

StoneX®

# Git Abyss

REPO <https://github.com/galkatie/GitAbyssWorkshop>





StoneX®

# Git Abyss

Katarzyna Gałwa, Rafał Karwat  
Stonex Financial Ltd

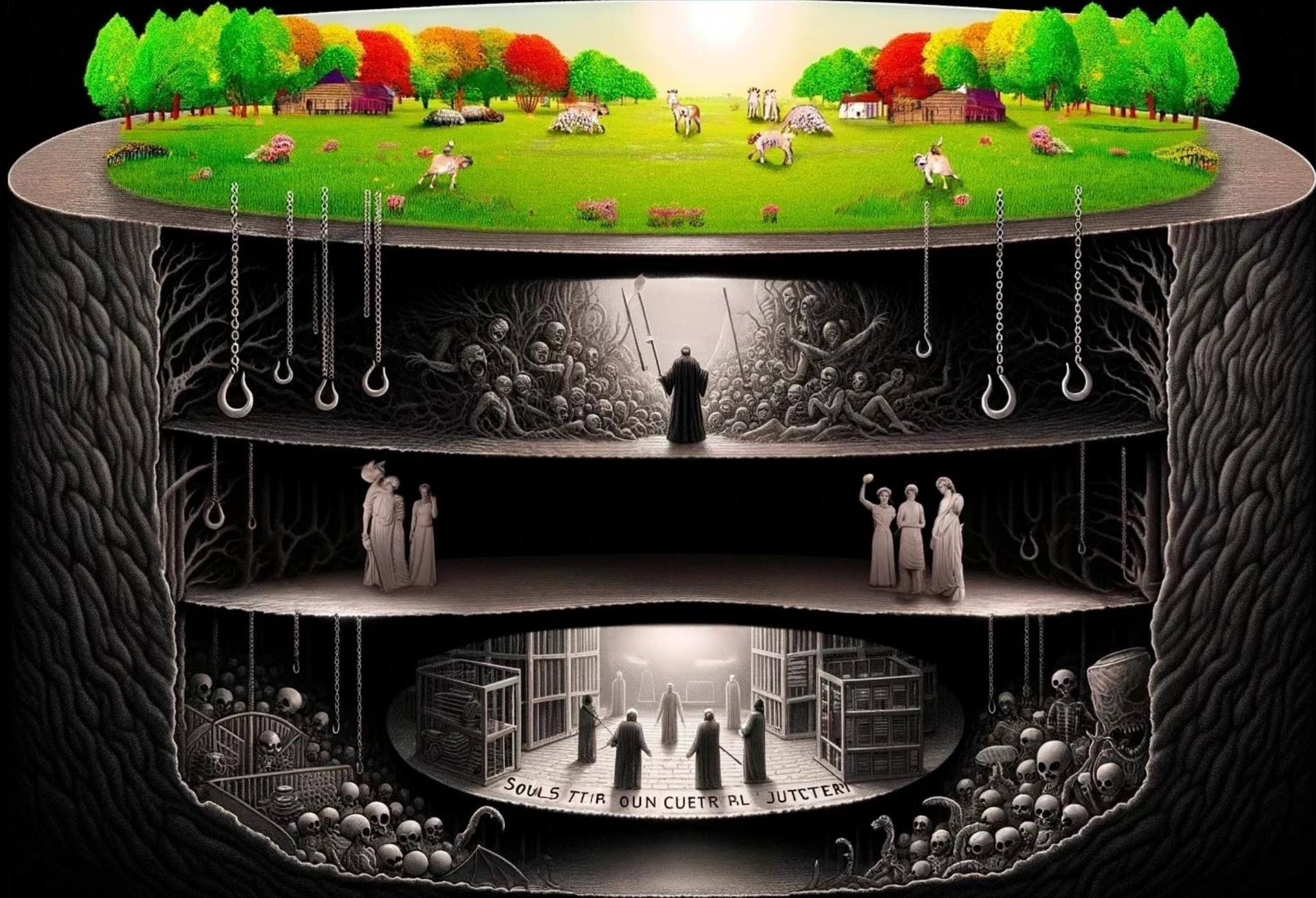






StoneX®





SOULS TIR DUN CUETR AL JUTCTERI



Quiz





git hooks



# Server hooks

## **update**

This hook is invoked before any changes are made to the repository.

## **post-receive**

This hook runs after the changes have been fully received and applied to the repository.

## **auto-gc**

the automatic garbage collection process that runs periodically to optimize the repository.





# Client hooks

## **commit**

Manages various stages of the commit process, including preparing, validating, and performing actions after a commit.

## **checkout**

Manages actions after checking out branches or updating files, ensuring the environment is set up properly and notifications are sent.

## **merge**

Deals with the merge process, handling validations before a merge and actions after a successful merge.

## **applypatch**

Handles actions related to applying patches with git am, allowing checks and notifications during the patch application process

## **rebase**

Handles actions related to rebasing, allowing checks and preparations before starting a rebase.

## **rewrite**

Handles actions related to rewriting commits, such as during an amend or rebase, ensuring checks and preparations before the rewrite process.





# Sum up

## Server

pre-receive

update

post-receive

post-update

pre-auto-gc

## Client

applypatch-msg

pre-applypatch

post-applypatch

pre-commit

prepare-commit-msg

commit-msg

post-commit

pre-rebase

post-checkout

post-merge

pre-push

pre-rebase

pre-auto-gc

post-rewrite



# Sum up

## Server

pre-receive

update

post-receive

post-update

pre-auto-gc

## Client

applypatch-msg

pre-applypatch

post-applypatch

pre-commit

prepare-commit-msg

commit-msg

post-commit

pre-rebase

post-checkout

post-merge

pre-push

pre-rebase

pre-auto-gc

post-rewrite



# Little example

Simple pre-commit



```
1  #!/bin/sh
2
3  # Get the current git username
4  USERNAME=$(git config user.name)
5
6  echo "Hello $USERNAME!"
7
8  exit 0
```

```
/c/git/workshop/GitAbyssWorkshop (main)
$ echo "First modification" > some-file.txt

/c/git/workshop/GitAbyssWorkshop (main)
$ git commit -am "First modification"
[main 3f27e9b] First modification
 1 file changed, 1 insertion(+), 1 deletion(-)

/c/git/workshop/GitAbyssWorkshop (main)
$ echo "Second modification" > some-file.txt

/c/git/workshop/GitAbyssWorkshop (main)
$ cp pre-commit .git/hooks/

/c/git/workshop/GitAbyssWorkshop (main)
$ git commit -am "Second modification"
Hello Rafal Karwat!
[main d26fb9f] Second modification
 1 file changed, 1 insertion(+), 1 deletion(-)
```



