

Д/р
14158

Одеський національний університет імені І. І. Мечникова
(повне найменування вищого навчального закладу)

Навчально-науковий інститут інформаційних та соціальних технологій
(повне найменування інституту/факультету)

кафедра системного програмного забезпечення та технологій дистанційного
навчання
(повна назва кафедри)

Дипломна робота

спеціаліста

на тему: «Сравнительный анализ и тестирование препроцессоров CSS»

«Порівняльний аналіз і тестування препроцесорів CSS»
«Comparative analysis and testing of CSS preprocessors»

Текст, рос.

Виконав: студент денної форми навчання
спеціальності 123 «Комп'ютерна інженерія»
Дерев'янка Андрій Олегович
(ППП)

Керівник канд. фіз.-мат. наук. доц. Шугайло Ю.Б.

Рецензент канд. фіз.-мат. наук., ст. н. с. Чечко В.Є.

Рекомендовано до захисту:

Протокол засідання кафедри

№ 10 від 31 травня 2017 р.

Захищено на засіданні ЕК № 4

протокол № 7 від 14.06 2017 р.

Оцінка Відмінно! А | 90
(за національною шкалою, шкалою ECTS, бали)

Завідувач кафедри

(підпис)

Альошин О.М.

Голова ЕК

(підпис)

Одеса – 2017

790276

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
РАЗДЕЛ 1. ПОСТАНОВКА ЗАДАЧИ	5
§1.1. Задача CSS препроцессора.....	5
РАЗДЕЛ 2. МЕТОДЫ И ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ВЕБ-РАЗРАБОТКИ	7
§2.1. Язык разметки HTML.....	7
§2.2. Каскадные таблицы стилей CSS.....	11
§2.3. Современные стандарты веб разработки.....	21
РАЗДЕЛ.3 ПРЕПРОЦЕССОРЫ CSS	25
§3.1. Препроцессор LESS.....	26
§3.2. Препроцессор SASS.....	29
§3.3. Сравнительный анализ синтаксиса.....	34
ВЫВОДЫ	37
ЛИТЕРАТУРА	38
ПРИЛОЖЕНИЕ А	39

ВВЕДЕНИЕ

На заре интернета визуальное представление документов целиком ложилось на плечи браузеров, которые использовали свои собственные встроенные стили для отображения различных HTML-тегов. Поскольку никакими стандартами на тот момент оформление не регламентировалось, производители браузеров в борьбе за неокрепшие умы пользователей украшали странички как могли, и в результате один и тот же документ мог разительно отличаться по внешнему виду в зависимости от ОС и браузера, в котором он отображался. Глядя на эту дикую вольницу, норвежский ученый-информатик Хокон Виум Ли в 1994 году предложил производителям браузеров концепцию каскадных таблиц стилей, призванную отделить оформление от структуры веб-страницы и целиком отдать его на откуп веб-разработчиков. Идея приглянулась свежесобранному консорциуму W3C, и после двух лет прений и согласований спецификация CSS 1.1 была выброшена прямо на поле боя разгоравшейся войны браузеров. В дальнейшем спецификация CSS пережила еще два мажорных релиза, обзавелась огромной кучей новых свойств и умений, но концептуально недалеко ушла от изначальных редакций: задание стилей, как и раньше, сводилось к монотонному перечислению свойств для каждого элемента индивидуально, браузеры исправно поставляли уникальные префиксозависимые функции. К тому же отсутствовали единые рекомендации по группировке (или сортировке) CSS-свойств внутри блоков, что, в свою очередь, порождало многочисленные противоречия. В таком виде CSS вступил в конец двухтысячных, своеобразный ренессанс фронтенд-разработки. CSS-разметка все чаще стала ложиться на плечи профильных фронтенд-разработчиков, которые быстро обнаружили, что в текущем виде CSS сложно поддается структуризации и автоматической генерации и не предоставляет никаких возможностей для гибкой и динамической работы со стилями элементов. Грядущий CSS3 обещал немного скрасить ситуацию, предложив

