Лекция 7- Язык РНР Веб-сервера Nginx, Apache

Тлеубаева А.О.

PHP

- PHP (расшифровывается как "PHP: Hypertext Preprocessor") это скриптовый язык программирования, который широко используется для разработки веб-приложений и динамических веб-сайтов. Вот некоторые основные характеристики PHP:
- Серверная сторона: РНР выполняется на сервере, и его задачей является обработка запросов от клиентов (браузеров) и создание динамических веб-страниц.
- Интеграция с HTML: PHP код может быть встроен непосредственно в HTML документы, что позволяет создавать динамические страницы.
- Разнообразие функций: РНР предоставляет множество встроенных функций для работы с базами данных, файлами, строками, изображениями и многими другими аспектами веб-разработки.
- Популярность: РНР очень популярен и имеет большое сообщество разработчиков, что делает его одним из самых используемых языков для веб-разработки.

РНР не работает сам по себе. Для работы с пользователем он должен быть настроен в связке с веб-сервером, например, Nginx или Apache. Если по-простому, то пользователь отправляет запрос к веб-серверу по протоколу НТТР. Веб-сервер определяет, к какому типу файла было совершено обращение, и, если это файл с расширением .php, то вебсервер передаёт запрос на обработку интерпретатором языка РНР, скрипт считывается, преобразуется в байт-код, выполняется и возвращает ответ веб-серверу. Веб-сервер передаёт сформированный результат пользователю. Этот результат, как правило, представляет собой динамически сгенерированную HTML-страничку.

Nginx

- Высокая производительность: Nginx известен своей выдающейся производительностью и способностью обрабатывать большие нагрузки. Он часто используется для высоконагруженных вебприложений.
- Асинхронная архитектура: Nginx работает в асинхронном режиме, что позволяет эффективно обрабатывать множество одновременных подключений.
- Прокси-сервер: Nginx также часто используется в качестве проксисервера, который направляет запросы клиентов к другим серверам, таким как серверы приложений или кэширующие серверы.
- Легковесный и малый объем памяти: Nginx потребляет меньше памяти, чем Apache, что делает его хорошим выбором для ограниченных по ресурсам серверов.

Nginx

- **Nginx** <u>веб-сервер</u> и почтовый <u>прокси-сервер</u>, работающий на <u>Unix</u>-подобных операционных системах (тестировалась сборка и работа на <u>FreeBSD</u>, <u>OpenBSD</u>, <u>Linux</u>, <u>Solaris</u>, <u>macOS</u>, <u>AIX</u> и <u>HP-UX</u>). Начиная с версии 0.7.52 появилась экспериментальная бинарная сборка под <u>Microsoft Windows</u>.
- Игорь Сысоев начал разработку в 2002 году. Осенью 2004 года вышел первый публично доступный релиз. С июля 2011 работа над nginx продолжается в рамках компании Nginx.

- Nginx позиционируется производителем как простой, быстрый и надёжный сервер, не перегруженный функциями.
- Применение nginx целесообразно прежде всего для <u>статических</u> <u>веб-сайтов</u> и как <u>обратного прокси-</u> <u>сервера</u> перед <u>динамическими сайтами</u>

• В **nginx** рабочие процессы обслуживают одновременно множество соединений, мультиплексируя их вызовами операционной системы select, epoll (Linux) и kqueue (FreeBSD). Рабочие процессы выполняют цикл обработки событий от дескрипторов. Полученные от клиента данные разбираются с помощью конечного автомата. Разобранный запрос последовательно обрабатывается цепочкой модулей, задаваемой конфигурацией. Ответ клиенту формируется в буферах, которые хранят данные либо в памяти, либо указывают на отрезок файла. Буфера объединяются в цепочки, определяющие последовательность, в которой данные будут переданы клиенту. Если операционная система поддерживает эффективные операции ввода-вывода, такие, как writev и sendfile, то **nginx** применяет их по возможности.

Apache

- Гибкость: Apache это старожил веб-серверов, и он предлагает множество модулей и настроек для настройки сервера под конкретные потребности.
- Широкое распространение: Apache долгое время был наиболее популярным веб-сервером и до сих пор широко используется.
- Модульная архитектура: Apache позволяет легко добавлять дополнительные модули, такие как модуль для обработки PHP (через модуль mod_php) или модуль для безопасности.
- .htaccess файлы: Apache поддерживает .htaccess файлы, которые позволяют настраивать доступ и правила перенаправления непосредственно из директорий сайта.