## Exercise 1

**Prevents commits  
to main or master**



# Prevents commits to main or master

## Exercise 1

To get branch name:

```
git symbolic-ref --short HEAD
```

Conditional:

```
if <con>; then  
  <body>
```

```
fi
```





```
1  #!/bin/sh
2
3  # Get the name of the current branch
4  BRANCH_NAME=$(git symbolic-ref --short HEAD)
5
6  # Prevent commits to master branch
7  ✓ if [ "$BRANCH_NAME" = "master" ] || [ "$BRANCH_NAME" = "main" ]; then
8      echo "Commits to the master branch are not allowed."
9      echo " Please switch to another branch."
10     exit 1
11 fi
```

# git show

Let the show begins



git bisect good [commit-hash or tag]

# 50 shades of commit

- Hash
- Tag name
- Branch name
- HEAD (@)

Revision selection commands:

- caret (^)
- tilde (~)
- reflog shortcuts
- above can be mixed



# 50 shades of commit

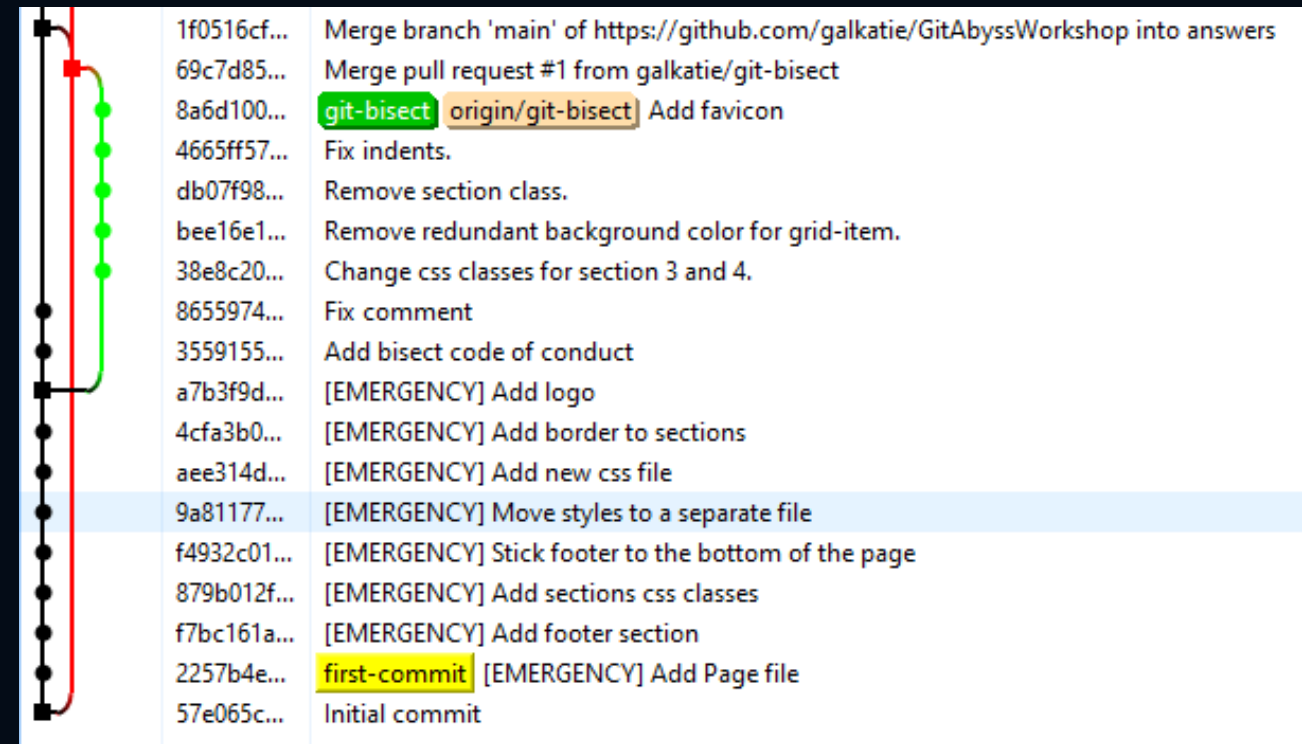
tilde (~)

```
$ git show 1f0516c~1 --oneline --no-patch  
8655974 Fix comment
```

```
$ git show 1f0516c~2 --oneline --no-patch  
3559155 Add bisect code of conduct
```

```
$ git show git-bisect~5 --oneline --no-patch  
a7b3f9d [EMERGENCY] Add logo
```

```
$ git show first-commit~1 --oneline --no-patch  
57e065c Initial commit
```





# 50 shades of commit

caret (^)

```
$ git show 1f0516c^2 --oneline --no-patch
69c7d85 Merge pull request #1 from galkatie/git-bisect
```

