Git and Github, as collaboration tools

Leonor Loureiro (@leonor-loureiro), GitHub

Leonor Loureiro

Software Engineer, Git Systems @ GitHub

@leonor-loureiro

Agenda

- What is a Version Control System?
- What is Git?
- Git vs GitHub, are they the same?
- How to configure Git for the first time
- Git Internals (git objects, references, branches, forks)
- Git Commands (pull, add, commit, push, stash, reset, restore)
- Merge Strategies (merge vs rebase)
- Git Branching Workflows
- Hands-on Practice
- GitHub Tools: Codespaces & Copilot

Version Control System (VCS)

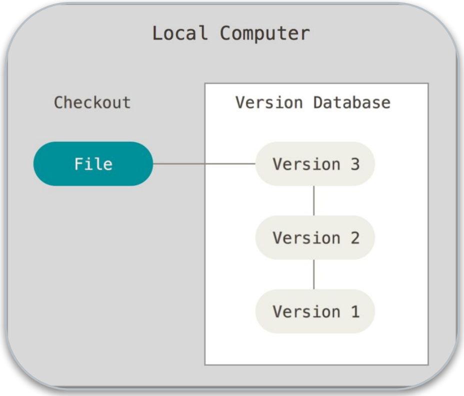
Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.

Version Control System (VCS)

Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.

Local VCS

RCS



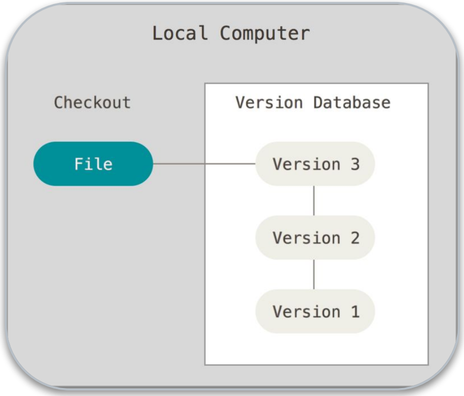
src: https://git-scm.com/book

Version Control System (VCS)

Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.

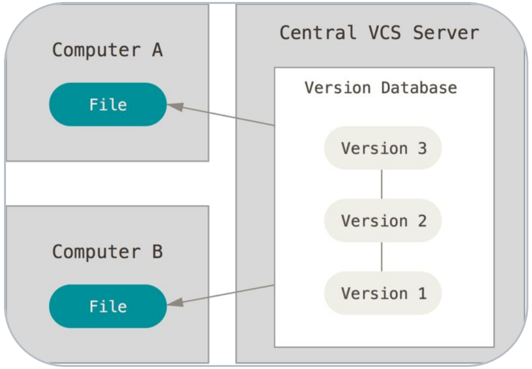
Local VCS

RCS



Centralized VCS

CVS, SVN, Perforce



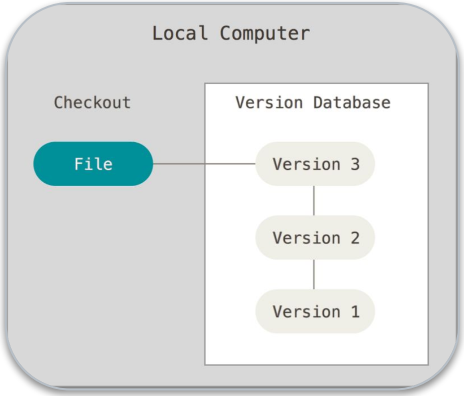
src: https://git-scm.com/book

Version Control System (VCS)

Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.

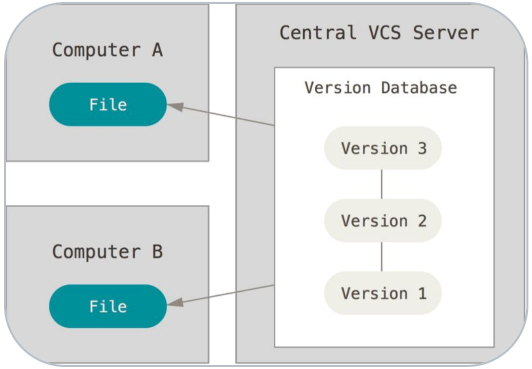
Local VCS

RCS



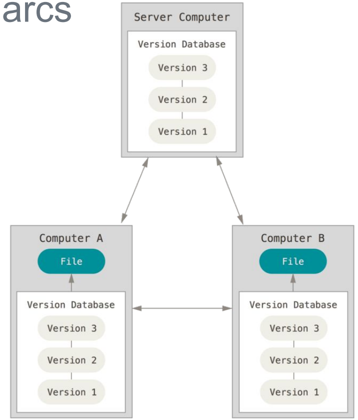
Centralized VCS

CVS, SVN, Perforce



Distributed VCS

Git, Mercurial, Bazaar or Darcs



src: https://git-scm.com/book

Git

Open-source distributed version control system. Currently, most widely used VCS in the world.

Git

Open-source distributed version control system. Currently, most widely used VCS in the world.

1 Small & Fast

Git

Open-source distributed version control system. Currently, most widely used VCS in the world.

1. Small & Fast
2. Able to handle large projects efficiently

Git

Open-source distributed version control system. Currently, most widely used VCS in the world.

1. Small & Fast
2. Able to handle large projects efficiently
3. Strong support for non-linear development

Git

Open-source distributed version control system. Currently, most widely used VCS in the world.

1. Small & Fast
2. Able to handle large projects efficiently
3. Strong support for non-linear development
4. Fully distributed

Git

Open-source distributed version control system. Currently, most widely used VCS in the world.

1. Small & Fast
2. Able to handle large projects efficiently
3. Strong support for non-linear development
4. Fully distributed
5. Provide data assurance

Git

A version control system used for tracking file changes.

GitHub

A web-based platform for developers to store code in the cloud.

Installing and configuring Git

Install Git

https://github.com/git-guides/install-git

Configuring Git

for the 1st Time

Identity

```
$ git config –global
user.name <full name>

$ git config –global
user.mail <email>
```

Set your username and email address

Editor

```
$ git config –global “code
–wait”

$ git config –global
“core.editor "nano"
```

Set your preferred code editor

Check settings

```
$ git config –list
user.name=Leonor Loureiro
user.email=leonor-loureiro@git
hub.com
core.editor=code
```

Check your configuration settings

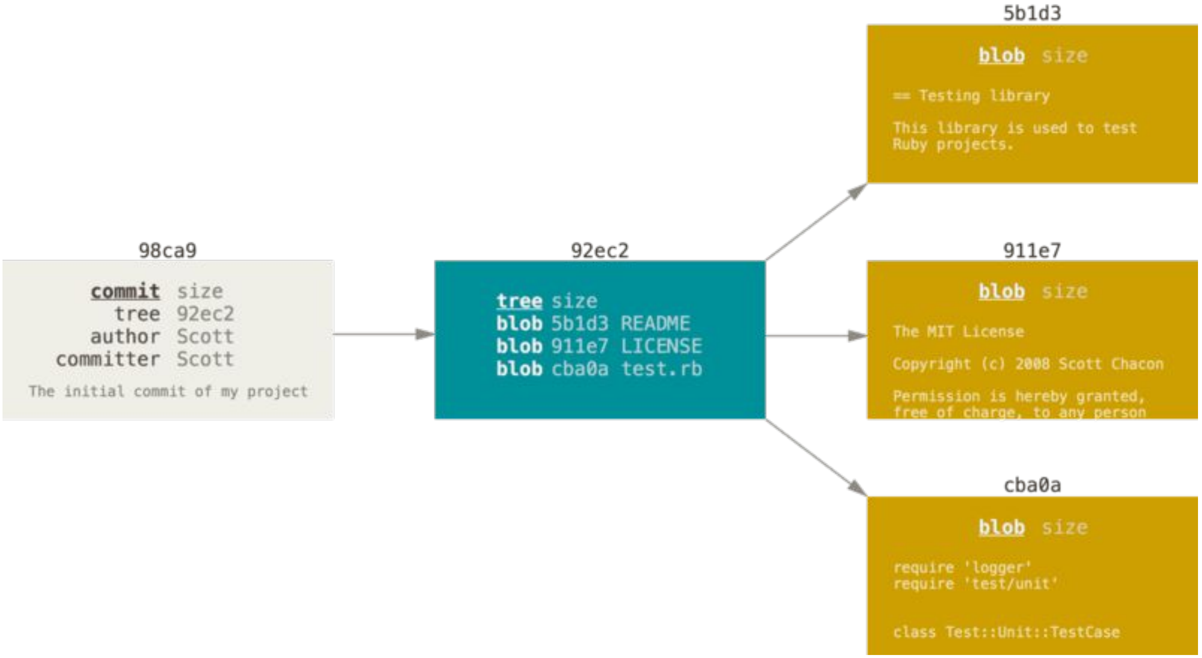
Git Fundamentals

Git Object Store

Git is key-value datastore (maps keys to values), that is persistent (i.e. stored in disk)

```
$ git init lip-workshop

$ ls .git/objects
info/  pack/

$ echo "Hello LIP" | git hash-object -w --stdin
8d05afb7cca6c65de29de3dd3b80fea3685d445a

$ ls .git/objects
8d  info  pack

$ git cat-file -p 8d05afb7cca6c65de29de3dd3b80fea3685d445a
Hello LIP

$ git cat-file -t 8d05afb7cca6c65de29de3dd3b80fea3685d445a
blob
```

Git Object Model



98ca9

```
     commit  size
       tree  92ec2
     author  Scott
  committer  Scott

The initial commit of my project
```

92ec2

```
tree size
blob 5b1d3 README
blob 911e7 LICENSE
blob cba0a test.rb
```

5b1d3

```
blob  size

== Testing library

This library is used to test
Ruby projects.
```

911e7

```
blob  size

The MIT License

Copyright (c) 2008 Scott Chacon

Permission is hereby granted,
free of charge, to any person
```

cba0a

```
blob  size

require 'logger'
require 'test/unit'


class Test::Unit::TestCase
```

- **Blob**: represent file contents
- **Tree**: represent directory listings, and provide the mapping between file names and their contents (i.e blobs).
- **Commit**: are snapshots in time

Let’s create our first commit

```
$ echo "LIP Workshop" > README.md
$ mkdir src && echo "echo 'Hello World'" > src/hello.sh
$ tree .
.
├── README.md
└── src
    └── hello.sh
$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
     README.md
     src/
```

Let’s create our first commit

```
$ git add .
< changes are added to the staging area >

$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
     new file:   README.md
     new file:   src/hello.sh
$ git commit -m “First commit!”
```

