

## Лабораторная работа №6.

### Работа с ветками

**Цель:** научиться работать с ветками.

1. Для работы с ветками в первую очередь необходимы две команды:

Создать ветку с помощью команды `git branch имя_ветки`

Переключиться на свежесозданную ветку с помощью команды `git checkout`

`<имя ветки>`

Создадим ветку feature и переключимся на нее.

```
пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (main)
$ git branch feature

пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (main)
$ git checkout feature
Switched to branch 'feature'

пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (feature)
$
```

Как мы видим, Git написал “Переключился на ветку feature” и сменил ветку main на новую созданную ветку.

Команда `git branch` без указания имени ветки выводит список всех локальных веток в текущей директории:

```
пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (main)
$ git branch
feature
* main
```

Если вы не так назвали ветку, то ее можно переименовать:

`git branch -m <oldname> <newname>`

2. Команда `git checkout -b` как бы объединяет вышеуказанные команды: она и создает ветку, и переключается на нее. Создадим новую ветку vetka и перейдем на нее, используя одну команду.

```
пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (feature)
$ git checkout -b vetka
Switched to a new branch 'vetka'

пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (vetka)
$ |
```

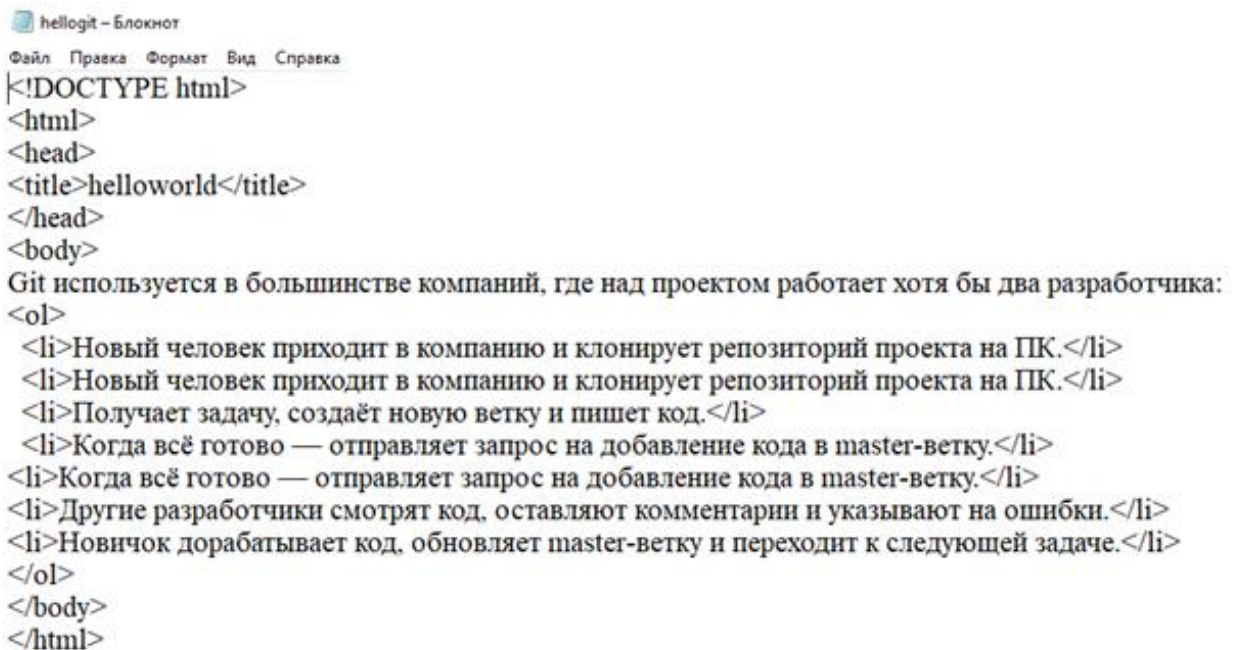
`git checkout -b <name> <remote>/<branch>` — переключиться на удаленную ветку

