

## Git

מערכת Version Control עבור קוד.

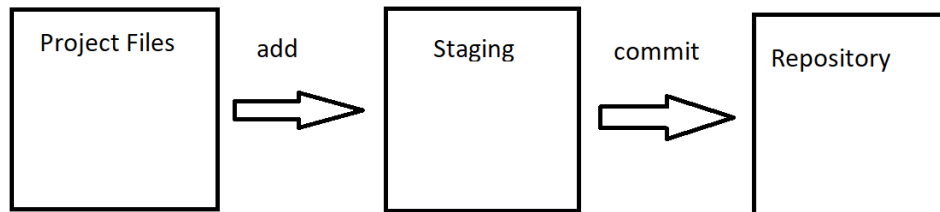
Version Control – ניהול גרסאות הקוד שלנו. גרסאות של קוד הפיתוח.

מערכות ישנות: SVN, TFS ועוד...

Git מכילה Repository – מאגר אחד ששומר את הקוד שלנו.

ניתן לפתח ולשמור את השינויים במאגר הזה.

המאגר הינו מאגר מקומי במחשב או ברשת.



## GitHub

זה אתר הבנוי לשמירת מאגרי Git ב-Cloud

ה-Could מספק שירותים נוספים, לדוגמה שעוד מתכנתים יוכלו לראות את הפרויקטים שלכם, לדוגמה, שעוד מתכנתים יוכלו "לתרום" לקוד שלכם. ועוד.

## GitHub Pages

זה אפשרות ב-GitHub להעלות לאור אתר Frontend בלבד (אין שם Backend).

## פקודות Git נפוצות:

- Install Git Bash: <https://git-scm.com/downloads>

- Config Commands:

git --version	Display git version
git config --global user.name "<username>"	Set username once
git config --global user.email "<email>"	Set email once
git config --global --unset user.name	Unsets username once
git config --global --unset user.email	Unsets email once
git config <command>	Show Specific config command (e.g. user.email)
git config --list	List all config commands
git help <command>	Same
clear	Clear window
exit	Close window

- Directories:

<code>pwd</code>	Show current directory
<code>cd ..</code>	Go one directory up
<code>cd /c/&lt;dir1&gt;/&lt;dir2&gt;</code>	Change directory (must be / and not \)
<code>ls</code>	List files and folders in the current directory
<code>ls -la</code>	List also hidden files
<code>git init</code> containing an existing project)	Initialize a new local Repository (in empty directory or a directory
- Staging & committing:	
<code>git add &lt;file&gt;</code>	Add a file to the stage
<code>git add &lt;dir&gt;</code>	Add a directory to the stage
<code>git add .</code>	Add all files and directories to the stage
<code>git commit -m "&lt;msg&gt;"</code>	Commit the stage to the repository (-m = message)
<code>git log</code>	List commit history
<code>git log -&lt;n&gt;</code>	List last n commits
<code>git log --oneline</code>	List commit history in one line
<code>git log --oneline -&lt;n&gt;</code>	List last n commit history in one line
<code>git log --pretty=format:"%an, %h, %s"</code> commit id, %s = commit message)	List commit history formatted (%an = username, %h =
<code>git status</code> repository	List file changes between local files and stage or stage and
<code>git diff</code>	List code changes between local files and stage
<code>git diff --staged</code>	List code changes between stage and commit
<code>rm &lt;file&gt;</code>	Remove a file from disk but not from stage or commit
<code>git rm &lt;file&gt;</code> commit	Remove a file from disk and from stage, but not from
<code>mv &lt;file&gt;</code>	Move (or rename) a file in disk but not in stage or commit
<code>git mv &lt;file&gt;</code> commit	Move (or rename) a file in disk and in stage, but not in
<code>git clean -f</code>	Delete all untracked files from disk (f = force)
<code>git reset head</code>	Unstage all files
<code>git reset head &lt;file&gt;</code>	Unstage a file
<code>git reset --hard</code>	Revert local files to last commit
<code>git tag &lt;tag-name&gt;</code>	Tag the last commit
<code>git tag -a &lt;tag-name&gt; -m "&lt;msg&gt;"</code> a = annotated)	Tag the last commit with date record, message and more (-
<code>git tag</code>	List all tags
<code>git checkout &lt;tag-name&gt;</code>	Checkout to that tag
<code>git checkout &lt;commit-id&gt;</code>	Checkout to that commit
<code>git checkout master</code>	Checkout to master

git checkout -- <file>	Restore a file from last commit into the disk
git checkout <commit-id> -- <file>	Add a file from that commit to current commit
git branch	List all branches
git branch <branch-name>	Create a new branch (but not checkout to it)
git checkout <branch-name>	Checkout to that branch
git checkout -b <branch-name>	Create a new branch + Checkout to it
git merge-base <branch-name1> <branch-name2>	Shows commit id of the most recent shared commit between those branches

- GitHub:

git remote add <push-name> <rep-address>	Connect local repository to GitHub
git push <push-name> <branch-name>	Push local branch-name to GitHub
git push	Continue pushing to GitHub repository
git pull	Pull project to local repository
git clone <rep-address>	Clone GitHub repository to some current directory