

Master

Hotfix

Release

Develop

Feature

Feature

v0.1

<> v0.2

v1.0

شاخه‌ها

مفاهیم و دستورات کار با شاخه‌ها در گیت

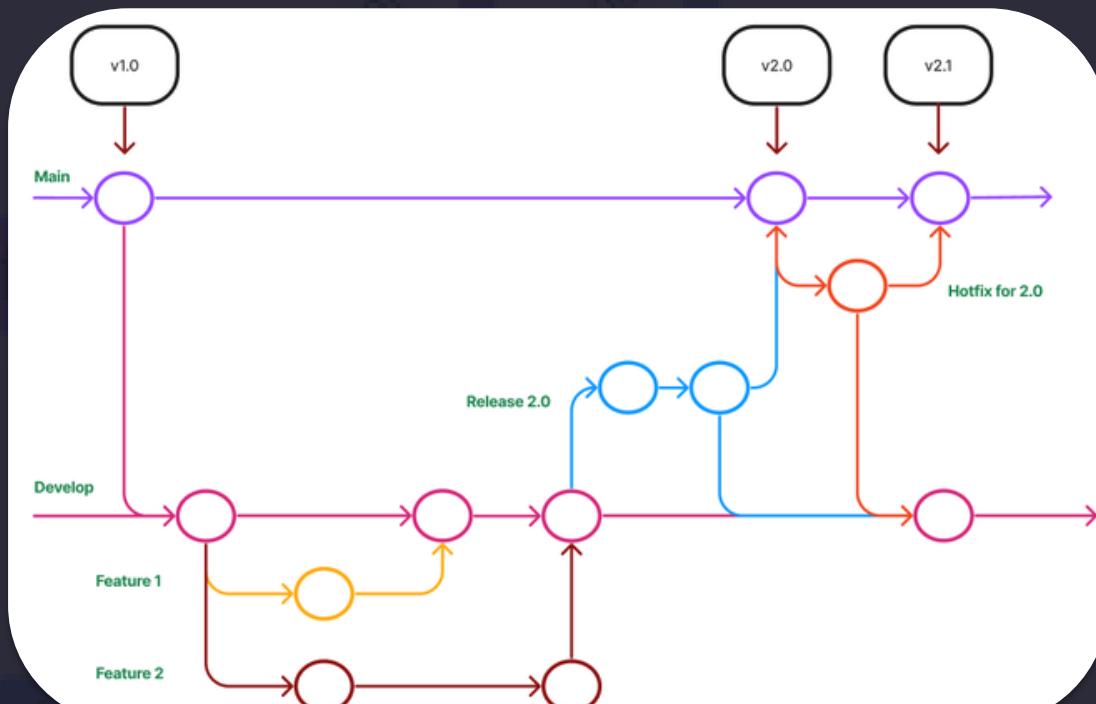
۴۰۱۳۲۰۰۴۸ محمدامین محمدی

دکتر علی طائی زاده

درس نرم افزار ا

main

v0.4.0



v0.4.1

v0.4.2

معرفی شاخه‌ها

شاخه‌ها در گیت به شما اجازه می‌دهند تا به صورت موازی روی ویژگی‌های مختلف کار کنید.
بدون اینکه روی کار یکدیگر تأثیر بگذارید.

