ЯЗЫКИ ВЕБ-ПРОГРАММИРОВАНИЯ

Выполнил: студент группы 12ПКС 9-4 Осипов Виктор

История развития WEB

История развития WEB

Веб (англ. *web* — паутина) — интернет-пространство.

World Wide Web (www, web, рус.: веб, Всемирная Паутина) — распределенная информационная система,
предоставляющая доступ к гипертекстовым документам по протоколу HTTP.

WWW — сетевая технология прикладного уровня стека TCP/IP, построенная на клиент-серверной архитектуре и использующая инфраструктуру Интернет для взаимодействия между сервером и клиентом (www).

Серверы www (веб-серверы) — это хранилища гипертекстовой (в общем случае) информации, управляемые специальным программным обеспечением.

Документы, представленные в виде гипертекста называются веб-страницами. Несколько веб-страниц,
объединенных общей тематикой, оформлением, связанных гипертекстовыми ссылками и обычно находящихся на одном и том же веб-сервере, называются веб-сайтом.

Для загрузки и просмотра информации с веб-сайтов используются специальные программы — браузеры,
способные обрабатывать гипертекстовую разметку и отображать содержимое веб-страниц.

Браузер может обратиться к веб-серверу по доменному имени или по ip-адресу, передавая в запросе идентификатор требуемого ресурса. Получив запрос от клиента, сервер находит соответствующий ресурс на локальном устройстве хранения и отправляет его как ответ. Браузер принимает ответ и обрабатывает его соответствующим образом, в зависимости от типа ресурса

Сеть ARPANET

История развития WEB

Сеть ARPANET.

Предпосылкой создания сети Интернет послужила потребность Министерства обороны США в надежной системе передачи информации во время холодной войны.

План создания компьютерной сети (названной «**ARPANET**») был представлен в октябре 1967 года, а в декабре 1969 была запущена в работу первая сеть из четырех компьютеров. Основная проблема при создании сети состояла в том, как соединить отдельные физические сети, не связывая сетевые ресурсы постоянными каналами.

Решение - разбиение запросов данных на небольшие фрагменты, «пакеты», которые могут быстро обрабатываться, не блокируя коммуникацию других частей — этот принцип все еще используется сегодня для работы Интернет.

Концепция получила широкое признание с появлением нескольких других сетей, использующих тот же самый метод коммутации пакетов ,сформировавшая основу первой университетской сети Великобритании . Эти сети, несмотря на наличие множества соединений, были в большей степени частными сетями, чем сегодняшней сетью Интернет.

Протокол TCP/IP

История развития WEB

Протокол TCP/IP.

Распространение различных сетевых протоколов стало вскоре проблемой при попытке заставить общаться все эти отдельные сети. Однако в поле зрения имелось решение — Роберт Кан, во время работы над проектом спутниковой пакетной сети ARPA, начал определять некоторые правила для более открытой сетевой архитектуры для замены используемого в ARPANET протокола. Позже, вместе с Винтоном Серфом из Станфордского университета, они создали систему, которая маскировала различия между сетевыми протоколами с помощью нового стандарта. В публикации, посвященной проекту спецификации в декабре 1974 года, он был назван «Internet Transmission Control Program».

Эта спецификация уменьшила роль сети и перенесла ответственность за поддержание целостности передачи на хост-компьютер. Конечным результатом этого было то, что стало возможно легко соединять почти любые сети. ARPA профинансировала разработку программного обеспечения и в 1977 году была проведена успешная демонстрация коммуникации трех различных сетей. К 1981 спецификация была завершена, опубликована и принята, и в 1982 году соединения ARPANET за пределами США были переведены на использование нового протокола «TCP/IP».

Появилась сеть Интернет. Встал вопрос о создании системы распределенного доступа к сетевым файлам.

