Как мы уже выяснили, HTML отвечает за то, ЧТО будет выведено на экран. А вот за то, КАК оно будет выглядеть и где будут располагаться разные части страницы, отвечает css. Проще говоря HTML - это структура и содержание, а css - внешний вид.

Cascading Style Sheets - язык описания внешнего вида html документа. Основной целью разработки css, являлось разделение логической структуры страницы от описания внешнего вида документа. Это позволяет, для одного и того же html документа, создавать представления адаптированные для людей с ограниченными возможностями, оптимизированные под мобильные устройства, предназначенные для печати. Кроме того, последние версии css позволяют накладывать анимацию и, фактически, рисовать.

Способов создать css существует немало, рассмотрим только один самый простой вариант. Опишем стиль всех тегов p (это абзац) на странице:

<style>

**p** {

background-color: blue; */\* Фон \*/*

margin: 10px; */\* Отступы от границ блока с абзацем \*/*

font-family: Arial; */\* Шрифт \*/*

}

</style>

В таком виде стиль вставляется прямо на html страницу. В реальных приложениях стили описывают в отдельных файлах, например style.css и с помощью специальных тегов подключают на страницу:

<link rel="stylesheet" href="/vendor/github-markdown.css">

Гораздо чаще стиль задается не для всех тегов, а только для определенных "классов":

<style>

.strange {

font-size: 1.4em;

color: blue;

}

</style>

<body>

<h1 class="strange">Этот заголовок голубого цвета</h1>

</body>

Аттрибут class есть у каждого тега, а значит можно изменить внешний вид и расположение всего что представлено на странице (иногда с ограничениями).

В двух предыдущих примерах, p и .strange называются селекторами. Селекторы определяют набор элементов, к которым будут применяться указанные за ними css правила:

* \* – любые элементы.
* div – элементы с таким тегом.
* #id – элемент с данным id.
* .class – элементы с таким классом.
* [name="value"] – элементы с таким значением соответствующего свойства
* :visited – псевдоклассы

Более того, их можно комбинировать, что может значительно помочь сократить описание стилей.

Попробуйте создать index.html со следующим содержанием и открыть его в браузере:

**<!DOCTYPE html>**

<html>

<head>

<style>

**body** {

color: red;

}

.note {

color: grey;

background-color: yellow;

font-weight: bold;

}

</style>

</head>

<body>

<p style="font-size: 20px; color: green;">Inline Styled Paragrap</p>

<div class="note">Note Class</div>

</body>

</html>

Всего с помощью css можно менять сотни свойств относящихся к документу.

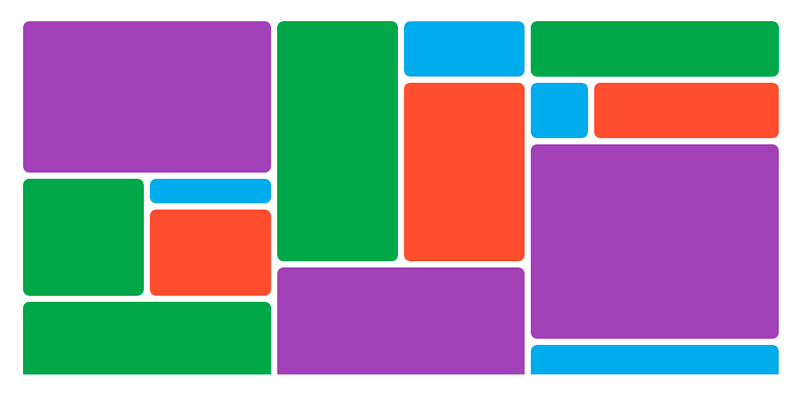
Каждый браузер содержит определенный набор стилей "по умолчанию". Этот набор применяется всегда и формирует базовое представление элементов. С одной стороны это избавляет от необходимости писать стили тогда, когда устраивает умолчание, но с другой появляются серьезные проблемы.

Этот набор отличается от браузера к браузеру. Все усугубляется тем, что и сами стили работают по разному в разных браузерах. В этот момент проявляется самая большая боль верстки. Необходимо знать тонкости старых и новых браузеров, причем, в современном мире, это не только браузеры для для обычных компьютеров, но так же браузеры под мобильные устройства и даже машины (например tesla) с телевизорами.

Есть [определенные инструменты](https://github.com/postcss/autoprefixer), позволяющие частично сгладить эту проблему, но системно, скорее всего, она не решится никогда и чем дальше, тем больше появляется подобных различий. Их можно увидеть на этой странице <http://caniuse.com/>.

