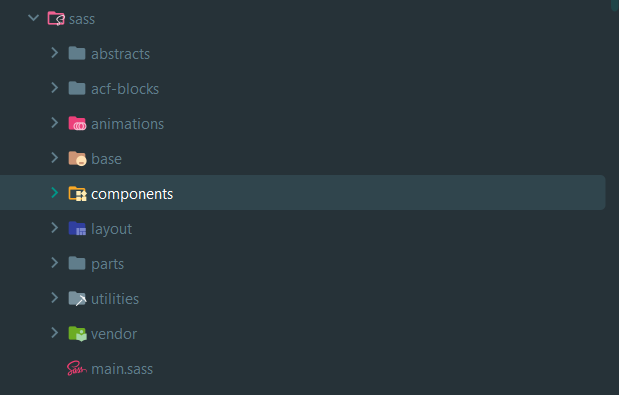
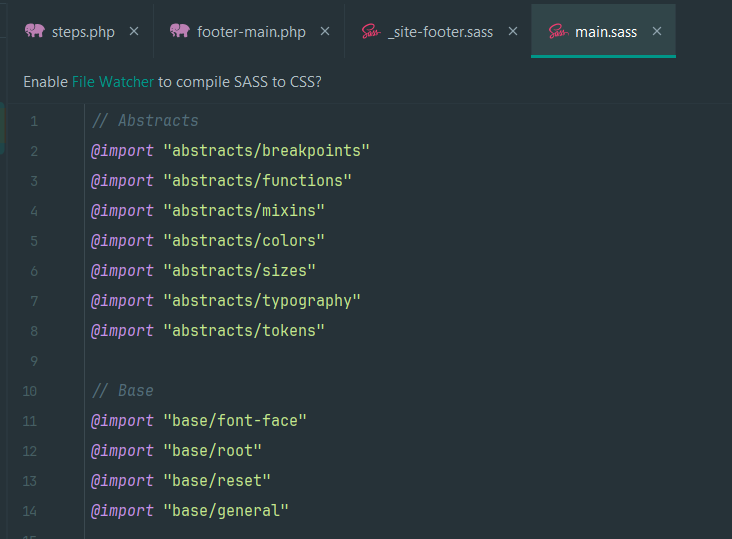
Структура папки:

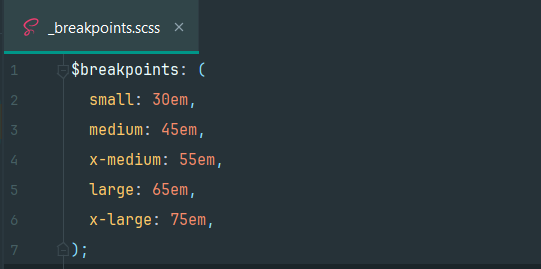


Все підключаємо в main.sass.



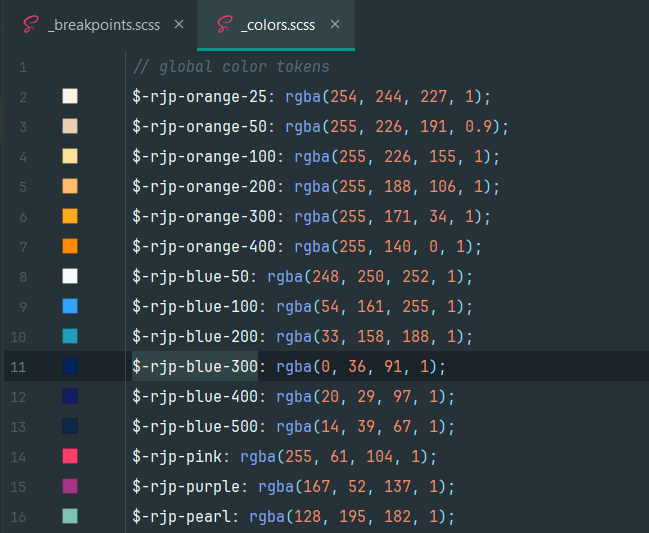
Папка abstracts.

Breakpoints.scss – вказуємо брейкпоінти, які будемо використовувати на всьому сайті.