В начале 1990 годов в качестве системы извлечения информации в глобальной сети использовалась система Gopher, которая предоставляла меню ссылок на файлы, компьютерные ресурсы и другие меню. Gopher была создана в Университете Миннесоты. В феврале 1993 университет объявил, что собирается требовать лицензионные отчисления за использование своей эталонной реализации сервера Gopher. Вследствие этого многие организации начали искать

Создание всемирной паутины

Создание Всемирной паутины.

Европейский совет по ядерным исследованиям в Швейцарии имел такое альтернативное решение. **Тим Бернерс-Ли** работал над системой управления информацией, в которой текст мог содержать связи и ссылки на другие работы, позволяя читателю быстро перемещаться от документа к документу. Он создал сервер CERN httpd для публикации документов такого вида, а также программу для их чтения, которую назвал «World Wide Web». Это программное обеспечение было выпущено впервые в 1991 году. 30 апреля 1993 года CERN выпустил исходный код World Wide Web во всеобщее достояние, поэтому кто угодно мог использовать или применять это программное обеспечение без всякой платы.

В 1994 году Тим Бернерс-Ли основал World Wide Web Consortium (W3C) в Массачусетском технологическом институте (Massachusetts Institute of Technology) при поддержке CERN, DARPA (в которую была переименована ARPA) и Европейской Комиссии. Консорциум W3C видел свою задачу в стандартизации протоколов и технологий, которые используются для создания Web, чтобы информационное содержание было доступно как можно большему числу жителей всего мира.

В течение нескольких следующих лет W3C опубликовал несколько спецификаций (называемых «рекомендациями»), включая HTML, формат изображений PNG (Portable Network Graphics), и каскадные таблицы стилей (CSS).

Языки WEB-программирования

Коротко о главном

Языки веб-программирования

Языки веб-программирования — это языки, которые в основном предназначены для работы с веб-технологиями. Языки веб-программирования можно условно разделить на две пересекающиеся группы: клиентские и серверные.

WEB-программирование

Веб-программирование — раздел программирования, ориентированный на разработку веб-приложений (программ, обеспечивающих функционирование динамических сайтов Всемирной паутины).

Клиентские языки программирования

Клиент - это сам пользователь, а точнее браузер установленный на вашем компьютере. Когда пользователь Интернета обращается с запросом к серверу, то он является клиентом.

Клиентские языки выполняются на компьютере пользователя (клиента). Если говорить более точно, то их выполняет сам браузер. Обычно клиентские языки встраиваются в html-код web-страницы. Таким образом, чтобы увидеть код, достаточно открыть веб-страницу с помощью любого текстового редактора или браузера.

Минусы:

Один из основных недостатков клиентских языков заключается в том, что они не могут взаимодействовать с сервером, на котором расположен сайт. Нет возможности сохранять и загружать информацию с него. Это значительно ограничивает применение клиентских языков. Например, о таких программах, которые должны сохранять и загружать определенную информацию с /на сервер: гостевых книгах, блоках комментариев, рейтингах, голосованиях и.т.д. можно вообще забыть. Кроме того, раз любой пользователь может очень легко просмотреть код программы, то о безопасности можно тоже забыть. Если Вы захотите закрыть доступ к некоторым файлам, с помощью таких языков, это вряд ли удастся. Очень просто просмотреть HTML – код web-страницы, чтобы узнать всю защищенную информацию и пароли.

Плюсы:

Они не отправляют данные на сервер, что делает их работу значительно быстрей.Не требуется никакого дополнительного программного обеспечения, все необходимое есть в браузере клиента. За пользование серверными языками компания, предоставляющая хостинг, требует денег (платный хостинг), а за клиентскими нет.

Клиентские языки программирования

Самыми распространенными клиентскими языками являются: **JavaScript** и **VisualBasicScript** **(VBS)**. Для того, чтобы браузер мог их понимать и выполнять в него встроен специальный инструмент – интерпретатор.

Интерпретатор — программа (разновидность транслятора), выполняющая *интерпретацию.*
Интерпретация — построчный анализ, обработка и выполнение исходного кода программы или запроса
(в отличие от компиляции, где весь текст программы, перед запуском, анализируется и транслируется в
байт-код, без её выполнения).

