

GIT Basics

Version management



git

Version Control

- version control is the management of changes to files.
 - In our case its our application code



What's Git

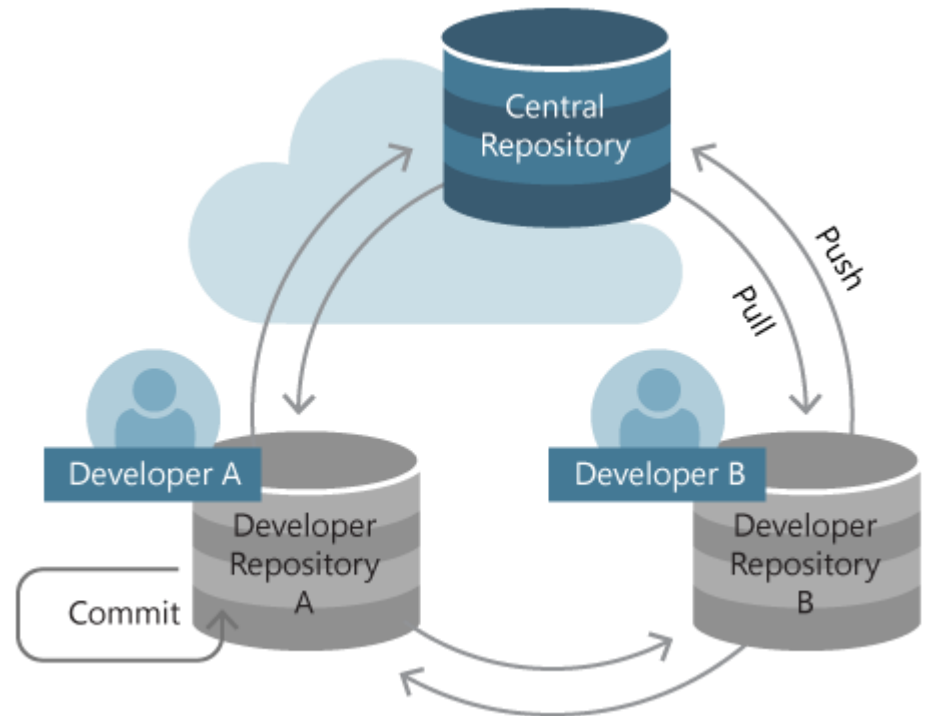
- **Git** is a free and open source distributed version control system
- Suitable to small to very large projects
- Most popular! (other options: svn, clearcase, etc)



git

Using Git

- When using Git, developers are **committing** their changes to their *local repositories*
- And **pushing** their commits to a central (*remote*) repository



GIT Basic commands

- Initialize a new empty local git repository:
`git init`
- Add all files to staged changes status:
`git add .`
- Save changes in your local repository:
`git commit -m "<some message>"`
- Connect a local repo to your github repositories
`git remote add origin "<url from github>"`
- Push all files from your local repository to remote
`git push //(on the first time add: <-u origin main>)`
- To clone a repo from github:
`git clone "<url from github>"`
- To get all updates from remote to your local repo:
`git pull`

Ignoring Files

- Ignoring files is a must in version management.
- .gitignore - contains file names to ignore while committing versions



GIT reset

- With git we can go to a previous version we committed.
- BE CAREFUL - IRREVERSIBLE
- `git reset --hard <commit ID>`



Github

- Remote git repository service

```
git remote add origin <git repository link>
```



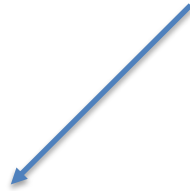
Collaborating with GIT



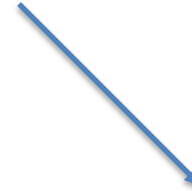
Team version control

- When controlling versions in a team

Commit local version



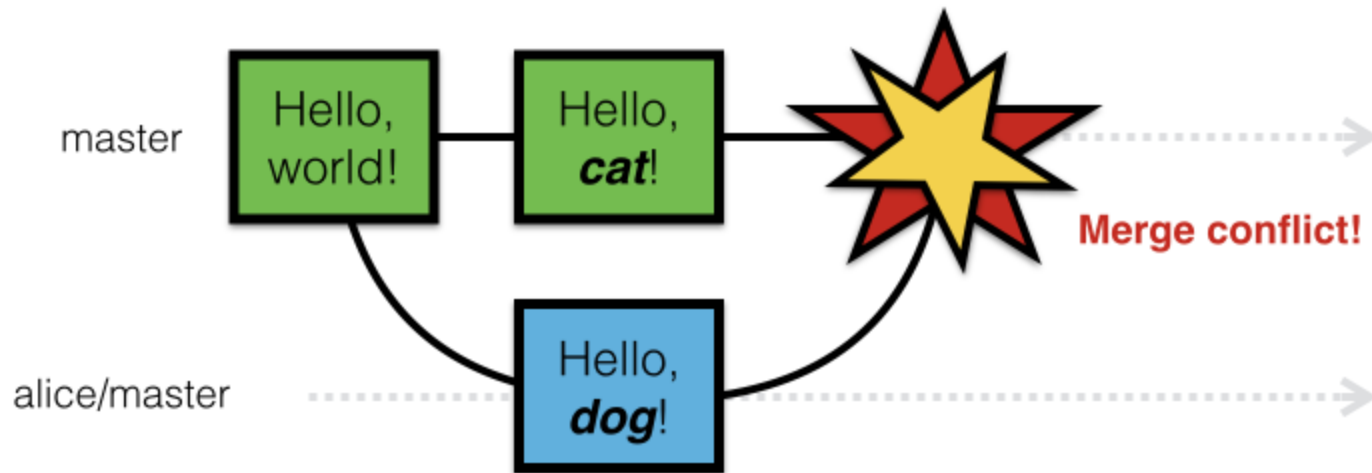
Push to update
remote
repository
`git push`