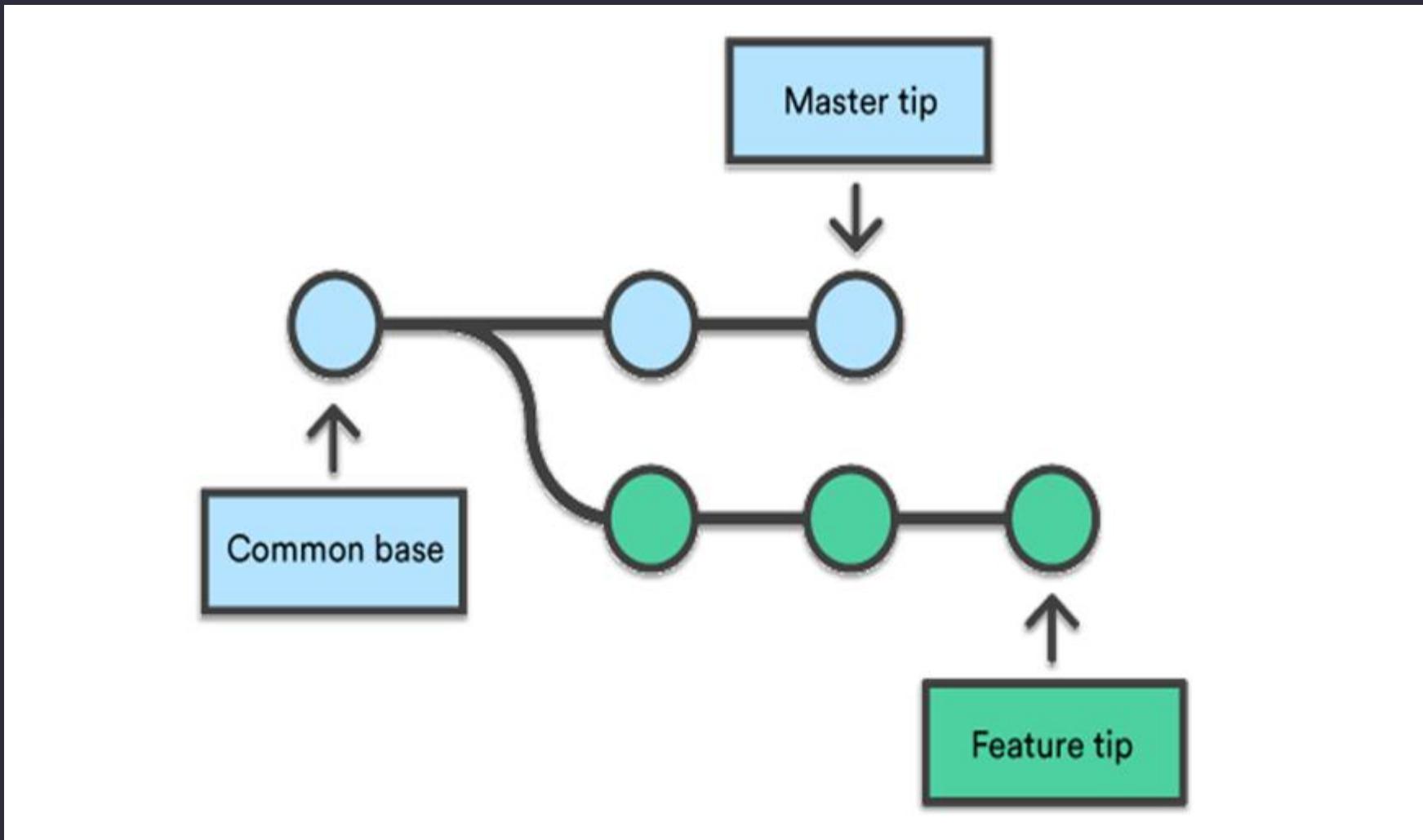
امکان کار همزمان چندین توسعه‌دهنده روی پروژه - توسعه همزمان و مستقل

شاخه اصلی همیشه پایدار و قابل استفاده باقی می‌ماند - حفظ پایداری شاخه اصلی

امکان آزمایش ویژگی‌های جدید بدون تأثیر بر کد اصلی - توسعه و آزمایش بدون ریسک

شاخه‌های محلی و ریموت





اتصال به گیت‌هاب

This terminal is Termite,
using a Basel6 Gruvbox Dark theme,
and using the font Hack at 11 pt.

It's also using git-bash-prompt, and
the icon at the bottom is from Thunar.

ایجاد شاخه‌های متصل به گیت‌هاب <>

✓ ~/apps/shed+js [master] 20:10

ایجاد شاخه محلی متصل به شاخه ریموت

git checkout -b



git push -u



اتصال شاخه محلی به شاخه ریموت

git push --set-upstream



روش جایگزین برای اتصال شاخه‌ها

اتصال مخزن محلی به گیت‌هاب



git remote add

افزودن مخزن ریموت به پروژه محلی

git remote -v



بررسی اتصال به مخزن ریموت

git branch -r



مشاهده شاخه‌های ریموت

git branch -a



مشاهده تمام شاخه‌های محلی و ریموت

نمونه عملی اتصال به گیت‌هاب

```
git remote add origin https://github.com/user/project.git $
```

```
git remote -v $
```

```
origin https://github.com/user/project.git (fetch)
```

```
origin https://github.com/user/project.git (push)
```

```
git checkout -b feature $
```

```
Switched to a new branch 'feature'
```

```
git push -u origin feature $
```

ایجاد و جابجایی بین شاخه‌ها

MINGW64:/c/source/stevenalexander.github.io

ایجاد شاخه جدید



ایجاد و جابجایی به شاخه



git checkout -b (قدیمی)



git switch -c (جدید)



ایجاد شاخه جدید و جابجایی به آن

git checkout



git switch (جدید)



git checkout - (شاخه قبلی)



git switch - (شاخه قبلی)



ایجاد شاخه از شاخه دیگر



git branch



git checkout -b



ایجاد شاخه جدید بر اساس شاخه موجود

xander.github.io (master)

GUI?

```
git checkout -b feature/login $
```

```
git branch $
```

```
* feature/login
```

```
main
```

```
git checkout main $
```

```
Switched to branch 'main'
```

```
git checkout - $
```

```
Switched to branch 'feature/login'
```

