

УДК 004.42(07)

Н.А. Христинець, А.А. Сахнюк, К.А. Свиридчук, О.І. Міскевич
Луцький національний технічний Університет**ВИКОРИСТАННЯ БЕМ-БЛОКІВ ПРИ СТВОРЕННІ САЙТІВ****Н.А. Христинець, А.А. Сахнюк, К.А. Свиридчук, О.І. Міскевич. Використання бем-блоків при створенні сайтів.**

Розкрито суть реалізації БЕМ-блоків при створенні сайтів. Досліджено актуальність заміни стандартної реалізації гіпертекстової розмітки HTML та стилей CSS на технологію БЕМ в середовищі розробки Drupal8.

Ключові слова: БЕМ-блок, CSS, HTML, Drupal 8, шаблон, фреймворк.

Н.А. Христинець, А.А. Сахнюк, Е.А. Свиридчук, О.І. Міскевич. Использование бэм-блоков при создании сайта.

Раскрыта суть реализации БЭМ-блоков при создании сайтов. Исследована актуальность замены стандартной реализации гипертекстовой разметки HTML и стилей CSS на технологию БЭМ в среде разработки Drupal8.

Ключевые слова: БЭМ-блок, CSS, HTML, Drupal 8, шаблон, фреймворк.

N.A. Khrystynets, A.A. Sakhnyuk, E.A. Sviridyuk, O.I. Miskevich. Use of bem-blocks when creating a site.

The essence of the implementation of BEM-blocks when creating websites is disclosed. The relevance of replacing the standard implementation of hypertext markup HTML and CSS styles with BEM technology in the Drupal8 development environment has been investigated.

Keywords: BEM-block, CSS, HTML, Drupal 8, template, framework.

Постановка проблеми. На сьогоднішній день існує проблема як спростити код і полегшити рефакторинг сайту. На сьогодні існує одна з найпоширеніших методик реалізації цієї проблеми – методологія БЕМ.**Аналіз досліджень** В роботах закордонних дослідників, розробників описується популярність використання методології БЕМ.

В публікації Harry Robertsа [4] зазначено, що БЕМ-блоки роблять все набагато жорсткішим і більш зв'язним, роблячи код простішим у обслуговуванні командами або навіть самостійно через певний проміжок часу.

В публікації Johan Ronsse [7] визначено, що найкраще застосовувати БЕМ-блоки краще в великих проектах, а в невеликих проектах БЕМ використовувати не потрібно, приводиться доцільність використання БЕМ-блоків в невеликих проектах.

Виклад основного матеріалу й обґрунтування отриманих результатів.

БЕМ – значення блоку, елемента, модифікатора – це методологія фронтального іменування, придумана розробниками в Яндексі. Це розумний спосіб назвати свої класи CSS, щоб дати їм більшу прозорість і значення для інших розробників. Вони набагато суворіші та інформативніші, що робить ідею іменування БЕМ ідеальною для команд розробників на великих проектах, які можуть тривати деякий час.

Важливо відзначити, що у дипломному проекті використано схему найменування, засновану на БЕМ, але змінену Гаррі Робертсом. Методи найменування, описані в цій публікації, не є оригінальними БЕМ, але є вдосконаленою версією. Якими б не були фактичні позначення, всі вони базуються на тих самих принципах БЕМ.

Найменування відповідає такому шаблону:

```
.block {}
.block__element {}
.block--modifier {}
```

- .block являє собою більш високий рівень абстракції або компонента.
- .block__element являє собою нащадка, .block що допомагає сформуванню .block в цілому.
- .block—modifier являє собою інший стан або версію .block.

Суть БЕМ полягає в тому, щоб розповісти іншим розробникам більше про те, що шматок розмітки робить. Читаючи якийсь HTML з деякими класами, можна побачити, як фрагменти пов'язані;

щось може бути лише компонентом, щось може бути нащадком або елементом цього компонента, і щось може бути варіацією або модифікатором цього компонента. Наприклад:

```
.person {}  
.person__hand {}  
.person--female {}  
.person--female__hand {}  
.person__hand--left {}
```

Доцільність використання БЕМ-блоків

Програмісти радять використовувати БЕМ, оскільки його корисність зарекомендувала себе з хорошої сторони, а також рекомендують іншим розглянути можливість його прийняття, оскільки блоки роблять все набагато жорсткішим і більш зв'язним, роблячи код простішим у обслуговуванні командами або навіть самостійно через певний проміжок часу.

Наприклад візьмемо логотип `.site-logo`. Уявіть, що ми хочемо мати святкову версію логотипу для нашого дизайну сайту Різдвяного. Ми могли б:

```
.site-logo {}  
.site-logo--xmas {}
```

З цього робимо висновки, що ми можемо швидко побудувати варіації речей, використовуючи позначення модифікатора.

