

УДК 004.023

Багнюк Н.В. к.т.н, доцент, Мельник К.В. к.т.н, доцент, Михайлуца П.С., студент
Луцький національний технічний університет

СУЧАСНІ ТЕХНОЛОГІЇ РЕАЛІЗАЦІЇ ДИНАМІЧНОГО ДИЗАЙНУ WEB СТОРІНОК

Багнюк Н.В., Мельник К.В., Михайлуца П.С. Сучасні технології реалізації динамічного дизайну web сторінок. У статті проведено огляд еволюції принципів побудови web сторінок, а також пропонується спосіб їх розробки з використанням скриптових мов програмування та дослідження принципу реалізації динамічної зміни властивостей різноманітних об'єктів в процесі її активної експлуатації.

Ключові слова: динамічний, дизайн, parallax, web сторінки, cms, framework, javascript.

Багнюк Н.В., Мельник К.В., Михайлуца П.С. Современные технологии реализации динамического дизайна web страниц. В статье проведен обзор эволюции принципов построения web страниц, а также предлагается способ их разработки с использованием скриптовых языков программирования и исследование принципа реализации динамического изменения свойств различных объектов в процессе ее активной эксплуатации.

Ключевые слова: динамический, дизайн, parallax, web страницы, cms, framework, javascript.

Bahniuk N.V., Melnyk K.V., Mykhailutsa P.S. Modern technologies of dynamic web page design. This article provides an overview of the evolution of the principles of building web pages, and provides a method of development using scripting languages, as well as research principles of the dynamic changes in the properties of different objects during its active operation.

Keywords: translation, dynamic, design, parallax, web pages, cms, framework, javascript.

Вступ. Сучасні рішення реалізації веб сторінок та загалом веб ресурсів базуються на використанні складного набору різноманітних інструментів та засобів розробки. Розвиток технології звичайної передачі гіпертексту та реалізації його на стороні клієнта притерпівся суттєвих змін за останні 30 років. Ринок послуг, які повинні задовольняти усім вимогам до наповнення сторінки також повинен відповідати вимогам щодо сучасних технологічних рішень.

Реалізація динамічного дизайну сторінки висуває низку вимог:

- використання скриптів в процесі експлуатації сторінки;
- зчитування поточного стану сторінки, положення мишки, прогортання;
- внесення змін «на льоту» в код сторінки або її параметри, наприклад, стиль;
- дизайн повинен пригортати увагу користувача.

Більшість сучасних рішень побудовані на основі JavaScript, що обґрунтовано його гнучкістю та набором модулів для роботи з складовими частинами сторінки та легкістю маніпуляції DOM об'єктами.

Постановка наукової проблеми. Метою даної роботи є аналіз існуючих способів розробки динамічного web дизайну в розрізі еволюції принципів розробки з використанням скриптових мов програмування та дослідження реалізації принципу динамічної зміни властивостей різноманітних об'єктів web сторінки в процесі її активної експлуатації.

Аналіз еволюції принципів розробки web сторінок та існуючих засобів.

Для розуміння принципу побудови динамічного дизайну web сторінки слід відмітити чотири основних етапи технологічного розвитку виконання сторінок, що відрізняються основним набором інструментів.