JavaScript, разработан компанией **Netscape** и первоначально использовался только для браузера **Netscape Navigator**. В настоящее время этот язык получил очень большую популярность. **VisualBasicScript (VBS)** это аналог клиентского языка от компании **Microsoft**.

Как уже говорилось ранее, все эти языки работают в обычных браузерах без всяких дополнительных модулей и плагинов. Самые распространенные браузеры это: Internet Explorer, Opera, Mozilla, Google Chrome.

Серверные языки программирования

Серверные языки выполняются непосредственно на самом сервере, специальной программой. Это значит, что для того, чтобы они работали не важно, каким браузером пользуется пользователь, все равно все вычисления будут проходить на удаленном компьютере (сервере).

Увидеть код программы на серверном языке для посетителя сайта вообще невозможно, он видит только результаты работы, которые будут уже представлены в качестве HTML-страницы.
Серверные языки предоставляют веб-программисту гораздо больше возможностей, чем клиентские. Используя их можно обмениваться данными с сервером, чего мы были лишены при использовании клиентских языков.

Наибольшую популярность, среди серверных языков получили: **PHP** и **Perl**.

Т.к. серверные языки взаимодействуют с удаленным компьютером (сервером), возникает необходимость где-то хранить результаты этого взаимодействия. Возникает вопрос: как и где хранить эти данные?
По мере роста любого сайта, приходиться хранить огромное количество информации. Именно для этой цели существует **База Данных (БД).** Это своего рода библиотека, где вся информация аккуратно разложена по полочкам. Но просто одной **БД** еще не достаточно. Когда информации становиться много, найти то, что нам необходимо становиться очень трудно, не говоря уже, уже о том, чтобы всю информацию обработать или извлечь только нужные данные. Так почему бы не поручить это компьютеру? Так и поступили. В настоящее время БД успешно обрабатываются компьютерами. Системы, которые за все это отвечают, называются **Системами Управления Базами Данных (СУБД).**

Серверные языки программирования

В них можно хранить всю необходимую Вам информацию, даже пароли (причем в зашифрованном виде). Взаимодействие пользователя с БД обеспечивает специальный язык запросов - **Structured Query Language (SQL)** (Язык структурированных запросов).
Самым известным **СУБД** является **MySQL.**