Let’s create our first commit

```
$ git log
commit e679a195d655c640b8ed4fbc146d673ff7223e6c (HEAD -> main)
Author: Leonor Loureiro <leonor-loureiro@github.com>
Date:   Sun Jul 2 16:04:41 2023 +0000

    First commit!
$ find .git/objects/ -type f
.git/objects/1f/9112685f31563969cec721058678199a1afb1f
.git/objects/0f/caeb1f6b4bdd9c5d206c0da09e5616fb39b26d
.git/objects/e6/79a195d655c640b8ed4fbc146d673ff7223e6c
.git/objects/8d/05afb7cca6c65de29de3dd3b80fea3685d445a
.git/objects/15/50528077df971a998c5a5d106c03b630255e91
.git/objects/cc/609ca9fd3039068e12621a6623d0964a02cf20
```

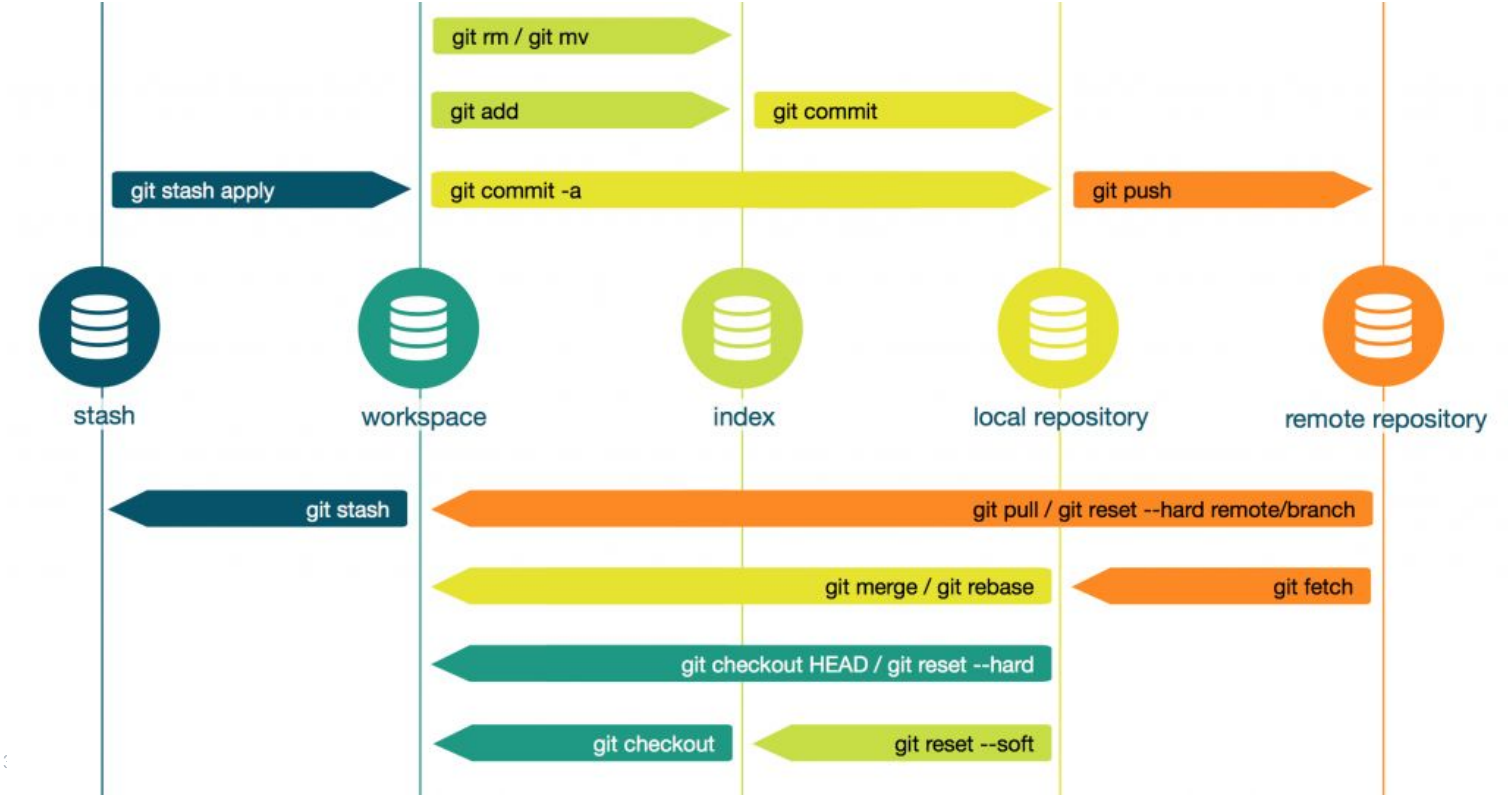
Let’s analyze the objects in the repo

```
$ git cat-file e679a19 -t
commit

$ git cat-file e679a19 -p
tree 0fcaeb1f6b4bdd9c5d206c0da09e5616fb39b26d
author Leonor Loureiro <leonor-loureiro@github.com> 1688313881 +0000
committer Leonor Loureiro <leonor-loureiro@github.com> 1688313881 +0000

First commit!
```

Let’s analyze the objects in the repo

```
$ git cat-file 0fcaeb1f6b4bdd9c5d206c0da09e5616fb39b26d -t
tree

$ git cat-file 0fcaeb1f6b4bdd9c5d206c0da09e5616fb39b26d -p
100644 blob 1550528077df971a998c5a5d106c03b630255e91   README.md
040000 tree cc609ca9fd3039068e12621a6623d0964a02cf20   src

$ git cat-file cc609ca9fd3039068e12621a6623d0964a02cf20 -p
100644 blob 1f9112685f31563969cec721058678199a1afb1f   hello.sh

$ git cat-file 1f9112685f31563969cec721058678199a1afb1f -p
echo 'Hello World'
```

Basic Git Commands



git rm / git mv

git add

git commit

git stash apply

git commit -a

git push

stash

workspace

index

local repository

remote repository

git stash

git pull / git reset --hard remote/branch

git merge / git rebase

git fetch

git checkout HEAD / git reset --hard

git checkout

git reset --soft

Git as a Collaboration Tool

Git References

Refs are simple ways to refer to commits, using a simple, easy to remember name.

- .git/refs/heads directory defines all of the local branches in your repository.
- .git/refs/tags directory defines all the tags in your repository.

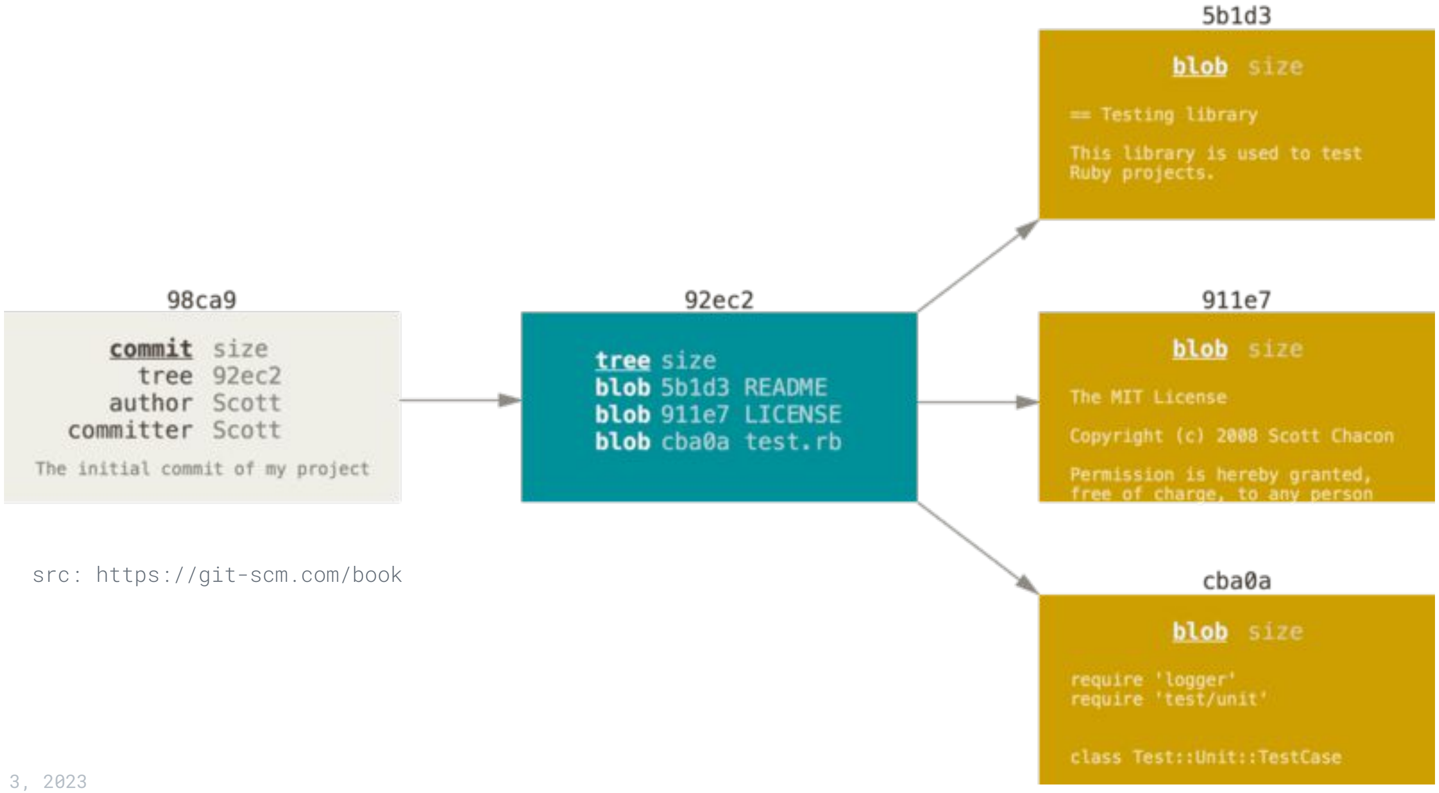
```
$ tree .git/refs/
.git/refs/
├── heads
│   └── main
└── tags
$ git show main
commit e679a195d655c640b8ed4fbc146d673ff7223e6c (HEAD -> main)
Author: Leonor Loureiro <leonor-loureiro@github.com>
Date:   Sun Jul 2 16:04:41 2023 +0000

    First commit!

diff --git a/README.md b/README.md <...>
```

Let’s create a dev branch

HEAD is a special reference that points to the current branch always.

```
$ cat .git/HEAD
ref: refs/heads/main

$ git branch -c dev
$ tree .git/refs
.git/refs/
├── heads
│   ├── dev
│   └── main
└── tags

$ git checkout dev
Switched to branch 'dev'

$ cat .git/HEAD
ref: refs/heads/dev
```

Git Branches



98ca9

```
     commit  size
       tree  92ec2
     author  Scott
  committer  Scott

The initial commit of my project
```

92ec2

```
tree size
blob 5b1d3 README
blob 911e7 LICENSE
blob cba0a test.rb
```

5b1d3

```
blob  size

== Testing library

This library is used to test
Ruby projects.
```