некоторые новые селекторы и встроенные средства для работы с анимацией, но необходимость в префиксах и отслеживании поддержки браузерами новых функций делала процесс еще более утомительным. В результате неумеренный энтузиазм Ruby- и JavaScript-разработчиков вылился в создание нескольких «семейств» CSS-препроцессоров, а также целых фреймворков на их основе. В большинстве своем препроцессоры CSS преследуют схожие цели: унификация браузерных префиксов и хаков, упрощение синтаксиса, вложенность селекторов и привнесение в описания стилей базовой логики. Таким образом, вместо сугубо декларативного перечисления свойств появляется возможность описывать стили, используя подходы и приемы, характерные для «полноценных» языков программирования: переменные, миксины (некоторое подобие шаблонных наборов свойств, позволяющее подстановки и замены внутри себя), а также циклы, условия и несложную математику.

ВЫВОДЫ

Целью моей работы был сравнительный анализ препроцессоров CSS. Каждый из препроцессоров, (Sass, Less), имеет свой уникальный подход к решению одной и той же задачи — дать разработчикам возможность использовать полезные функции, при этом сохраняя кроссбраузерность и чистоту кода.

Но так же у препроцессоров есть несколько существенных минусов. На изучение препроцессоров необходимо время. Для опытного разработчика изучение большинства возможностей препроцессоров занимает пару часов. Но если у разработчика нет навыков программирования кроме как CSS/HTML и/или отсутствуют знания основной логики программирования (циклы, функции, области видимости), это, возможно, займёт больше времени. Командная строка — это ещё одно препятствие для CSS/HTML разработчиков. Но не в этом цель статьи, я буду конкретно говорить о злоупотреблении этими возможностями. Я попробую объяснить большую часть основных проблем, которые я вижу, когда каждый раз кто-то показывает мне пример кода, или я вижу проект, написанный с использованием любого из этих препроцессоров.

Используя новый слой абстракции, разработчик не осознаёт, что он создаёт сильно дублирующийся код. Всегда сжимайте файлы CSS и JavaScript. Gzip очень хорошо сжимает дублирующиеся данные, так что эта проблема может быть несущественной на публикации. Необходимо учитывать, что сгенерированный CSS-код будет тяжелее поддерживать в случае, если будущие разработчики по каким-то причинам решат прекратить использование препроцессора и просто обновят сгенерированный CSS (может, у них не будет доступа к исходным файлам или не будет опыта работы с препроцессорами).

ЛИТЕРАТУРА

1. Shamir A. How to share a secret // Com. Of the ACM. - 1979. - Vol. 22, №11. - P.612-613.
2. Blakley G.R. JavaScript for Css preprocessing// Proc. Of AFIPSNasional ComputerConference. -1979. - 48. - P.313-317
3. Ященко В.В. Эффективное программирование. - Санкт-Петербург: МЦНМО, 2001. - 237 с.
4. Шнайер Б.Основа JavaScript. - М.: Изд-во Триумф, 2003. - 816 с.
5. Чмора А. Современные стандарты веб разработки. - М.: Гелиос АРВ, 2001. - 244 с.
6. Андрущенко А.С., Уривский А.В., Пьянов С.М. Основы самостоятельного изучения CSS: «НИР «Пронто-И», 2013. – 170 с.
7. Блэккли Р.Г. Проблемы передачи информации. – 1997. – Т.33. - №3. – С. 42-46.
8. Carnin E.D., Greene J.W., Hellman M.E. On Secret Sharing Systems // IEEE Trans. Inform. Theory. – 1983. – V.29. - №1. – P.231-241.
9. Спельников А.Б. HTML, CSS, JavaScript– Вест.
10. C. Asmuth, J. Bloom. A modular approach to CSS// Information Theory, IEEE Transactions on. — 1983. — В. 2. — Т. 29.
11. L. Harn, C. Lin. Detection and identification of cheaters in (t, n) . – Des. Codes – 52(1) – 2009 – P. 15-24