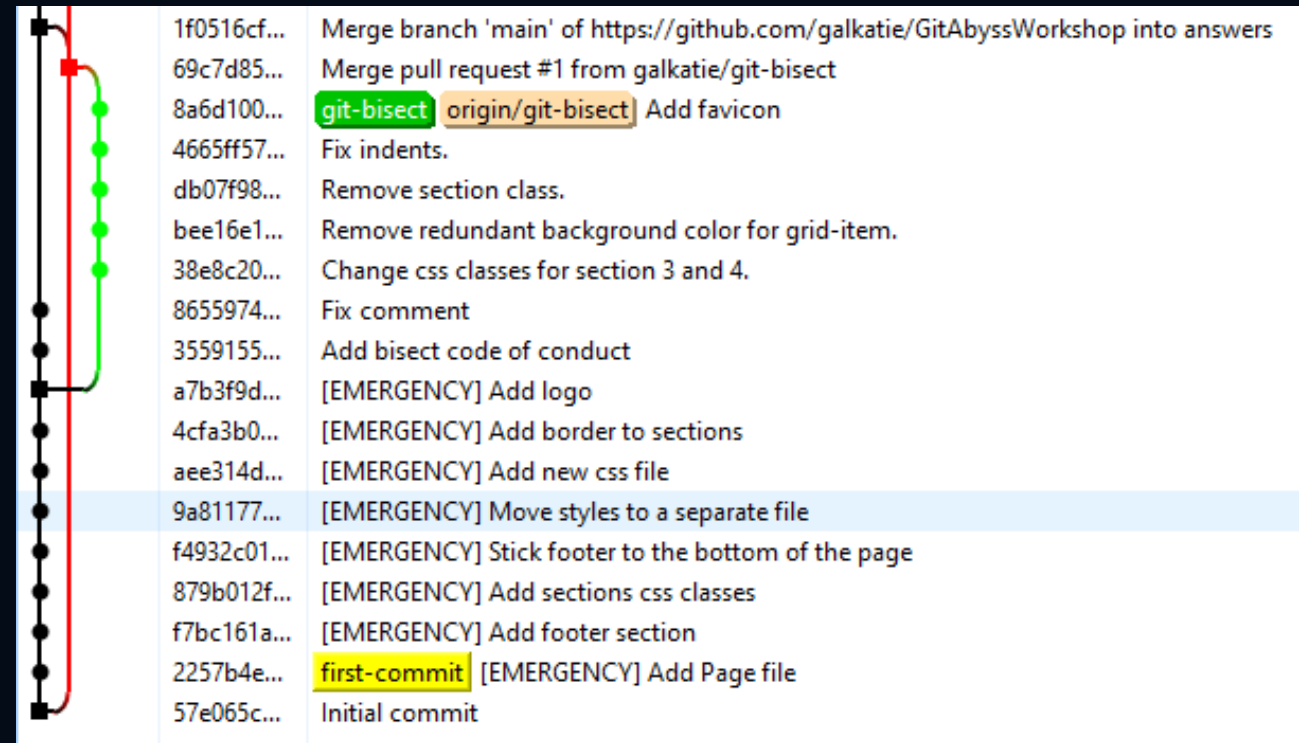
commit^

```
$ git show 1f0516c^ --oneline --no-patch
8655974 Fix comment
```

commit^^

```
$ git show 1f0516c^^ --oneline --no-patch
3559155 Add bisect code of conduct
```

commit^2



# 50 shades of commit

## reflog shortcuts

HEAD@{yesterday}

HEAD@{"1 month 2 days 12 hours ago"}

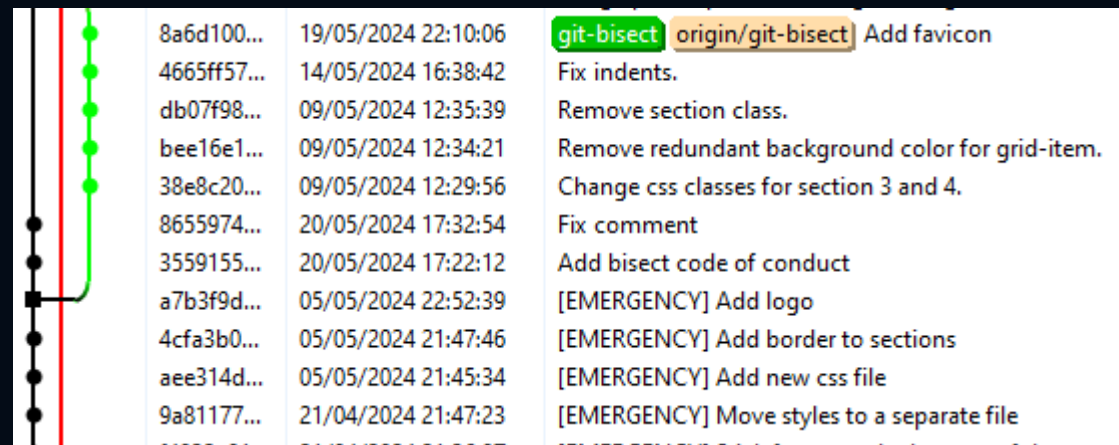
```
$ git show git-bisect@{"1 month 2 days ago"} --oneline --no-patch
4cfa3b0 [EMERGENCY] Add border to sections
```

```
$ git show git-bisect@{"1 month 2 days 12 hours"} --oneline --no-patch
9a81177 [EMERGENCY] Move styles to a separate file
```

HEAD@{2}

```
$ git reflog
226747a (HEAD -> main, origin/main, origin/TestBranch, origin/HEAD) HEAD@{0}: checkout: moving from git-bisect to main
8a6d100 (origin/git-bisect, git-bisect) HEAD@{1}: checkout: moving from 2257b4e37d8208f847a27dc95695964a65353648 to git-bisect
2257b4e (tag: first-commit) HEAD@{2}: checkout: moving from git-bisect to first-commit
8a6d100 (origin/git-bisect, git-bisect) HEAD@{3}: checkout: moving from main to git-bisect
226747a (HEAD -> main, origin/main, origin/TestBranch, origin/HEAD) HEAD@{4}: checkout: moving from answers to main
```

```
$ git show HEAD@{2} --oneline --no-patch
2257b4e (tag: first-commit) [EMERGENCY] Add Page file
```



8a6d100...	19/05/2024 22:10:06	git-bisect origin/git-bisect Add favicon
4665ff57...	14/05/2024 16:38:42	Fix indents.
db07f98...	09/05/2024 12:35:39	Remove section class.
bee16e1...	09/05/2024 12:34:21	Remove redundant background color for grid-item.
38e8c20...	09/05/2024 12:29:56	Change css classes for section 3 and 4.
8655974...	20/05/2024 17:32:54	Fix comment
3559155...	20/05/2024 17:22:12	Add bisect code of conduct
a7b3f9d...	05/05/2024 22:52:39	[EMERGENCY] Add logo
4cfa3b0...	05/05/2024 21:47:46	[EMERGENCY] Add border to sections
aee314d...	05/05/2024 21:45:34	[EMERGENCY] Add new css file
9a81177...	21/04/2024 21:47:23	[EMERGENCY] Move styles to a separate file



