1. Назначение сервиса GitHub:

GitHub – это платформа для совместной работы над программным кодом. Она предоставляет ряд инструментов для управления версиями кода, совместного редактирования, отслеживания изменений и общения между разработчиками.

Основные функции:

Хостинг репозиториев: Хранение и управление проектами с кодом.

Версионный контроль: Отслеживание всех изменений в коде, возможность вернуться к любой предыдущей версии.

Сотрудничество: Разделение задач, совместная работа над кодом, общение в рамках проекта.

Открытый доступ: Возможность опубликовать свой код открыто для всего мира или ограничить доступ.

2. Назначение сервиса Github Desktop:

GitHub Desktop – это графический интерфейс для работы с Git. Он упрощает использование командной строки Git, позволяя выполнять основные операции с репозиториями, ветками и коммитами.

Основные функции:

Простой интерфейс: Облегчает работу с Git для новичков.

Визуализация: Показывает структуру репозитория, изменения в коде, историю коммитов.

Основные операции: Создание коммитов, пушей, пулреквестов, мерджей.

Интеграция с GitHub: Обеспечивает плавный переход между GitHub Desktop и сайтом GitHub.

Термины Git и GitHub:

1. Репозиторий:

Определение: Директория (папка) на вашем компьютере или на сервере, которая содержит все файлы проекта, историю изменений и метаданные.

Пример: Репозиторий на GitHub, хранящий исходный код вашего сайта.

2. Рабочая область и хранилище:

Определение:

Рабочая область: Место, где вы редактируете файлы проекта (как правило, на вашем локальном компьютере).

Хранилище: Место, где хранится история всех изменений и метаданные проекта (как правило, на сервере, например, на GitHub).

Пример:

Вы открываете проект в редакторе, изменяете файл - это рабочая область.

Сохраняете изменения в хранилище, чтобы сохранить историю изменений.

3. Форк:

Определение: Копия репозитория, которую вы можете изменять независимо от оригинального репозитория.

Пример: Вы нашли интересный проект на GitHub и хотите внести свои изменения - вы делаете форк репозитория.

4. Ветка:

Определение: Временная копия репозитория, используемая для внесения изменений без влияния на основную версию.

Пример: Вы хотите добавить новую функцию, но не хотите сразу вносить изменения в основную версию. Вы создаете ветку, делаете изменения, а затем сливаете ветку с основной.

5. Мастер:

Определение: Основная ветка репозитория, которая обычно содержит стабильную версию проекта.

Пример: Вы используете мастер для выпуска новых релизов проекта.

6. Коммит:

Определение: Сохранение изменений в репозитории. Каждый коммит имеет уникальный идентификатор и описание изменений.

Пример: Вы исправили ошибку в коде и сделали коммит с описанием "Исправлена ошибка".

7. Пул:

Определение: Получение изменений из другого репозитория в ваш локальный.

Пример: Вы хотите обновить свой локальный репозиторий с последними изменениями, сделанными в оригинальном репозитории.

8. Пуш:

Определение: Отправка изменений из локального репозитория в хранилище на сервере.

Пример: Вы завершили работу над новой функцией и сделали пуш в репозиторий на GitHub.

9. Пулреквест:

Определение: Запрос на слияние ветки из вашего репозитория в оригинальный репозиторий.

Пример: Вы создали новую ветку, внесли изменения и создали пулреквест, чтобы автор проекта мог просмотреть ваши изменения и принять их.

10. Мердж:

Определение: Слияние ветки с другой веткой (чаще всего с мастер-веткой).

Пример: Автор проекта одобрил ваш пулреквест, и ваша ветка была мерджен в мастер-ветку.

11. Кодревью:

Определение: Визуальное представление изменений в коде, сделанных в коммите.

Пример: Вы смотрите на кодревью и видите, какие строчки кода были добавлены, удалены или изменены.