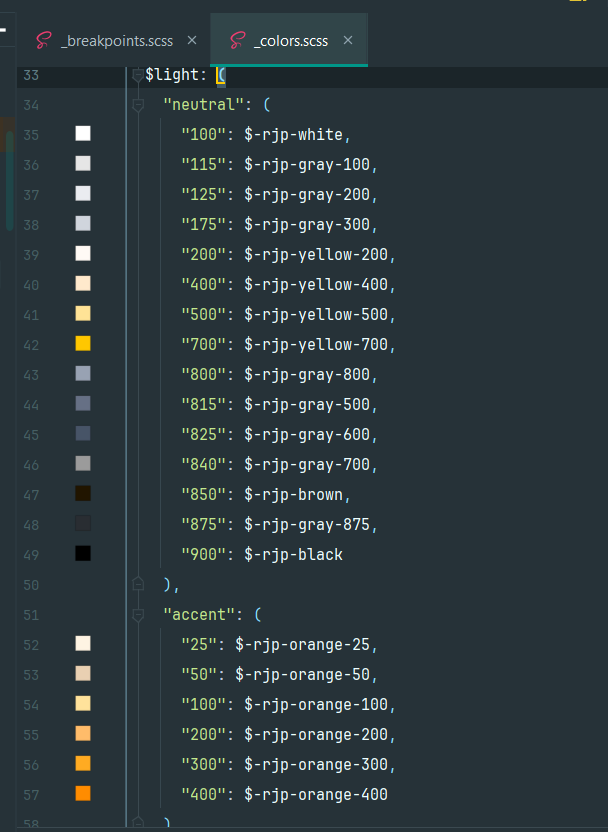


Colors.scss.

Спочатку вказуємо всі кольори:



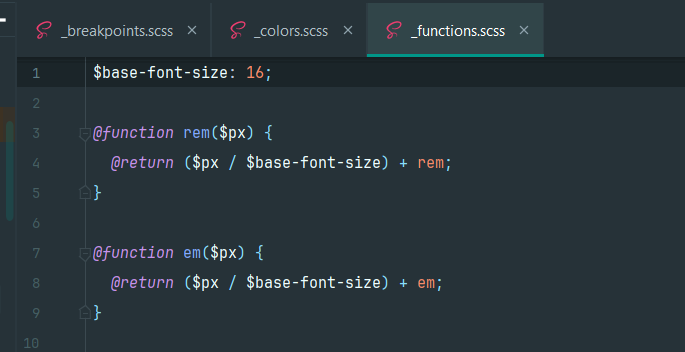
Потім розбиваємо кольори на групки в асоціативному масиві. Наприклад, primary, accent, neutral.



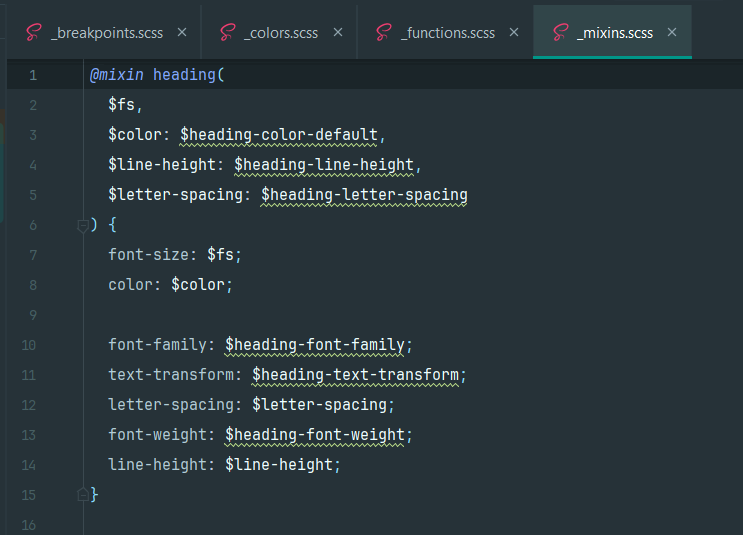
Нижче стоврюємо змінні (для зручності, щоб потім всюди не писати var):



\_functions.scss містить функції:

