**ШПАРГАЛКА ПО ОСНОВНЫМ КОМАНДАМ GIT**

| **Команда** | **Описание** |
| --- | --- |
| **git add** | Добавляет содержимое рабочей директории в индекс (*staging area*) для последующего коммита. |
| **git status** | Показывает состояния файлов в рабочей директории и индексе: какие файлы изменены, но не добавлены в индекс; какие ожидают коммита в индексе. |
| **git commit** | Команда берет все данные, добавленные в индекс с помощью***git add***, и сохраняет их снимок во внутренней базе данных, а затем сдвигает указатель текущей ветки на этот снимок. |
| **git clean** | Удаление мусора из рабочей директории. |

**ШПАРГАЛКА ПО ОСНОВНЫМ КОМАНДАМ ВЕТВЛЕНИЯ И СЛИЯНИЯ**

|  |  |
| --- | --- |
| **git branch** | Управление ветками (перечисление, создание, удаление, переименование). |
| **git checkout** | Перемещение между ветками и выгрузки их содержимого в рабочую директорию. |
| **git merge** | Слияние одной или нескольких веток. |
| **git stash** | Временное сохранение всех незакоммиченных изменений для очистки рабочей директории без необходимости коммитить незавершенную работу в новую ветку. |

**ШПАРГАЛКА ПО СОВМЕСТНОЙ РАБОТЕ И ОБНОВЛЕНИЮ**

|  |  |
| --- | --- |
| **git fetch** | Извлекает и сохраняет локально из удаленного репозитория все изменения, которых пока у вас нет. |
| **git pull** | Работает как комбинация команд ***git fetch*** и ***git merge***, то есть *Git* вначале забирает изменения из указанного удалённого репозитория, а затем пытается слить их с текущей веткой. |
| **git push** | Используется для установления связи с удаленным репозиторием (сервером), вычисления локальных изменений, отсутствующих в нём, и собственно их передачи в репозиторий. Этой команде нужно право на запись в репозиторий, поэтому она использует аутентификацию. |

**Несколько «стандартных» слов**

Есть слова, которые уже воспринимаются разработчиками как определенные действия:

* + - **init** — инициализация;
    - **add** — добавление;
    - **delete** — удаление;
    - **update** — изменение;
    - **fix** — исправление;
    - **refactor** — рефакторинг кода приложения;
    - **style** — исправление опечаток, форматирования;
    - **docs** — всё, что касается документации;
    - **test** — всё, что связано с тестированием;
    - **merged, fix conflict** — слияние, решение конфликта.