

Понятие страницы сайта. Что такое HTML?

Настройка среды разработки

Цели занятия

- разобраться с тем, что такое страница сайта;
- дать определение термину HTML и узнать какие задачи он решает;
- установить необходимые для работы программы и настроить их для дальнейшей работы

Что такое страница сайта?

Страница веб-сайта — самостоятельная часть веб-ресурса, представляющая собой текстовый файл в формате HTML, который доступен в сети Интернет по уникальному адресу (URL).

URL (Uniform Resource Locator) — адрес, который выдан уникальному ресурсу в интернете. В теории, каждый корректный URL ведёт на уникальный ресурс. Такими ресурсами могут быть HTML-страница, CSS-файл, изображение и т.д. На практике, существуют некоторые исключения, когда, например, URL ведёт на ресурс, который больше не существует или был перемещён.

Подробнее: [что такое URL?](#)

Что такое HTML?

HTML (Hypertext Markup Language) — код, который используется для структурирования и отображения веб-страницы и её контента. Например, контент может быть структурирован внутри множества

параграфов, маркированных списков или с использованием изображений и таблиц данных. Данный язык не является языком программирования, а считается языком разметки, и используется для того, чтобы объяснить браузеру как отображать вашу страницу.

Подробнее: [что такое HTML?](#)

Необходимые программы и инструменты

Google Chrome — один из самых популярных браузеров в мире, на февраль 2022 года его доля занимает чуть более 62%.

Используется программистом для просмотра результата во время разработки / [Скачать](#)

Chrome DevTools — встроенные в браузер инструменты для разработчиков, содержат редактор кода, дебаггер, наглядную систему управления стилями, инструменты для проверки адаптивности сайта под различные устройства и автоматической проверки сайтов на качество (производительность, SEO, адаптация под людей с ограниченными возможностями и т.п.) / [Подробнее](#)

VS Code — редактор исходного кода, разработанный компанией Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. / [Скачать](#)

Данный редактор разработан на языке JavaScript и поэтому весь функционал редактора доступен даже онлайн без скачивания на компьютер / [Открыть](#)

GitLab — это инструмент для хранения и управления репозиториями Git. Он дает возможность выполнять совместную разработку силами нескольких команд, применять обновления кода и откатывать изменения, если это необходимо / [Зарегистрироваться](#)

В разных операционных системах может потребоваться установка дополнительных программ, чтобы иметь возможность работать с Git, например, для Windows необходимо [установить Git Bash](#), для Linux можно воспользоваться [инструкцией](#), а также есть [инструкция для MacOS](#)

Figma — онлайн-сервис для разработки интерфейсов и прототипирования с возможностью организации совместной работы в режиме реального времени. Используется дизайнерами для создания макетов и прототипов, а фронтенд-разработчики используют Figma для просмотра данных макетов и отдельных параметров каждого элемента / [Зарегистрироваться](#)

Первоначальная настройка Git

Когда Git установлен в вашей системе, самое время настроить среду для работы с Git под себя. Это нужно сделать только один раз — при обновлении версии Git настройки сохранятся. Но, при необходимости, вы можете поменять их в любой момент, выполнив те же команды снова.

- Открыть терминал внутри VS Code (можно с помощью интерфейса или горячими клавишами: нажать `Ctrl + `` для Windows или `⌘Ctrl + `` для macOS)
- Указать ваше имя с помощью команды в терминале:
`git config --global user.name "Yury Kudin"`
- Указать вашу почту с помощью команды:
`git config --global user.email yury.kudin@yandex.ru`

Основные команды для работы с Git

`git clone https://gitlab.com/Kundin/test.git` — скопировать репозиторий для локальной работы

`git add .` — внести в индекс все изменённые файлы

`git commit -m "Initial commit"` — создать коммит с сообщением "Initial Commit" (перед выполнением коммита необходимо добавить все файлы, то есть выполнить команду `git add .`)

`git pull` — получить изменения из репозитория

`git push` — загрузить изменения в репозиторий

`git status` — посмотреть состояние репозитория, если имеются какие-то изменения, то в консоли они будут показаны все затронутые файлы