Apache

- **Apache HTTP-сервер** (является искажённым сокращением от <u>англ.</u> *а patchy server*; среди русских пользователей общепринято переводное *апа́ч*) <u>свободный веб-сервер</u>.
- Apache является <u>кроссплатформенным ПО</u>, поддерживает операционные системы <u>Linux</u>, <u>BSD</u>, <u>Mac OS</u>, <u>Microsoft Windows</u>, <u>Novell NetWare</u>, <u>BeOS</u>.
- Основными достоинствами Apache считаются надёжность и гибкость конфигурации. Он позволяет подключать внешние модули для предоставления данных, использовать <u>СУБД</u> для <u>аутентификации</u> пользователей, модифицировать сообщения об ошибках и т. д. Поддерживает <u>IPv4</u>.

- Система конфигурации Apache основана на <u>текстовых конфигурационных файлах</u>. Имеет три условных уровня конфигурации:
- Конфигурация сервера (httpd.conf). Директивы конфигурации сгруппированы в три основных раздела:
 - директивы, управляющие процессом Apache в целом (глобальное окружение);
 - директивы, определяющие параметры «главного» сервера, или сервера «по умолчанию», который отвечает на запросы, которые не обрабатываются виртуальными хостами (определяют также установки по умолчанию для всех остальных виртуальных хостов);
 - установки для виртуальных хостов, позволяющие обрабатывать запросы Web однимединственным сервером Apache, но направлять по раздельным адресам IP или именам хостов.
- Конфигурация виртуального хоста (httpd.conf с версии 2.2, extra/httpd-vhosts.conf).
- Конфигурация уровня каталога (<u>.htaccess</u>).
- Имеет собственный язык конфигурационных файлов, основанный на блоках директив. Практически все параметры ядра могут быть изменены через конфигурационные файлы, вплоть до управления МРМ. Большая часть модулей имеет собственные параметры.
- Часть модулей использует в своей работе конфигурационные файлы операционной системы (например <u>/etc/passwd</u> и <u>/etc/hosts</u>).
- Помимо этого, параметры могут быть заданы через ключи командной строки.

- Разработка проектов на РНР требует наличие сервера. Сервер может быть двух видов:
- удаленный сервер (хостинг);
- локальный сервер.

• Удаленный сервер — это сервер, что находится на неком удаленном ПК. Доступ к такому серверу имеете вы и все участники сети Интернет. При публикации сайта на удаленный сервер вы выгружаете его для всеобщего обозрение. Все смогут перейти к вам на сайт. Такая услуга выгрузки зачастую платная и приобретается у разных хостинг провайдеров.

• Локальный сервер — это сервер, что устанавливается на ваш компьютер. Такой сервер доступен только вам. Его зачастую используют в целях разработки проекта. Именно такой сервер будем использовать мы в ходе всего курса.

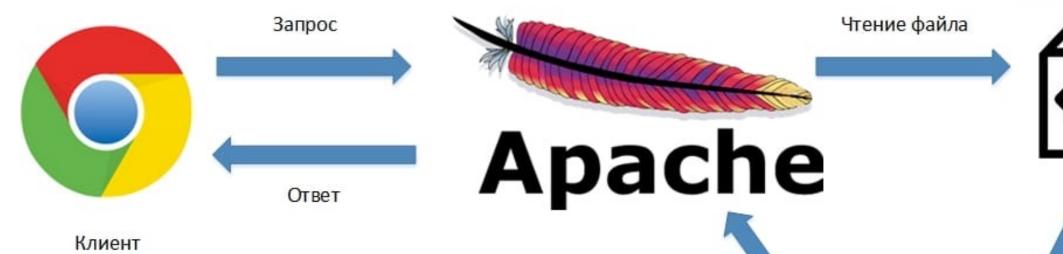
Для его установки нужна специальная программа. Вы можете использовать любую удобную вам. Ниже несколько из них:

- Программа <u>МАМР</u>. Удобная и быстрая, но ее работа может быть не точной на ОС Виндовс. Рекомендуем для установки на Мак ОС;
- Программа OpenServer. Схожая программа, что идеально работает на ОС Виндовс;
- Программа <u>XAMPP</u>. Работает на всех ОС, но имеет немного сложный интерфейс. Рекомендуем к установке на ОС Linux.