911e7

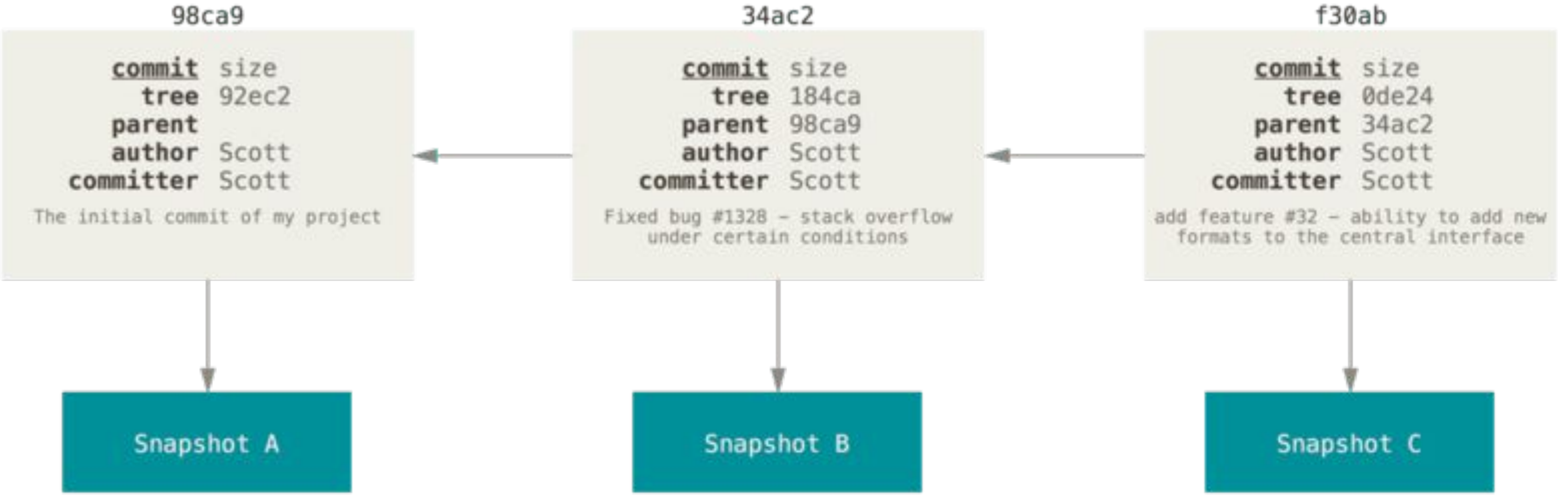
```
blob  size

The MIT License

Copyright (c) 2008 Scott Chacon

Permission is hereby granted,
free of charge, to any person
```

cba0a

```
blob  size

require 'logger'
require 'test/unit'


class Test::Unit::TestCase
```