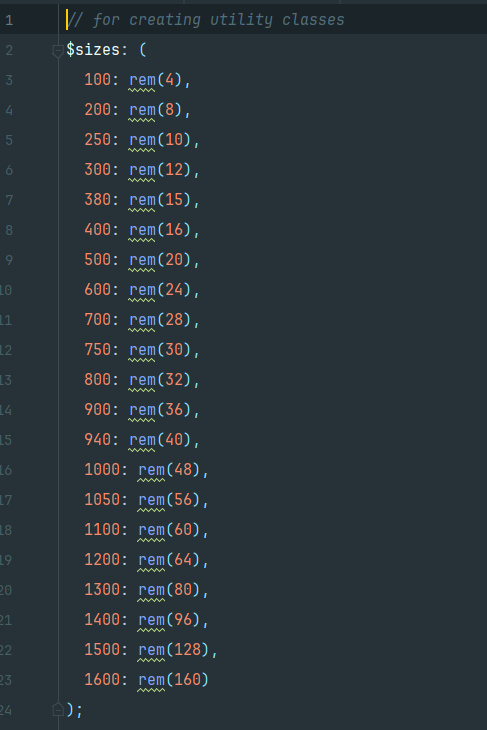


\_mixins.scss містить корисні міксіни:

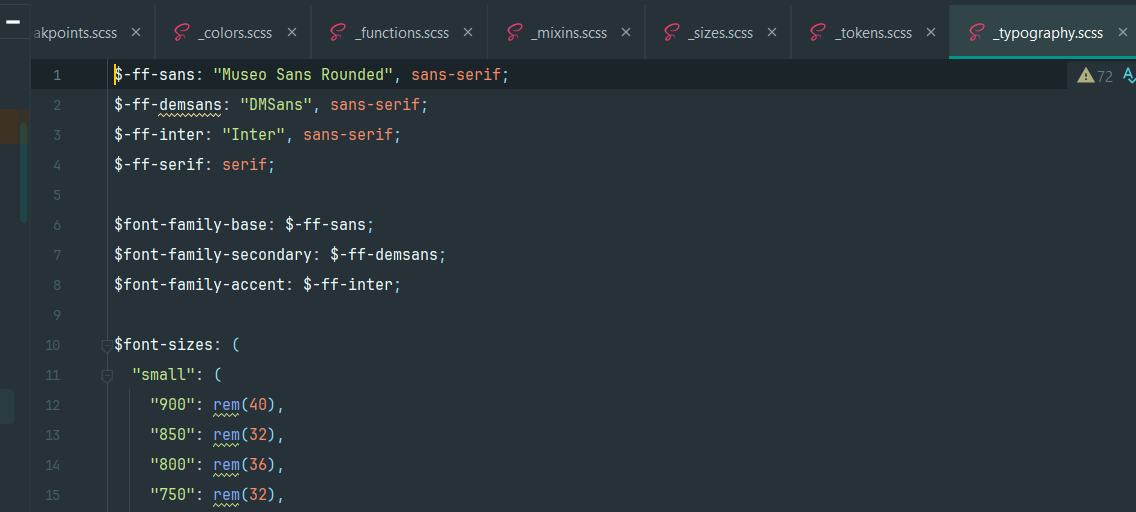


Sizes.scss.

Sizes генеруються для всього – utilities/spacing.

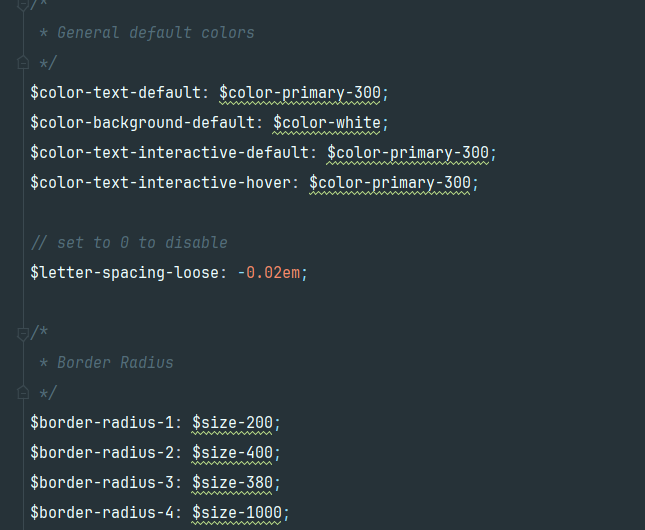


\_typography.scss – тут містяться всі налаштування для тексту.