Доцільність використання БЕМ-блоків в малих проектах

Методологія БЕМ надає правила організації веб-проектів, незалежно від їх розміру або кількості розробників в команді. Навіть якщо у вашій команді двоє людей і ви верстаєте односторінкові сайти, БЕМ дозволяє:

- Повторно використовувати верстку
 - Невеликі однотипні проекти можуть мати схожу структуру. Наприклад, посадочні сторінки (landing) – різні зовні, однакові всередині. Для їх створення можна використовувати готові шаблони.
 - У межах однієї сторінки використовуються однакові блоки: кілька кнопок, випадаючих списків або меню. Їх можна взяти з готової бібліотеки або реалізувати свою бібліотеку і використовувати у всіх проектах.
- Швидко прототіпіровать верстку
 - Прототип сайту створюється з блоків. Замість верстки в БЕМ-проект ви відразу проектуєте інтерфейс з готових блоків.
- прискорити розробку
 - Рівні перевизначення дозволяють підключати бібліотеки і доопределять блоки, не залежати від оновлень бібліотеки.
 - БЕМ-проект можна швидко почати з шаблонного проекту `project-stub` або `БЕМ-express`.
- Чи не залежати від конкретного розробника
 - Однакова структура всіх проектів, одні правила організації коду, ізольовані блоки полегшують передачу коду між розробниками.
- прискорити рефакторинг
 - БЕМ-проект влаштований таким чином, що зміни в одному блоці можна застосувати до всіх блоків в проекті. При цьому немає необхідності знати всі можливі випадки використання цього блоку.
 - Система іменування БЕМ-сутностей дозволяє вкласти сенс в імена і зробити їх максимально інформативними для розробника, тобто писати самодокументуємий код.
- Прискорити і спростити зміну дизайну за рахунок рівнів перевизначення.
 - Модифікувати CSS / JS навіть в односторінковому проекті.

Принципи роботи з елементами**Вкладеність**

- Блоки можна вкладати один в одного.
- Допустима будь-яка вкладеність блоків.

Належність

Елемент – завжди частина блоку і не повинен використовуватися окремо від нього.

Необов'язковість

Елемент – необов'язковий компонент блоку. Не у всіх блоків повинні бути елементи.

Приклад сторінки Medoc сайту Profinteco з використанням БЕМ-блоків:

```
{% set БЕМ_block = 'app' %}
<div class="{{ БЕМ_block }}">
  {% include '@blogger/include/header.html.twig' with {'show_dlog_hero': false} %}
  {% include '@blogger/include/pages/medoc.html.twig' %}
  <main{{ main_layout_attributes.addClass(БЕМ_block ~ '__content') }}>
    <div class="main-layout__inner">
      {{ page.content }}
      {{ page.sidebar_first }}
    </div>
  </main>

  {% include '@blogger/include/footer.html.twig' %}
</div>
```



Рис. 1. Результат файлу header.html.twig

На прикладі сторінки можна побачити, що всі принципи роботи з елементами виконані. Також можливе використання інших файлів які використовуються в розробці і потрібні для розробки сторінки. Наприклад файли header.html.twig і footer.html.twig являються стандартними для всіх сторінок сайту, тому вони перевикористовуються в кожній сторінці сайту. Файл medoc.html.twig(код доданий нижче) є власним блоком сторінки. Тому при створенні нової сторінки просто змінюються чи додаються власні блоки.

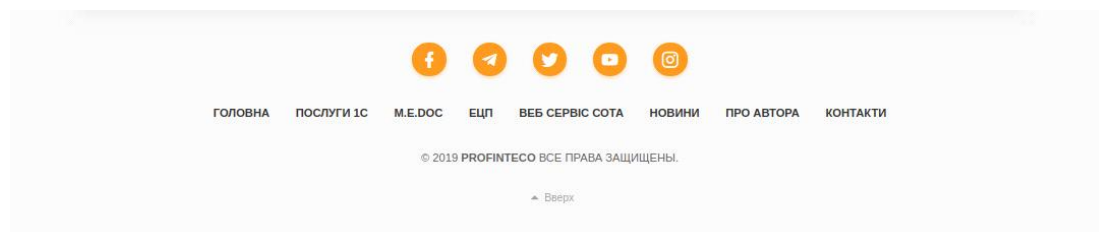


Рис. 2. Результат файлу footer.html.twig

Код файлу **medoc.html.twig**:

```
{% set БЕМ_block = 'medoc' %}
```

```
<main class="{{ БЕМ_block }}">
  <div class="{{ БЕМ_block }}__welcome">
    <div class="{{ БЕМ_block }}__welcome-video-wrapper">
      <video poster="{{ medoc_image_uri|image_style('thumbnail') }}" autoplay loop muted
class="{{ БЕМ_block }}__welcome-video">
        <source src="{{ file_url(medoc_video_uri) }}" type="video/mp4">
        {{ 'Your browser does not support the video tag'|t }}
      </video>
    </div>
    <div class="{{ БЕМ_block }}__container">
      <div class="{{ БЕМ_block }}__welcome-inner">
        <h1 class="{{ БЕМ_block }}__welcome-title">M.E.Doc</h1>

        <div class="{{ БЕМ_block }}__welcome-description">
          <p>Система електронного документообігу.</p>
        </div>
      </div>
    </div>
  </div>
</main>
```

