**Вопрос 1. Почему HTML и CSS нельзя считать полноценными языками программирования?**

1. Попробую ответить на этот вопрос своими словами, как я это понимаю.

Язык программирования (далее – ЯП) – это некая система формальная знаковая система, которая умеет или должна уметь программировать (по сути писать) некую определенную логику. ЯП описывает функцию, при его помощи можно производить различные вычисления.

HTML и CSS описывают скорее представление, а не функцию. Их основная задача – отображение данных, а не их обработка (как в случае с ЯП). И, конечно, на HTML и CSS нельзя произвести вычисления. HTML интерпретируется клиентами (браузером, например).

Хотя, конечно, строго говоря, это так же языки – разметки документа (HTML) и декларативный язык описания внешнего вида документа (CSS), написанного на HTML – по сути набор стандартов (CSS).

1. Подкрепим мое видение официальными источниками=)

Обратимся к определению ЯП[[1]](#footnote-1).

ЯП – это формальная знаковая система, предназначенная для записи компьютерных программ. Язык программирования определяет набор лексических, синтаксических и семантических правил, задающих внешний вид программы и **действия**, которые выполнит исполнитель (компьютер) под её управлением *(источник – открытые интернет-источники, Википедия*).

Другими словами, ЯП – это определенное множество заранее определенных, однообразных и понятных исполнителю инструкций для выполнения их устройством. В данном случае исполнителем может выступать машина (компьютер), программист, интерпретатор, компилятор. При этом должны соблюдаться определенные правила или требования:

* инструкций должно быть ограниченное количество, и все их должны знать и понимать однозначно;
* инструкции должны выстраиваться определенным образом для получения определенных результатов, и все должны об этом знать;
* должны существовать правила написания инструкций и все должны их знать; каждая конструкция языка должна однозначно соотносить то, что написано, с тем, что требовалось обозначить.

Проще говоря, ЯП умеет или должен уметь программировать некую логику.

К ЯП применим и такой критерий, как ‘тьюринг-полный язык’[[2]](#footnote-2). ЯП считается тьюринг-полным, если на нем можно реализовать любую вычислимую функцию (то есть если любая вычислимая функция, которая вам нужна, может быть записана на этом языке и решена его исполнителем).

Теперь «навесим» это определение и свойства на языки HTML, CSS. Получается:

HTML, CSS – языки, так как они имеют характеристики или свойства языка: это система знаков, имеет свой синтаксис, семантику, лексику.

При этом к HTML, CSS не применимо определение ‘тьюринг-полного языка’, так как на них можно совершить только некоторую ограниченную совокупность действий, интерпретируемых клиентом.

Невозможно обработать данные, только отобразить.

*Источники:*

1. Материалы курса лекций TechMind
2. Википедия
3. Habr.com

**Вопрос 2. Какой формат верстки вы порекомендуете для блога?**

Мы говорим о блоге, а значит, такой тип информации будет «читаться» на различных устройствах (планшетах, ПК, мобильных телефонах) + контент может быть абсолютно разнообразным (типы информации – картинки, текстовые блоки, динамические кадры, интерактивный контент, видео и т.д.).

Также мы говорим об идеальной ситуации: у нас есть команда хороших разрабов (верстальщик в нашем случае) и есть деньги у Заказчика.

! По критерию изменяемости, конечно, стоит выбрать отзывчивую верстку, так как в отличие от адаптивной, например, (что тоже вполне может подойти), она не требует четкого определения (прописывания) размеров, при которых будет происходить изменение отображения. То есть без учета конфигурации.

По критерию конфигурации текста, на мой взгляд, наиболее удачным будет выбор семантической верстки. По сути та же блочная, но на HTML 5. Текст страницы будет разбит на блоки не просто структурно, а с учетом семантики (то есть использование блоков по назначению – с наименованием). Выбор этой верстки упростит работу разработчика и одновременно информация для пользователя (читателя) визуально будет более восприимчива.

! Если же Заказчик располагает достаточным количеством средств (денежных), я бы рекомендовала предусмотреть и мобильную версию сайта (блога), так как скорее всего таким продуктом будут часто пользоваться с телефона. Но здесь нужны вести диалог с Заказчиком, так как мобильная версия 1) удорожает разработку и 2) подчеркивает основные функции продукта (это все же абсолютно иной продукт получится).

**Вопрос 3. Как снизить нагрузку на Back-end с помощью Javascript?**

JavaScript – основной язык программирования для Front-end. Самый популярный язык на текущий момент. Кроме основных функций JS, данный ЯП позволяет реализовывать логику на стороне клиента. И это основная функция JS как фронтэндного ЯП.

То есть вся логика, которая касается клиентской части, может настраиваться, изменяться и храниться на клиенте (например, в Local Store, Session Store, Cookie – локальные БД для клиентской части), тем самым немного разгружая Back-end.

Например, настройка логики отображения таблиц или другого контента на клиенте -🡪 сохранение в локальные БД клиентской части (не складываем в Back-end) 🡪 при последующих входах на страницу будут отображаться настроенный контент (без обращения к Back-end’у). Еще одним примером может служить валидация отдельных данных (полей, формы…) на клиенте (по сути та же предустановка), без отправки данных для проверки на сервер и затем возвращения клиенту.

Динамические страницы (сгенерированные или сформированные по шаблону в процессе исполнения запроса пользователя). В админке осуществляется, например, настройка страницы, и при следующем входе на сайт будет отображаться тот или иной шаблон. Это может осуществлять через Back-end (генерироваться на фронте, а затем передаваться на бэк (HTML-документ), сохраняться, заполняться на бэке 🡪 передается в браузер), а может осуществляться на клиентской части в случае использования различных JS фреймворков. Но надо понимать, что это скорее кастомная настройка, то есть без участия бэка это не будет работать на любом устройстве или в любом браузере (то есть это актуально, например, для какого-то браузера конкретного, на котором была установлена эта настройка).

Условно сюда можно отнести и технологию AJAX. Асинхронный обмен данными с веб-сервером (без перезагрузки страницы, в фоновом режиме). То есть получение данных условно порционно, постепенно, без больших, тяжелых запросов на сервер.

1. Обращаться к общепринятым определениям HTML и CSS пока не будем, так как в них содержится достаточно однозначная трактовка языков как языков НЕ программирования, а языков разметки и описания внешнего вида документа (выражаясь более бытовым языком). [↑](#footnote-ref-1)
2. Алан Тьюринг - английский математик, который придумал абстрактный вычислитель — машину Тьюринга и дал определение множества функций, вычислимых посредством машин Тьюринга. [↑](#footnote-ref-2)