HTML

HTML

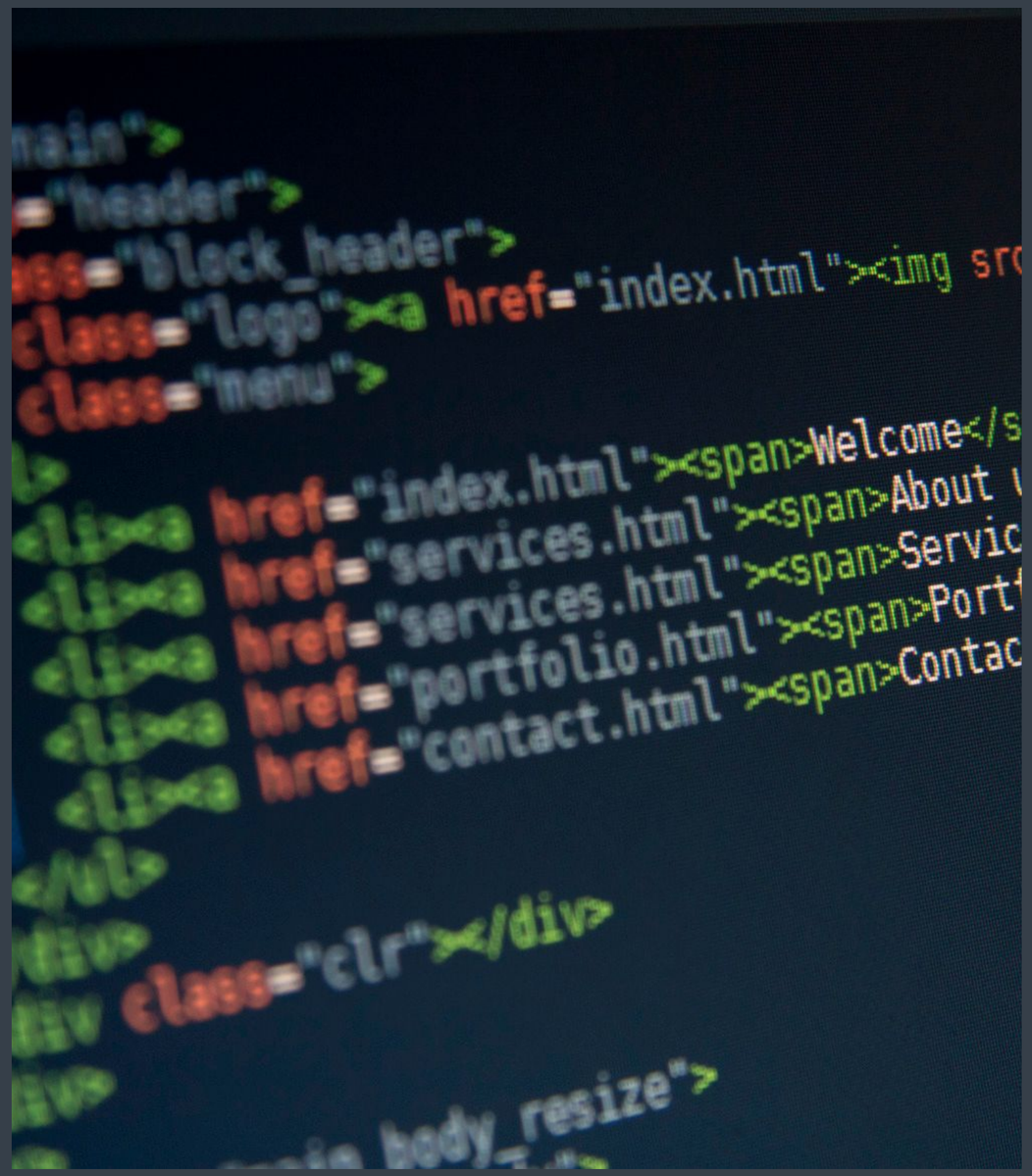
HTML - это сокращение от "HyperText Mark-up Language/язык гипертекстовой разметки"

Говоря кратко, HTML был изобретён в 1990 году учёным, Тимом Бёрнсом-Ли (Tim Berners-Lee), и предназначался для облегчения обмена документами между учёными различных университетов. Проект имел больший успех, чем Tim Berners-Lee мог ожидать. Этим изобретением HTML он заложил основы современной сети Internet.

HTML это язык, который позволяет представлять информацию (например, научные исследования) в Internet. То, что вы видите при просмотре страницы в Internet, это интерпретация вашим браузером HTML-текста.

HTML — теговый язык разметки документов. Любой документ на языке HTML представляет собой набор элементов, причём начало и конец каждого элемента обозначается специальными пометками — тегами.

В настоящее время Консорциум Всемирной паутины разработал HTML версии 5 , который расширяет HTML для лучшего представления семантики различных типичных страниц , например форумов, сайтов аукционов и т.д.



HTML5

HTML5 — это не продолжатель языка разметки гипертекста, а новая открытая платформа, предназначенная для создания
веб-приложений использующих аудио, видео, графику, анимацию и многое другое. Он расширяет, улучшает и рационализирует
разметку документов, а также добавляет единый API для сложных веб-приложений

SVG

SVG — язык разметки масштабируемой векторной графики, созданный Консорциумом Всемирной паутины (W3C) и входящий в подмножество расширяемого языка разметки XML, предназначен для описания двумерной векторной и смешанной векторно/растровой графики в формате XML.