src: https://git-scm.com/book

Git Branches



98ca9

commit size
tree 92ec2
parent
author Scott
committer Scott

The initial commit of my project

34ac2

commit size
tree 184ca
parent 98ca9
author Scott
committer Scott

Fixed bug #1328 - stack overflow under certain conditions

f30ab

commit size
tree 0de24
parent 34ac2
author Scott
committer Scott

add feature #32 - ability to add new formats to the central interface

Snapshot A

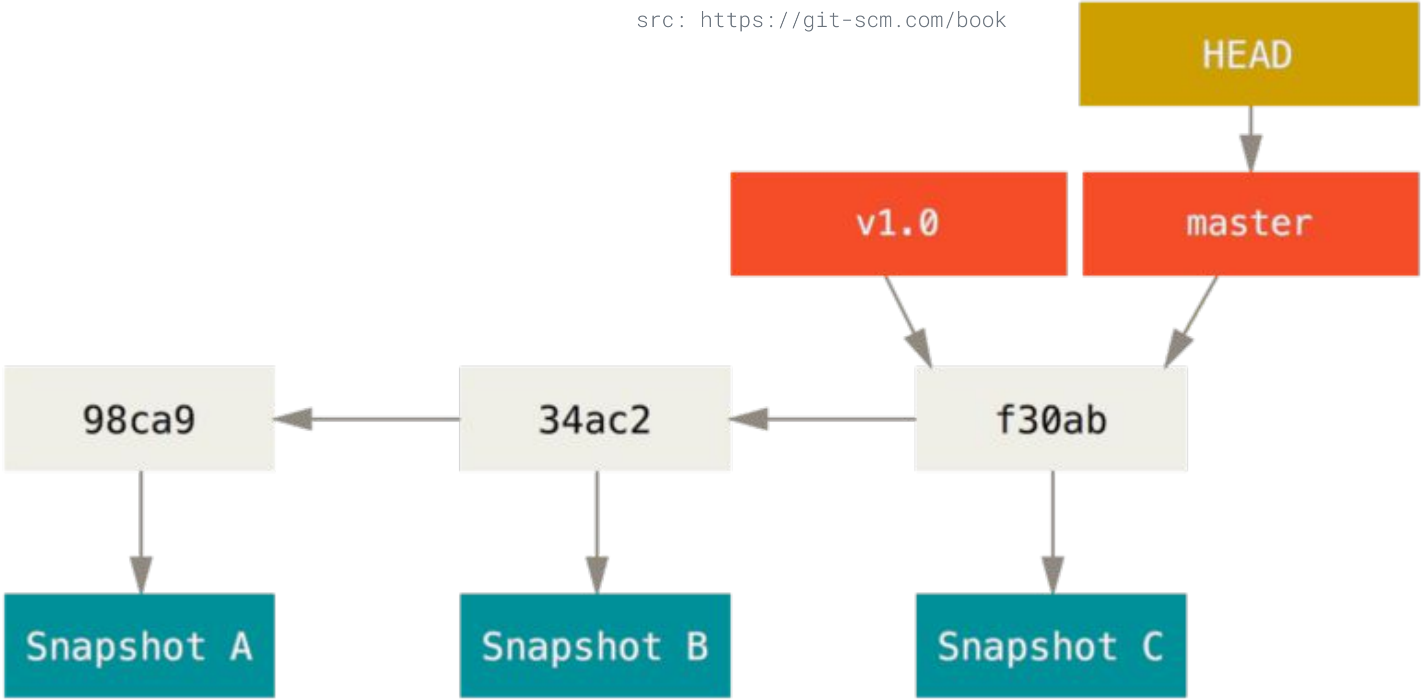
Snapshot B

Snapshot C

src: https://git-scm.com/book

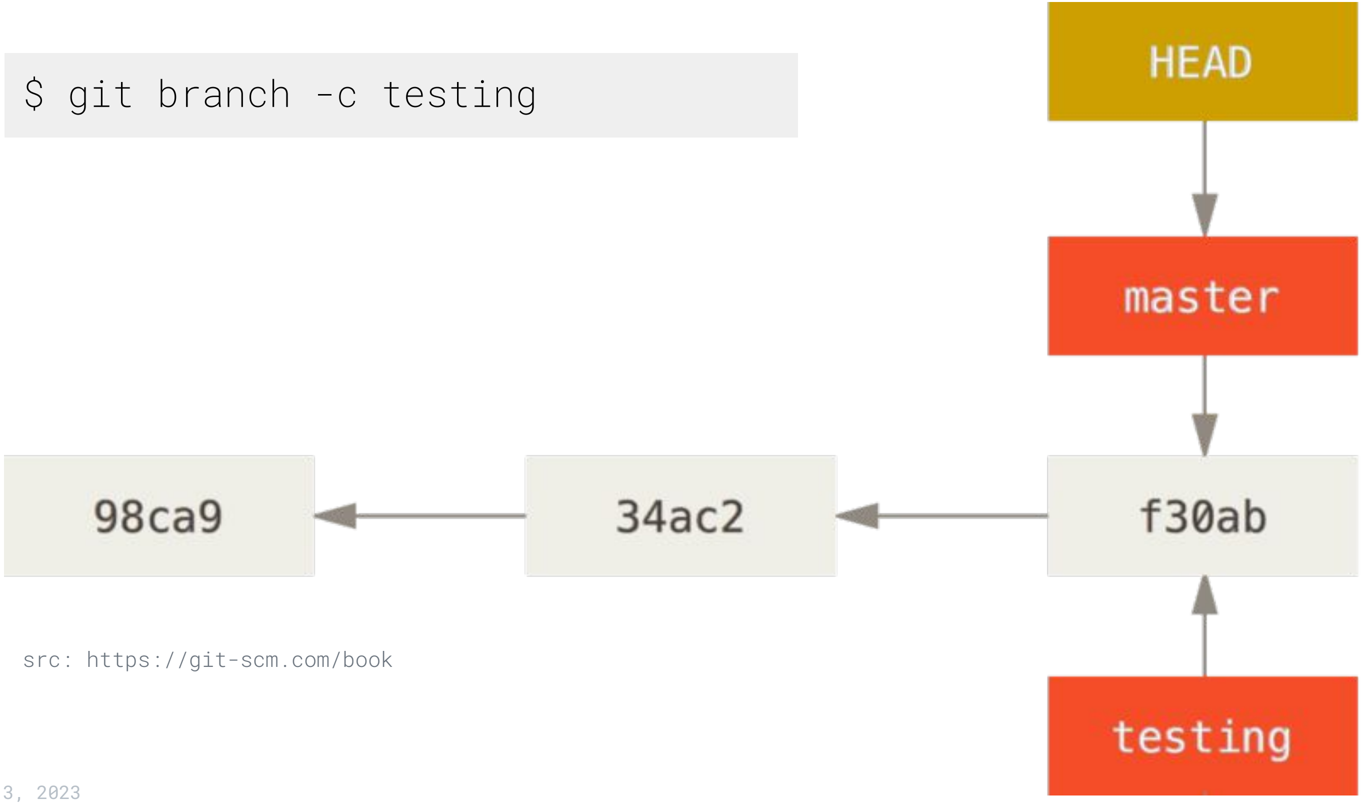
Git Branches

src: https://git-scm.com/book



Git Branches

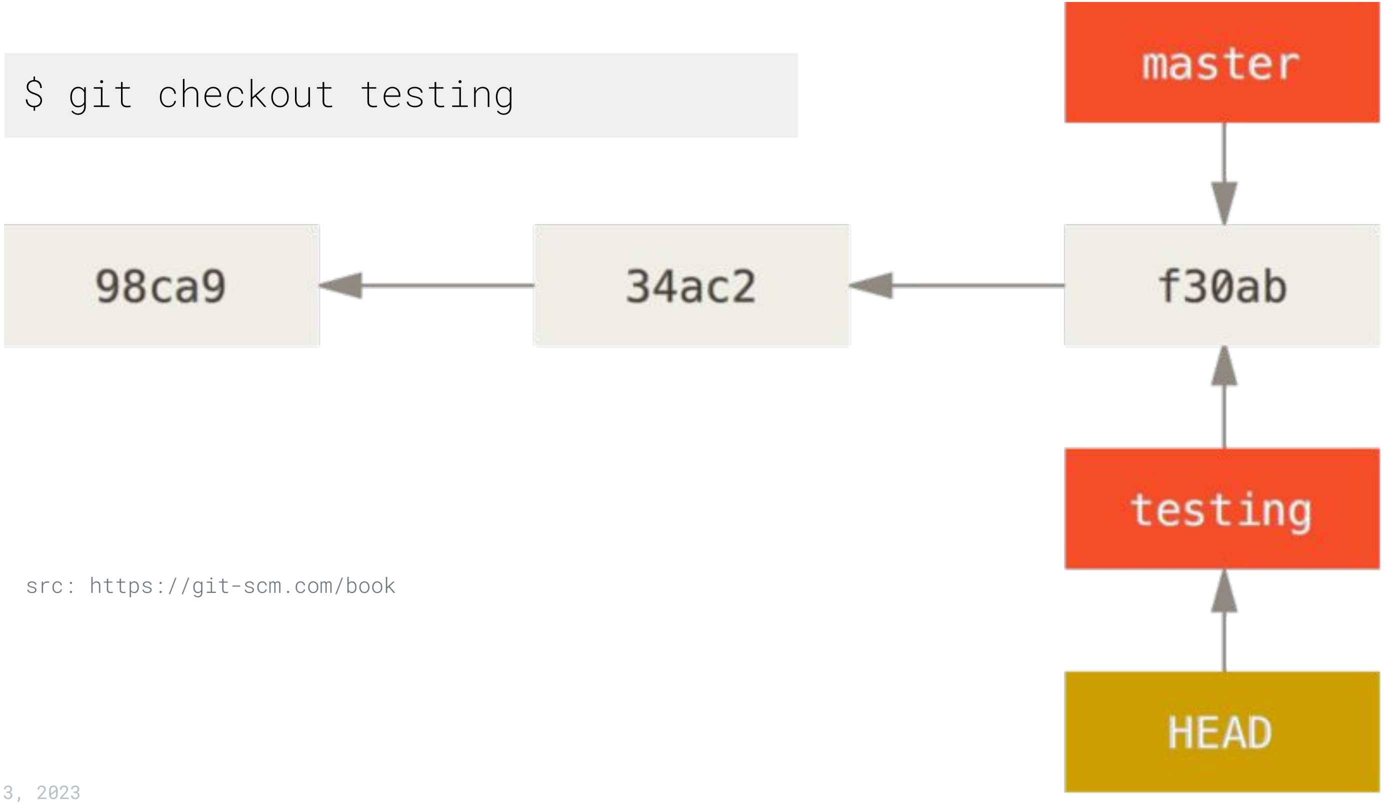
```
$ git branch -c testing
```



src: https://git-scm.com/book

Git Branches

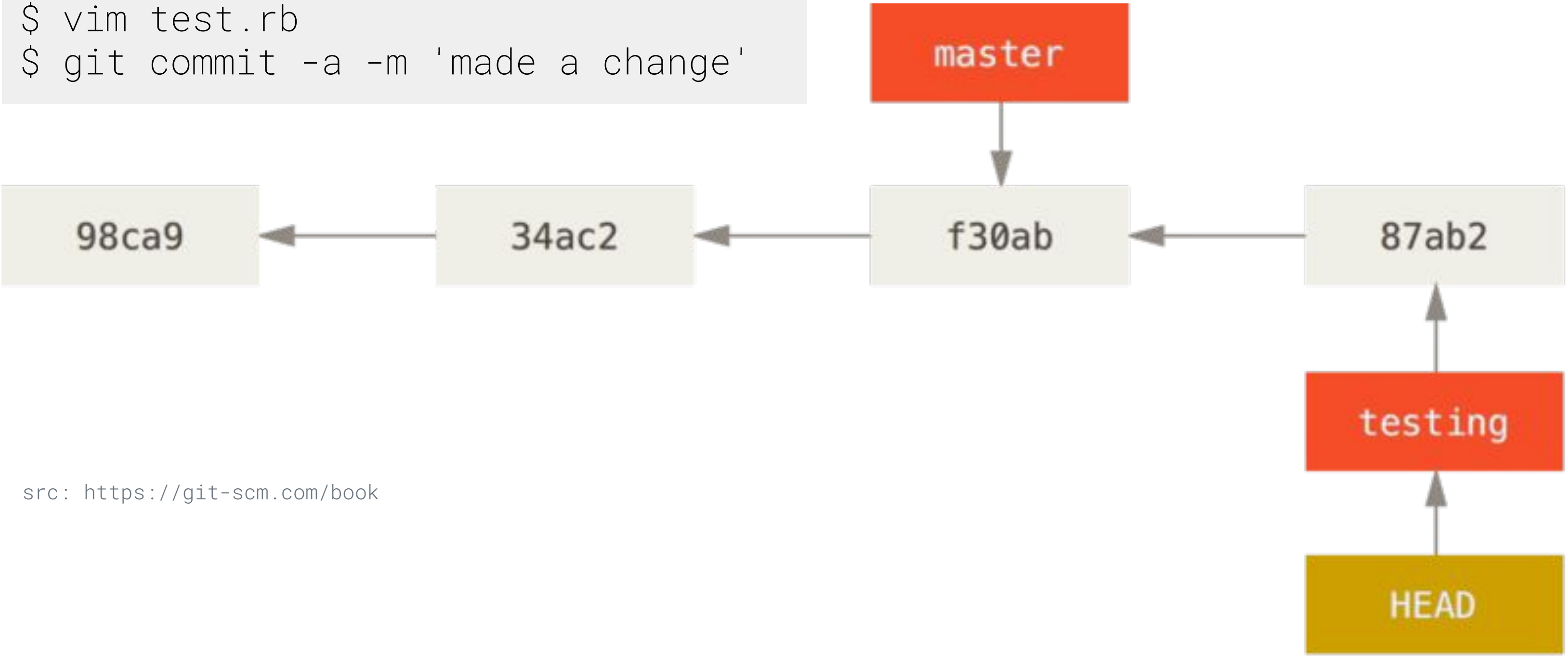
```
$ git checkout testing
```



src: https://git-scm.com/book

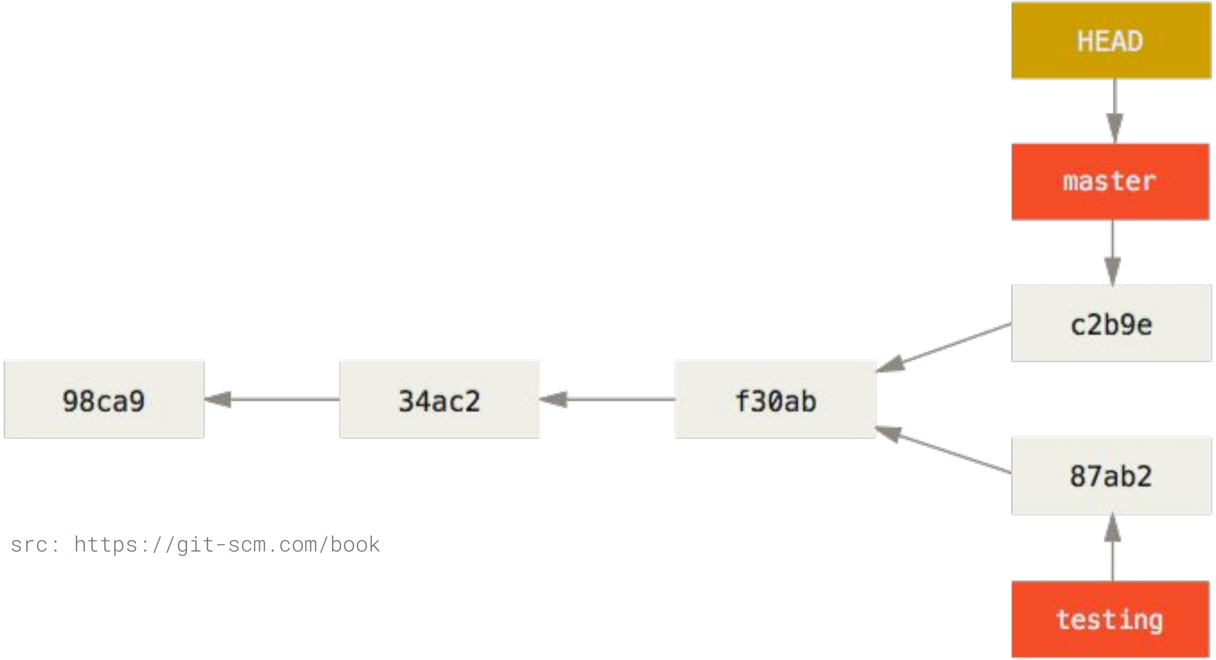
Git Branches

```
$ vim test.rb
$ git commit -a -m 'made a change'
```



src: https://git-scm.com/book

Git Branches

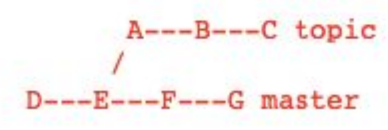


src: https://git-scm.com/book

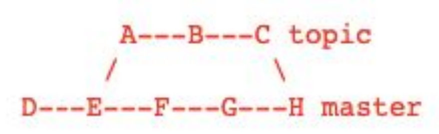
Git Merge

Join two or more development histories together

```
$ git checkout master
$ git merge topic
```



```
      A---B---C topic
     /
D---E---F---G master
```



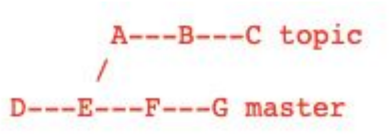
```
      A---B---C topic
     /         \
D---E---F---G---H master
```

src: https://git-scm.com/book

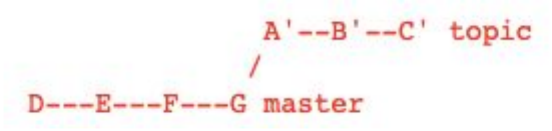
Git Rebase

Re-apply commits on top of another base tip

```
$ git checkout topic
$ git rebase master
```



```
      A---B---C topic
     /
D---E---F---G master
```

