

## Использование надстройки LESS как динамический язык стилей

*Круглик Роман Игоревич*

*Приамурский государственный университет им. Шолом-Алейхема*

*Студент*

*Брыкова Анастасия Леонидовна*

*Приамурский государственный университет им. Шолом-Алейхема*

*Студент*

### Аннотация

В статье рассмотрен метод использования надстройки LESS. Разобран пример создания и динамическое изменение переменных. Наглядно показаны функции осветления и затемнения.

**Ключевые слова:** Программирование, динамичность, CSS, LESS.

### Use the add-in as a LESS dynamic language styles

*Kruglik Roman Igorevich*

*Sholom-Aleichem Priamursky State University*

*Student*

*Brykova Anastasia Leonidovna*

*Sholom-Aleichem Priamursky State University*

*Student*

### Abstract

In article describes the method of using add-ons LESS. The example create and dynamically changing variables. Clearly shows the functions of lightening and darkening.

**Keywords:** Programming, dynamic, CSS, LESS.

Каждый разработчик при использовании каскадной таблицы стилей как CSS сталкивался с проблемами сложного управления и ограниченностью ресурсов. С помощью надстройки LESS мы можем не запоминать и не переписывать по несколько раз один и тот же код, например #87CEFA для получения светло-голубого цвета, а передавать его в виде переменной, записанной заранее. При правильном использовании данного метода можно ускорить скорость вёрстки в несколько раз.

LESS - это динамический язык стилей. Его первая версия была написана на Ruby, однако в последующих версиях было решено отказаться от использования этого языка программирования в пользу JavaScript. Less - это

вложенный метаязык: валидный CSS будет валидной less-программой с аналогичной семантикой.

Исследования в использовании различных методов ускорения разработки не заканчиваются и по сей день. В статье А.И.Дмитрийчук выделяет модуль CSS Grid Layout для построения интерфейсов на основе двумерной сетки [1]. И.В.Смагина, О.А.Романова [2] предоставляют сравнительный анализ языка разметки HTML и каскадных таблиц стилей CSS. Н.А.Христинец, Я.П.Цяпич [3] рассказывают о таких расширениях как HAML/SASS. В статье [4] Е.В.Крылов, В.К.Аникин, А.А.Воронин рассмотрели подходы к повышению быстродействия интерпретации CSS. В.В.Бастрикина [5] показала наиболее известные адаптивные CSS фреймворки для ускоренной разработки.

Чтобы начать работать с данным методом нужно скачать Js файл с официального сайта разработчика <http://lesscss.org/> и привязать его к проекту как и любой другой js файл:

```
<script src="less.js" type="text/javascript"></script>
```

Теперь создадим файл с расширением less и привяжем его:

```
<link rel="stylesheet/less" type="text/css" href="style.less">
```

Подключаемые LESS файлы должны подключаться выше чем js файл с расширением.

Переменные в LESS работают так же как в большинстве других языков программирования. Создадим 3 поля для рисования типа canvas и на них посмотрим возможности работы с переменными. Для начала опишем переменные, которые начинаются с символа @. После чего они могут быть использованы во всём файле. Если разработчик захочет изменить цвет, то ему нужно отредактировать переменную, с которой связан данный цвет. В примере ниже показано присваивание переменных и их вызов:

```
@color-purple: #a76fe4;
@color-green: #64d45d;
@color-turquoise: #7ae8d9;
#canvas-top{
    background: @color-purple;
}
#canvas-center{
    background: @color-green;
}
#canvas-bottom{
    background: @color-turquoise;
}
```

Посмотрим на результат. (см. рис. 1)



Рисунок 1. Результат закрашивания

Теперь наши цвета могут динамически изменять сколько бы не было у нас полей и вызовов.

Существует очень много возможностей для работы с цветами при помощи LESS. Попробуем использовать ещё одну возможность. Функции `lighten` (освещение) и `darken` (затемнение) очень широко используются при разработке. Закрасим все наши `canvas` одинаковым цветом для наглядного виденья изменений. Первый прямоугольник осветим на 20%, второй затемним на 20%, а третий оставим в оригинальном состоянии.

```
@color-purple: #a76fe4;
#canvas-top{
  background: lighten(@color-purple, 20%);
}
#canvas-center{
  background: darken(@color-purple,20%);
}
```

```
#canvas-bottom{  
  background: @color-purple;  
}
```

Посмотрим на результат (см. рис. 2)



Рисунок 2. Результат управления цветом

Не всегда использование LESS надстройки является правильным. Зачем создавать сложную структуру для маленьких проектов или для одной всего лишь манипуляции. Цель состоит в том, чтобы объяснить, как используя LESS можно помочь сделать наш CSS более динамичным. CSS препроцессоры позволяют нам создавать и управлять CSS более эффективно. Даже имея хорошее понимание CSS, LESS позволит нам использовать его более продуктивно. Всё же данный метод ускоряет разработку сокращая количество кода и улучшая его структуру.

**Библиографический список**

1. Дмитрийчук А.И. CSS GRID LAYOUT - следующий шаг В CSS // Міжнародний науковий журнал інтернаука (Київ). 2017. №9. С. 35-36.
2. Смагина И.В., Романова О.А. Сравнительная характеристика приемов работы с html и css при создании веб-документов // Научные записки орелгизт. 2012. №1. С. 444-447.
3. Христинець Н.А., Цяпич Я.П. HAML/SASS Як Розширення для HTML/CSS при верстці веб-сайтів // Комп'ютерно-інтегровані технології: освіта, наука, виробництво .2014. №16-17.С. 46-50.
4. Крылов Е.В., Аникин В.К., Воронин А.А. Исследование подходов к повышению быстродействия интерпретации CSS //Адаптивні системи автоматичного управління. 2008. №13. С. 135-143.
5. Бастрикина В.В. Сравнительный анализ адаптивных css фреймворков // Актуальные проблемы авиации и космонавтики. 2016. №12. С. 595-596.