

1. Краткая история развития World Wide Web. Основные составляющие WWW
2. Язык разметки гипертекста HTML. История развития. Основные понятия. Структура документа
3. Язык разметки гипертекста HTML. Теги заголовочной части документа. Теги форматирования текста. Графика в HTML-документах. Форматирование таблиц.
4. Каскадные таблицы стилей CSS. Селекторы. Специфичность селекторов. Применение селекторов для выбора DOM-элементов.
5. Каскадные таблицы стилей CSS. Позиционирование элементов.
6. Каскадные таблицы стилей CSS. Flexible Box Layout Module. Свойства flex-элемента и flex-контейнера.
7. Каскадные таблицы стилей CSS. Медиа-запросы. Медиа-функции.
8. Каскадные таблицы стилей CSS. Единицы измерения.
9. Каскадные таблицы стилей CSS. Блочная модель.
10. Язык разметки гипертекста HTML. Элемент DIV и его свойства. Гипертекстовые ссылки. URL.
11. Язык разметки гипертекста HTML. Формы в HTML. Теги форм.
12. Понятие DOM (Document Object Model) и его связь с HTML-документом. Методы доступа к элементам DOM и различия между ними.
13. Понятие события DOM. Способы их обработки с помощью JavaScript.
14. Функции JavaScript для работы с DOM-элементами, их применение.
15. Понятие делегирования событий (event delegation). Применение делегирования событий при обработке событий.
16. Понятие объектно-ориентированное программирования (ООП), его основные принципы и подходы в Javascript. Понятие прототипа объекта (prototype), его применение в JavaScript.
17. Способы создания нового объекта с помощью конструктора в Javascript. Основные особенности конструктора в JavaScript. Способы создания методов и свойств у объектов в JavaScript.
18. Понятие наследования в ООП. Способ реализации наследования с помощью прототипов в JavaScript.
19. Понятие класса в JavaScript. Основные различия класса и прототипа.
20. Принципы работы ключевого слова this в JavaScript, его связь с контекстом выполнения?
21. Основные возможности ES6 (ECMAScript 2015) в сравнении с классическим Javascript. Примеры использования этих возможностей. Типы данных в ES6, их применение.
22. Понятие стрелочных функций (arrow functions) в JavaScript, их отличия от обычных функций.
23. Основные принципы работы модульной система (module system) в ES6. Подходы и практики ее применения в организации кода.
24. Понятие генератора (generator) в JavaScript, способы их применения для создания итераторов.
25. Основные принципы работы деструктуризации (destructuring) в JavaScript, области ее применения.
26. Принципы и подходы в работе с промисами (promises) в управлении асинхронным кодом в JavaScript.
27. Понятие фреймворка. Основные особенности и преимущества Vue.js для разработки веб-приложений. Способы создания компонентов в Vue.js. Основные методы жизненного цикла компонента.
28. Способы передачи данных из родительского компонента в дочерний компонент в Vue.js.
29. Понятие реактивности в Vue.js, ее связь с обновлением DOM.
30. Методы обработки событий в компонентах Vue.js.

31. Способы обработки данных пользовательского ввода в компонентах Vue.js.
32. Принципы работы системы компонентов в Vue.js и организация их взаимодействия.
33. Понятие глобальных миксинов (global mixins) в Vue.js, их применение.
34. Принципы работы модификатора v-model в Vue.js, его применение.
35. Понятие асинхронных компонентов (async components) в Vue.js. Применение асинхронных компонентов в оптимизации загрузки приложения.
36. Понятие Docker, основные преимущества его использования. Порядок установки и настройки Docker на локальной машине.
37. Порядок действия для написания простого docker-compose. Основные команды для сборки и запуска контейнеров, описанных в docker-compose.yml файле.
38. Подходы к настройке сетей для контейнеров в docker-compose.yml файле. Способы настройки связи между контейнерами в docker-compose.yml файле.
39. Способы просмотра состояния и журнала контейнеров, запущенных с использованием docker-compose. Способы остановки и удаления контейнеров, запущенных с использованием docker-compose.
40. Понятие, особенности и возможности языка PHP для разработки веб-приложений.
41. Основные методы взаимодействия между клиентской и серверной частями при использовании PHP.
42. Способы загрузки файлов на сервер с помощью PHP. Основные функции, используемые для обработки загруженного файла. Основные методы безопасности, применяемые при обработке загружаемых файлов с помощью PHP.
43. Подходы, функции, правила обработки ошибок к установке соединения с базой данных из PHP.
44. Понятие ООП (объектно-ориентированное программирование) в языке PHP. Основные преимущества ООП в разработке программного обеспечения на PHP.
45. Основные элементы класса в PHP. Способы создания объекта в PHP. Операции с объектами. Основные подходы и принципы при проектировании классов и взаимодействии между ними в PHP.
46. Понятие наследования в ООП и его применение в PHP для создания иерархии классов.
47. Понятие абстрактных классов и интерфейсы в PHP, и их применение для определения общего поведения и структуры классов.
48. Понятие полиморфизма в ООП. Способы реализации полиморфизма в PHP с помощью переопределения методов.
49. Подходы к использованию пространства имен (namespaces) в PHP для организации кода и предотвращения конфликтов имен.
50. Понятие асинхронного программирования. Основные преимущества для взаимодействия с сервером в PHP. Методы и средства PHP для асинхронного взаимодействия с сервером.
51. Понятие фреймворка в PHP. Преимущества фреймворка Laravel для разработки веб-приложений.
52. Компоненты для установки и настройки Laravel на локальном сервере.
53. Способы, функции и классы Laravel для загрузки и обработки файлов на сервере.
54. Создание и применение моделей в Laravel для работы с базой данных. Методы и отношения между моделями.
55. Операции хранения, поиска и удаления данных с использованием моделей и запросов в Laravel.
56. Методы и функции Laravel для удобной и эффективной работой с базой данных, включая миграции, сидеры и фабрики.
57. Настройка поиска и фильтрации данных в Laravel с использованием запросов Eloquent.
58. Функции Laravel для обработки и валидации пользовательского ввода перед сохранением в базу данных.

59. Методы и функции Laravel для управления файлами и папками на сервере, включая загрузку, хранение и удаление файлов.
60. Понятие авторизации и аутентификации веб-приложений. Способы реализации в Laravel.
61. Настройка и применение механизма аутентификации пользователей в Laravel.
62. Создание и управление ролями и правами пользователей в Laravel.
63. Методы и функции Laravel для создания и обработки токенов аутентификации (API-токенов).
64. Создание и настройка API-маршрутов в Laravel для обеспечения взаимодействия с внешними системами через API.
65. Методы и функции Laravel для валидации и фильтрации входных данных при использовании API.
66. Инструменты и практики для обеспечения безопасности при работе с авторизацией, разграничением прав и созданием API в Laravel.
67. Применение командной строки (CLI) в Laravel для выполнения задач и управления приложением. Применение Artisan для выполнения команд и задач из командной строки.
68. Создание и регистрация собственных команды в Laravel с помощью Artisan.
69. Методы и функции Laravel для работы с расписанием (Scheduling) и выполнения задач по расписанию. Способы настройки расписания задач в Laravel, указывая время и интервалы выполнения задач.
70. Применение механизма очередей (Queue) в Laravel для выполнения задач асинхронно или в фоновом режиме. Способы добавления и настройки драйверов очереди в Laravel.