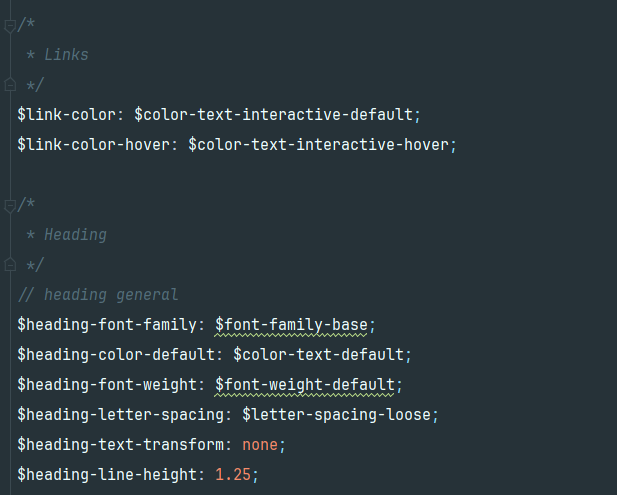


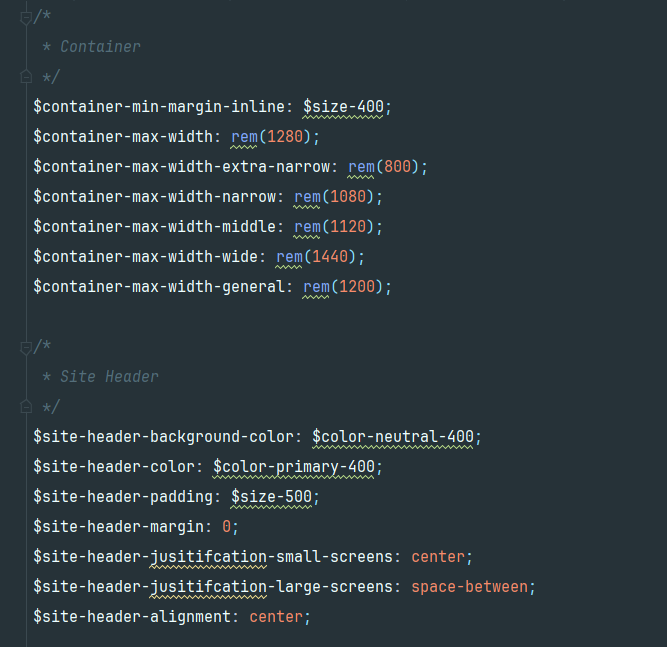
\_tokens.scss – файл, де все збирається в купу.

Ми тут використовуємо всі наші створені змінні.



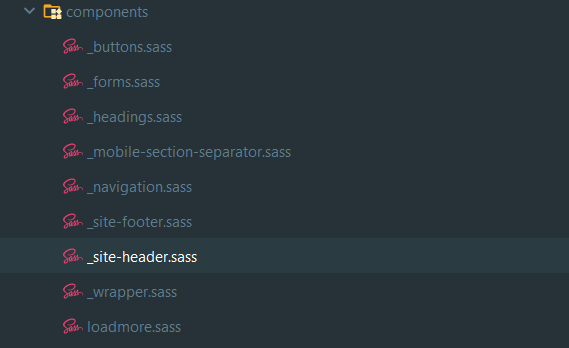
Тут також прописуємо значення для різних компонентів:





І вже ці змінні використовуємо в компонентах.

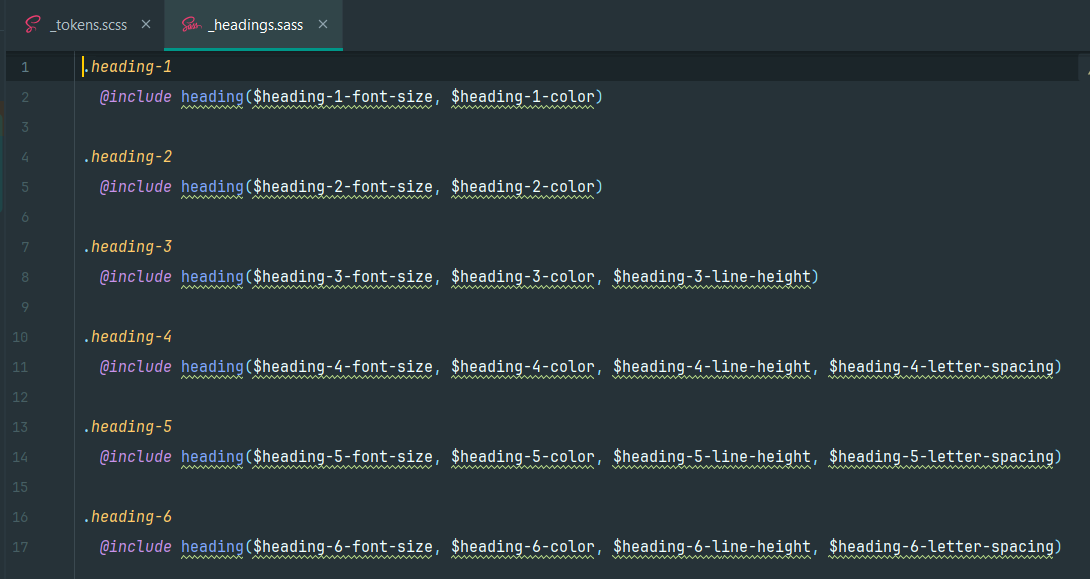
Папка components.