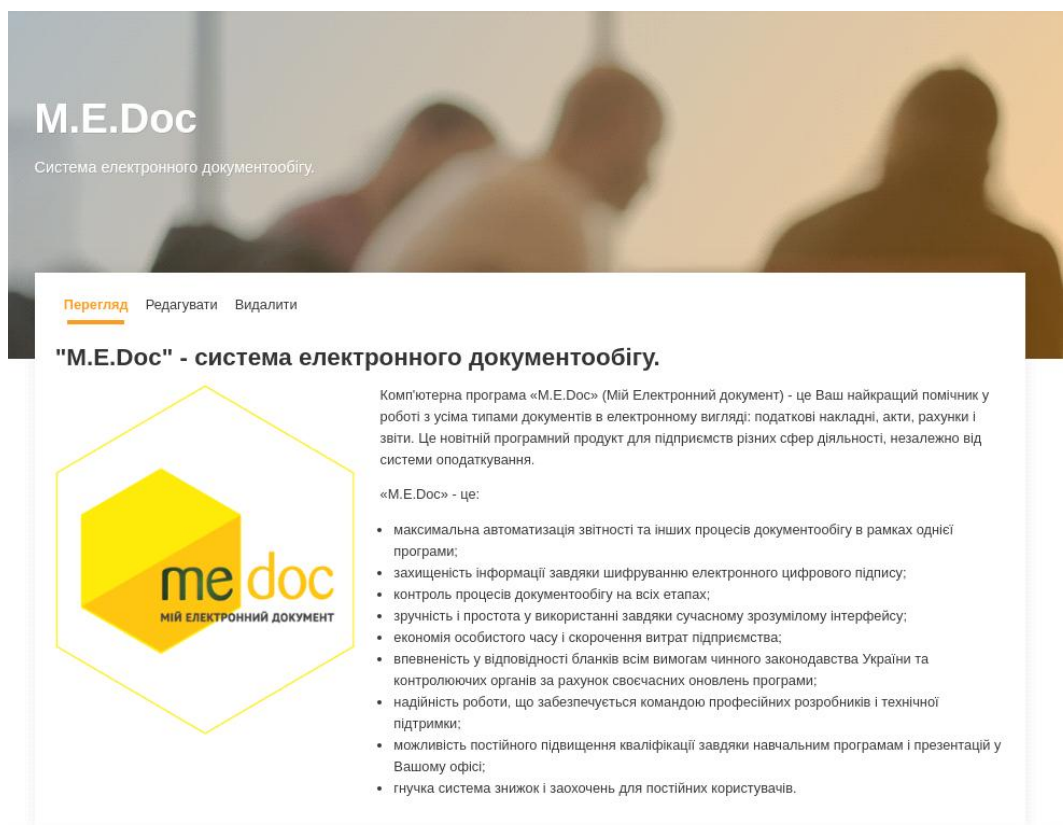


Рис. 3 – Результат файлу medoc.html.twig

Недоліки використання БЕМ-блоків

Одним з основних недоліків БЕМ є тому, що код має дещо дивний вигляд. Якщо програміст зациклений виключно на зовнішньому вигляді коду, або якщо код стає невиправдано важким для

розуміння, або важким для читання, то, можливо, програмісту доведеться двічі подумати, перш ніж використовувати його, але якщо він "просто виглядає дивно", тоді технологія БЕМ-блоків повинна бути розглянута при розробці проекту. Технологія БЕМ виглядає дивно, але потужність, яку вона надає, значно переважає будь-які недоліки, пов'язані з його зовнішнім виглядом на порядок.

Другим недоліком є те що верстка можлива тільки блоками. У кожному великому проекті є дрібні елементи, як кнопки, дропдауни, тайтли, субтайтли, секції і т.д. Але в БЕМі їх немає, так як будь-який елемент без блоку використовувати теж заборонено.

Висновки та перспективи подальшого дослідження.

Отже, це БЕМ щодня надзвичайно корисна, потужна і проста угода про найменування, яка полегшує читання і розуміння вашого інтерфейсного коду, легше працювати, легше масштабувати, більш надійне і явне і набагато суворіше.

Для всіх БЕМ виглядає трохи дивним, це є надзвичайно цінним доповненням до панелі інструментів розробника, незалежно від проекту.

1. Addison Berry, Angela Byron, Bruno De Bondt. Using Drupal (2nd Edition). – O'Reilly. 2012. – 500p.
2. Al-Darwish, N.: PageGen: An Effective Scheme for Dynamic Generation of Web Pages. Information and Software Technology 45(10), 15 July 2003, Pages 651-662
3. Cynthia McCourt. Drupal: The Guide to Planning and Building Websites. – Wrox. 2011. – 504 p.
4. <https://ru.BEM.info/methodology/naming-convention/>
5. <https://csswizardry.com/2013/01/mindBEMding-getting-your-head-round-BEM-syntax/>
6. <https://ru.BEM.info/methodology/quick-start/>
7. <https://web-creator.ru/articles/BEM>
8. <https://frontender.info/learning-to-love-BEM/>