**Средства Web-разработки**

Современные средства разработки в обасти создания, дизайна и сопровождения web-страниц представляют столь огромные возможности для всего вышеописанного что требуют от пользователей для создания сайтов порой не уступающих написанными полу-профессионалами, минимального багажа знаний и толику желания.

И начать стоит с получения удобного и мощного инструмента. Здесь все зависит от потребностей. Это может быть легковесный, но модульный **Brackets**.Для адаптивной верстки удобен **Adobe DreamWeaver**. Если вы рассчитываете много работать с PHP и JS, то вам подойдет **PhpStorm**. На крайний случай все можно проделывать и в Блокноте, дело лишь в удобстве.

Помочь оптимизировать рутинную работу с разработкой способны task-менеджеры. Программы **Gulp**, **Grunt**, **Broccoli**, коии хоть и относительно сложны в установке и настройке, но помогают существенно сократить время на сборку проекта, автоматически сохранять внесенные изменения и отправлять их на сервер. Надо отметить что для небольших проектов их плюсы несущественны.

CSS-препроцессоры и -постпроцессоры могут помочь с решением проблем, которые не решены и в нынешних версиях CSS, к примеру - отображение векторной графики, и вообщем ускорят работу с этим языком. Самые популярные из пре-процессоров на момент 2019-го года - **Sass**, **Stylus** и **JSS**. Представители группы постпроцессоров - **CSSO**, **PostCSS**.

Примеры **Markup/UI-фреймворков**, которые превратят практически любого желающего в веб-дизайнера среднего уровня - **Bootstrap**, **Elemental UI**, **Material Lite**. Удается им это за счет бесчетного множества модулей и шаблонов, созданных на их основе, удобных систем разметки, заранее заложенных стилей и возможности редактировать их прямо в HTML коде путем добавления и редактирования классов. На их основе удобно строить и небольшие красивые сайты-одностраничники (из-за удобства работы с CDN, ныне использующимся очень часто) и большие проекты (поскольку их можно установить автономно и редактировать, подстраивая под свои нужды).

Если же вы даже не можете организовать рабочий процесс или придумать систему именования, то можете перенять это из сервисов предоставляющих собственную методологию разработки - **BEM**, **OOCSS**. В них можно найти информацию по составлению, ведению и работе с большинством популярных языков. Удобная файловая структура позволит сделать все предельно понятным и читаемым для кого угодно. Они помогут “правильно” начать работу и будут полезны новичкам или тем, кто не хочет заново изобретать велосипед.

Одним из финальных этапов разработки является проверка и оптимизация работы кода. И для этого придуманы программы, тестирующие различные параметры работы JS кода или запросов. Они прямо “укажут” на недостатки, после чего останется их всего-лишь решить.

Среди всего этого обилия программ и сервисов упрощающих жизнь важно не забывать что программист - профессия *человека*. И если что-то не получается понять или воплотить в коде - лучше поинтересоваться у знающих людей, желающих помочь. Для этого есть форумы и чаты, такие как **chats**, **IRC**, **meetups**. Но помните о золотом правиле - сначала искать в интернете, потом - проверить документацию языка/программы, затем - еще раз поискать и только если ваше любопытсво не исчерпано, а код по-прежнему не работает - вам вероятнее всего помогут. Программисты довольно доброжелательны, в большинстве своем.

Хочу оговориться - в данной статье CMS я рассматривать не буду, поскольку они относительно просты и пояснения в работе не требуют. Установил с нужным шаблоном, подогнал под себя - и все работает. Визуальные средства разработки - это отдельная тема. Да, они делают все красиво, но код, генерируемый ими, очень не оптимизирован и нелогичен для человека, что делает очень неудобным чтение кода таковой страницы.

А теперь стоит обсудить языки веб-программирования давно закрепившиеся в этой области программирования. Первые места по популярности и регулярности использования занимают HTML, CSS, JS, PHP и СУБД MySQL. Давайте разберем назначение и возможности, которые они могут предоставить.

HTML - основа любого сайта, “скелет сайта”. HTML довольно слаб сам по себе, но незаменим в тандеме с другими языками. Основные средства разметки позволяют создавать строчные и блочные элементы с различными классами для взаимодействия в основном с CSS и ID, позволяющие работать в JS с отдельными элементами.

CSS отвечает за визуальное отображение объектов, внешнее оформление нашего “скелета”. Все многообразие дизайнерский решений которое можно увидеть в интернете - это он. CSS за счет создания и возможности изменения классов отлично работает с JS. Хотя CSS и сам по себе может реализовывать какие-то простые операции типа изменения внешнего вида при наведении на элемент, создания градиента, анимации и задания ее скорости, создания и трансформации геометрических фигур.

JS - скрипты, функции и все вот это. Это “внутренние органы”, выполняющие определенный скрытый код не видимый пользователю, но явно заставляющий все работать. Возможности этого языка описывать бесполезно - они растут каждый день, расширяясь за счет надстроек и фреймворков. Универсальная и многофункциональная мощь, позволяющая при знаниях и желании реализовать любую задумку.

PHP - имеет свои задачи. Если применять его к нашему человеку - “связь с космосом”. Именно такое определение - поскольку он позволяет работать с удаленными БД, извлекая и преобразуя информацию для работы с ней на машине клиента. Особое удобство предоставляет система запросов позволяющая включить код любого файла в состав данного, что удобно для добавления элементов, присутствующих на всех страницах сайта.

MySQL являет собой создание, хранение изменение БД, удобно используемых на сайте через PHP запросы. Реляционная система хранения, удобство создания ключей и связи отношений - это лишь немногие плюсы.