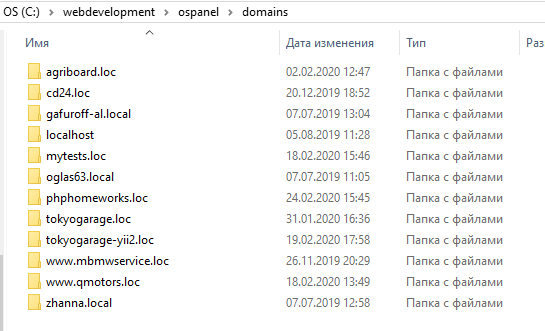
1. Установить программное обеспечение: веб-сервер, базу данных, интерпретатор, текстовый редактор. Проверить, что все работает правильно.

Софт, который я использую в работе и учебе:



2. Выполнить примеры из методички и разобраться, как это работает.

Посмотрел примеры из методички, отдельно каждый проработать в виде коде не стал, в связи с тем, что на протяжении последних трёх лет только этим и занимаюсь. Чтобы не быть голословным предоставляю список директорий с проектами, над которыми я сейчас работаю, либо постоянно, либо время от времени. Если понадобится кусок кода из какого-либо проектов, готов предоставить.



**agriboard.loc** – сельскохозяйственная доска объявлений, проект выполняемый на заказ, строится на основе фреймворка yii2, моя задача это базы данных, бэкэнд и часть фронтенда;

**cd24.loc** – учебный проект, разрабатывался изначально в рамках курса по командной разработке, теперь время от времени работаю над ним когда время есть. Моя задача, в рамках этого проекта, это разработка парсеров сайтов доноров. Парсеры написаны в виде консольных модулей на PHP, запускаются из консоли через утилиту nohup, парсеры работают циклично на сервере, пока не поступит сигнал отключения…

**gafuroff-al.local** – мой персональный сайт (статический).

**mytests.loc** – здесь отрабатываю различные языковые приемы, мелкие утилиты и прочие конструкции языка.

**oglas63.local** – локальная версия моей собственной доски объявлений, когда-то была написана на основе CMS e107, сейчас переписываю на yii2.

**phphomeworks.loc** – создал для выполнения домашних заданий в рамках данного курса.

Далее 4 папки это локальные копии проектов с которыми я работаю в рамках моей основной работы: **tokyogarage.loc** – локальная копия на 1C Битрикс; **tokyogarage-yii2.loc** – копия будущего предыдущего сайта, пересоздается на основе yii2; **www.mbmservice.loc** и **www.qmotors.loc** – написаны на неизвестной MVC CMS под названием Reactor.

**zhanna.local** – персональный сайт моей жены, создан на основе DataLifeEngine.

3. Объяснить, как работает данный код:

<?php

$a = 5;

$b = '05';

var\_dump($a == $b); // Почему true?

var\_dump((int)'012345'); // Почему 12345?

var\_dump((float)123.0 === (int)123.0); // Почему false?

var\_dump((int)0 === (int)'hello, world'); // Почему true?

?>

var\_dump($a == $b); // Почему true?

== нестрогое сравнение, строка при приводится к числу, в результате сравнение истинно;

var\_dump((int)'012345');

преобразование строки в число, при этом впереди стоящий ноль, не несет никакой нагрузки в составе числа и отбрасывается;

var\_dump((float)123.0 === (int)123.0);

=== строгое сравнение, которое учитывает не только значение, но и тип данных. В результате при разных типах данных возвращается ложь;

var\_dump((int)0 === (int)'hello, world');

при попытке преобразовать буквенную строку в число возвращается ложь, возвращаемое значение приводится к нулю, в результате получаем два ноля в типом данных число, поэтому сравнение истинно;

4. Используя имеющийся HTML-шаблон, сделать так, чтобы главная страница генерировалась через PHP. Создать блок переменных в начале страницы. Сделать так, чтобы h1, title и текущий год генерировались в блоке контента из созданных переменных. Построить шаблон 3-я способами, ВАЖНОЕ!

Все решения находятся в корневой директории **/answer\_4/;**

5. \*Используя только две переменные, поменяйте их значение местами. Например, если a = 1, b = 2, надо, чтобы получилось b = 1, a = 2. Дополнительные переменные использовать нельзя.

Решение находится в PHP файле **answer\_5.php**