# Default structure

## commit metadata

```
commit 2257b4e37d8208f847a27dc95695964a65353648
Author: galkatie <32791965+galkatie@users.noreply.github.com>
Date:   Sun Apr 21 21:19:17 2024 +0200

[EMERGENCY] Add Page file
```

## diff of changes

```
diff --git a/GitBisect/grid.css b/GitBisect/grid.css
index 204caf4..c18649f 100644
--- a/GitBisect/grid.css
+++ b/GitBisect/grid.css
@@ -5,4 +5,12 @@
 .grid-item{
     border: 1px solid black;
+    background-color: blue;
+    color:bisque;
+}
+
+.grid-item:hover {
+    border: 1px solid black;
+    background-color: black;
+    color: white;
+}
\ No newline at end of file
```

# Commit metadata

Predefined formats(`--pretty=[option]`):

`oneline`

`short`

`medium` (default)

`full`

`email`





```
$ git show ba60590 --pretty=oneline --no-patch
ba605901f4bd40a22e6335c5dad75d03f6ac3317 Add info about documentation to readme
```

```
$ git show ba60590 --pretty=short --no-patch
commit ba605901f4bd40a22e6335c5dad75d03f6ac3317
Author: galkatie <galwa.katarzyna@gmail.com>
```

Add info about documentation to readme

```
$ git show ba60590 --pretty=medium --no-patch
commit ba605901f4bd40a22e6335c5dad75d03f6ac3317
Author: galkatie <galwa.katarzyna@gmail.com>
Date: Sat Jun 8 08:37:15 2024 +0200
```

Add info about documentation to readme

```
$ git show ba60590 --pretty=full --no-patch
commit ba605901f4bd40a22e6335c5dad75d03f6ac3317
Author: galkatie <galwa.katarzyna@gmail.com>
Commit: galkatie <galwa.katarzyna@gmail.com>
```

Add info about documentation to readme

```
$ git show ba60590 --pretty=email --no-patch
From ba605901f4bd40a22e6335c5dad75d03f6ac3317 Mon Sep 17 00:00:00 2001
From: galkatie <galwa.katarzyna@gmail.com>
Date: Sat, 8 Jun 2024 08:37:15 +0200
Subject: [PATCH] Add info about documentation to readme
```

# Commit metadata

Custom format

`--format="custom format"`

Chosen placeholders:

commit hash (`%H`, `%h`)

author (`%aN`, `%an`, `%ae`)

date (`%ai`, `%at`, `%as`)

subject (`%s`)

format:

color (`%C(red)`)

decoration (`bold`, `italic`)

\*see documentation for many other options



```
/c/git/workshop (workshop-test-cases)
```

```
$ git show 4665ff5 --format="%h %aN %ai %B" --no-patch  
4665ff5 galkatie 2024-05-14 16:38:42 +0200 Fix indents.
```

```
/c/git/workshop (workshop-test-cases)
```

```
$ git show 4665ff5 --format="%C(green)%H %C(u! magenta)%aN %C(italic blue)%ar %C(reset%B" --no-patch  
4665ff57e2dabe889f795da41762bb4bdbec18de galkatie 4 weeks ago Fix indents.
```

```
/c/git/workshop (workshop-test-cases)
```

```
$ git show first-commit --format="%C(fff0080)%t %C(8080c0)%ae %C(22dba8)%as%reset : %s" --no-patch  
d96ca0b 32791965+galkatie@users.noreply.github.com 2024-04-21 : [EMERGENCY] Add Page file
```



# Diff formatting

Default git diff format

No diff information (**--no-patch**)

Only list of files (**--name-only**)

Raw diff format (**--raw**)

Statistics (**--stat**, **--numstat**, **--shortstat**)

Information about changes in submodule (**--submodule**)



```
/c/git/workshop (workshop-test-cases)
$ git show 4665ff5 --oneline --name-only
4665ff5 Fix indents.
GitBisect/Page.html
```

```
/c/git/workshop (workshop-test-cases)
$ git show 4665ff5 --oneline --raw
4665ff5 Fix indents.
:100644 100644 adc18b6 c8ff024 M          GitBisect/Page.html
```

```
/c/git/workshop (workshop-test-cases)
$ git show 4665ff5 --oneline --stat
4665ff5 Fix indents.
GitBisect/Page.html | 36 ++++++-----
1 file changed, 18 insertions(+), 18 deletions(-)
```

```
/c/git/workshop (workshop-test-cases)
$ git show 4665ff5 --oneline --numstat
4665ff5 Fix indents.
18      18      GitBisect/Page.html
```

```
/c/git/workshop (workshop-test-cases)
$ git show 4665ff5 --oneline --shortstat
4665ff5 Fix indents.
1 file changed, 18 insertions(+), 18 deletions(-)
```



## Exercise 3

# Use of revision selection





# Use of revision selection

## Exercise 3

Checkout to branch '**workshop-test-cases**'.

Starting from **17ad2db** and using only revision selection commands find commit named '**Add bisect code of conduct**' and display it

[Optional] Display result in format:

- abbreviated commit hash - in chosen color
- date without hour - in chosen color in *italic*
- author name - bold in chosen color
- subject - no special formatting

Diff format – full statistics



```
/c/git/workshop (workshop-test-cases)
```

```
$ git show 17ad2db~3^2~8^^2~2 --format="%C(brightred)%h %C(italic blue)%as %C(reset magenta bold)%ae%Creset %s" --stat
7026850 2024-05-20 galwa.katarzyna@gmail.com Add bisect code of conduct
```

```
Dont look up/Bisect/bisect_typo.js      | 26 ++++++
Dont look up/Bisect/bisect_typo.ps1     | 13 ++++++
Dont look up/Bisect/bisect_typo.sh      | 14 ++++++
Dont look up/Bisect/how to use scripts.txt | 22 ++++++
4 files changed, 75 insertions(+)
```

```
/c/git/workshop (workshop-test-cases)
```

```
$ git show 17ad2db~3^2~9^2~2 --format="%C(brightred)%h %C(italic blue)%as %C(reset magenta bold)%ae%Creset %s" --stat
7026850 2024-05-20 galwa.katarzyna@gmail.com Add bisect code of conduct
```

```
Dont look up/Bisect/bisect_typo.js      | 26 ++++++
Dont look up/Bisect/bisect_typo.ps1     | 13 ++++++
Dont look up/Bisect/bisect_typo.sh      | 14 ++++++
Dont look up/Bisect/how to use scripts.txt | 22 ++++++
4 files changed, 75 insertions(+)
```

# git submodule

This is not the solution you're looking for



# Consider first

Repository or dependency manager  
(Nuget, Maven, RubyGems, Npm, and more...)