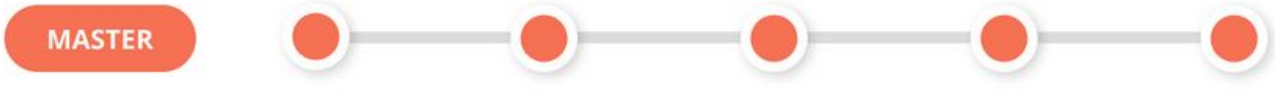


```
              A'--B'--C' topic
             /
D---E---F---G master
```

src: https://git-scm.com/book

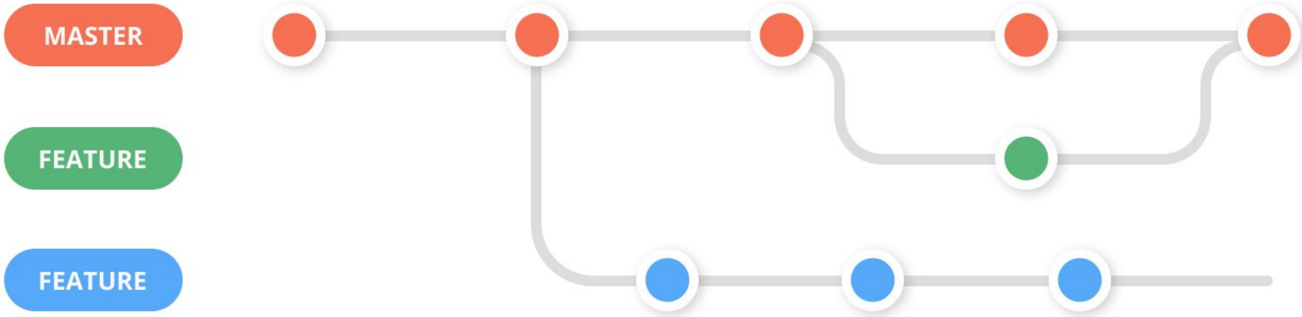
Git Workflows

Basic Git Workflow



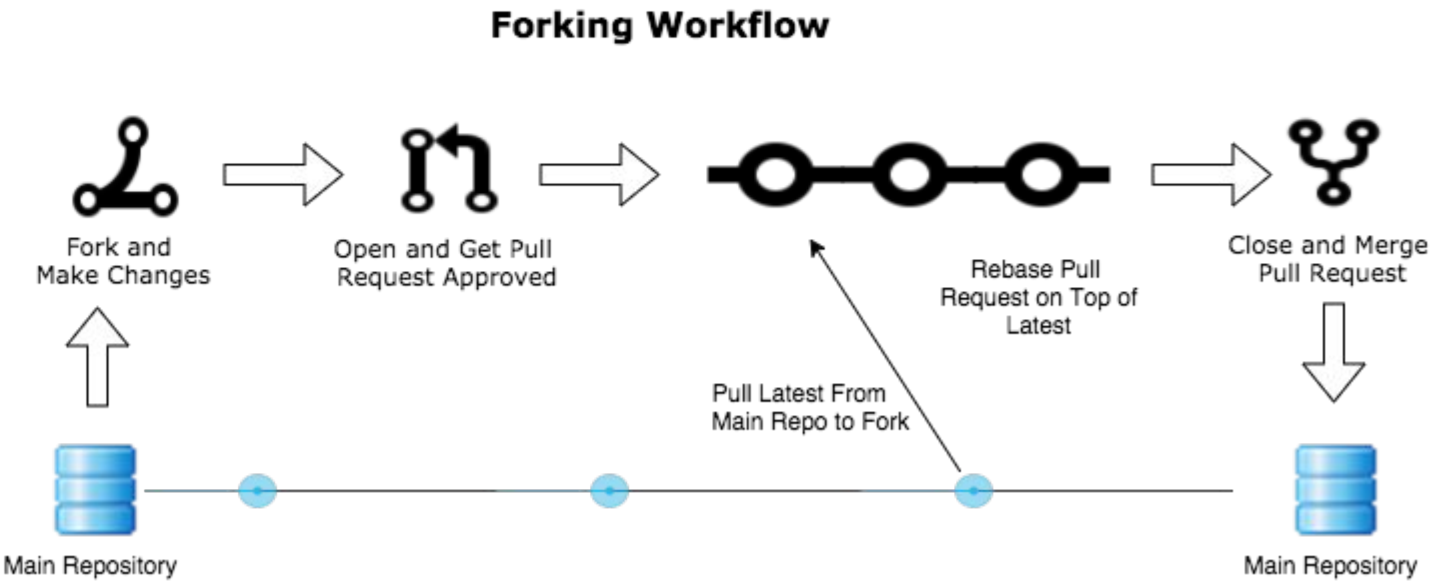
src:https://rovitpm.com/5-git-workflows-to-improve-development/

Git Feature Branch Workflow



src:https://rovitpm.com/5-git-workflows-to-improve-development/

Forking Workflow


Forking Workflow
Fork and Make Changes
Open and Get Pull Request Approved
Rebase Pull Request on Top of Latest
Close and Merge Pull Request
Pull Latest From Main Repo to Fork
Main Repository
Main Repository

https://docs.rhodecode.com/RhodeCode-Enterprise/collaboration/workflow-fork.html

Best Practices

Write short, detailed commit messages

Why is a change being made, how does it address the issue and what effects does it have.

Use branches

Avoid pushing directly to master/main. Every a separate branch for every logically different feature.

Update working branch frequently

Prevent bugs, rework and minimize conflicts.

Request feedback early and often

Ensure code and decision quality.

Hands-on Practice

- Fork a public repository
- Push a change to a new feature branch
- Open a Pull Request against the original repository
- Resolve a merge conflict
- Merge the Pull Request

GitHub Codespaces

Instant, cloud-based development environment that uses a container to provide you with common languages, tools, and utilities for development.

- Easy setup and configuration
- Faster onboarding
- Consistent & customizable development environments
- Enhanced security

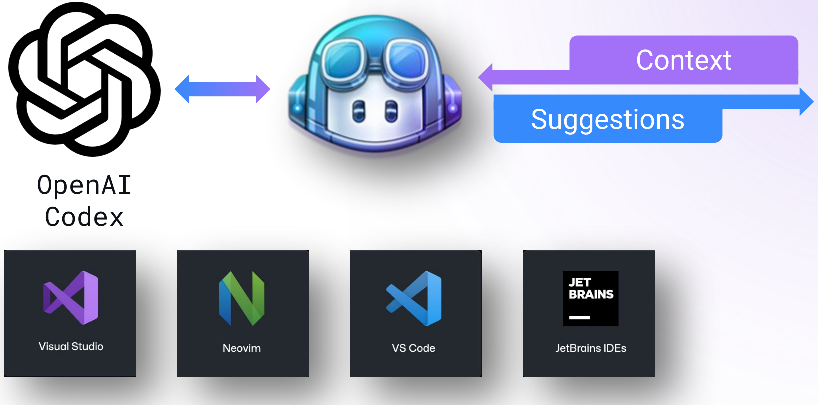
GitHub Copilot

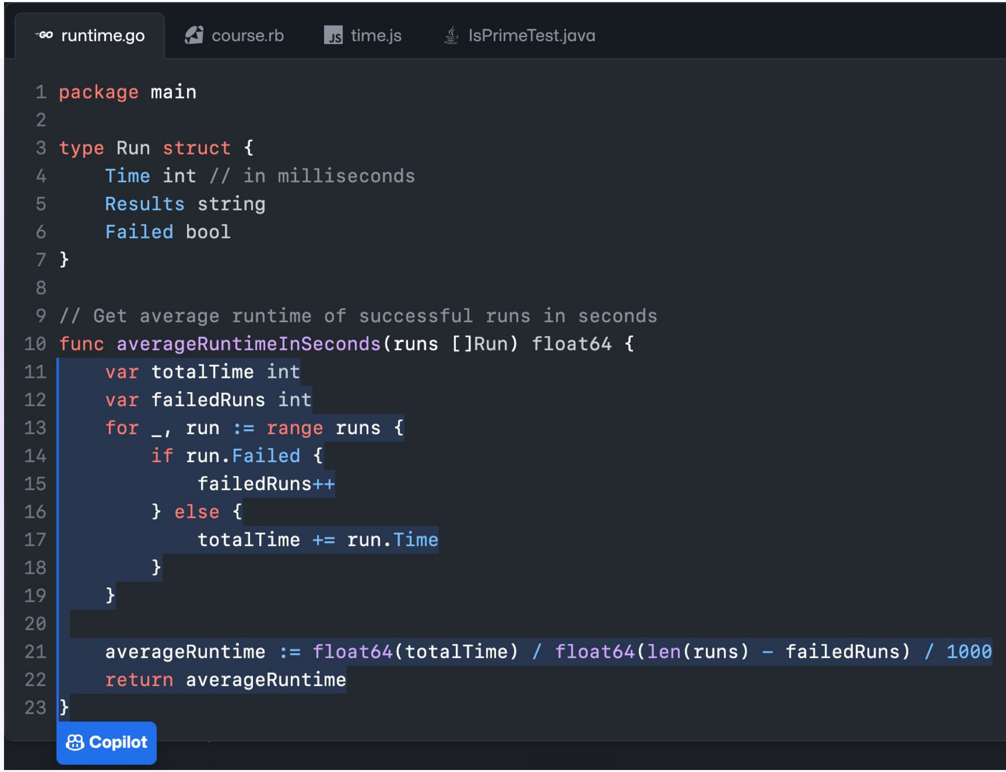
Offers code suggestions, contextualized to your project

Convert comments to code

Autofill for repetitive code

Show alternatives





```go
package main

type Run struct {
    Time int // in milliseconds
    Results string
    Failed bool
}

// Get average runtime of successful runs in seconds
func averageRuntimeInSeconds(runs []Run) float64 {
    var totalTime int
    var failedRuns int
    for _, run := range runs {
        if run.Failed {
            failedRuns++
        } else {
            totalTime += run.Time
        }
    }

    averageRuntime := float64(totalTime) / float64(len(runs) - failedRuns) / 1000
    return averageRuntime
}
```

Q&A

Resources & Links

- Pro Git Book - https://git-scm.com/book/en/v2
- Git Cheat Sheet - https://training.github.com/downloads/github-git-cheat-sheet/
- GitHub Skills -https://skills.github.com/
- GitHub Standard Fork & Pull Request Workflow - https://gist.github.com/Chaser324/ce0505fbed06b947d962
- Resolve Merge Conflicts Workshop - https://github.com/skills/resolve-merge-conflicts
- Copilot Codespaces Integration - https://github.com/skills/copilot-codespaces-vscode
- Getting started with Codespaces: https://docs.github.com/en/codespaces
- Copilot documentation: https://docs.github.com/en/copilot

Let’s make a change in branch dev

```
$ echo "Learning to use branches" >> README.md

$ git add -p

$ git commit -m “Update README.md”

$
```

Brand guidelines & accessibility

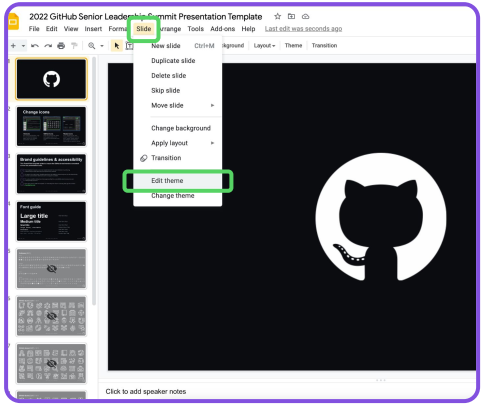
Our goal is to create a user friendly template so all Hubbers have the opportunity to create a presentation that looks as if a designer created it. Please be sure to follow guidelines and template structures to ensure brand and accessibility compliance.