Семантика

Семантические замены для использования универсальных блочных ( ) и строчных () элементов, например, <nav> (блок навигации по сайту), <footer> (обычно относится к нижней части страницы или последней строке HTML кода) или <audio> и <video> вместо <object>

API

Возможности DOM расширены и фактически используемые свойства задокументированы.
Элемент холст для непосредственного метода рисования в 2D,
контроль над проигрыванием медиафайлов;
File API: возможность загрузки документа через выбор (тег <input type="file">) или перетаскиванием (Drag-and-drop);
тип MIME и регистрация обработчика протокола;
Микроданные.

CSS

CSS

CSS (англ. Cascading Style Sheets — каскадные таблицы стилей) — формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

Правила CSS пишутся на формальном языке CSS и располагаются в таблицах стилей, то есть таблицы стилей содержат в себе правила CSS. Эти таблицы стилей могут располагаться как в самом веб-документе, внешний вид которого они описывают, так и в отдельных файлах, имеющих формат CSS. (По сути, формат CSS — это обычный текстовый файл. В файле .css не содержится ничего, кроме перечня правил CSS и комментариев к ним.) То есть, эти таблицы стилей могут быть подключены, внедрены в описываемый ими веб-документ четырьмя различными способами:

1) когда таблица стилей находится в отдельном файле, она может быть подключена к веб-документу посредством тега , располагающегося в этом документе между тегами и . (Тег будет иметь атрибут href, имеющий значением адрес этой таблицы стилей). Все правила этой таблицы действуют на протяжении всего документа;

   ```
   <link rel="stylesheet"
   href="/public/fonts/font-awesome/css/font-awesome.min.css">
   ```

2) когда таблица стилей описана в самом документе, она может располагаться в нём между тегами (которые, в свою очередь, располагаются в этом документе между тегами и). Все правила этой таблицы действуют на протяжении всего документа;

   ```
   <style>body{color: red;}</style>
   ```

CSS

3) когда таблица стилей описана в самом документе, она может располагаться в нём в теле какого-то отдельного тега (посредством его атрибута style) этого документа. Все правила этой таблицы действуют только на содержимое этого тега.

```
<p style="color:red;"></p>
```

Стили CSS по сравнению с HTML имеют гораздо больший спектр возможностей по дизайну элементов страниц сайта. Простыми методами изменяется цвет фона элемента, добавляется рамка, устанавливается шрифт текста, определяются размеры и относительное расположение элемента страницы, и другие элементы дизайна сайта.

Для создания структуры страниц сайта используется метод блочной верстки. При помощи тэга DIV с присвоенным ему стилем создается структура страницы сайта. В данное время это наиболее распространенный метод создания страниц сайта.

CSS3

CSS3

По сути это все тот же CSS, с набором новых аргументов, которые дают дополнительные возможности в плане различных эффектов. Например, свечение текста. Многие браузеры не поддерживают такие свойства, поэтому рекомендуется использовать
такие эффекты в минимальных количествах.

Скругленные рамки

SVG — язык разметки масштабируемой векторной графики, созданный Консорциумом Всемирной паутины (W3C) и входящий в подмножество расширяемого языка разметки XML, предназначен для описания двумерной векторной и смешанной векторно/растровой графики в формате XML.

Анимация

CSS3-анимация придаёт сайтам динамичность. Она оживляет веб-страницы, улучшая взаимодействие с пользователем.
Создание анимации начинается с установки ключевых кадров правила @keyframes. Кадры определяют, какие свойства на каком шаге будут анимированы. Каждый кадр может включать один или более блоков объявления из одного или более пар свойств и значений.

Градиенты

CSS3-градиент представляет собой переходы от одного цвета к другому.
Градиенты создаются с помощью функций linear-gradient() и radial-gradient().
Линейный градиент создается с помощью двух и более цветов, для которых задано направление, или линия градиента.
Если направление не указано, используется значение по умолчанию — сверху-вниз.
Цвета градиента по умолчанию распределяются равномерно в направлении, перпендикулярном линии градиента.

