

1 месяц

- PHP
- LINUX
- COMPOSER
- ФОРМЫ
- БАЗЫ ДАННЫХ
- HTTP
- GIT
- MVC
- ACTIVERECORD
- DATAMAPPER

[PHP]Архитектура протокола HTTP и Web
[PHP]Принципы работы веб-приложений
[PHP]Linux-системы и рабочее окружение
[PHP]Принципы работы PHP7

[Linux]Основы
[Linux]Работа с файлами и каталогами
[Linux]Работа с Git
[Linux] Установка пакетов
[Linux]Конфигурация LAMP
[Linux]Устранение неисправностей

[Git]Основы работы и уст-во систем контроля версий
[Git]Создание веток и переключения между ветками
[Git]Commit
[Git]Rebase
[Git]Push/Pull
[Git]SSH/HTTP-доступ

[Form]Дизайн форм и извлечение отправленных данных на сервере
[Form]Валидация клиентских данных
[Form]Защита данных перед обработкой и фильтрация
[Form]Отображение формы с ошибками

[Symfony]Создание первого проекта
[Symfony]Symfony Flex
[Symfony]Управление зависимостями через Composer
[Symfony]Бандлы (Bundles) и Паки (pack)
[Symfony]Работа с БД на примере ORM Doctrine
[Symfony]Работа с шаблонами на примере Twig
[Symfony]Symfony Best Practices
[Symfony]Symfony Components
[Symfony]Маршрутизация в Symfony приложениях
[Symfony]Формы в Symfony приложениях
[Symfony]Кеширование в Symfony приложениях
[Symfony]Безопасность в Symfony приложениях
[Symfony]Консольные команды в Symfony приложениях

[Twig]Наследование, блоки, вывод данных
[Twig]Функции и фильтры
[Twig]Написание кастомных функций и фильтров
[Twig]Дебаг

[REST API]Конфиг
[REST API]Сериализаця
[REST API]CRUD
[REST API]JWT

[OOP]Введение в OOP
[OOP]Паттерн MVC (Model-View-Controller)
[OOP]Паттерны Data Mapper и Active Record
[OOP]Внедрение зависимостей (Dependency Injection)
[OOP]Имплементация паттернов на практике

[Database]Принцип работы реляционных баз данных
[Database]Принципы ACID
[Database]Определение таблиц и полей
[Database]Создание связей между записями с помощью первичных и внешних ключей
[Database]Написание запросов к БД на языке SQL
[Database]Работа с базой данных с помощью Workbench/PhpStorm
[Database]Отладка запросов с помощью EXPLAIN

[Composer] Принцип работы пакетного менеджера
[Composer]Установка Composer
[Composer]Подключение зависимостей
[Composer]Обновления зависимостей
[Composer]Установка зависимостей
[Composer]Packagist для поиска пакетов

[Filesystem]Создание, запись и удаление файлов/папок
[Filesystem]Поиск по файлам и директориям
[Filesystem]Сортировка

[Doctrine]OOP подход при работе с БД
[Doctrine]Обновление схемы БД
[Doctrine]Работа с фикстурами
[Doctrine]Сохранение, редактирование, обновление и удаление объектов из БД
[Doctrine]Настройка связей между сущностями (OneToMany, ManyToOne, ManyToMany)
[Doctrine]Производительность и кеширование
[Doctrine]Behaviors (Tree, Translatable, Loggable, Sluggable, Timestampable)

[Doctrine]OOP подход при работе с БД
[Doctrine]Обновление схемы БД
[Doctrine]Работа с фикстурами
[Doctrine]Сохранение, редактирование, обновление и удаление объектов из БД
[Doctrine]Настройка связей между сущностями (OneToMany, ManyToOne, ManyToMany)
[Doctrine]Производительность и кеширование

[MakerBundle] Конфиг
[MakerBundle]Генерация CRUD
[MakerBundle]Кастомизация

[Routing]Создание маршрутов
[Routing]Прием и передача перменных
[Routing]Фильтрация данных
[Routing]Внутренняя генерация роутов