This template is based on our core brand elements to assist Hubbers in producing communications that align with the GitHub brand system.

The colors and font sizing have been approved by the accessibility team to ensure we are actively being inclusive.

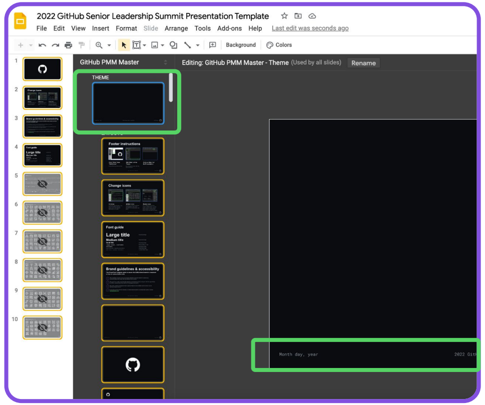
If you have issues with the template, please reach out to via our slack channel #visual-communications

Footer instructions


2022 GitHub Senior Leadership Summit Presentation Template
File Edit View Insert Format Slide Arrange Tools Add-ons Help Last edit was seconds ago
New slide Ctrl+M
Duplicate slide
Delete slide
Skip slide
Move slide
Change background
Apply layout
Transition
Edit theme
Change theme
Click to add speaker notes

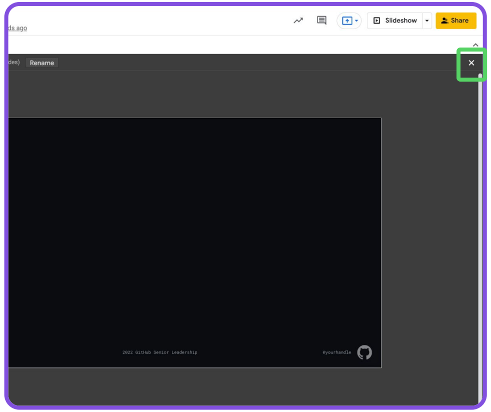
Click “Slide” then “Edit theme”

Found at the top of the tool bar


2022 GitHub Senior Leadership Summit Presentation Template
File Edit View Insert Format Slide Arrange Tools Add-ons Help Last edit was seconds ago
GitHub PMM Master
Editing: GitHub PMM Master - Theme (Used by all slides)
Rename
THEME
Month day, year
2022 Git

Edit the footer on Slide 0 of the theme

By changing the date, talk title, and GitHub handle of the speaker(s)



Click the X at the top right to close out and return to presentation

Font guide

X-large #	Arial Bold 60pt
Large title	Arial Bold 44pt
Medium title	Arial Bold 36pt
Small title	Arial Bold 28pt
X-small title	Arial Bold 20pt
Large dates, usernames	Roboto Mono 16pt
Information	Arial Regular 16pt
Small information, supporting descriptions	Arial Regular 14pt
X-small information, supporting descriptions	Arial Regular 12pt
Small dates, usernames, information	Roboto Mono 12pt

Color palette

You can also always find the color palette here! ➡️

Background

Background
#FFFFFF

Text

Text, Shapes
Gray-09 #0d1117

Text, Shapes
Gray-07 #6a737d

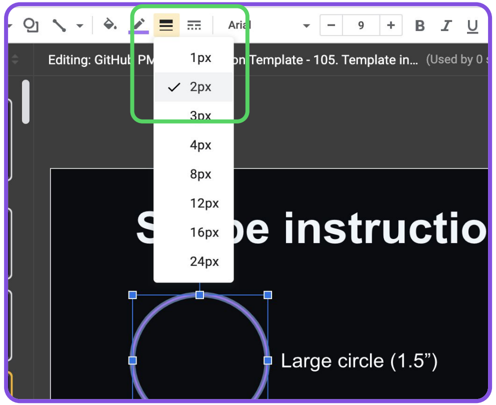
Shape Outline

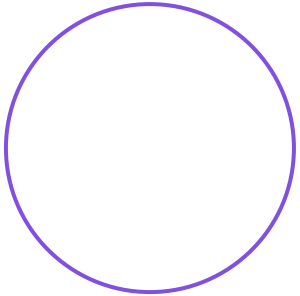
Shape outline
Gray-03 #6a737d

Shapes

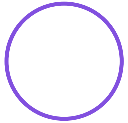
Shapes Purple-05 #8250DF	**Shapes** Purple-07 #512A97	**Shapes** Indigo-05 #545DF0	**Shapes** Indigo-07 #2C33A5
Shapes Blue-05 #096BDE	**Shapes** Blue-07 #033F8F	**Shapes** Teal-04 #339D9B	**Shapes** Teal-06 #136061
Shapes Green-04 #2EA043	**Shapes** Green-06 #116329	**Shapes** Pink-05 #BF3989	**Shapes** Pink-07 #772057

Shape instructions

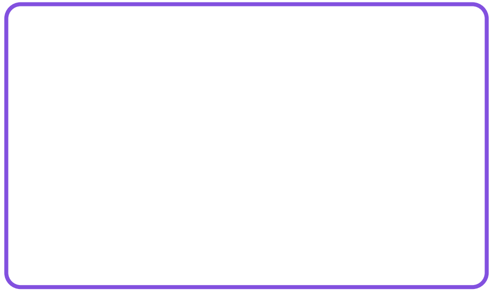




Large circle
(1.5”)



Small circle
(0.6”)

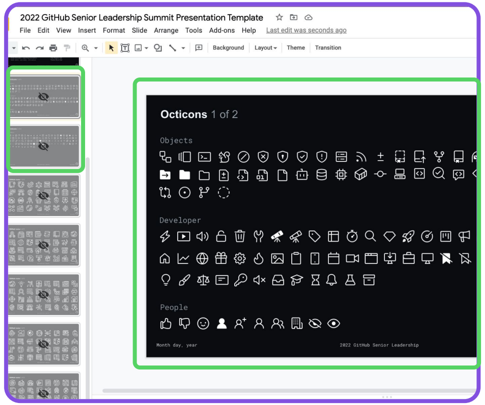


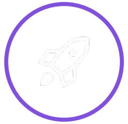
Rectangle with
rounded corners

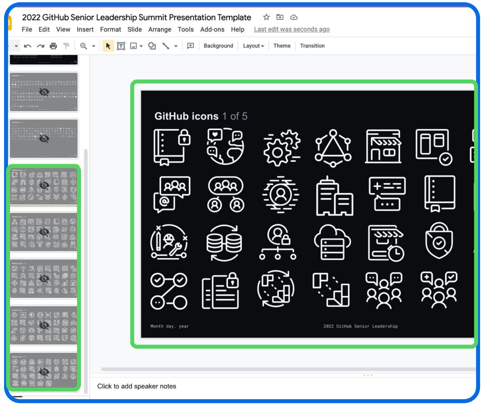
When creating rounded corner rectangles, lines, or circles, all line widths or outlines should be 2px

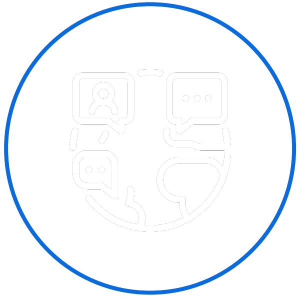
Either create your own shape and follow the shape instructions or feel free to copy and paste from here! Click “Slide” then “Edit theme” and then copy and paste the intended shape. Click the “x” at the top right to exit theme and return to presentation view.

Icon instructions


2022 GitHub Senior Leadership Summit Presentation Template
File Edit View Insert Format Slide Arrange Tools Add-ons Help Last edit was seconds ago
Background Layout Theme Transition
Octicons 1 of 2
Objects
Developer
People
Month day, year
2022 GitHub Senior Leadership




2022 GitHub Senior Leadership Summit Presentation Template
File Edit View Insert Format Slide Arrange Tools Add-ons Help Last edit was seconds ago
Background Layout Theme Transition
GitHub icons 1 of 5
Month day, year
2022 GitHub Senior Leadership
Click to add speaker notes



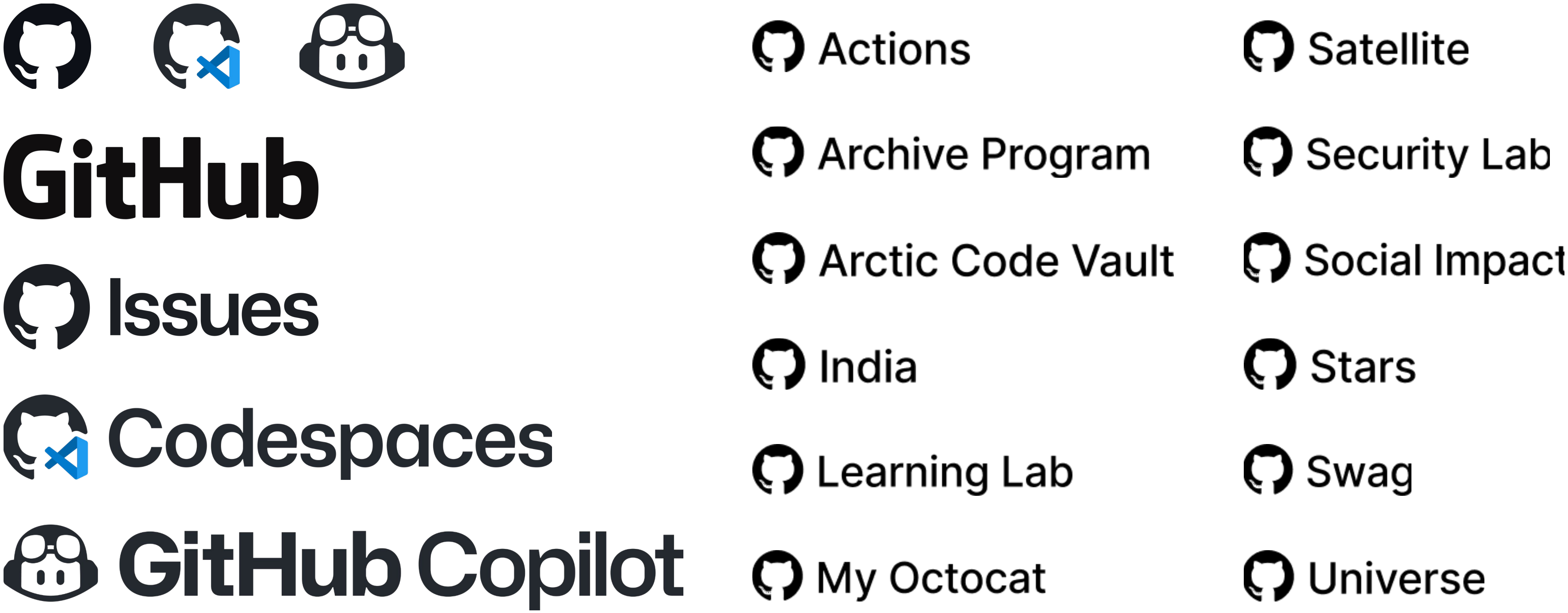
Small circle (0.6”)

When you see the small circle, find an Octicon from the Octicon pages and copy/paste into the center of the circle. No need to resize.