WRITE A TITLE IN THIS SECTION

Full Width Text Sample

First: What is a Fast-Fish? Alive or dead a fish is technically fast, when it is connected with an occupied ship or boat, by any medium at all controllable by the occupant or occupants,—a mast, an oar, a nine-inch cable, a telegraph wire, or a strand of cobweb, it is all the same. Likewise a fish is technically fast when it bears a waif, or any other recognized symbol of possession; so long as the party waiting it plainly evince their ability at any time to take it alongside, as well as their intention so to do. First: What is a Fast-Fish? Alive or dead a fish is technically fast, when it is connected with an occupied ship or boat, by any medium at all controllable by the occupant or occupants,—a mast, an oar, a nine-inch cable, a telegraph wire, or a strand of cobweb, it is all the same. Likewise a fish is technically fast when it bears a waif, or any other recognized symbol of possession; so long as the party waiting it plainly evince their ability at any time to take it alongside, as well as their intention so to do.

First: What is a Fast-Fish? Alive or dead a fish is technically fast, when it is connected with an occupied ship or boat, by any medium at all controllable by the occupant or occupants,—a mast, an oar, a nine-inch cable, a telegraph wire, or a strand of cobweb, it is all the same. Likewise a fish is technically fast when it bears a waif, or any other recognized symbol of possession; so long as the party waiting it plainly evince their ability at any time to take it alongside, as well as their intention so to do. First: What is a Fast-Fish? Alive or dead a fish is technically fast, when it is connected with an occupied ship or boat, by any medium at all controllable by the occupant or occupants,—a mast.

JavaScript

JavaScript

Javascript – это язык программирования, с помощью которого веб-страницам придается интерактивность. С его помощью создаются приложения, которые включаются в HTML-код (например, анкеты или формы регистрации, которые заполняются пользователем).

С помощью Javascript можно изменять страницу, изменять стили элементов, удалять или добавлять теги. С его помощью можно узнать о любых манипуляциях пользователя на странице (прокрутка страницы, нажатие любой клавиши, клики мышкой, увеличение или уменьшение рабочей области экрана…) Через него можно к любому элементу HTML-кода получить доступ и делать с этим элементом множество манипуляций. Можно загружать данные не перезагружая страницу, выводить сообщения, считывать или устанавливать cookie и выполнять множество других действий.

Вся уникальность данного языка программирования заключается в том, что он поддерживается практически всеми браузерами и полностью интегрируется с ними, а все что можно сделать с его помощью – делается очень просто. Ни одна другая технология не вмещает в себе все эти преимущества вместе.