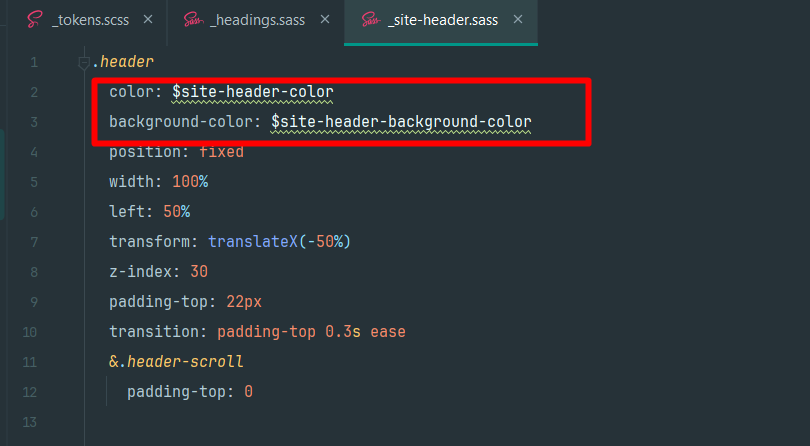
Headings.scss:

Використовуємо змінні з файлу токенів.

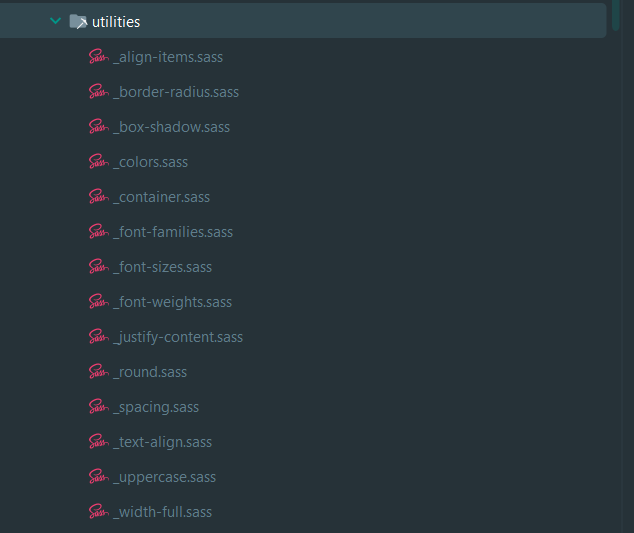
Додатково використовуємо міксіни.



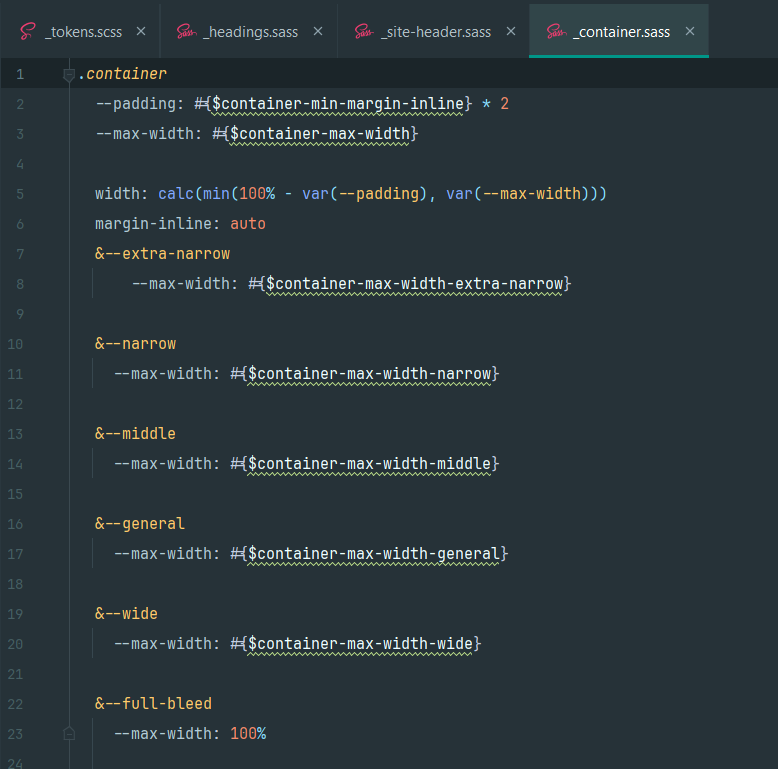
Або header.scss:



Папка utilities.

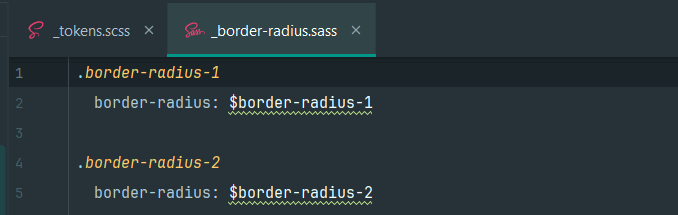


Наприклад, налаштування контейнеру.



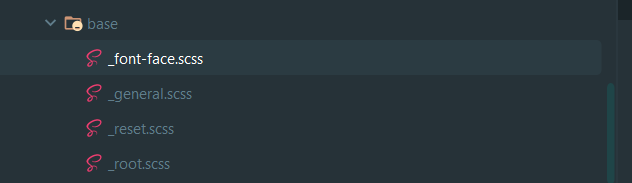
Також використовуємо змінні, які прописали в токенах.

Або border radius:



Папка base.

Тут загальні налаштування:



Grid сітка:

A screenshot of a computer

Description automatically generated

Ось в токенах юзаємо:

A screenshot of a computer program

Description automatically generated

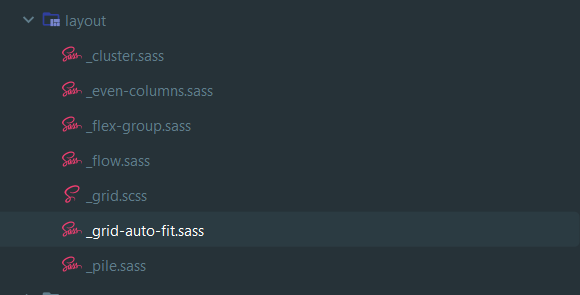
В index. Php є приклад:

A screenshot of a computer

Description automatically generated

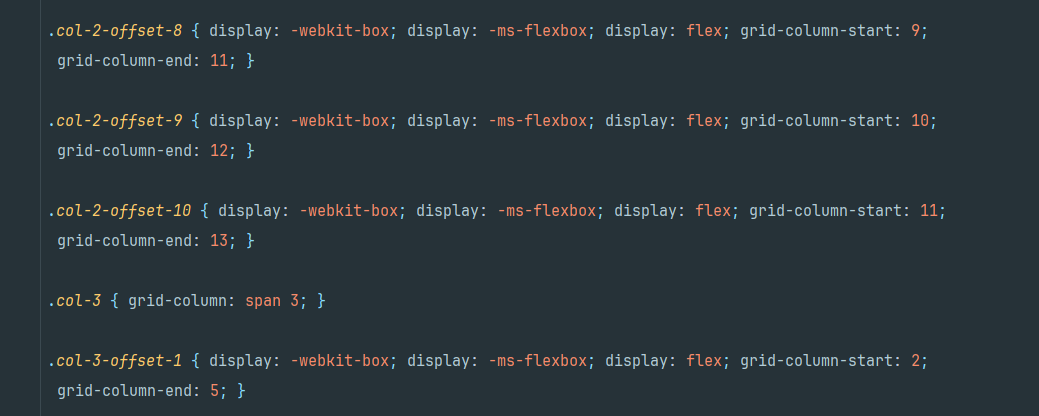
Col-5-offset-2

Загалом можна просто проглянути папку layout.



Згенеровані класи.

