Индивидуальный проект. 1 этап.

Дисциплина - Операционные системы

Оширова Юлия Николаевна, НКАбд-02-22

Содержание

# 1 Цель работы

Научиться создавать сайт на Хьюго, создать необходимые репозитории.

# 2 Задание

№1 Первый этап процесса (Хьюго, новый репозиторий).

№2 Второй этап процесса.

# 3 Теоретическое введение

Хьюго - это один из генераторы статических сайтов с открытым исходным кодом самый популярный, который к тому же совершенно бесплатный. Обладая поразительной скоростью и гибкостью, этот генератор позволяет быстро и легко создавать веб-сайты.

Общая характеристика Хьюго

* Высокая скорость. Как указано на их веб-сайте, это самый быстрый инструмент в своем роде. В среднем сайт создается менее чем за секунду.
* Надежное управление контентом и правила гибкости. Хьюго - мечта контент-стратега. Хьюго поддерживает неограниченное количество типов контента, таксономий, меню, динамического контента на основе API и т. д., все без надстроек.
* Шорткоды предлагают нам возможность использовать синтаксис Markdown, обеспечивая большую гибкость.
* Встроенные шаблоны. Этот конструктор статических веб-сайтов имеет общие шаблоны для быстрого создания нашей работы. Hugo поставляется с готовыми шаблонами для быстрой работы с SEO, комментариев, аналитики и других функций.
* Пользовательские выходы. Позволяет нам генерировать наш контент в нескольких форматах, включая JSON или AMP, и, таким образом, облегчить создание контента.
* Доступно более 300 тем, давая нам надежную систему тем, которую легко реализовать, но способную создавать даже самые сложные веб-сайты. Может быть брошен взглянуть на темы доступно на сайте проекта.

# 4 Выполнение лабораторной работы

***Первый этап***

Первое, что нам нужно, это скачать файл хьюго, чтобы генерировать страницу сайта, ссылку можно найти в описании к индивидуальному проекту:

![Скачивание Хьюго](data:application/octet-stream;base64,)

Скачивание Хьюго

Скачиваем файл hugo\_extended последней версии (110) в tar.gz, разархивируем и перенесем в текущую папку. Копируем этот файл и вырежем.

В домашней папке создаем пустую папку «bin» и скопируем этот файл туда:

![Создание папки “Бин” и скопируем файл туда](data:application/octet-stream;base64,)

Создание папки “Бин” и скопируем файл туда

Создаем репозиторий на основе шаблона, переходим по ссылке и нажимаем «use this template». Указываем имя «blog»:

![Создание репозитория на основе шаблона](data:application/octet-stream;base64,)

Создание репозитория на основе шаблона

После этого клонируем его в терминале. Появится каталог «blog», а внутри уже будут все файлы:

![Клонирование репозитория в терминале](data:application/octet-stream;base64,)

Клонирование репозитория в терминале

![Все файлы появились](data:application/octet-stream;base64,)

Все файлы появились

Переходим в блог, можем видеть все файлы с помощью команды «ls -l».

Пишем команду «~»:

![Выполнение команды](data:application/octet-stream;base64,)

Выполнение команды

С помощью команды mc удаляем каталог «public»:

![Удаление “Паблик”](data:application/octet-stream;base64,)

Удаление “Паблик”

Выполним команду «~server»:

![Выполнение команды](data:application/octet-stream;base64,)

Выполнение команды

Видим ссылку в выводе, копируем и вставляем в браузере, видим сайт, который доступен только нам:

![Сайт, который доступен только нам](data:application/octet-stream;base64,)

Сайт, который доступен только нам

***Второй шаг***

Переносим сайт на еще один репозиторий. Надо будет создать его. Чтобы это сделать, надо перейти в гитхаб, создадим репозиторий со специальным названием, в моем случае «joshirova.github.io»:

![Создание репозитория с именем аккаунта на Гитхаб](data:application/octet-stream;base64,)

Создание репозитория с именем аккаунта на Гитхаб

Переходим в терминал, клонируем репозиторий с помощью команды «git clone —rescursive (ваша ссылка на репозиторий)»:

![Клонируем репозиторий](data:application/octet-stream;base64,)

Клонируем репозиторий

Перейдя в него, выполняем команду «git checkout -b main»:

![Выполнение команды](data:application/octet-stream;base64,)

Выполнение команды

После этого создаем пустой файл с помощью команды «touch README.md». Добавляем файл на гитхаб с помощью уже знакомых команд с прошлого и нынешнего семестра:

![Создаем пустой файл и добавляем на Гитхаб](data:application/octet-stream;base64,)

Создаем пустой файл и добавляем на Гитхаб

Проверив в браузере на гитхабе в репозитории, удостоверились, что файл появился.

Переходим обратно в blog, выполним команду, которая подключит новый репозиторий к папке public внутри нашего blog (заранее исправив «#public/ в комментировании .gitignore):

![Подключаем новый репозиторий к папке Паблик](data:application/octet-stream;base64,)

Подключаем новый репозиторий к папке Паблик

С помощью команды «~/bin/hugo» в папке public автоматически появились файлы:

![С помощью команды осуществилось появление файлов](data:application/octet-stream;base64,)

С помощью команды осуществилось появление файлов

Синхронизуем эти файлы с репозиторием, возвращаемся в public, делаем стандартные действия:

![Отправка на Гитхаб](data:application/octet-stream;base64,)

Отправка на Гитхаб

Обновляем репозиторий и видим, что все файлы появились.

Копируем ссылку на наш новый сайт, подождём секунд 20 и позже появится полноценный сайт, который в дальнейших этапах индивидуального проекта мы будем редактировать:

![Наш будущий сайт](data:application/octet-stream;base64,)

Наш будущий сайт

# 5 Выводы

Я создала сайт на Хьюго, а также нужные репозитории.

# Список литературы

[1. Что такое Хьюго?](https://ubunlog.com/ru/hugo-generador-sitios-web-estaticos/)

[2. Первый этап индивидуального проекта](https://vk.com/away.php?utf=1&to=https%3A%2F%2Fyoutu.be%2FOpsSv0RE3C4) ::: {#refs} :::