Utilize pipelines or other CI/CD features

Git Subtree



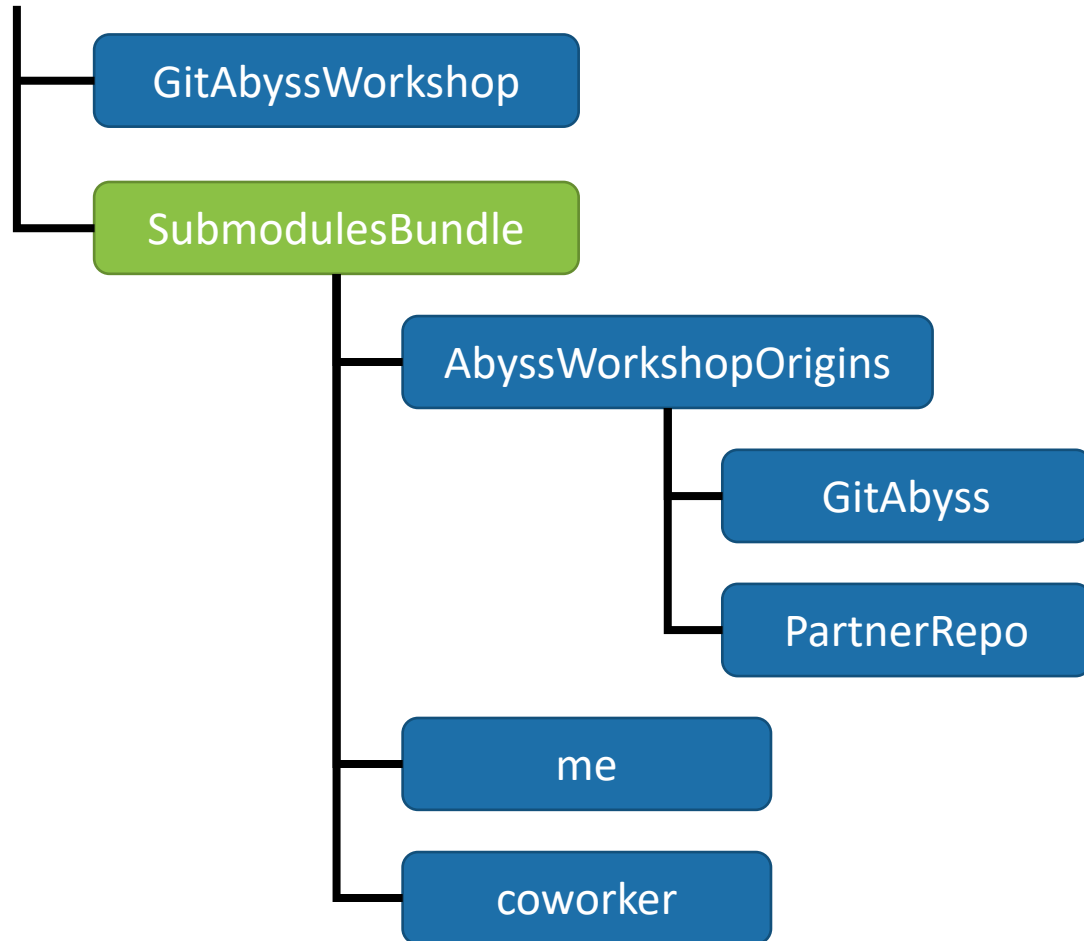
# Before we (sub)start

```
git checkout submodules  
./init.sh  
cd ../SubmodulesBundle/
```



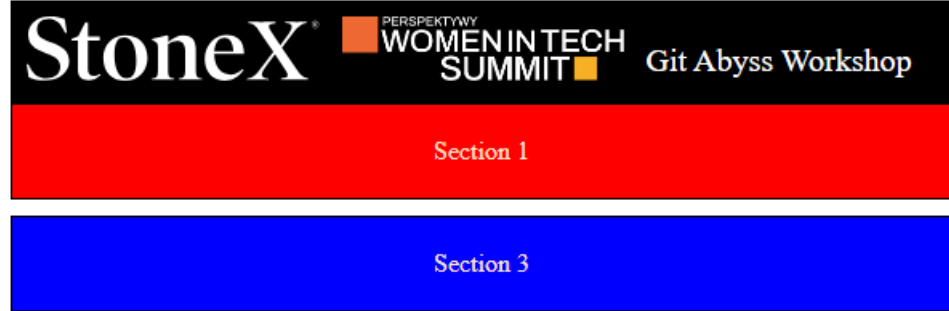


# Before we (sub)start





# Before we (sub)start



# Adding new submodule

```
/c/git/workshop/SubmodulesBundle/me (submodules)
$ git submodule add ../../AbyssWorkshopOrigins/PartnerRepo ./plugins/partner
Cloning into 'C:/git/workshop/SubmodulesBundle/me/plugins/partner'...
done.
```

```
/c/git/workshop/SubmodulesBundle/me (submodules)
$ git status
On branch submodules
Your branch is up to date with 'origin/submodules'.
```

```
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .gitmodules
    new file:   plugins/partner
```

```
/c/git/workshop/SubmodulesBundle/me (submodules)
$ ls plugins/partner/
partner-logo.png  remote.css
```

# Adding new submodule





# Adding new submodule

```
/c/git/workshop/SubmodulesBundle/me (submodules)
$ git config --global status.submoduleSummary true

/c/git/workshop/SubmodulesBundle/me (submodules)
$ git status
On branch submodules
Your branch is up to date with 'origin/submodules'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .gitmodules
    new file:   plugins/partner

Submodule changes to be committed:

* plugins/partner 0000000...0a867a9 (5):
  > Everything works fine, doesn't it?
```