Large circle (1.5”)

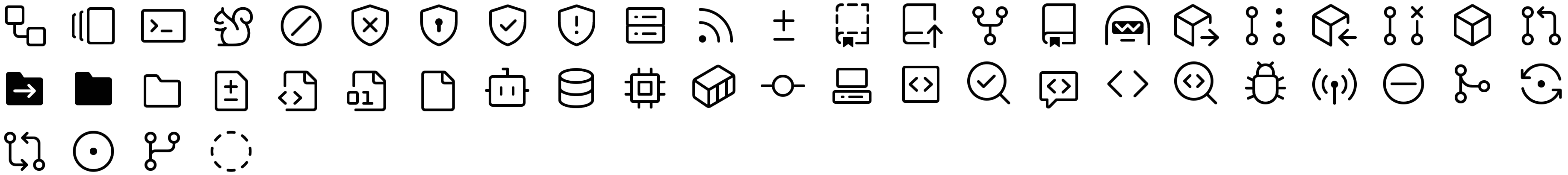
When you see the large circle, find a GitHub icon from the GitHub icon pages and copy/paste into the center of the circle. No need to resize.

GitHub logos

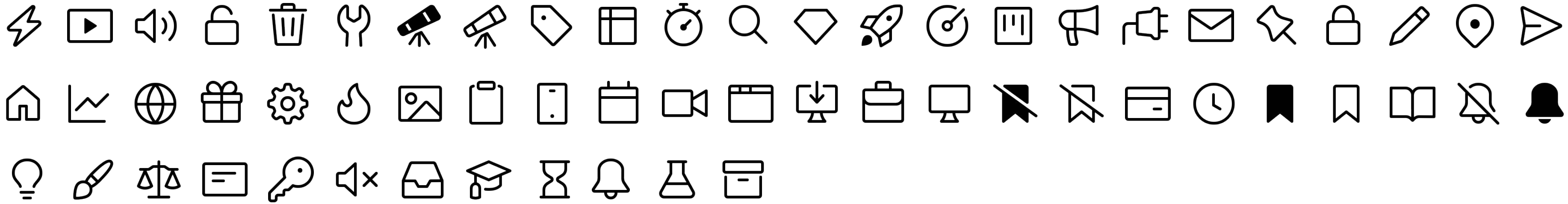


Octicons 1 of 2

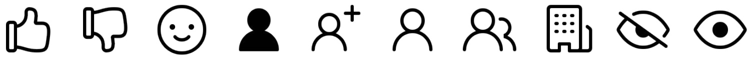
Objects



Developer

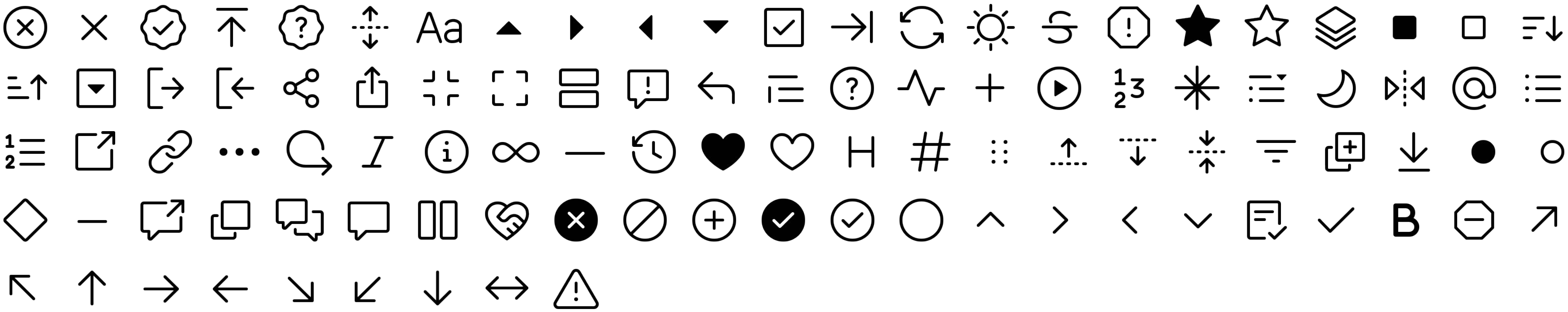


People

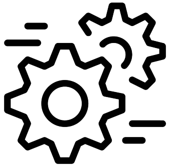


Octicons 2 of 2

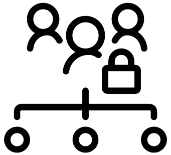
Symbols

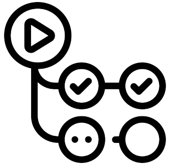


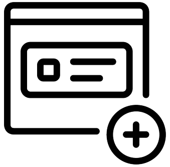
GitHub icons 1 of 5


API

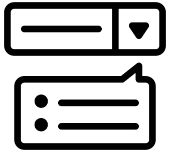
Accelerated SDLC

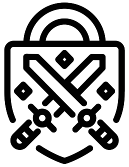

Access and permission

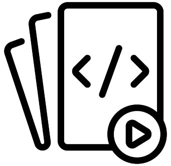

Actions

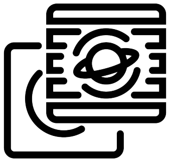

Add tasks to columns

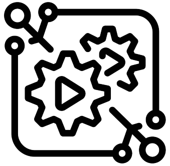
Admin mentoring

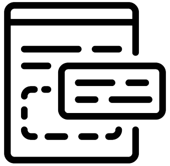

Administration


All for one

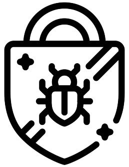

Any language

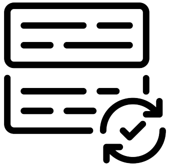

Augmented reality

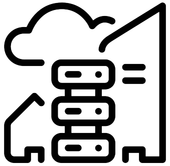

Automate your workflow

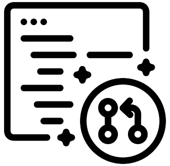

Autonomous coding

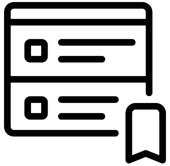
Blockchain


Bug Hunter


Build

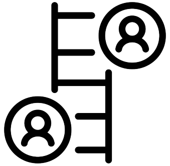

Business


CLI easy creation

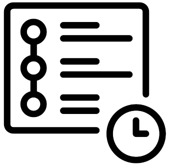

CLI seamless navigation

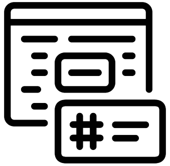
CLI status check

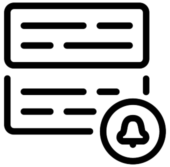
CVE record

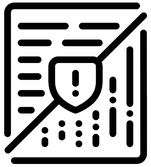

Career Ladder

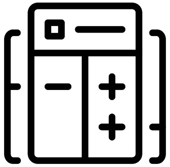
Celebration

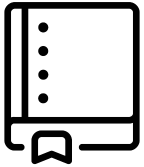

Changelog

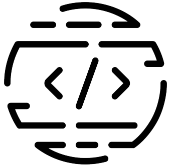

Code Nav

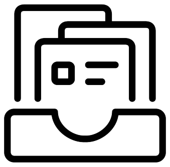

Code alert

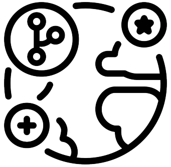

Code as data

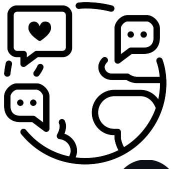

Code diff


Code hosting

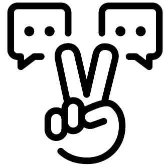

Code

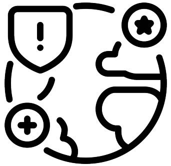

Collection

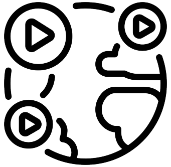

Community actions

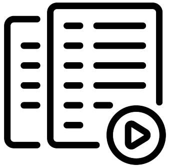

Community forum

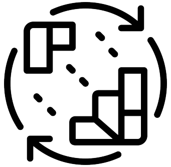
GitHub icons 2 of 5

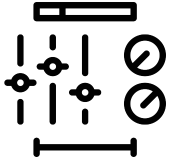

Community management

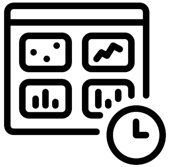

Community-led security approach

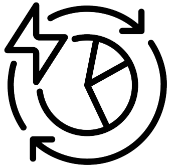

Community-powered

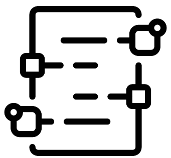

Configuration as code

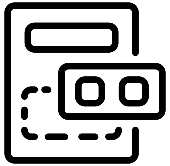

Continuous integration

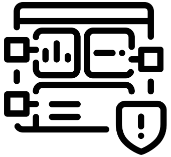

Customized training

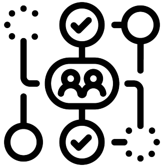

Data-driven recommendations

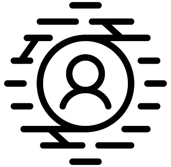

Data-driven

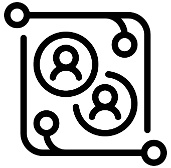

Dependencies

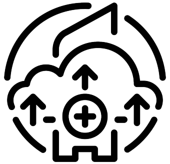

Dependencies 2

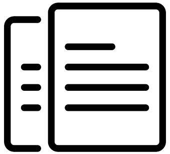

Dependency insight dashboard

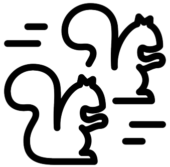

DevOps workflow

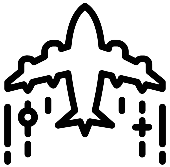

Developer identity

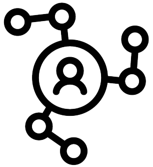

Developers

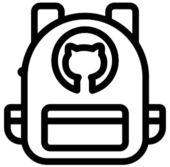

Disaster recovery

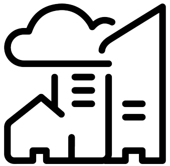

Documentation

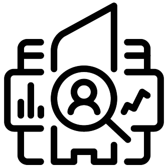

Double ship it

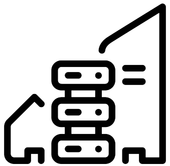

Dreamlifter

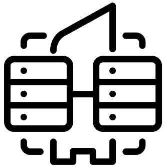

Ecosystem

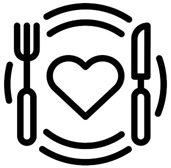

Education

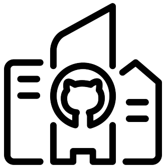

Enterprise Cloud

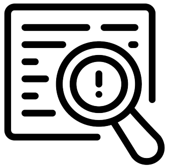

Enterprise Observability

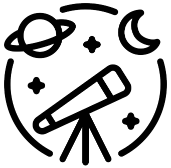

Enterprise Server

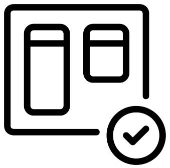

Enterprise availability

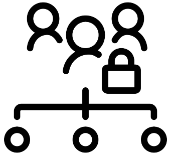

Enterprise dinner

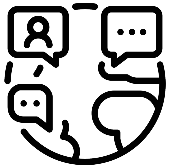

Enterprise

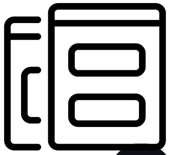

Existing vulnerability


Explore

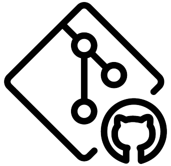

Finish your project

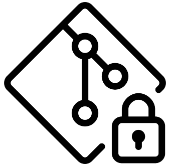

Flexible permission

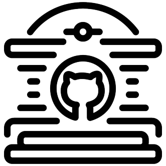

Forum

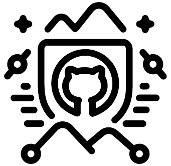

Get organized

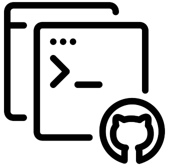
GitHub icons 3 of 5

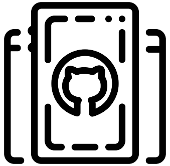

Git + GitHub


Git encryption

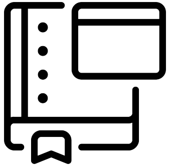

GitHub Archive Program

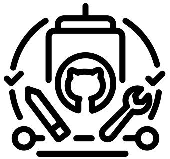

GitHub Arctic Code Vault


GitHub CLI


GitHub Mobile

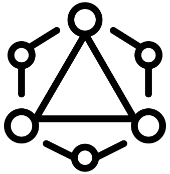
GitHub One


GitHub Pages

