федеральное государственное автономное образовательное

учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Лист №1, всего листов \_\_\_\_.

**ОТВЕТЫ НА ЭКЗАМЕНАЦИОННЫЕ ВОПРОСЫ**

по программе магистратуры

*Веб-технологии*

*Факультет ПИиКТ*

Дата проведения экзамена 04 августа 2021 г. Билет №\_\_\_\_\_\_\_

Ф.И.О. поступающего:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата рождения\_\_\_\_\_\_\_\_\_\_\_\_\_

Данное поле заполняется экзаменатором

|  |  |  |  |
| --- | --- | --- | --- |
|  | Вопрос № 1 | Вопрос № 2 | Экзамен |
| Оценка |  |  |  |

Экзаменатор

**Вопрос билета № 1.**

**Вопрос билета № 2.**

**Статические веб-сайты. Обзор JAMstack.**

**Статические веб-сайты**

**Сайт**, или **веб-сайт** (от [анг.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA" \o "Английский язык) *website*: *web* — «паутина, сеть» и *site* — «место», букв. — «место, сегмент, часть в сети»), также **веб-узел**, — одна или несколько логически связанных между собой  веб-страниц; также место расположения контента сервера. Обычно сайт в Интернет представляет собой массив связанных данных, имеющий уникальный адрес и воспринимаемый пользователями как единое целое. Веб-сайты называются так, потому что доступ к ним происходит по протоколу HTTP.

Веб-сайт как система [электронных документов](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D1%8B%D0%B9_%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82) ([файлов](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB) данных и [кода](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4)) может принадлежать частному лицу или организации и быть доступным в [компьютерной сети](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C) под общим [доменным именем](https://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%B8%D0%BC%D1%8F) и [IP-адресом](https://ru.wikipedia.org/wiki/IP-%D0%B0%D0%B4%D1%80%D0%B5%D1%81) или локально на одном компьютере. В статье [журнала](https://ru.wikipedia.org/wiki/%D0%96%D1%83%D1%80%D0%BD%D0%B0%D0%BB) «Хозяйство и право» также было высказано мнение, что каждый сайт имеет своё название, которое при этом не следует путать с доменным именем[[2]](https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82#cite_note-2). С точки зрения [авторского права](https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D1%80%D1%81%D0%BA%D0%BE%D0%B5_%D0%BF%D1%80%D0%B0%D0%B2%D0%BE) сайт является составным произведением, соответственно название сайта подлежит охране наряду с названиями всех прочих произведений.

Все сайты в совокупности составляют [Всемирную паутину](https://ru.wikipedia.org/wiki/%D0%92%D1%81%D0%B5%D0%BC%D0%B8%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D1%83%D1%82%D0%B8%D0%BD%D0%B0), где [коммуникация](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BC%D1%83%D0%BD%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C) (паутина) объединяет сегменты информации мирового сообщества в единое целое — [базу данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) и коммуникации планетарного масштаба. Для прямого доступа клиентов к сайтам на [серверах](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) был специально разработан [протокол](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB_%D0%BF%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D1%87%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) [HTTP](https://ru.wikipedia.org/wiki/HTTP).

**Статический сайт** — это сайт, который состоит из неизменяемых веб-страниц. Исходный код таких страниц состоит только из HTML и возможно ещё CSS и JS. Чтобы обновить или дополнить информацию на статичной веб-странице, нужно самостоятельно вручную внести изменения в её исходный код. Когда веб-сервер получает запрос от клиента на какой-нибудь ресурс (изображение, [HTML документ](https://puzzleweb.ru/html/00_teacher2.php) и др.), он находит этот ресурс и просто отправляет его клиенту.

**Плюсы статического сайта:**

* Возможность прямого просмотра HTML-документа в браузере, без использования промежуточного ПО.
* Для создания страницы не обязательно знать языки [веб-программирования](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5).
* Хорошая кэшируемость страницы.
* Быстрая скорость загрузки страниц.
* Минимальные требования к веб-серверу и минимальная нагрузка на него.
* Простота переноса на другой сервер или локальный компьютер.
* Возможность прямого просмотра файла в браузере, без использования промежуточного ПО ([веб сервера](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80), [CGI](https://ru.wikipedia.org/wiki/CGI)).