# Adding new submodule

```
/c/git/workshop/SubmodulesBundle/me (submodules)  
$ cd plugins/partner/  
  
/c/git/workshop/SubmodulesBundle/me/plugins/partner (main)  
$ ls  
partner-logo.png  remote.css
```

# Adding new submodule

```
/c/git/workshop/SubmodulesBundle/me (submodules)
$ git add -A

/c/git/workshop/SubmodulesBundle/me (submodules)
$ git commit -m "Add partner submodule"
[submodules cd9fa98] Add partner submodule
 2 files changed, 4 insertions(+)
 create mode 100644 .gitmodules
 create mode 160000 plugins/partner

/c/git/workshop/SubmodulesBundle/me (submodules)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 438 bytes | 438.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
To C:/git/workshop/SubmodulesBundle/./AbyssWorkshopOrigins/GitAbyss
 2fc7e76..cd9fa98  submodules -> submodules
```



# Getting new submodule

```
/c/git/workshop/SubmodulesBundle/coworker (submodules)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 418 bytes | 18.00 KiB/s, done.
From C:/git/workshop/SubmodulesBundle/./AbyssWorkshopOrigins/GitAbyss
    2fc7e76..cd9fa98  submodules -> origin/submodules
Updating 2fc7e76..cd9fa98
Fast-forward
 .gitmodules      | 3 +++
 plugins/partner  | 1 +
 2 files changed, 4 insertions(+)
 create mode 100644 .gitmodules
 create mode 160000 plugins/partner

/c/git/workshop/SubmodulesBundle/coworker (submodules)
$ ls plugins/partner/
```

# Getting new submodule

```
/c/git/workshop/SubmodulesBundle/coworker (submodules)  
$ cat .gitmodules  
[submodule "plugins/partner"]  
    path = plugins/partner  
    url = ../../AbyssWorkshopOrigins/PartnerRepo
```

# Getting new submodule

```
/c/git/workshop/SubmodulesBundle/coworker (submodules)
$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = false
    bare = false
    logallrefupdates = true
    symlinks = false
    ignorecase = true
[remote "origin"]
    url = C:/git/workshop/SubmodulesBundle/./AbyssWorkshopOrigins/GitAbyss
    fetch = +refs/heads/*:refs/remotes/origin/*
[branch "submodules"]
    remote = origin
    merge = refs/heads/submodules
[submodule "plugins/partner"]
    active = true
    url = C:/git/workshop/SubmodulesBundle/./AbyssWorkshopOrigins/PartnerRepo

/c/git/workshop/SubmodulesBundle/coworker (submodules)
$ ls plugins/partner/
```



# Getting new submodule

```
/c/git/workshop/SubmodulesBundle/coworker (submodules)
$ git submodule update
Cloning into 'C:/git/workshop/SubmodulesBundle/coworker/plugins/partner'...
done.
Submodule path 'plugins/partner': checked out '0a867a9ac8e6b5dc895542bc25f61c'

/c/git/workshop/SubmodulesBundle/coworker (submodules)
$ ls plugins/partner/
partner-logo.png  remote.css

/c/git/workshop/SubmodulesBundle/coworker (submodules)
$ cd plugins/partner/

/c/git/workshop/SubmodulesBundle/coworker/plugins/partner ((0a867a9...))
```

# Making commit with submodule

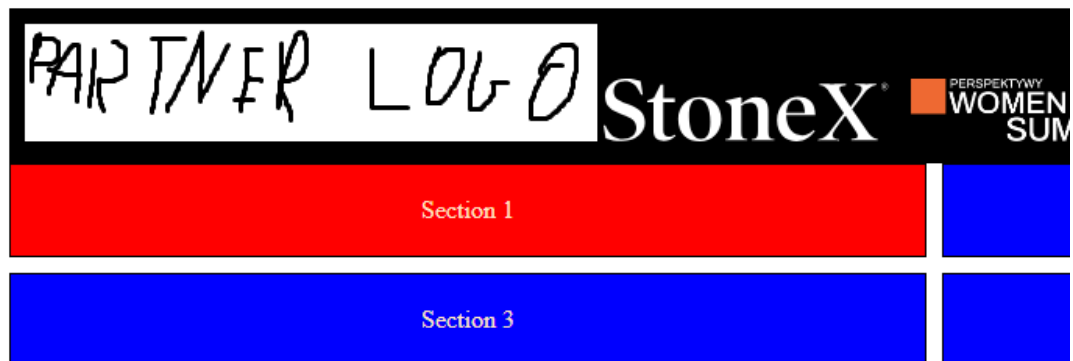
```
/c/git/workshop/SubmodulesBundle/me (submodules)
$ echo ".$(cat plugins/partner/remote.css)" > plugins/partner/remote.css

/c/git/workshop/SubmodulesBundle/me (submodules)
$ git status
On branch submodules
Your branch is up to date with 'origin/submodules'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
  (commit or discard the untracked or modified content in submodules)
        modified:   plugins/partner (modified content)

no changes added to commit (use "git add" and/or "git commit -a")
```

# Adding new submodule





# Making commit with submodule

```
/c/git/workshop/SubmodulesBundle/me (submodules)
$ cd plugins/partner/

/c/git/workshop/SubmodulesBundle/me/plugins/partner (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   remote.css

no changes added to commit (use "git add" and/or "git commit -a")
```

# Making commit with submodule

```
/c/git/workshop/SubmodulesBundle/me/plugins/partner (main)
$ git commit -am "Fixing the bug in submodule"
[main 5f3b4e9] Fixing the bug in submodule
 1 file changed, 1 insertion(+), 1 deletion(-)
/c/git/workshop/SubmodulesBundle/me/plugins/partner (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 333 bytes | 333.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
To C:/git/workshop/SubmodulesBundle/./AbyssWorkshopOrigins/PartnerRepo
 0a867a9..5f3b4e9  main -> main
```

# Making commit with submodule

```
/c/git/workshop/SubmodulesBundle/me/plugins/partner (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

/c/git/workshop/SubmodulesBundle/me (submodules)
$ git status
On branch submodules
Your branch is up to date with 'origin/submodules'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   plugins/partner (new commits)

no changes added to commit (use "git add" and/or "git commit -a")
```