НТТР-сервер

- обслуживание неизменяемых запросов, индексных файлов, автоматическое создание списка файлов, кэш дескрипторов открытых файлов
- акселерированное проксирование без кэширования, простое распределение нагрузки и отказоустойчивость
- поддержка кеширования при акселерированном проксировании и FastCGI
- акселерированная поддержка <u>FastCGI</u> и <u>memcached</u>-серверов, простое распределение нагрузки и отказоустойчивость
- модульность, фильтры, в том числе сжатие (gzip), byte-ranges (докачка), chunked-ответы, HTTP-<u>аутентификация</u>, <u>SSI</u>-фильтр
- несколько подзапросов на одной странице, обрабатываемые в SSI-фильтре через прокси или FastCGI, выполняются параллельно
- поддержка <u>SSL</u>
- поддержка <u>PSGI</u>, <u>WSGI</u>
- экспериментальная поддержка встроенного <u>Perl</u>

Файл с РНР-Кодом



Передача на выполнение интерпретатору

Передача результата работы серверу



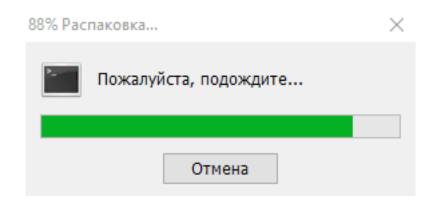
Интерпретация кода

- Этот процесс можно значительно ускорить, подключив расширение-акселератор. Оно занимается тем, что кэширует получившийся байт-код, и пока файл остаётся неизменным сразу начинается его выполнение. Начиная с версии РНР 5.6 акселератор OPcache включён в дистрибутив по умолчанию.
- Здесь стоит упомянуть, почему РНР так хорош для веб-разработки он по умолчанию содержит в себе огромное число инструментов для работы с запросами, средства для работы с огромным количеством баз данных, работу с файлами. Всё это активно используется современными сайтами.

- Первым делом идём на официальный сайт OpenServer и скачиваем программу. Я рекомендую скачать BASIC-комплектацию, для наших нужд её вполне хватит. Однако можете посмотреть список программ, которые включены в другие пакеты. Возможно, найдёте что-то полезное, будь то какойнибудь редактор кода или GUI-клиент для СУБД MySQL.
- Ссылка на оф. сайт: https://ospanel.io/

Установка

Запускаем установщик и задаём директорию для установки программы. Установка представляет собой простую распаковку.



После завершения установки окно молча закроется.

Настройка

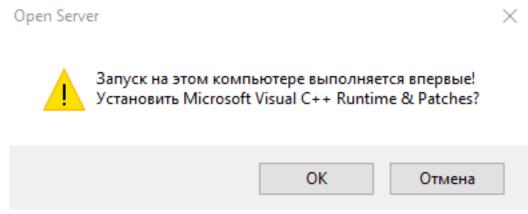
Переходим в папку с распакованным ПО: C:\OpenServer.

Здесь мы видим файлы для запуска программы OpenServer для разных версий системы и 3 папки:

- •domains папка, предназначенная для хранения Ваших сайтов. Внутри неё мы и будем создавать наши проекты, например: myproject.loc;
- •modules папка, содержащая в себе исполняемые файлы модулей и необходимые для их работы библиотеки, дополнительные файлы;
- •userdata папка, в которой находятся пользовательские данные СУБД, файлы кэша, логи, а также файлы конфигурации модулей.

Запустим один из ехе-файлов в корне, в зависимости от разрядности вашей ОС. В моём случае это **Open Server x64.exe**.

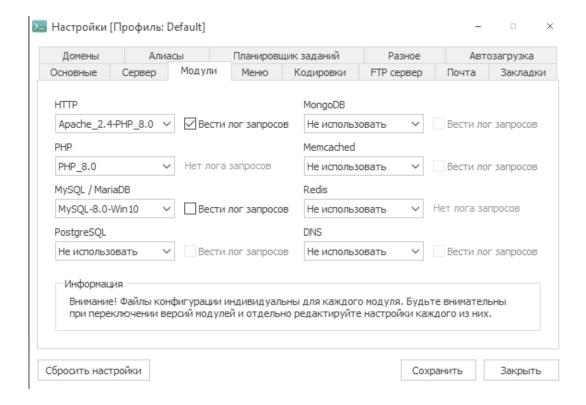
Выбираем язык и соглашаемся с установкой необходимых библиотек:



Перезагружаем компьютер.



