цього не доводилося працювати з ними), на мою думку, це гарна можливість спростити собі розробку та підтримку і масштабування великих проєктів, особливо таких, функціонал яких (кнопки, вигляд тексту, поля вводу і т. д.) часто повторюється. Також, наприклад, коли ми використовуємо для одного типу кнопок одне значення заокруглення, меж, заднього фону, а для інших інші - для цього достатньо створити декілька змінних, і не ламати собі постійно голову над тим, де яке заокруглення, на скільки пікселів, який колір заднього фону та інші параметри. Тобто, якщо тобі необхідно змінити, наприклад, значення меж для всіх кнопок - не треба, як в звичайному CSS шукати їх вручну та виправляти - достатньо всього змінити значення однієї заданої змінної, і всі параметри, де вона застосовується автоматично приймуть її самостійно. Також, мені сподобалось створення міксинів та вкладеність. Можна

створити один міксин для тексту, інший - для кнопки, ще один для логотипу. І, знову ж таки, щоб не доводилося переписувати той же код, можна просто викликати міксин та ввести або свої значення, або залишити ті, які були задані за замовчуванням (при його створенні). Вкладеність теж значно полегшує орієнтацію в коді, адже вона відображає структуру самого HTML. І, мною були застосовані деякі вбудовані функції LESS - а саме `darken` та `lighten` - для роботи з кольорами, з ними можна взяти відомий нам колір і, або затемнити, або, відповідно, висвітлити на задану кількість відсотків. Саме для того, щоб по декілька десятків (а то і сотень) разів не переписувати та не виконувати одну й ту ж роботу та дії, дані препроцесори значно полегшують роботу розробників.

Щодо переваг (суб'єктивних) SASS (`.sass` та `.scss`) то я хотів би почати саме з `.scss` - він має синтаксис, який дуже схожий на стандартний CSS, де використовуються фігурні дужки та крапки з комами, що, як на мене, полегшує перехід та подальше використання і масштабування, особливо для розробників, які до цього працювали з CSS. У `.sass` же, навпаки, немає потреби у застосуванні ні фігурних дужок, ні крапок з комою - для мене це не є недоліком чи перевагою - з одного боку, це зменшує кількість рядків коду, і робить його не таким об'ємним (до речі, я обрав використання цієї версії для 4 макету, який займає найменше коду, для меншого редагування), а з іншого, для деяких розробників та особливо складних та об'ємних проєктів це може виявитись не так зручно. Але, так як ми маємо вибір між `.scss` та `.sass` то кожен може їх порівняти, застосувати та зробити для себе висновок, з чим йому краще працювати. Особисто для мене простішою виявилася робота із використанням `.scss`.

На рахунок LESS я також можу сказати, що він мені сподобався - синтаксис у нього схожий на CSS, також із фігурними дужками та крапками з комами, і так як я спочатку виконав завдання для SASS, з цим препроцесором у мене не виникло проблем, вони схожі між собою, але з деякими невеликими відмінностями - в SASS оголошення власної змінної починається з `$`, міксина - з `@mixin` та його назви, використання - з `@include` та назви, у LESS же, відповідно, власна змінна починається з `@`, міксин з `.`, а його виклик також із крапки та назви. Але все ж, до явних мінусів я б відніс набагато менше ком'юніті та навчальних матеріалів в мережі (у порівнянні з тим же SASS).

Загалом, підсумовуючи, мені сподобалася ця робота, вона познайомила мене з новими препроцесорами, їх сутністю, застосуванням та можливостями. Також, я бачив, що у варіантах завдання є ще один препроцесор - Stylus, сподіваюсь, що колись також матиму змогу із ним познайомитися. Але, на даний момент, при розробці я б віддав перевагу SASS, а саме його версії .scss.