# Making commit with submodule

```
/c/git/workshop/SubmodulesBundle/me (submodules)
$ git commit -am "Updating submodule to fixed commit"
[submodules f267138] Updating submodule to fixed commit
1 file changed, 1 insertion(+), 1 deletion(-)

/c/git/workshop/SubmodulesBundle/me (submodules)
$ git status
On branch submodules
Your branch is ahead of 'origin/submodules' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

# Getting update with submodule

```
/c/git/workshop/SubmodulesBundle/coworker (submodules)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 291 bytes | 11.00 KiB/s, done.
From C:/git/workshop/SubmodulesBundle/./AbyssWorkshopOrigins/GitAbyss
    cd9fa98..f267138  submodules -> origin/submodules
Fetching submodule plugins/partner
From C:/git/workshop/SubmodulesBundle/./AbyssWorkshopOrigins/PartnerRepo
    0a867a9..5f3b4e9  main      -> origin/main
Updating cd9fa98..f267138
Fast-forward
 plugins/partner | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

/c/git/workshop/SubmodulesBundle/coworker (submodules)
$ cat plugins/partner/remote.css
header::before {
```

# Getting update with submodule

```
/c/git/workshop/SubmodulesBundle/coworker/plugins/partner ((0a867a9...))
$ git show --pretty=oneline --no-patch -4
0a867a9ac8e6b5dc895542bc25f61cccaf2ef4d0 (HEAD, main) Everything works fine, doesn't it?
91c1a6ea16a6dcd21ee1a24274a70e431d863e47 merge
ae7057e91350e743d8d8d1e45e036bdeffb748c7 Fix
b6e9325afa21289aea93edef518dee1ebfe47f4f Remote css

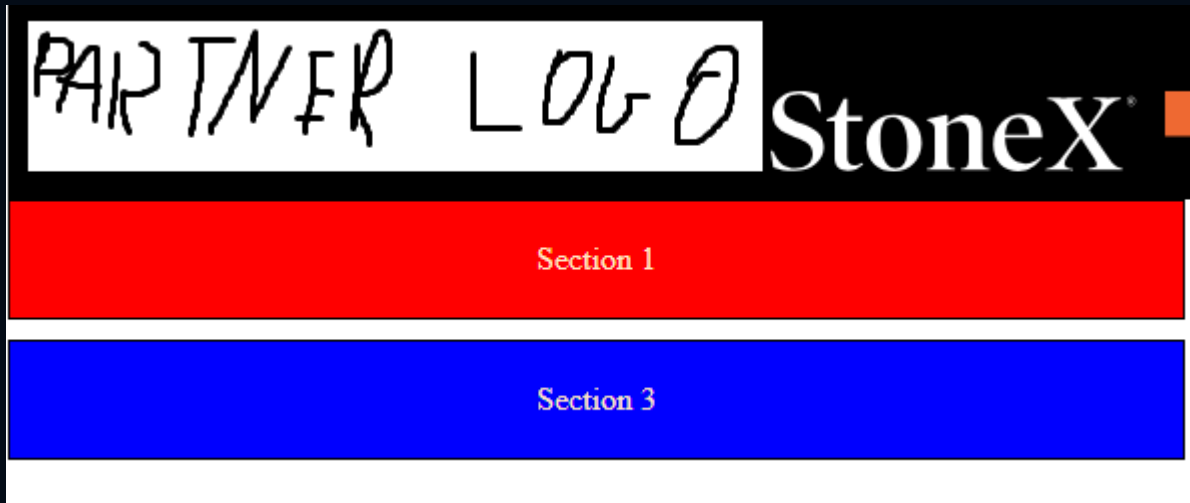
/c/git/workshop/SubmodulesBundle/coworker (submodules)
$ git pull --recurse-submodules
Fetching submodule plugins/partner
Already up to date.
Submodule path 'plugins/partner': checked out '5f3b4e95e3d563db5c461458a876745f364bc00c'

/c/git/workshop/SubmodulesBundle/coworker/plugins/partner ((5f3b4e9...))
$ git show --pretty=oneline --no-patch -4
5f3b4e95e3d563db5c461458a876745f364bc00c (HEAD, origin/main, origin/HEAD) Fixing the bug
0a867a9ac8e6b5dc895542bc25f61cccaf2ef4d0 (main) Everything works fine, doesn't it?
91c1a6ea16a6dcd21ee1a24274a70e431d863e47 merge
ae7057e91350e743d8d8d1e45e036bdeffb748c7 Fix
```



# Getting update with submodule

```
/c/git/workshop/SubmodulesBundle/coworker (submodules)  
$ cat plugins/partner/remote.css  
.header::before {  
    content: '';  
    display: inline-block;
```



# Removing submodule

```
/c/git/workshop/SubmodulesBundle/me (submodules)  
$ git rm plugins/partner  
rm 'plugins/partner'
```

# Removing submodule

```
/c/git/workshop/SubmodulesBundle/me (submodules)
$ git status
On branch submodules
Your branch is up to date with 'origin/submodules'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    plugins/partner

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   .gitmodules

Submodule changes to be committed:

* plugins/partner 5f3b4e9...0000000:

/c/git/workshop/SubmodulesBundle/me (submodules)
$ cat .gitmodules
```



# Removing submodule

```
/c/git/workshop/SubmodulesBundle/me (submodules)
$ git commit -m "Remove submodule"
[submodules 953d9b7] Remove submodule
2 files changed, 4 deletions(-)
delete mode 160000 plugins/partner

/c/git/workshop/SubmodulesBundle/me (submodules)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 266 bytes | 266.00 KiB/s, done.
Total 3 (delta 1), reused 1 (delta 0), pack-reused 0
To C:/git/workshop/SubmodulesBundle/./AbyssWorkshopOrigins/GitAbyss
f267138..953d9b7  submodules -> submodules
```

# Someone removed submodule

```
/c/git/workshop/SubmodulesBundle/coworker (submodules)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 1 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 246 bytes | 10.00 KiB/s, done.
From C:/git/workshop/SubmodulesBundle/./AbyssWorkshopOrigins/GitAbyss
    f267138..953d9b7  submodules -> origin/submodules
warning: unable to rmdir 'plugins/partner': Directory not empty
Updating f267138..953d9b7
Fast-forward
 .gitmodules      | 3 ---
 plugins/partner  | 1 -
 2 files changed, 4 deletions(-)
 delete mode 160000 plugins/partner
```

# Someone removed submodule

```
/c/git/workshop/SubmodulesBundle/coworker (submodules)
$ git status
On branch submodules
Your branch is up to date with 'origin/submodules'.