`git checkout [filename]` — вернуть файл в первоначальное состояние, если он еще не был добавлен в индекс коммита

3. Git хранит специальный указатель **HEAD** — это указатель на текущую локальную ветку. При помощи известной команды `git log -- oneline --decorate` посмотрим куда указывают указатели веток:

```
пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (vetka)
$ git log --oneline --decorate
be72de3 (HEAD -> vetka, origin/main, main, feature) Second Commit
5566069 my first commit
69664c1 First Commit
```

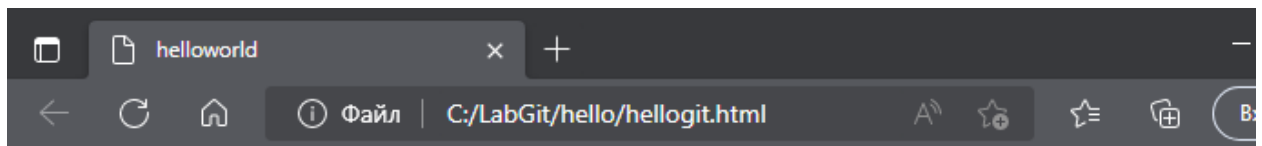
4. Создадим новый файл `hellogit.html`, с которым далее будем работать.



hellogit - Блокнот

Файл Правка Формат Вид Справка

```
<!DOCTYPE html>
<html>
<head>
<title>helloworld</title>
</head>
<body>
Git используется в большинстве компаний, где над проектом работает хотя бы два разработчика:
<ol>
<li>Новый человек приходит в компанию и клонирует репозиторий проекта на ПК.</li>
<li>Новый человек приходит в компанию и клонирует репозиторий проекта на ПК.</li>
<li>Получает задачу, создаёт новую ветку и пишет код.</li>
<li>Когда всё готово — отправляет запрос на добавление кода в master-ветку.</li>
<li>Когда всё готово — отправляет запрос на добавление кода в master-ветку.</li>
<li>Другие разработчики смотрят код, оставляют комментарии и указывают на ошибки.</li>
<li>Новичок дорабатывает код, обновляет master-ветку и переходит к следующей задаче.</li>
</ol>
</body>
</html>
```



Git используется в большинстве компаний, где над проектом работает хотя бы два разработчика:

1. Новый человек приходит в компанию и клонирует репозиторий проекта на ПК.
2. Новый человек приходит в компанию и клонирует репозиторий проекта на ПК.
3. Получает задачу, создаёт новую ветку и пишет код.
4. Когда всё готово — отправляет запрос на добавление кода в master-ветку.
5. Когда всё готово — отправляет запрос на добавление кода в master-ветку.
6. Другие разработчики смотрят код, оставляют комментарии и указывают на ошибки.
7. Новичок дорабатывает код, обновляет master-ветку и переходит к следующей задаче.

5. Мы находимся в ветке `vetka` и у нас создан новый файл, который не залит на GitHub.

Добавим содержимое файла в индекс, введя команду `git add имя_файла`.

Далее закоммитим изменения в новую ветку, введя команду `git commit -m "Commit new vetka"`. И используя команду `git push`, внесем изменения в репозиторий GitHub.

```
пользователь@LAPTOP-27F0P5I5 MINGW64 /c/LabGit/hello (vetka)
$ git add hellogit.html

пользователь@LAPTOP-27F0P5I5 MINGW64 /c/LabGit/hello (vetka)
$ git status
On branch vetka
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   hellogit.html

пользователь@LAPTOP-27F0P5I5 MINGW64 /c/LabGit/hello (vetka)
$ git commit -m "Commit new vetka"
[vetka 47e6e48] Commit new vetka
1 file changed, 18 insertions(+)
create mode 100644 hellogit.html

пользователь@LAPTOP-27F0P5I5 MINGW64 /c/LabGit/hello (vetka)
$ git push
fatal: The current branch vetka has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin vetka
```