Перший етап характеризувався статичними сторінками, які базувались на гіпертекстовій розмітці сторінки (HTML, XHTML, DHTML) (рис. 1). Використання таких технологій з плином часу вимагало залучення великих ресурсів для створення сторінок. Чим складніша сторінка (наповнена великою кількістю даних), тим більше часу необхідно було затрачати для її побудови та обслуговування.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Title</title>
6 </head>
7 <body>
8   <h1>Hello World!</h1>
9 </body>
10 </html>
```

Рис. 1. Приклад гіпертекстової розмітки сторінки

На підтримку розробникам сторінок було запропоновано використання динамічно-генерованого вмісту сторінок та використання технології збереження даних у базах даних, що спростило керування такими ресурсами, а процес написання нових сторінок супроводжувався реалізацією алгоритмів, які забезпечували збір та розміщення даних на сторінці з використанням алгоритмів та функцій мов програмування. Залучення на другому етапі таких мов, як наприклад, PHP змінили підхід до створення сторінок загалом. Основну вимогу до знання чи використання гіпертекстової розмітки не було відмінено, її розширили, давши можливість модульного розширення функціоналу [1].

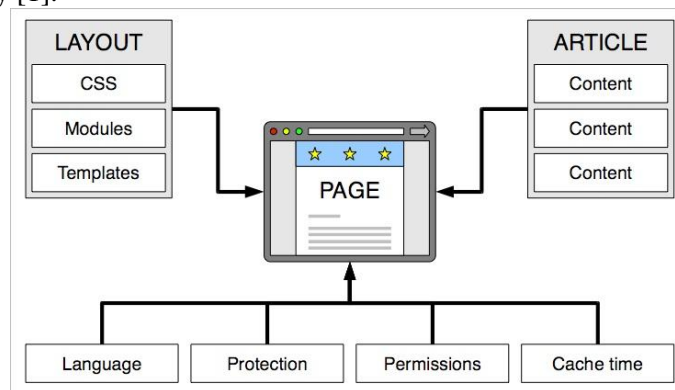


Рис. 2. Структура сторінки, згенерованої системою управління вмістом

Використання мов програмування дало можливість ще більш спростити процес розробки об'ємних ресурсів в мережі Інтернет та використати повноцінну модель «клієнт-сервер», що в свою чергу дало старт третьому етапу еволюції, а саме – використанню систем управління вмістом CMS та системам, побудованих на базі Framework технології (робочих середовищ розробки). Такі системи надають набір спеціалізованого або базового функціоналу для побудови профільних, унікальних ресурсів. Управління вмістом повністю покладається на ядро такої системи і в залежності від її типу або спеціалізації потребує набору інструментів для його роботи. Наприклад, однією з найпопулярніших сьогодні систем є WordPress, що побудована на основі PHP, може використовувати велику кількість модулів з набором різноманітних функцій, або JavaScript, система AngularJS, яка є також модульною, але являє собою звичайний набір файлів, реалізованих на мові JavaScript. Структуру CMS показано на рисунку 2. Для прикладу, серед середовищ можна взяти сучасний YII 2, що побудована на базі PHP або середовища мови Python: Django, Flask, Tornado. Структуру веб додатків, побудованих на основі Framework, показано на рисунку 3. Як CMS, так і Framework системи генерують вміст сторінки для відображення її у браузері клієнта ресурсу, основна ж їх відмінність є лише місце генерації цього вмісту – сервер або ж клієнтський браузер[2].