JavaScript

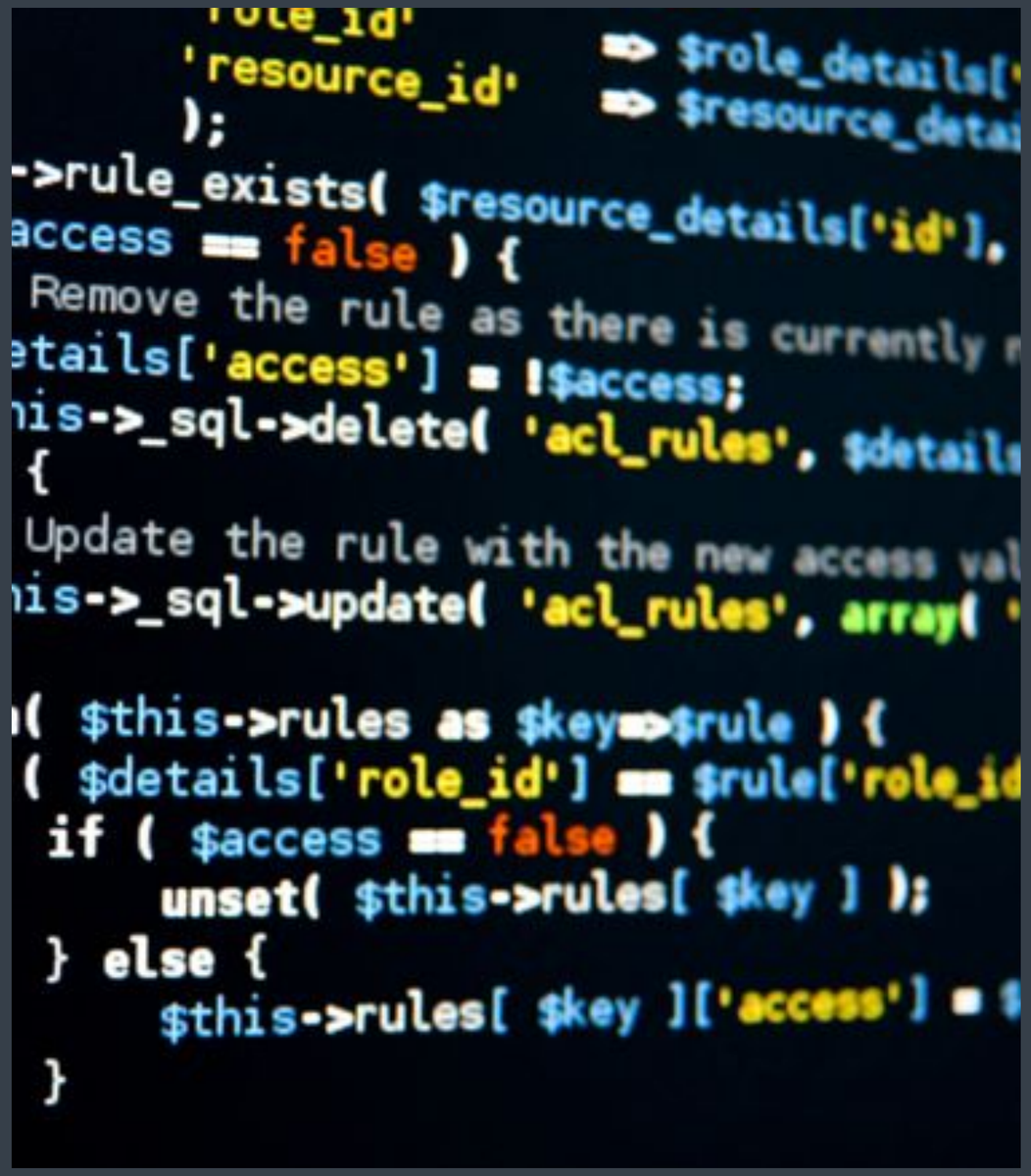
Для добавления JavaScript-кода на страницу, можно использовать теги , которые рекомендуется, но не обязательно, помещать внутри контейнера . Контейнеров <script> может быть сколько угодно.
Так же можно подключить и внешние файлы с JS кодом,желательно перед закрывающимся тегом <body> </body>.

PHP

Языки WEB- программирования

PHP

PHP – это широко используемый язык сценариев общего назначения с открытым исходным кодом. Говоря проще, PHP это язык программирования, специально разработанный для написания web-приложений (сценариев), исполняющихся на Web-сервере. Аббревиатура PHP означает “Hypertext Preprocessor (Препроцессор Гипертекста)".



Популярность в области построения веб-сайтов определяется наличием большого набора встроенных средств для разработки веб-приложений. Основные из них:

1) автоматическое извлечение POST и GET-параметров, а также переменных окружения веб-сервера в предопределённые массивы;

2) взаимодействие с большим количеством различных систем управления базами данных (MySQL, MySQLi, SQLite, PostgreSQL, Oracle (OCI8), Oracle, Microsoft SQL Server, Sybase, ODBC, mSQL, IBM DB2, Cloudscape и Apache Derby, Informix, Ovrimos SQL, Lotus Notes, DB++, DBM, dBase, DBX, FrontBase, FilePro, Ingres II, SESAM, Firebird / InterBase, Paradox File Access, MaxDB, Интерфейс PDO);

3) автоматизированная отправка HTTP-заголовков;

4) работа с HTTP-авторизацией;

5) работа с cookies и сессиями;

6) работа с локальными и удалёнными файлами, сокетами;

7) обработка файлов, загружаемых на сервер;

8) работа с Xforms.