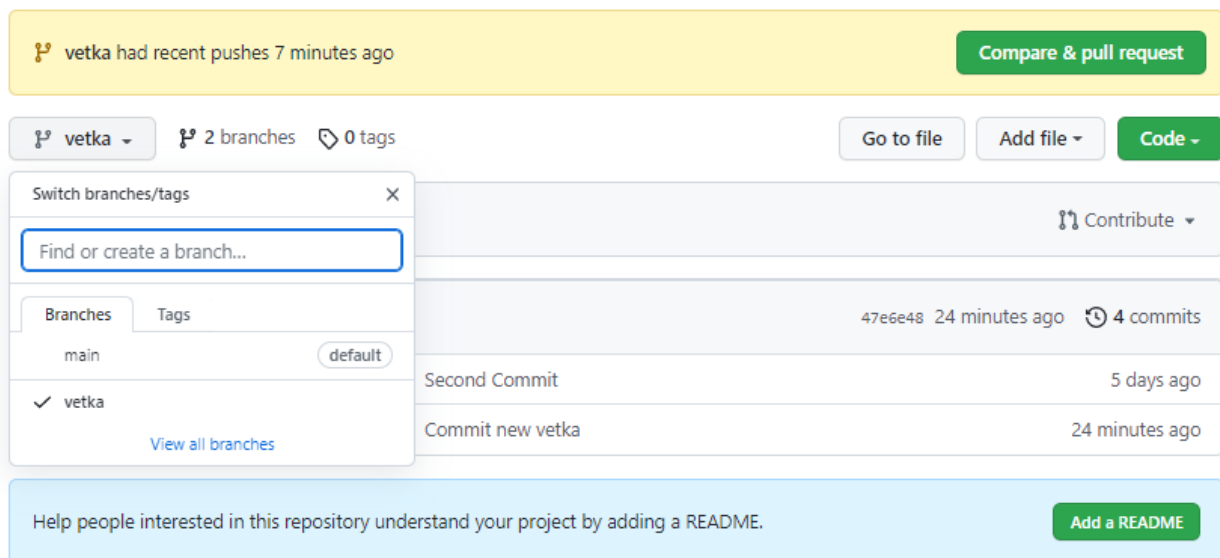
Возникла ошибка: Текущая ветка `vetka` не имеет вышестоящей ветки. Git подсказывает нам, какую команду использовать: `git push --set-upstream origin vetka`.

```

пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (vetka)
$ git push --set-upstream origin vetka
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 875 bytes | 291.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for ' vetka ' on GitHub by visiting:
remote:   https://github.com/for-you-nastya/studying-Git/pull/new/vetka
remote:
To https://github.com/for-you-nastya/studying-Git.git
 * [new branch]      vetka -> vetka
Branch 'vetka' set up to track remote branch 'vetka' from 'origin'.

```

Посмотрим, что изменилось на GitHub:



6. Посмотрим список всех созданных веток:

`git branch -a`

```

пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (vetka)
$ git branch -a
feature
main
* vetka
remotes/origin/feature
remotes/origin/main
remotes/origin/vetka

