****

[ornaorg.github.io](http://ornaorg.github.io/)

**OrnaJS** — свободная JavaScript библиотека для динамической стилизации HTML элементов без прямого написания JS или CSS кода. Стилизация осуществляется методом прибавления классов к HTML элементу, как и в привычных CSS-библиотеках. В отличие от СSS-библиотек, Orna не ограничена количеством ранее прописанных в ней классов, а работает как парсер, динамически обрабатывает класс и задает стиль. Стилизация в OrnaJS основана по принципу Atomic CSS, из чего следует, что вид элемента создается по частям (атомам), но не одним классом. Орну можно сравнить с атрибутом style в HTML, но с наследованием, сокращениями и событиями. Имя Orna сокращение от Ornament (украшение).

В первые два месяца после публикации, хотя Орна предлагает нестандартный метод стилизации страниц, ее успели скачать более 2000 разработчиков, что говорит о растущей популярности динамической стилизации.

OrnaJS и Atomic CSS

**Atomic CSS** — методология написания CSS кода, которая соблюдает принцип «один класс = одно свойство». При написании стилей по методологии Atomic CSS, разработчик создает универсальные классы, которые в большинстве случаев несут только одно свойство и его значение. Она призвана убрать повторяемость, многословность, запутанность при работе CSS и также с CSS препроцессорах, Less, Sass. CSS является фундаментом для создания стилей он дает возможность сделать надстройки и разрабатывать удобные архитектуры для конкретных разработчиков, именно поэтому у него полная форма свойств и нет строгой архитектуры. архитектура Atomic CSS разрабатывается командой из Yahoo, OrnaOrg и всеми энтузиастами.  
  
Проект OrnaJS начался в 2015 году и изначально разрабатывался как инструмент для удобной реализации Atomic CSS архитектуры, но потом был подвержен реформации и сейчас соблюдает от части правила из Atomic CSS и от части из React CSS.

**React CSS** — реактивный CSS, в отличие от методологии Atomic CSS работает с динамическими стилями, которые создаются в процессе работы с элементами. React CSS методология, которая широко используется при создании SPA, где элементы разметки создаются динамически и изначально их еще нет. Команда Orna создала отдельную библиотеку для полной реализации Atomic CSS, которая предполагает уже описание атомарные стили, а не их динамическое создание, Orna4Node. Orna4Node автоматически генерирует CSS файл на основе классов прописанных в элементах, в отличие от OrnaJS которая не создает CSS файл, а на прямую, методами JavaScript обращается к DOM дереву. Orna4Node похожа по работе на Atomizer от Yahoo!..  
  
OrnaJS и jQuery

OrnaJS работает вместе с библиотекой jQuery, из которой вынесла основные селекторы и выборки по элементам. Orna работает с первой версией jQuery, со второй и с третьей, выбор поставлен на разработчика.  
  
**bgc\_red\_div:even** — класс задает красный цвет фона всем четным, дочерним div элементам.  
  
**bgc\_red\_div:odd** — класс задает красный цвет фона всем нечетным, дочерним div элементам.  
  
**bgc\_red\_div:first** — класс задает красный цвет фона первому дочернему div элементу.  
  
**bgc\_red\_div:last** — класс задает красный цвет фона последнему дочернему div элементу.  
  
**bgc\_red\_option:selected** — класс задает красный цвет фона всем выбраним дочерним элементам option.  
  
**bgc\_red\_input[type=text]** — класс задает красный цвет фона всем дочерним элементам input с типом текст.  
  
**bgc\_red\_#menu** — класс задает красный цвет фона дочернему элементу с id menu.  
  
**bgc\_red\_.menu**— класс задает красный цвет фона всем дочерним элементам с классом menu.

Основное предназначение

Основное предназначение библиотеки-парсера OrnaJS упростить работу при стилизации элементов разметки. Орна помогает избавиться от повторяемости стилей, чрезмерной многословности и что основное от работы над структуризацией СSS кода. Так как Орна переняла основные идеи от методологии Atomic CSS, при работе с ней все атомарно, то есть стиль собирается частицами и все стили ложатся независимо, чисто, без необходимости усердного рефакторинга. Также с использованием Орны больше нет необходимости задавать стили элементам при помощи jQuery или Native JavaScript, так как в Орне это гораздо проще, из-за уровня абстракции, что работает по принципу водопада

*html --> ornajs --> jquery --> native javascript*.