Layouting

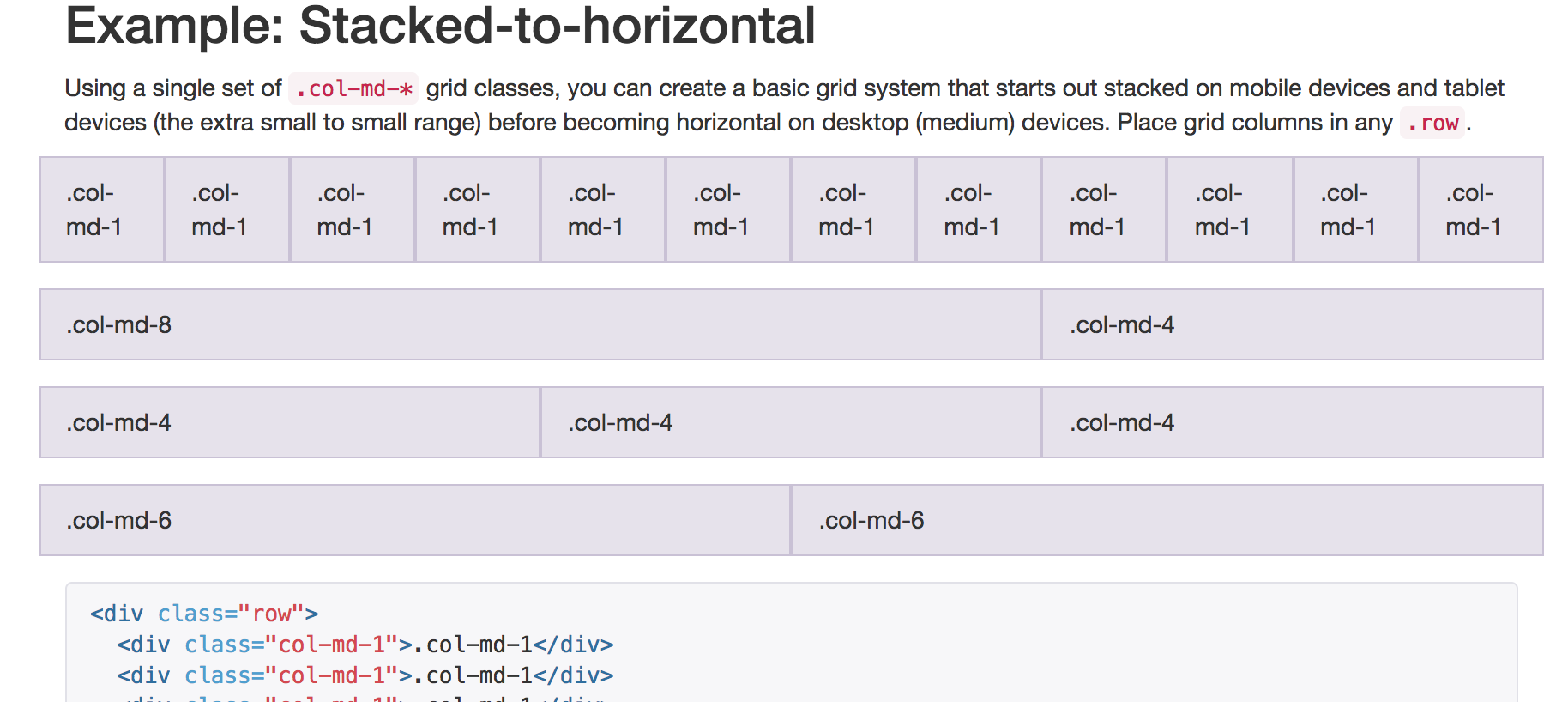
Отдельным большим блоком в css является позиционирование. Эта тема, наверное, самая сложная и крайне важная. Благодаря определенному набору свойств можно творить чудеса. Располагать элементы как угодно на странице, менять их размеры и вид, в зависимости от ширины экрана и даже скрывать.



Twitter Bootstrap

Раньше, каждый верстальщик копировал из проекта в проект заготовки css, для упрощения разработки (а кто-то делает это и сейчас). В конце концов, появились люди, которые увидели закономерности и смогли их обобщить. Так повились первые css фреймворки. Их центральным элементом была модульная сетка, набор стилей, в простейшем случае, позволяющий разбить все пространство на независимые блоки. По сути это каркас, на который уже можно нанизвывать конкретные элементы.