```

Самостоятельно перейдите на другую ветку и посмотрите все ветки и в какой находимся мы:

```

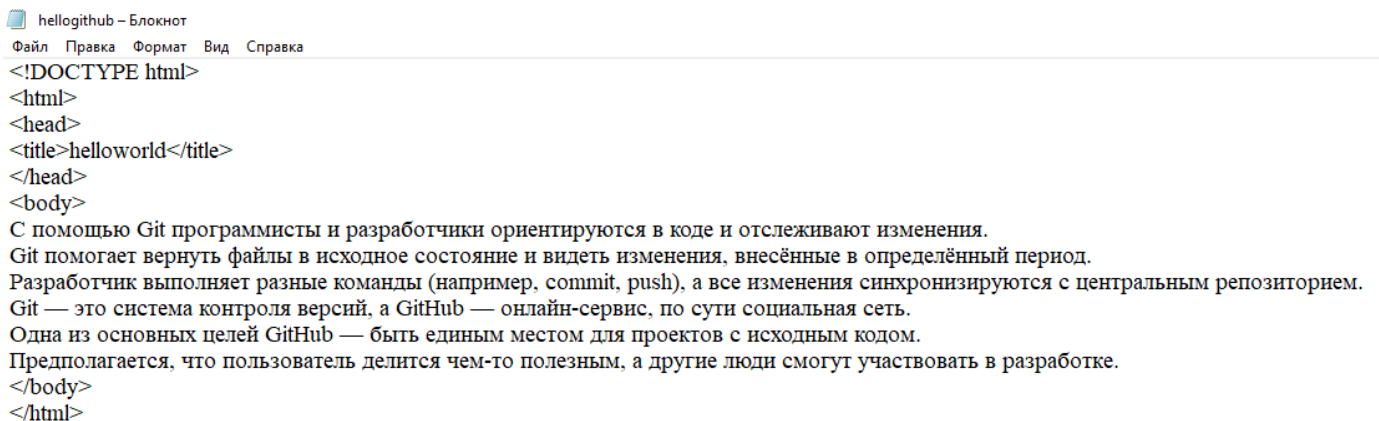
пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (vetka)
$ git checkout feature
Switched to branch 'feature'
Your branch is up to date with 'origin/feature'.

пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (feature)
$ git branch -a
* feature
  main
  vetka
remotes/origin/feature
remotes/origin/main
remotes/origin/vetka

```

7. Удаление веток происходит при помощи команды `git branch -d <имя ветки>`. Этот ключ выдаст предупреждение, если в ветке находятся изменения, которые не «слились» с основной веткой. Если хотите удалить без предупреждений, тогда нужно использовать ключ `-D`.

8. Самостоятельно создайте новый файл и поместите его на ветку `feature`, посмотрите изменения на GitHub.



```

hellogithub - Блокнот
Файл  Правка  Формат  Вид  Справка
<!DOCTYPE html>
<html>
<head>
<title>helloworld</title>
</head>
<body>
С помощью Git программисты и разработчики ориентируются в коде и отслеживают изменения.
Git помогает вернуть файлы в исходное состояние и видеть изменения, внесённые в определённый период.
Разработчик выполняет разные команды (например, commit, push), а все изменения синхронизируются с центральным репозиторием.
Git — это система контроля версий, а GitHub — онлайн-сервис, по сути социальная сеть.
Одна из основных целей GitHub — быть единым местом для проектов с исходным кодом.
Предполагается, что пользователь делится чем-то полезным, а другие люди смогут участвовать в разработке.
</body>
</html>

```

```

пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (vetka)
$ git checkout feature
Switched to branch 'feature'

пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (feature)
$ git add hellogithub.html

пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (feature)
$ git commit -m "Commit feature hellogithub"
[feature 100a191] Commit feature hellogithub
1 file changed, 10 insertions(+)
create mode 100644 hellogithub.html

пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (feature)
$ git push
fatal: The current branch feature has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin feature

пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (feature)
$ git push --set-upstream origin feature
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 875 bytes | 291.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature' on GitHub by visiting:
remote:   https://github.com/for-you-nastya/studying-Git/pull/new/feature
remote:
To https://github.com/for-you-nastya/studying-Git.git
 * [new branch]      feature -> feature
Branch 'feature' set up to track remote branch 'feature' from 'origin'.

```

feature
3 branches
0 tags
Go to file
Add file
Code

Switch branches/tags
Find or create a branch...

Branches

Tags

main

feature

vetka

default

View all branches

Contribute

100a191 12 минут назад 4 commits

Second Commit 5 дней назад

Commit feature hellogithub 12 минут назад

Add a README