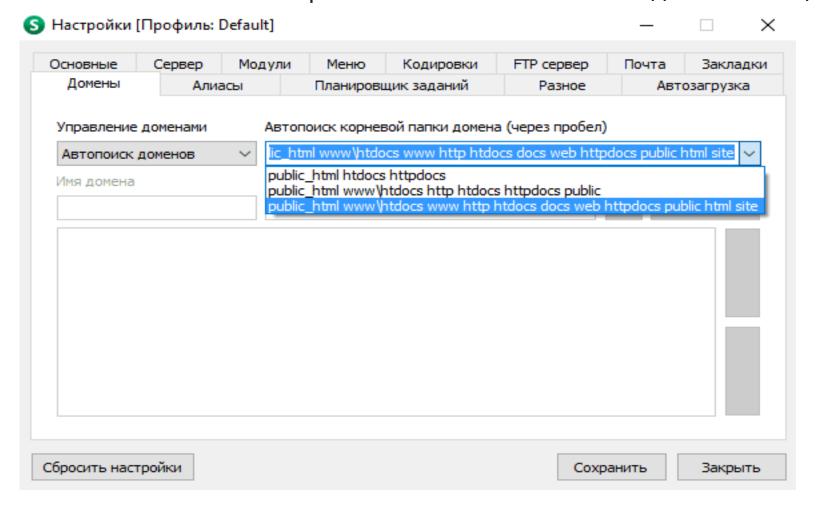
После перезагрузки снова запускаем наш ехе`шник. В трее появляется значок с красным флажком. Жмём по нему правой кнопкой мыши и выбираем пункт "Настройки". Откроется главное окно программы. Перейдём во вкладку "Модули" и выберем версию РНР: **PHP 8** и версию Арасhe: **Apache-PHP-8**.



Затем перейдём во вкладку "Домены", выберем "автопоиск доменов" и в качестве корневой папки домена

выберем пункт с наибольшим числом вариантов. Вы также можете дописать сюда свои

варианты.



После чего нажимаем "Сохранить" и закрываем окно настроек.

Hello World

- Перейдём в папку "domains" и создадим директорию для нашего проекта. Назовём его "myproject.loc". Внутри создадим директорию "www". А внутри неё файл "index.php". Откроем его и впишем следующий код:
- <?php echo 'Hello world!';</p>
- Теперь можно запустить сервер и проверить работу скрипта.
- Если при старте сервера появилась ошибка: Файл C:\Windows\system32\drivers\etc\hosts недоступен для записи - стоит закрыть программу и запустить её от имени администратора.
- Флажок в трее должен стать зелёным. Откроем браузер и перейдём по адресу http://myproject.loc/
- Если же вместо странички вы увидели 404-ошибку сервера, то скорее всего ваш файл назван не index.php a index.php.txt, при этом .txt не отображается в имени. Тогда нужно включить показ расширений для всех файлов и затем убрать у файла приписку .txt в конце. После этого обновить страничку в браузере и порадоваться результату.

PHPStorm

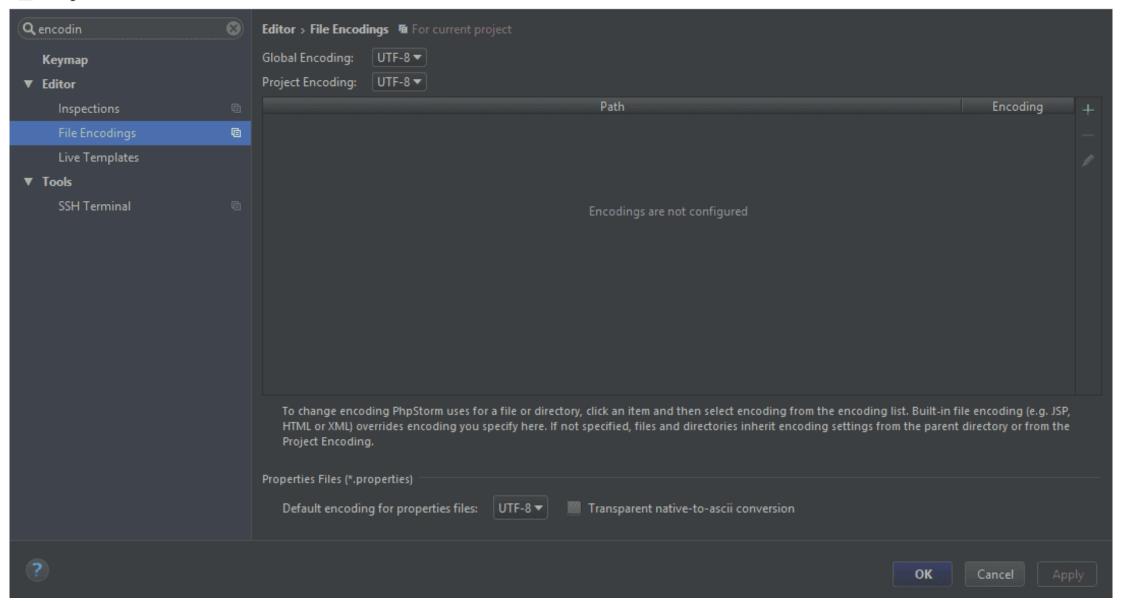
Установка

Итак, скачиваем программу по ссылке и запускаем установщик. Установка проходит стандартно и как правило не требует никаких дополнительных телодвижений.