Использование

Использовать Орну можно и для полной стилизации странниц или как дополнение к основным каркасным стилям. В современной тенденции веб-разработки, где много динамики и состояний, разработчики все больше и больше отходят от статической HTML и СSS разметки, в сторону их динамического создания в процессе взаимодействия пользователе с интерфейсом. OrnaJS может быть использована в контексте с react.js, riot.js, handlebars.js, angular.js, bootstrap.  
  
Подключение библиотек:

<head>

<script src="jquery.js"></script>

<script src="orna.js"></script>

</head>

Структура классов в OrnaJS

property\_value  
color\_red  
  
property\_value\_event  
color\_red\_click  
  
property\_value\_childtagname  
color\_red\_div  
  
property\_value\_event\_childtagname  
color\_red\_click\_div  
  
property\_value\_childtagname\_event  
color\_red\_div\_click

Стилизация

<div class="bgc\_black c\_yellow h\_100px br\_20px shadow\_0\_0\_20px\_0\_red">Hello! I'm decorated by OrnaJS...</div>

<div class="bgc\_black\_a c\_yellow\_a h\_30px\_a br\_20px\_a">Hello! My (a)link decorated by OrnaJS...

<a href="#">1</a>

<a href="#">2</a>

<a href="#">3</a>

<a href="#">4</a>

</div>

DOM дерево

Если динамически изменить DOM дерево?

1. Все новые элементы автоматически следуют стилям, которые были указаны через родительские элементы для дочерних, без перезапуска Орны.  
  
Пример:

<div class = "text-decoration\_none\_a border\_none\_button">

<a>link</a>

<button>Click</button>

</div>

2. Если были добавлены новые элементы уже с прописанными в них классами OrnaJS, то чтобы эти классы присвоились надо перезапустить OrnaJS.  
  
Для этого запустить нужно функцию createatom(); Без аргумента функция стилизует все новые элементы, минуя старые, с аргументом конкретный новый элемент.  
  
В OrnaJS подключаются любой font, который есть на компьютере в системных шрифтах, или загружен прямо в СSS и в его названии нет пробелов, так как пробел OrnaJS понимает, как конец класса. Два шрифта Arial и Times New Roman подключаются одним словом (arial, times) просто для быстрого доступа, а вообще нужно писать класс: 

<span class="ff\_Verdana">Text</span>

<span class="font-family\_Verdana">Text</span>

Демо

[codepen.io/DimaPopov/pen/dXyZQR](http://codepen.io/DimaPopov/pen/dXyZQR)  
[codepen.io/DimaPopov/pen/JKEaby](http://codepen.io/DimaPopov/pen/JKEaby)  
[fiddle.jshell.net/dimapopov/hokLfqqo](https://fiddle.jshell.net/dimapopov/hokLfqqo/)  
[www.youtube.com/watch?v=k4mkK\_3YUk8](https://www.youtube.com/watch?v=k4mkK_3YUk8)

Ссылки

[ornaorg.github.io](http://ornaorg.github.io/)  
[ornaorg.github.io/AtomicCSS.html](http://ornaorg.github.io/AtomicCSS.html)  
[www.npmjs.com/package/ornajs](https://www.npmjs.com/package/ornajs)  
[github.com/OrnaOrg](https://github.com/OrnaOrg)  
[github.com/OrnaOrg/OrnaJS](https://github.com/OrnaOrg/OrnaJS)