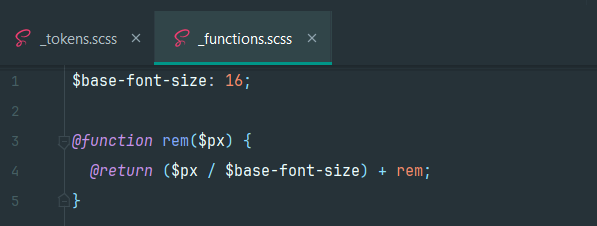
Можемо перейти в папку dist/main.css і подивитись, що згенерувалось.



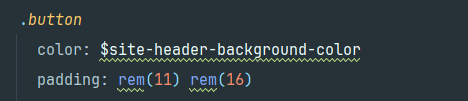
Ці класи потім можемо використовувати.

Зумінг.

Ми використовуємо функцію rem, замість того щоб писати px.



Приклад:



Значення буде вираховуватись від глобального розміру шрифта.

Чим менше екран, ти меньше шрифт, тим менще будуть падінги у блока.

**Приклад як налаштувати під різні розміри дизайну**.

Я зазвичай верстав під дизайн 1920 року, але зараз проект із полотном 1440 року.

A screenshot of a website

Description automatically generated

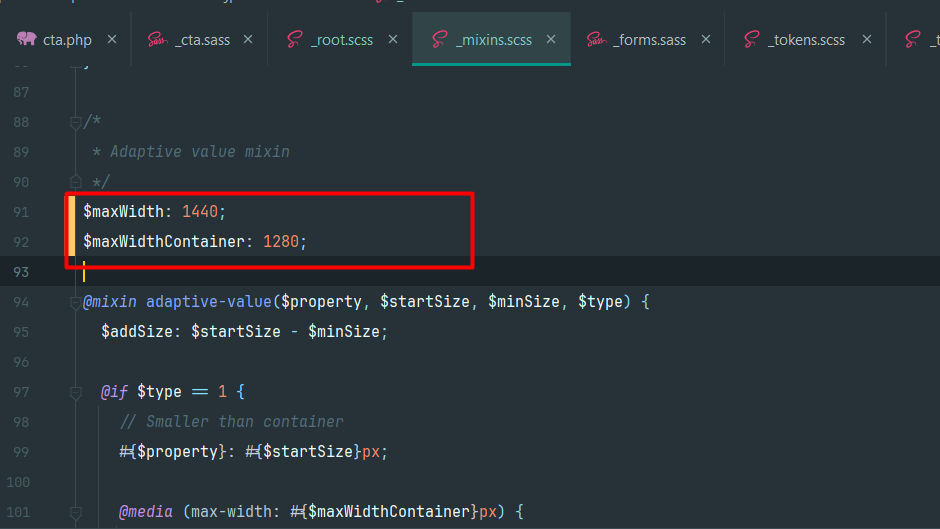
Нам треба змінювати font-size, тому що rem вираховується від нього.

Типово, що менше екран, то менше шрифт, то менше rem.



Змінюємо шрифт адаптивно на скрінах більше ніж мідіум. (Можно налаштувати по різному, не обовзяково саме такий брейкпоінт)

І найголовніше треба було внести змін до файлу міксинів:



maxWidth – наше полотно.

maxWidthContainer – максимальна ширина контейнера.

Тепер як воно працюватиме:

• На екрані 1440 пікселів – font-size: 16px;

• На екрані менше 1440 пікселів, font-size плавно зменшується

• На екрані більше 1440 пікселів, font-size плавно збільшується

Ось приклад.

Контейнер дизайну 1080px.

A screenshot of a computer

Description automatically generated

Зменшуємо екран:

A screenshot of a computer

Description automatically generated

Збільшуємо

A screenshot of a web page

Description automatically generated

JS:

Також я щойно змінив трохи структуру папок js. Можливо, хтось помічав, що Gulp трохи по-дурному збирав js з папки /js/vendor і в результаті приходилося підключати скрипти через CDN. Зараз я прибрав vendor-js з галп-збірки і створив окрему папку /lib, куди **можна кидати мініфікований js бібліотек**. В папці js тепер немає папок /custom, /vendor. У папці /js зберігатиметься весь кастомний js.



Підключення в вордпрес тепер таке.

Кастомні скрипти збираються в один файл і він підключається в самому кінці. Перед цим ми бібліотеки вже вручну підключаємо. Додав за-замовчуванням наші найпопулярніші: gsap, scroll-trigger, lenis, swiper