Дальнейшее их развитие привело к созданию таких монстров как [Bootstrap](http://getbootstrap.com/). Bootstrap - это больше чем просто css фреймворк. Кроме модульной сетки, Bootstrap поставляется с большим количеством компонентов, благодаря которым можно собрать сайт практически любой сложности. Ко всему этому стоит добавить то, что он автоматически хорошо работает и для разных размеров экранов и для мобильных устройств. А благодаря огромному комьюнити, он развивается огромными темпами.



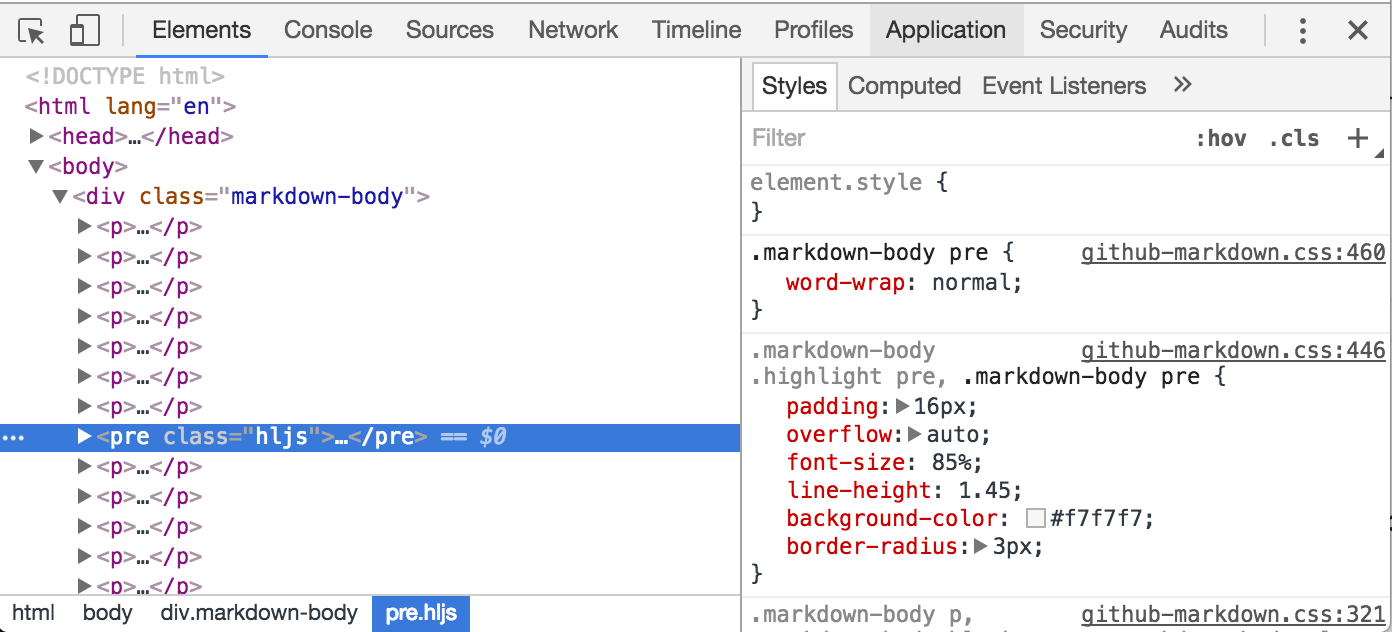
Хекслет, целиком и полностью построен на Bootstrap'e. Такой подход позволяет нам не отвлекаться на непрофильную деятельность и ускорить разработку проекта в разы.

Полезные ссылки

* [Справочник по HTML и CSS](https://webref.ru/)
* [Изучение лейаутинга](http://ru.learnlayout.com/)
* [Twitter Bootstrap](http://getbootstrap.com/)

DevTools

Инструментарий разработчика предлагаемый браузерами, позволяет просматривать не только html структуру, но и css. Более того, вы можете очень быстро проводить эксперименты, изменяя css прямо тут же.



Заключение

Как видно, владение css на должном уровне, совсем не тоже самое что и понимание HTML. Чтобы хорошо его применять, недостаточно прочитать справочник. Нужно целенаправленно учиться и практиковаться. Однако это не означает, что все его знают. Большинство веб-разработчиков знают css так себе. Все же это прерогатива верстальщиков, да и фреймворки, такие как, Bootstrap, позволяют оставаться в неведении.

**ТЕСТЫ**

За что отвечает css?

За внешний вид документа

За расположение элементов документа на странице

К чему будет применен данный стиль?

**p** {

color: white;

}

Ко всем абзацам

Как нужно назвать стиль чтобы он применился ко всем элементам с классом white?

Начало формы

 {

color: white;

}

.white

Что можно сделать с помощью css?

Настройку внешнего вида

Адаптацию страницы под разные экраны

Анимацию