В настоящее время PHP используется сотнями тысяч разработчиков. Согласно рейтингу корпорации TIOBE, базирующемся на данных поисковых систем, в мае 2016 года PHP находился на 6 месте среди языков программирования.

PHP

Сервер и браузер общаются, посылая друг другу запросы по особому протоколу - HTTP. Соединение может инициировать только браузер. Он посылает серверу запрос - показать такой-то файл. Сервер клиенту файл посылает.
Только так и происходит. Клиент запросил - сервер отдал. И забыл сразу о клиенте. Отсюда становится понятным ответ на вопрос, можно ли точно узнать, сколько юзеров сечас на сайте. Нельзя. потому, что "на сайте" нету ни одного. Они соединяются, запрашивают страницу, и отсоединяются. Не имеют постоянного соединения с сервером. Узнать можно только примерно, записывая время каждого соединения и выбирая записи за определенный промежуток времени.

PHP выполняется на сервере. Браузер посылает серверу запрос на страницу с php кодом. Сервер отдает эту страницу на исполнение интерпретатору PHP, интерпретатор генерирует HTML код, отдает серверу, а сервер посылает клиенту. Никакого PHP кода в браузер не попадает (это важно! Это значит, что увидеть исходный код PHP скрипта невозможно!). Единственный способ отправить что-то скрипту - это кликнуть по ссылке или нажать на кнопку в форме. Так, чтобы PHP обрабатывал какие-то действия пользователя в браузере - невозможно.
PHP остался на сервере, ждать новых запросов с данными для обработки. PHP, но не скрипт! Скрипт, который выполнялся, отдавая пользователю страницу, **завершил** работу. Все данные, которые были в нем - пропали. Именно поэтому, если какая-то переменная нужна при последующих вызовах скрипта, ее надо этому скрипту передать снова.

Способов, предоставляемых протоколом HTTP, немного. Это важная информация. Никаких других способов нет. На практике используются два:
GET - это когда данные передаются в адресной строке, например, когда пользователь жмет ссылку.
POST - когда он нажимает кнопку в форме.
Сформировали страницу со ссылкой или с формой методом GET - запрос придет GET-ом. Сформировали с формой, в которой указан метод POST - придет POST-ом.Определить, какой способ следует применять, очень просто. Если форма служит для запроса некой информации, например - при поиске, то ее следует отправлять методом GET. Чтобы можно было обновлять страницу, можно было поставить закладку и или послать ссылку другу.
Если же в результате отправки формы данные записываются или изменяются на сервере, то следует их отправлять методом POST, причем обязательно после обработки формы надо перенаправить браузер методом GET. Так же, POST может понадобиться, если на сервер надо передать большой объём данных (у GET он сильно ограничен), а так же, если не следует "светить" передаваемые данные в адресной строке (при вводе логина и пароля, например). Но в любом случае, после обработки POST надо всегда перенаправлять браузер на какую-нибудь страницу, пусть ту же самую, но уже без данных формы, чтобы при обновлении страницы они не записывались повторно.

Самое главное, что надо помнить: сервер по своей инициативе обратиться к клиенту не может. Мы можем только по факту запроса выдать что-то браузеру - либо страницу, либо команду запросить другой ресурс.

Плюсы и минусы WEB-разработки

Плюсы:

- востребованность на рынке труда;
- творческая работа, так как каждая задача уникальна, и для ее решения существует несколько способов;
- возможность совмещения с учебой; возможность стать специалистом в молодом возрасте;
- не всегда требуется диплом; свобода в принятии решений (как решать ту или иную задачу решает сам web-программист);
- возможность удаленной работы.

Плюсы и минусы WEB-разработки

Минусы:

- в небольших проектах web-программист совмещает функции нескольких специалистов, частые переключения с одной задачи на другую;
- ненормированный рабочий день.

THANKS FOR

COMING!

BY FRONTEND DEVELOPER **OSIPOV VICTOR**