Виділяють також четвертий етап розвитку, який характеризується запуском нової версії гіпертекстової мови розмітки сторінок HTML5. Загалом, дана нова «мова» є гібридом мови програмування та звичайної розмітки, що дозволяє викинути зі схеми додаткові інструменти,

наприклад Flash додатки, які несуть за собою низку проблем з безпекою та споживають велику кількість ресурсів на стороні клієнта для їх відображення. Можемо спостерігати поступове залучення такої технології побудови ресурсів, наприклад, на ресурсі YouTube, де успішно використовується HTML відео програвач. Великі компанії такі, як Amazon, Google, Cisco також розпочали перехід веб інтерфейсів на дану мову. Наприклад, Amazon впровадив новий інтерфейс управління масивами серверів, основна маса коду якого реалізована лише з використанням HTML5.

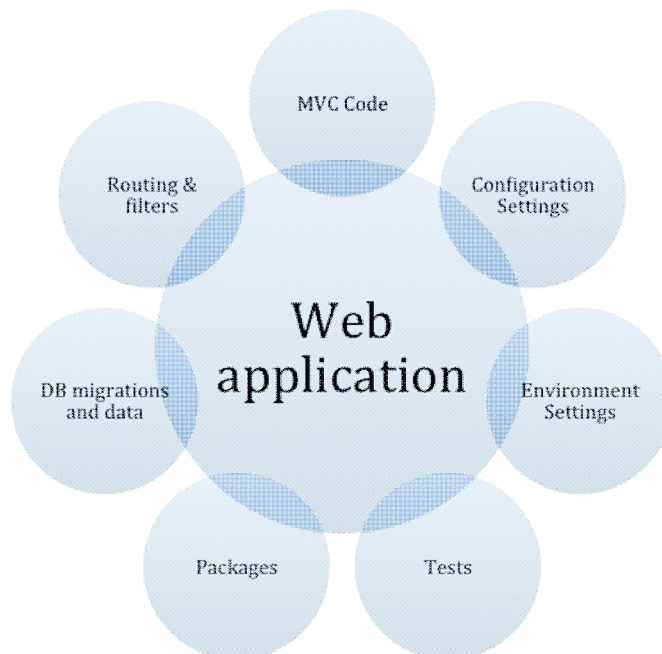


Рис. 3. Основні складові Web додатків, побудованих на базі Framework

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.

Для побудови динамічного дизайну сторінки використовується скриптовий код, що реагуватиме на зміну параметрів на сторінці або в процесі перегляду сторінки. З розвитком технологій та поширенням мобільних платформ (планшетів та сучасних телефонів), дуже актуальних є зміна вмісту або стану сторінки саме в процесі прогортання її, так званого «скролу», що дозволить пригорнути більшу увагу до самої сторінки. В галузі електронного маркетингу – пригортання уваги до продукту є одним з основних засобів ведення успішного бізнесу. Для реалізації цієї технології пропонується залучити технологію Parallax. Структура побудови зображення використовує багатоплановий, принцип як показано на рисунку 4.

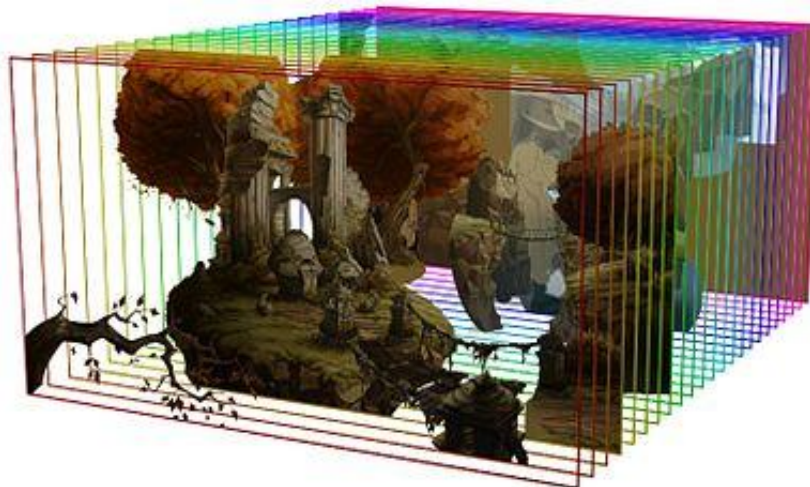


Рис. 4. Шари зображення Parallax

Кожному з шарів зображення можна задати свою динаміку та напрямок руху. Для акцентування уваги на окремих елементах сторінки використовується ефект впливання елементів сторінки в процесі її до низу або до верху. Кожен з шарів рухається з своєю швидкістю та змінюється як показано на рисунку 5.

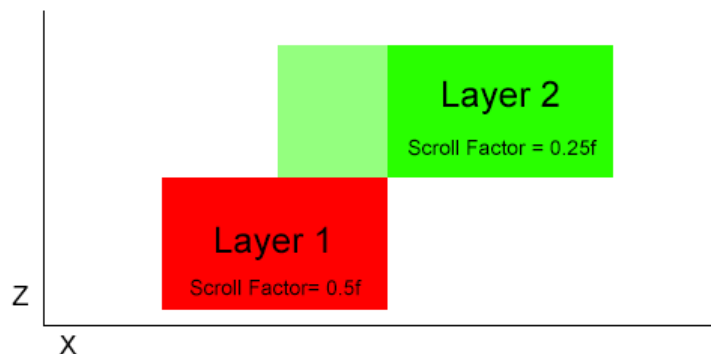


Рис. 5. Нашарування об'єктів на сторінці

Для реалізації таких ефектів використовується принцип ігор початку 90-х для ігрових консолей того часу. Найпростішим варіантом є використання зображень, як шпалер на сторінці, кожне з яких буде знаходитись на своєму шарі зі своїм стилем CSS. Рух таких шарів виконує алгоритм, що розміщується в скрипті для цієї сторінки. Наприклад, для реалізації 4 слайдів для прогортання на сторінці можна створити блок з 4 секцій в HTML розмітці сторінки:

```
<main>
  <section id="slide-1" class="homeSlide">
    <div class="bcg">
      <div class="hsContainer">
        <div class="hsContent">
          <h2>Найпростіший спосіб пригорнути увагу є...</h2>
        </div>
      </div>
    </div>
  </section>
  <section id="slide-2" class="homeSlide">
    <div class="bcg">
      <div class="hsContainer">
        <div class="hsContent">
          <h2>використати динамічний дизайн за допомогою...</h2>
        </div>
      </div>
    </div>
  </section>
  <section id="slide-3" class="homeSlide">
    <div class="bcg">
      <div class="hsContainer">
        <div class="hsContent">
          <h2>використання Parallax ефекту лише...</h2>
        </div>
      </div>
    </div>
  </section>
  <section id="slide-4" class="homeSlide">
    <div class="bcg">
      <div class="hsContainer">
        <div class="hsContent">
          <h2>додавши його на сторінку.</h2>
        </div>
      </div>
    </div>
  </section>
</main>
```

```
</main>
```

Кожному з блоків, як вказано у розмітці, задано свій клас та унікальну назву для спрощення стилізації їх у розмітці даної сторінки, наприклад, для визначення положення першої секції можна задати:

```
<section id="slide-1" class="homeSlide">
  <div class="bcg"
    data-center="background-position: 50% 0px;"
    data-top-bottom="background-position: 50% -100px;"
    data-anchor-target="#slide-1">
    <div class="hsContainer">
      <div class="hsContent"
        data-center="bottom: 200px; opacity: 1"
        data-top="bottom: 1200px; opacity: 0"
        data-anchor-target="#slide-1 h2">
        <h2> Найпростіший спосіб пригорнути увагу є...</h2>
      </div>
    </div>
  </div>
</section>
```

Як видно з коду, положення задається абсолютними значення в пікселях для початкового розташування та позиції зміни за допомогою data-attributes. Для управління динамічною передачею стану та зміни положення слід використати скрипт. Для отримання параметрів data-attributes, можна їх зчитати з документа за допомогою скрипта[3]:

```
/^data(?:-?(_\w+))?(?:-?(?:\d*\.\d+p?)? (?:-
?(start|end|top|center|bottom))?(?:-?(top|center|bottom))?$;/;
```

За допомогою параметрів data-center та data-top-bottom браузер отримує директиву на зсув позиції центру картинки першого слайду на 100 пікселів, доки не перейде до низу першого слайду. Перехід на наступний слайд відбудеться при досягненні параметру data-top за допомогою якоря data-anchor-target, що в свою чергу вказує на об'єкт, який має з'явитись.

Динаміку прогортання можна задати набором функцій, визначивши початкову позицію, як нуль, та кінцеву, як одиницю, проміжні ланки задати звичайним математичним розрахунком наприклад[4]:

```
begin: function() {return 0;};
end: function() {return 1;};
```

Для лінійної прокрутки:

```
linear: function(p) {return p;};
```

Для прокрутки в квадратній прогресії:

```
quadratic: function(p) {return p * p;};
```

Для прокрутки в кубічній прогресії:

```
cubic: function(p) {return p * p * p;};
```

Для повороту:

```
swing: function(p) {return (-Math.cos(p * Math.PI) / 2) + 0.5;};
```

Інші типи прокрутки:

```
sqrt: function(p) {return Math.sqrt(p);};
```

```
outCubic: function(p) {return (Math.pow((p - 1), 3) + 1);};
```

```
bounce: function(p) {
  var a;
  if(p <= 0.5083) {
    a = 3;
  } else if(p <= 0.8489) {
    a = 9;
  } else if(p <= 0.96208) {
```

```
    a = 27;  
  } else if(p <= 0.99981) {  
    a = 91;  
  } else {  
    return 1;  
  }  
  return 1 - Math.abs(3 * Math.cos(p * a * 1.028) / a);  
}
```

де p – множник заданої прокрутки.

Завдяки даному підходу можна реалізувати різноманітні ефекти появи об'єктів на сторінці та їх зникнення. Приведемо приклад горизонтальної прокрутки, де за допомогою не складних математичних перетворень положення об'єкту на сторінці, його кут повороту та швидкість появи можна змінювати за потреби у будь-якому положенні у трьохвимірному просторі.

Висновки

Запропонований метод створення динамічного дизайну для сайту є дуже популярним для невеликих проектів та великих компаній. Загальний об'єм даних, які підлягають трансформації на сторінках, проаналізованих у дослідженнях для цієї статті, напряду залежить від складності їх реалізації. Існують окремі проекти, які повністю базуються на технології Parallax та використовують велику частину її можливостей. Більшість існуючих рішень залучають динаміку зміни об'єктів на сторінці лише частково.

Дана технологія є чудовим рішенням для сайтів електронної комерції, фотоальбомів, одно-сторінкових ресурсів. Нажаль, технологію Parallax не об'єктивно використовувати в різноманітних інформаційно-навантажених системах, наприклад, інформаційних порталах чи панелях управління. Основним її недоліком є складність розрахунку позицій об'єктів та їх зміни при побудові їх на базі CMS або ж Framework. Цікавим рішенням для останніх платформ може стати інтеграція таких ефектів до структури розрахунків серверної частини фреймворку, що може в рази спростити процес генерації сторінок.

Існуючі скрипти, реалізовані на мові JavaScript для роботи з технологією Parallax, дозволяють її інтеграцію у більшість систем управління вмістом, що спрощує процес реалізації динамічного дизайну на різнотипних проектах.

1. Jon Duckett "HTML and CSS: Design and Build Websites" - John Wiley & Sons, U.S.A, том 9, №3, Nov 2011.
2. Dr. Andy Williams "Wordpress for Beginners 2017: A Visual Step-by-Step Guide to Mastering Wordpress", Kindle Edition, Dec 2016.
3. Douglas Crockford "JavaScript: The Good Parts" – Wilmington, DE, U.S.A May 2014
4. Ved Antani. " Mastering JavaScript". Packt Publishing Ltd. Birmingham UK, Jan 2016.