ف شاخه‌های ریموت



git fetch --prune

نگام‌سازی شاخه‌های محلی و ریموت و حذف شاخه‌های ریموت حذف شده

حذف شاخه از مخزن ریموت

ش جایگزین برای حذف شاخه از مخزن ریموت

10 *article, aside, details, figcaption, figheader, figtext, form, header, h1, h2, h3, h4, h5, h6, hr, input, main, menu, output, section, summary, time, mark, audio, video* {
11 *b, u, i, center,*
12 *dl, dt, dd, ol, ul, li,*
13 *fieldset, form, label, legend,*
14 *table, caption, tbody, tfoot, thead, tr, th, td,*
15 *article, aside, canvas, details, embed,*
16 *figure, figcaption, footer, header, hgroup,*
17 *menu, nav, output, ruby, section, summary,*
18 *time, mark, audio, video {*

حذف شاخه‌ها

حذف شاخه‌های محلی



```
git branch -d
```

فقط اگر ادغام شده باشد) حذف امن شاخص

git branch -D



حتی اگر ادغام نشده باشد) حذف اجباری شاخه

git push



نمونه عملی حذف شاخه‌ها



```
git branch $
```

```
main
```

```
* feature/login
```

```
git checkout main $
```

```
Switched to branch 'main'
```

```
git branch -d feature/login $
```

```
Deleted branch feature/login (was 3a2b1c4).
```

```
git push origin --delete feature/login $
```

```
To github.com:user/repo.git
```

```
- [deleted] feature/login
```

ادغام و تداخلات

ادغام Three-way

زمانی که هر دو شاخه تغییرات داشته‌اند و گیت باید یک کامیت ادغام جدید ایجاد کند

git checkout target-branch

<>

git merge branch-name

<>



ادغام Fast-forward

زمانی که شاخه هدف هیچ تغییری نداشته و فقط شاخه منبع پیشرفت کرده است

git checkout target-branch

<>

git merge branch-name

<>

Uncommitted changes

Markus
c31094a
Update test_Sub

Markus
17:17 - Commit - c31094a - Update test_Sub
17:21 - Merge - 89829b7 into B
There was a merge conflict.

Show in Explorer

Edit in External Editor

External Merge

Stage

Unstage

Discard Changes

Remove Untracked Files

Ignore

تداخلات در ادغام



```
zean_7@BlueBytes:~/testing_merge_conflict$ git checkout master
Switched to branch 'master'
zean_7@BlueBytes:~/testing_merge_conflict$ echo "Append this text to initial commit" >> test_file.txt
zean_7@BlueBytes:~/testing_merge_conflict$ git add .
zean_7@BlueBytes:~/testing_merge_conflict$ git commit -m "appended some text to initial commit"
[master 5f30568] appended some text to initial commit
1 file changed, 1 insertion(+)
zean_7@BlueBytes:~/testing_merge_conflict$ git merge new_branch_for_merge_conflict
Auto-merging test_file.txt
CONFLICT (content): Merge conflict in test_file.txt
Automatic merge failed; fix conflicts and then commit the result.
zean_7@BlueBytes:~/testing_merge_conflict$
```

در این موارد باید به تداخلات زمانی رخ می‌دهند که گیت نتواند به طور خودکار تغییرات را ادغام کند: صورت دستی تداخلات را حل کنید

باز کردن فایل‌های دارای تداخل

یافتن و حل تداخلات مشخص شده با علامت

حذف علامت‌های تداخل

اضافه کردن فایل‌های اصلاح شده با دستور **git add**

تکمیل ادغام با دستور **git commit**

عملیات Push و Pull

عملیات Pull



git pull

درایافت و ادغام تغییرات از شاخه ریموت متناظر

git pull master



درایافت و ادغام تغییرات از شاخه اصلی مخزن ریموت

git pull



درایافت و ادغام تغییرات از شاخه مشخص مخزن ریموت

کنید stash کنید commit، تغییرات محلی خود را pull قبل از

عملیات Push



git push

ارسال تغییرات شاخه فعلی به مخزن ریموت

git push --all



ارسال تمام شاخه‌ها به مخزن ریموت

git push



ارسال شاخه مشخص به مخزن ریموت مشخص

کرده‌اید commit، مطمئن شوید که تغییرات محلی خود را push قبل از

```
git add . $
```

```
git commit -m "Add new feature" $
```

```
[feature/login 2a3b4c5] Add new feature
```

```
1 file changed, 5 insertions(+)
```

```
git push origin feature/login $
```

```
Enumerating objects: 5, done.
```

```
To github.com:user/repo.git
```

```
1a2b3c4..2a3b4c5 feature/login -> feature/login
```

```
git checkout main $
```

```
Switched to branch 'main'
```

```
git pull origin main $
```

```
Updating 3d4e5f6..4e5f6g7
```

```
Fast-forward
```