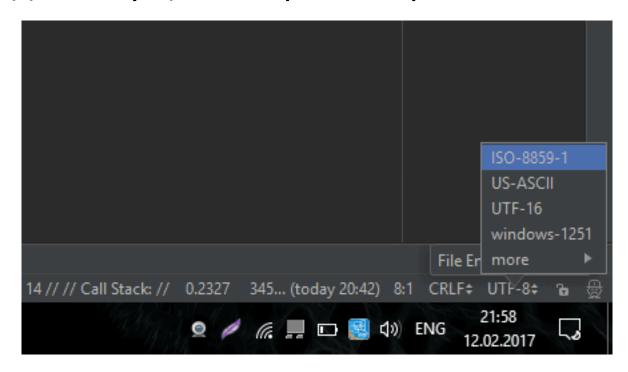
После установки запускаем программу и открываем созданный нами в прошлом уроке проект. Жмём File->Open и выбираем папку с проектом (C:\OpenServer\domains\myproject.loc). Первым делом нужно задать правильную кодировку для всех файлов в проекте. Для этого идём в File->Settings.

Слева в меню разворачиваем пункт **Editor**, и выбираем вложенный пункт **File Encodings**. Задаём следующие настройки:

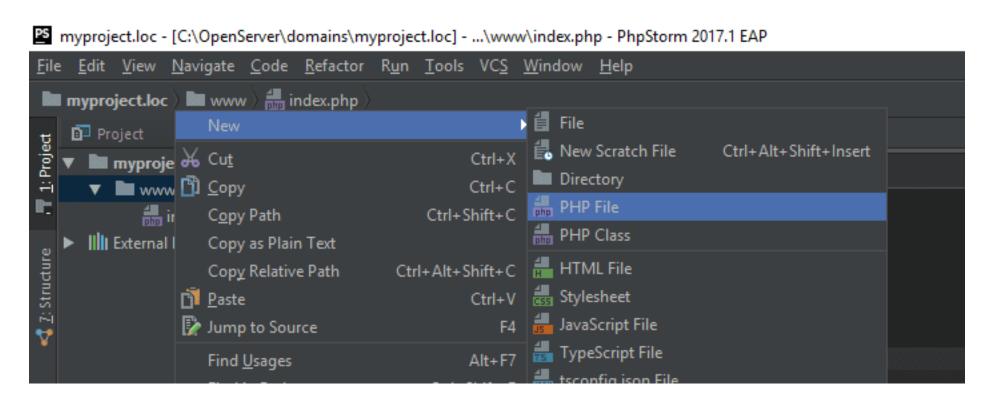
- Global Encoding: UTF-8
- Project Encoding: UTF-8
- **Default** encoding **for** properties files: UTF-8



- И сохраняем изменения.
- Теперь давайте удалим ранее созданные нами файлы, которые сейчас находятся в кодировке **Windows-1251**. Да, кодировку всегда можно сменить в правом нижнем углу программы, смена производится для текущего открытого файла.



Теперь давайте создадим новый файл в нашем проекте с названием **index.php**. Для этого слева в структуре проекта щёлкаем правой кнопкой мыши по папке **www**, выбираем **New->PHP File** Вводим имя и нажимаем **Create**.



Выражение в языке РНР

- Выражение в языке PHP это вообще любая строка, число, объект, массив, или что-то ещё, что представлено в явном виде или в виде каких-либо вычислений. Например:
- 'строка с пробелами'
- 7
- 2 + 2
- 5 / 2
- Всё вышеперечисленное является выражениями. При этом выражение должно иметь какое-либо значение. Для примеров выше это, например, строка с текстом, целое число 7, целое число 4 и, наконец, дробное число 2.5.
- Как вы понимаете, сами по себе выражения бесполезны, пока они не используются. Вся прелесть **PHP** в том что он может встраиваться в **HTML**.

- Так вот, **PHP** может быть встроен прямо в **HTML**. Давайте отредактируем наш с вами файл **index.php**. Впишем в него следующий код:
- <!DOCTYPE html> <html> <head> <meta charset="UTF-8"> <title>Пример встраивания PHP в HTML</title> </head> <body> <h1>Пример встраивания PHP в HTML</h1> 2 + 2 = <?php echo 2 + 2; ?> </body> </html>