**Минусы статического сайта:**

* При увеличении количества веб-страниц, поддерживать статический сайт становится труднее, так как появляются трудности с внесением изменений в код веб-страниц. Например, если надо внести какое-нибудь однотипное изменение, придётся редактировать код каждого HTML-документа отдельно, что в свою очередь является источником дополнительных время затрат, вероятности обновить не всё и допустить где-нибудь ошибки.
* Невозможность динамической генерации содержимого.
* Невозможность полноценной поддержки посетителей (выбор внешнего вида, cookie).
* Для наполнения сайта информацией необходимо получать доступ к файлам-страницам посредством [FTP](https://ru.wikipedia.org/wiki/FTP), или сторонним [веб-скриптам](https://ru.wikipedia.org/w/index.php?title=%D0%92%D0%B5%D0%B1-%D1%81%D0%BA%D1%80%D0%B8%D0%BF%D1%82&action=edit&redlink=1), позволяющим редактировать страницы.
* При большом количестве страниц (файлов), если возникает необходимость внести однотипные изменения (дизайн, оформление, добавление новых разделов) необходимо использовать стороннее ПО (утилиты).

**Обзор JAMstack**

Что понимают разработчики в рамках подхода JAMStack? Это, в основном, стек веб-технологий, который состоит из:

* **«J» (JavaScript)** - язык кодирования для рендеринга на стороне пользователя;
* **«A» (API)** - интерфейс прикладного программирования, позволяющий интегрировать дополнительные функции на стороне сервера;
* **«М» (разметка)** - основа сайта, его основной каркас.

Что мы получаем в результате такого подхода? Мы получаем сайты, которые не «привязаны» к одному веб-серверу, но основаны на полнофункциональном CDN. Весь кеш сохраняется на стороне клиента, что значительно повышает скорость загрузки страницы. Все версии веб-проекта также хранятся в Git. Вот почему даже если какой-либо сервер выходит из строя, любой клиентский репозиторий можно скопировать на другой сервер, чтобы возобновить работу сайта.

### **Когда JAMstack должен заменить традиционный рабочий процесс?**

* Ваш динамичный веб-сайт сложно и дорого масштабировать;
* Ваш динамический веб-сайт плохо синхронизировать со сторонними решениями (ERP, CRM-системы);
* Вы чувствуете, что ваша CMS ограничивает вашу гибкость при работе на UX;
* Вы понимаете, что CMS - не самый безопасный вариант;
* Вы не хотите, чтобы скорость загрузки страницы напрямую зависела от веб-сервера;
* У вас возникли проблемы с мобильной версией вашего динамического веб-сайта;
* Вы хотите работать с современными технологиями (такими как React, Vue и т. д.);
* Вы не хотите беспокоиться о SEO из-за недостатков на стороне сервера;
* Вы хотите подключить свой бэкэнд к облаку.

### **Принципиальные различия между традиционными динамическими сайтами и решениями JAM**

Теперь давайте углубимся в детали и выясним, чем традиционный подход к веб-разработке отличается от подхода JAMStack на практике. Таким образом, традиционный рабочий процесс подразумевает, что вся работа разработчика, в некоторой степени, сосредоточена вокруг хостинг-соединения.

Все усложняется из-за чрезмерных подключений в результате запросов к веб-серверу, базе данных, CMS и т. д. Кроме того, содержимое может обновляться через устаревший протокол передачи данных FTP, что снижает общую надежность и скорость работы веб-сайта.

С точки зрения JAMStack все намного проще. Во-первых, вам не нужно зависеть от хостинга, а это означает, что разработчики получают больше гибкости при создании любого другого компонента JAM.

Во-вторых, ресурсы, основанные на этом подходе, лучше подходят для глобально распределенных целевых аудиторий с быстрым обновлением данных, достигаемым благодаря специфике CDN. Наконец, для создания сайтов используются передовые статические генераторы, которые на шаг опережают необходимость использования централизованных систем контроля версий.

### **Как JAMStack решает проблемы других новых технологий веб-сайтов?**

Давайте рассмотрим некоторые практические аспекты реализации стека программирования JAM, которые четко указывают на отсутствие многих проблем, с которыми сталкиваются большинство создателей динамических веб-сайтов.