Pull from **remote**
repository to
update **local**
repository
`git pull`

Git Conflicts

- A conflict-marked area begins with <<<<<< and ends with >>>>>>.
- These are also known as the conflict markers.
- The two conflicting blocks themselves are divided by a =====.



Git Conflicts

```
index.html  styles.css
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>GC Merge Demo</title>
5    <link href="styles.css" rel="stylesheet" type="text/css" />
6  </head>
7  <body>
8  <====<==== HEAD
9    <h1>Grand Circus Merge Demo</h1>
10   =====
11    <h1 class="header">Merge Demo</h1>
12    >>>>>> 1d46372af5a97f8ef05b9eecb82712382cc5f31c
13    <p>
14      Demo the merge.
15    </p>
16    <p class="footer">
17      Grand Circus Detroit
18    </p>
19  </body>
20  </html>
21
```

Collaborating with GIT

Add collaborates to your github repository:

1) Go to your repository and then click on the settings button.

The screenshot shows the GitHub repository page for 'Avivissachar73 / Touch-the-nums'. The repository has 1 commit, 1 branch, 0 packages, 0 releases, 1 environment, and 1 contributor. The 'Settings' button is circled in red, and a red arrow points to it from the right. Another red arrow points to the 'Settings' button from below. The repository description is 'No description, website, or topics provided.' The repository contains the following files and commits:

File	Commit	Time
css	update bgc	14 days ago
js	my first commit	14 days ago
.gitignore	adding gitignore	14 days ago
index.html	my first commit	14 days ago

At the bottom, there is a prompt to 'Add a README'.

Collaborating with GIT

2) In the settings tab click on “Manage access”, (here you might be asked to re enter your password in order to continue).

The screenshot shows the GitHub interface for the repository 'Avivissachar73 / Touch-the-nums'. At the top, there are buttons for 'Unwatch' (1), 'Star' (0), and 'Fork' (0). Below this is a navigation bar with tabs for 'Code', 'Issues' (0), 'Pull requests' (0), 'Actions', 'Projects' (0), 'Wiki', 'Security', 'Insights', and 'Settings'. The 'Settings' tab is selected and highlighted with an orange bar. On the left side of the 'Settings' page, there is a sidebar with a list of options: 'Options', 'Manage access', 'Branches', 'Webhooks', 'Notifications', 'Integrations & services', 'Deploy keys', 'Secrets', 'Actions', 'Moderation', and 'Interaction limits'. The 'Manage access' option is circled in red, and a red arrow points to it from the right. The main content area of the 'Settings' page is titled 'Settings' and contains sections for 'Repository name' (with a text input field containing 'Touch-the-nums' and a 'Rename' button), 'Template repository' (with a checkbox and a description), and 'Social preview' (with a description and a 'Download template' link).

Collaborating with GIT

- 3) Click the big green “Invite a Collaborator” button.

Manage access

- Branches
- Webhooks
- Notifications
- Integrations & services
- Deploy keys
- Secrets
- Actions

Moderation

- Interaction limits

PUBLIC REPOSITORY

This repository is public and visible to anyone.

[Manage](#)

DIRECT ACCESS

0 collaborators have access to this repository. Only you can contribute to this repository.

Manage access

You haven't invited any collaborators yet

If you're using GitHub Free, you can add unlimited collaborators on public repositories, and up to three collaborators on private repositories owned by your personal account. [Learn more](#)

Invite a collaborator

Collaborating with GIT

- 4) In the collaborators section find your collaborator (by github username) and click on his name, then click on the big green “Add Collaborator” button.

The image displays two screenshots of the GitHub 'Who has access' dialog for the repository 'Touch-the-nums'. The left screenshot shows the search bar with 'vyaron' entered, and a list of results including 'Yaron Biton'. The right screenshot shows the same dialog with 'Yaron Biton' selected and the 'Add vyaron to Touch-the-nums' button highlighted in green. Red arrows and circles indicate the steps: clicking the 'Who has access' section, searching for the collaborator, selecting the collaborator, and clicking the 'Add Collaborator' button.