2 месяц

- ШАБЛОНИЗАЦИЯ
- РОУТИНГ
- ORM
- ФРЕЙМВОРКИ
- КОНСОЛЬ
- ЛОГИРОВАНИЕ
- БЕЗОПАСНОСТЬ
- CI

[Monolog]Логирование
[Monolog]Обработчики
[Monolog]Форматеры
[Monolog]Процессоры

[DomCrawler]Отправка HTTP запросов
[DomCrawler]Прием HTTP ответов
[DomCrawler]Парсинг HTML/XML данных

[Blackfire]Конфигурация
[Blackfire]Профилирование цепочек вызова php-кода и ресурсов (БД, сеть, память, процессор)
[Blackfire]Разбор отчетов и графов

[Docker]Установка и принцип работы
[Docker]Uml config
[Docker]Запуск и управление контейнерами
[Docker]Docker в контексте GitLab CI

[Testing]PHPUnit
[Testing]Selenium
[Testing]Travis/CircleCI
[Testing]GitlabCI/BitbucketCI

[Console]Создание консольных команд
[Console]Стилизация вывода
[Console]Дебаг
[Console]Консольные приложения

[Security]Разграничение прав доступа (ACL)
[Security]Файрволл и авторизация
[Security] LDAP
[Security]CSRF защита
[Security]Конфигурация

[BestPractices]Фильтрация и валидация пользовательских данных
[BestPractices]Безопасные SQL запросы для предотвращения SQL-инъекций
[BestPractices]Установка Composer
[BestPractices]Рефакторинг кода для ухода от дублирования (DRY, KISS)
[BestPractices]Стандарты кодирования и соглашения (Code Style, PSR)
[BestPractices]Contribution в Open Source библиотеки
[BestPractices]SOLID/XP/TRIZ

[Continuous Integration]Gilab CI
[Continuous Integration]Unit и Functional-тесты
[Continuous Integration]Code style
[Continuous Integration]Статическая проверка кода
[Continuous Integration]Auto-deployment
[Continuous Integration]Привязка коммитов к задачам

[Message Queueing]RabbitMQ
[Message Queueing]Gearman

3-6 месяц

- УПАКОВКА ВАШЕГО РЕЗЮМЕ
- СОБЕСЕДОВАНИЯ
- ПРАКТИКА НА БЕОМОВ ПРОЕКТЕ

[HighLoad]Что надо знать о BIG DATA и HighLoad
[HighLoad]Как оптимизируются запросы в БД
[HighLoad]Почему архитектура критическа важна
[HighLoad]Шардинг, репликация, партиционирование
[HighLoad]Горизонтальное масштабирование
[HighLoad]Отказоустойчивость (availability)
[HighLoad]Nginx + php-fpm
[HighLoad]Нагрузочное тестирование

[WebSocket]Ratchet/React
[WebSocket]WebSocketBundle
[WebSocket]Implementation WebSocket server and client side
[WebSocket]WAMP proto
[WebSocket]Swoole

[Практика]Выбор и проектирование БД
[Практика]Реализация бизнес-логики
[Практика]Настройка репозитория и CI
[Практика]Командная работа
[Практика]Покрытие кода тестами
[Практика]Сдача проекта

[Symfony]Кастомные системы авторизации. Symfony Guard
[Symfony]Управление локалями с Symfony Locale
[Symfony]Кеширующие бекенды с Symfony Cache
[Symfony]Переводы с Symfony Translation
[Symfony]Symfony Service Container

[Docker]Dockerizing workflow
[Docker]Orchestration
[Docker]Pull
[Docker]Create image

[Contributing]Github Social Coding
[Contributing]Fork, Pull Request
[Contributing]Ваша open source активность
[Contributing]Как находить репозитории для contributing
[Contributing]Как правильно контрибутировать и оформлять pull request-ы

[Marketing]Упаковка и оформление опыта в резюме (CV)
[Marketing]Поиск и подбор проектов
[Marketing]Разбор и прохождение интервью
[Marketing]Прохождение исп срока