Например, при развертывании статического ресурса вы можете интегрироваться практически с любым доступным сервисом, где могут храниться данные. В настоящее время существует множество таких решений (например, [JAMStack Amazon CDN](https://aws.amazon.com/cloudfront/" \t "_blank)), которые устраняют проблему поиска высокопроизводительного сервера.

Это напрямую влияет на скорость и общую производительность сайта. В довершение всего, из-за того, что каждая страница сайта постоянно загружается прямо на стороне клиента, весь контент уменьшается в размерах, что также ускоряет процесс.

Дополнительным преимуществом статических ресурсов является их надежность. Даже если какой-нибудь любопытный хакер попытается взломать ваш сайт, он не сможет сделать больше, чем просто скопировать HTML-файлы. Наконец, все версии хранятся в Git, а контент - на серверах CDN, что устраняет головную боль постоянной модерации на стороне сервера.

#### ****JAMStack преимущества****

* SEO по умолчанию;
* Сверхбыстрый доступ из-за отсутствия динамического контента в браузере;
* Нет необходимости в сервере приложений - все может быть основано на одном [Nginx](https://www.nginx.com/" \t "_blank);
* Даже если веб-сервер отключается, все приложения будут отображаться в автономном режиме.
* Даже если веб-сервер отключается, все приложения будут отображаться в автономном режиме благодаря функции Service Workers, которая позволяет при необходимости выполнять различные автономные процессы.

#### ****JAMStack недостатки****

* Процесс преобразования кода в статический контент может занять до 20 минут (может быть решен в ближайшее время и зависит от количества страниц);
* JAM-Решения - это, в основном, пользовательские проекты на стороне клиента, то есть вы не сможете использовать готовые шаблоны;
* Этот подход вообще не подходит для суперинтерактивных решений (например, сложные онлайн-редакторы фотографий, онлайн-разработчики сайтов / демо-сборки и т. д.);
* База данных может получать значительную нагрузку при каждом развертывании (весь контент создается через базу данных).

### **Инструменты для реализации JAM-проектов с нуля: Netlify, Forestry и другие**

Наиболее широко используемыми инструментами для отработки подхода JAMStack, которые упоминаются в каждом другом руководстве JAMStack, являются React, Vue и Angular. Все они основаны на JavaScript, однако вам придется пройти через определенные особенности, чтобы начать работать с ними (особенно это касается Angular). Конечно, вы можете выбрать более простой маршрут и использовать генераторы статических сайтов, такие как JAMStack CMS, например: [Gatsby](https://www.gatsbyjs.org/" \t "_blank), [Next.js](https://nextjs.org/), [Nuxt](https://nuxtjs.org/" \t "_blank) и [Gridsome](https://gridsome.org/" \t "_blank).

* Особое внимание уделяется Gatsby - многофункциональному решению для поддержки сложных сайтов и блогов. Некоторые из его конечных преимуществ включают в себя:
* встроенная поддержка пакета модуля React.js & Webpack;
* предсказательная выборка;
* мгновенное нажатие;
* поддержка быстрой ссылки по умолчанию preload;
* уже оптимизирован для пользователя;
* прогрессивное улучшение по умолчанию;
* хорошая экосистема;
* бюджетный.

Чтобы помочь вам более эффективно выбирать утилиты, [здесь](https://www.staticgen.com/) вы можете просмотреть галерею совместимых утилит и посмотреть, на каких языках кодирования основывается любая другая часть программного обеспечения.

Кроме того, не забудьте создать учетную запись на GitHub, чтобы у вас было место для хранения версий вашего проекта, а также на Netlify - сервисе для автоматизации создания, развертывания и управления статическими сайтами. Forestry и DatoCMS - это решения с возможностями, аналогичными возможностям Netlify CMS.

### **Инструменты, которые помогут вам преобразовать ваш старый сайт на основе технологии JAM**

Последняя конференция JAMStack, которую с удовольствием посетили наши разработчики, была частично посвящена вопросам рефакторинга старой версии сайта с использованием подхода JAMStack.

Это не самый простой процесс, который, однако, из-за доступных решений CMS (таких как [Gatsby](https://www.gatsbyjs.org/" \t "_blank)), может оказаться совсем не болезненным для вашей команды разработчиков.