Collaborating with GIT

5) Your collaborator will receive invite to his email account

The screenshot shows the GitHub interface for managing repository access. On the left is a sidebar with navigation links: Options, Manage access (highlighted), Branches, Webhooks, Notifications, Integrations & services, Deploy keys, Secrets, and Actions. Below these are links for Moderation and Interaction limits. The main content area is titled 'Who has access' and includes a 'Beta' badge and a link to 'Learn more or give us feedback'. It contains two panels: 'PUBLIC REPOSITORY' (with an eye icon) stating 'This repository is public and visible to anyone.' with a 'Manage' link, and 'DIRECT ACCESS' (with a people icon) stating '1 has access to this repository. 1 invitation.' Below this is a 'Manage access' section with an 'Invite a collaborator' button. It features a 'Select all' checkbox, a 'Type' dropdown, a search bar 'Find a collaborator...', and a list of collaborators. One collaborator, 'Yaron Biton', is listed with a status of 'Pending Invite' and a trash icon. At the bottom are 'Previous' and 'Next' navigation buttons.

Options

Manage access

Branches

Webhooks

Notifications

Integrations & services

Deploy keys

Secrets

Actions

Moderation

Interaction limits

Who has access

Beta [Learn more or give us feedback](#)

PUBLIC REPOSITORY

This repository is public and visible to anyone.

[Manage](#)

DIRECT ACCESS

1 has access to this repository. 1 [invitation](#).

Manage access

[Invite a collaborator](#)

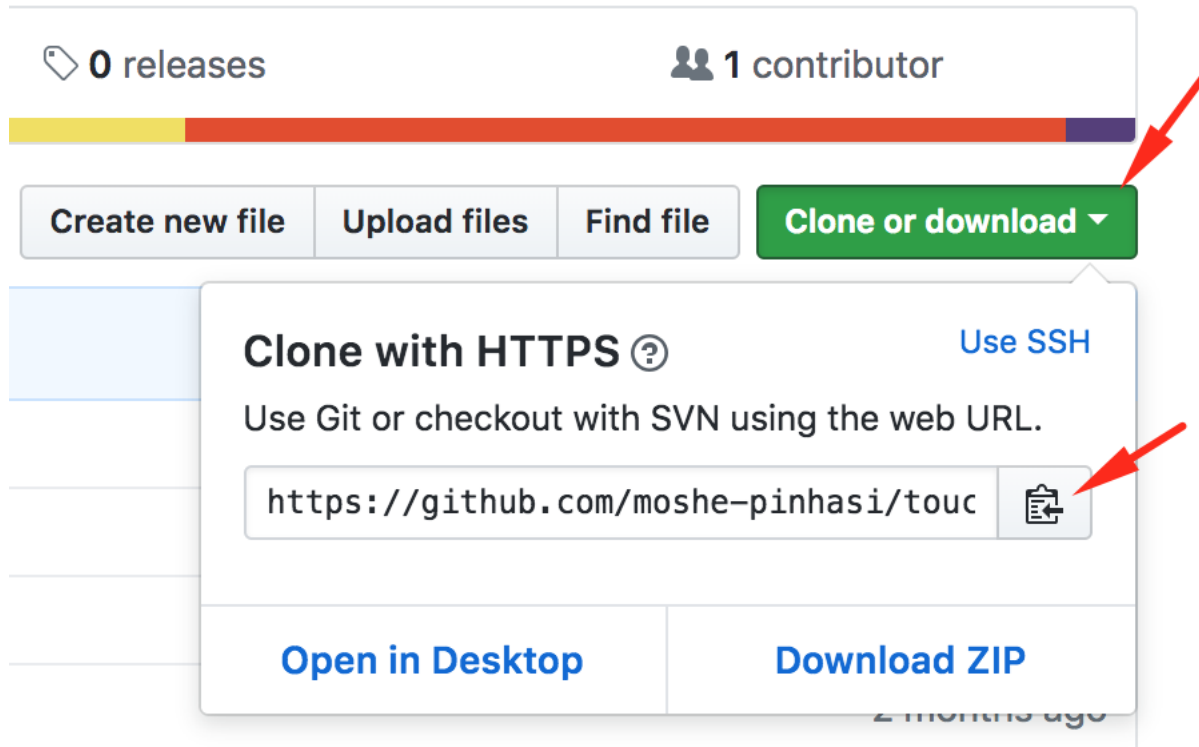
☐ Select all Type ▾

<input type="checkbox"/>	Yaron Biton Awaiting vyaron's response	Pending Invite	
--------------------------	--	----------------	--

[Previous](#) [Next](#)

Collaborating with GIT

6) Clone the repository to your local machine



Review VSCode GIT Support

Git and VSCode



Ya Welly,

You know github.

Enjoy it