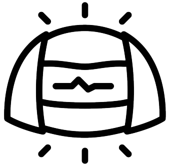

GitHub Transformative services

GitHub for business

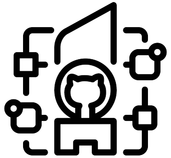
GitHub for teams

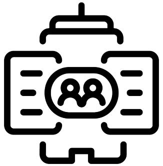

GraphQL

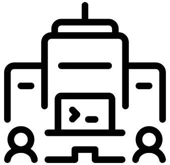
Hosting

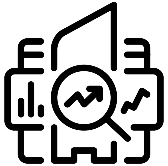

Hubot

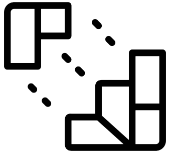
Ideas

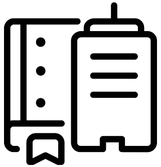

Implementation

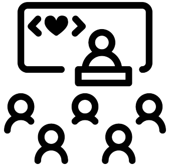

Innersource starter

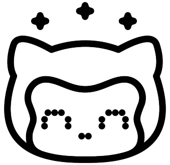

Innersource

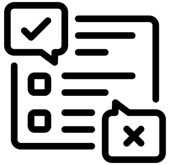

Insights


Integrations


Internal repos


Learning GitHub


Learning Lab for Orgs


Learning Lab

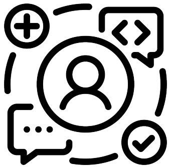
Live logs

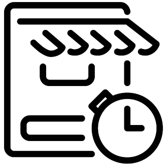
Locked documents

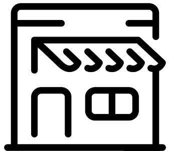
Locked repo

Machine learning

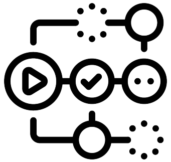
Magnifying glass

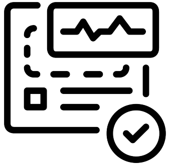

Maintainer

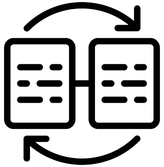

Marketplace trial

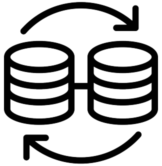

Marketplace

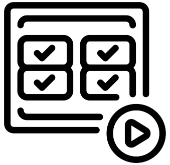
GitHub icons 4 of 5

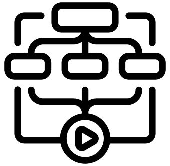

Matrix build

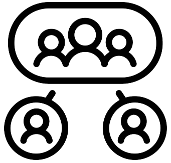

Metrics implementation

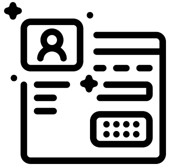

Migration

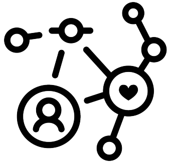

Migrations

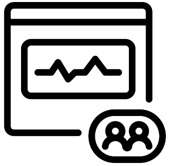

Multi-container testing

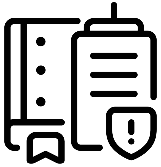

Multiple OS & Containers

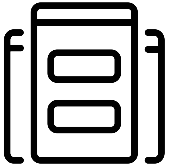

Nested teams

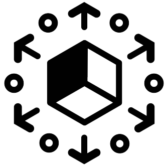

New user

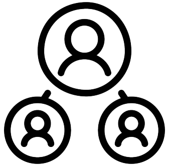

Newsfeed

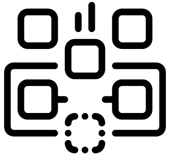

Org insights dashboard

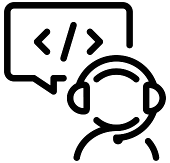

Org wide security policies

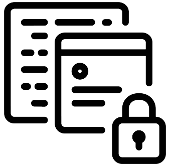

Organized by project status

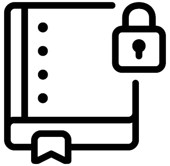

Packages

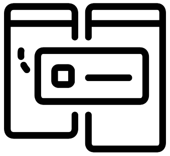

Permissions

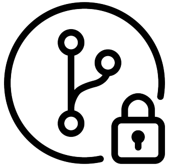

Platform

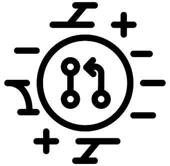

Premium support

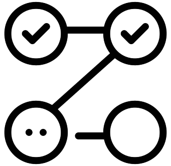

Private pages

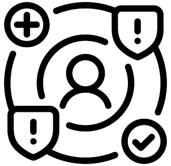

Private repos

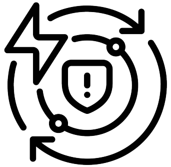

Project management

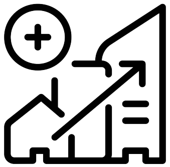

Protected branches

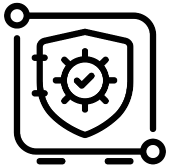

Pull request

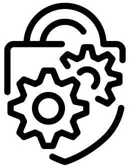

Required status check

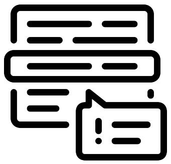

Responsible vulnerability report

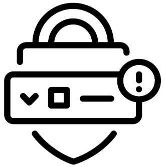

Revolutionary security engine

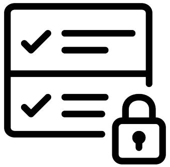

Scale

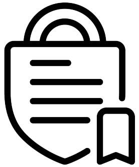

Secret store

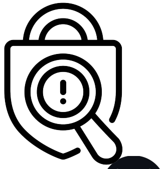

Security advisory API


Security advisory

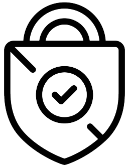

Security alert

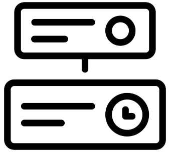

Security and administration

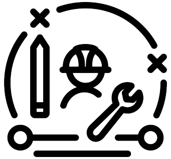

Security policy

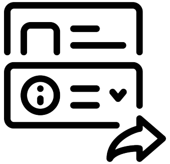

Security researcher

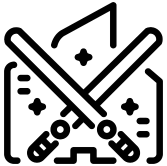
GitHub icons 5 of 5

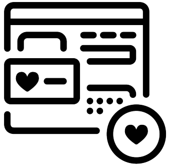

Security

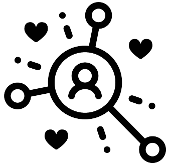

See project activity

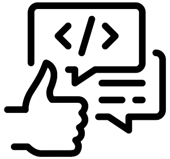

Service account engineer

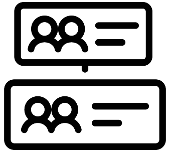

Share work

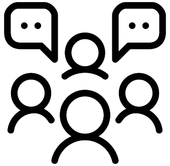

Skywalker

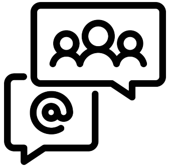

Sponsors in product

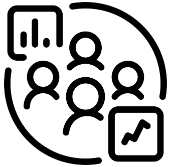

Sponsors

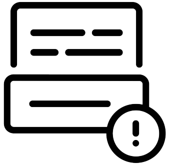

Suggested changes

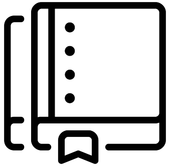

Team activity

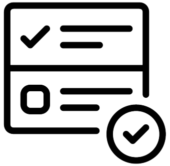

Team discussions

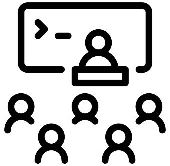

Team mentions

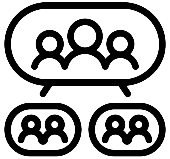

Team metrics

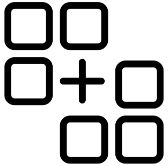

Token scanning

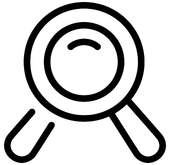

Topic

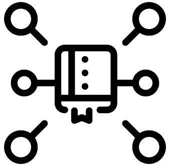

Track progress

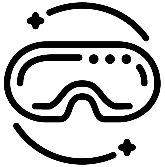

Train the trainer

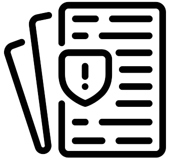

Unified business

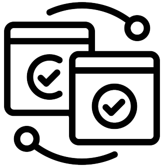

Unified contributions

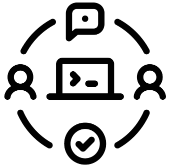

Unified search


Used by


Virtual reality


Vulnerability data


Webhooks


Workflow consultation

We want your feedback!

Our goal is to offer you a template which makes it simple to create a beautiful presentation.Tell us if we succeeded or if there are ways in which we could improve this tool for you!

Feedback survey