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

nothing added to commit but untracked files present (use "git add" to track)

/c/git/workshop/SubmodulesBundle/coworker (submodules)
$ ls plugins/partner/
partner-logo.png  remote.css
```



# Someone removed submodule

```
/c/git/workshop/SubmodulesBundle/coworker (submodules)
$ rm -rf plugins/partner/

/c/git/workshop/SubmodulesBundle/coworker (submodules)
$ git status
On branch submodules
Your branch is up to date with 'origin/submodules'.

nothing to commit, working tree clean
```

# Who did this?

1. Search manually history
2. Using git blame
3. Using git bisect



# Git blame

Basic usage:

`git blame "file_path"`

```
$ git blame "GitBisect\Page.html" --color-by-age
2257b4e3 (galkatie 2024-04-21 21:19:17 +0200 1) <!DOCTYPE html>
879b012f (galkatie 2024-04-21 21:33:51 +0200 2) <html lang="en">
4665ff57 (galkatie 2024-05-14 16:38:42 +0200 3)     <head>
4665ff57 (galkatie 2024-05-14 16:38:42 +0200 4)         <title>Git bisect </title>
4665ff57 (galkatie 2024-05-14 16:38:42 +0200 5)         <link rel="stylesheet" href="grid.css">
4665ff57 (galkatie 2024-05-14 16:38:42 +0200 6)         <link rel="stylesheet" href="styles.css">
4665ff57 (galkatie 2024-05-14 16:38:42 +0200 7)     </head>
4665ff57 (galkatie 2024-05-14 16:38:42 +0200 8)     <body>
4665ff57 (galkatie 2024-05-14 16:38:42 +0200 9)         <div class="header">
4665ff57 (galkatie 2024-05-14 16:38:42 +0200 10)             
4665ff57 (galkatie 2024-05-14 16:38:42 +0200 11)             <span class="title">Git Abyss workshop</span>
4665ff57 (galkatie 2024-05-14 16:38:42 +0200 12)         </div>
4665ff57 (galkatie 2024-05-14 16:38:42 +0200 13)         <div class="grid-container">
4665ff57 (galkatie 2024-05-14 16:38:42 +0200 14)             <div class="grid-item section1">Section 1</div>
4665ff57 (galkatie 2024-05-14 16:38:42 +0200 15)             <div class="grid-item section2">Section 2</div>
4665ff57 (galkatie 2024-05-14 16:38:42 +0200 16)             <div class="grid-item">Section 3</div>
4665ff57 (galkatie 2024-05-14 16:38:42 +0200 17)             <div class="grid-item">Section 4</div>
4665ff57 (galkatie 2024-05-14 16:38:42 +0200 18)         </div>
4665ff57 (galkatie 2024-05-14 16:38:42 +0200 19)         <div class="footer">This is page footer</div>
4665ff57 (galkatie 2024-05-14 16:38:42 +0200 20)     </body>
2257b4e3 (galkatie 2024-04-21 21:19:17 +0200 21) </html>
```

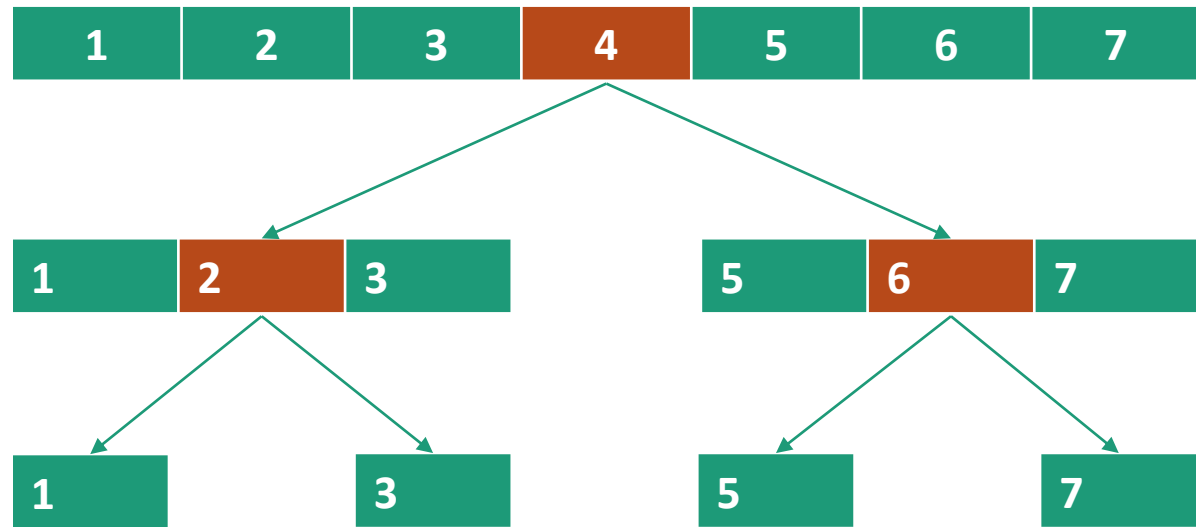
<https://git-scm.com/docs/git-blame>

# git bisect

New lease of blaming



# Binary search





git bisect good [commit-hash or tag]

# Quick start

1. Start bisecting  
`git bisect start`
2. mark it as buggy commit  
`git bisect bad commit-hash-tag`
3. Mark good commit  
`git bisect good commit-hash-tag`





# What's next?

1. Follow the instructions
2. If the current commit has the bug use  
`git bisect bad`
3. If it doesn't, mark it as good:  
`git bisect good`
4. Repeat points 2 and 3 till git will show you the result

If you finished investigation use:

`git bisect reset`





# Tips for Effective Use

Narrow the range

Automate testing:

```
git bisect start  
git bisect bad  
git bisect good commit-hash-tag  
git bisect run [test-script]
```

in script exit with:

0 if commit is good

1 otherwise

Use script language you prefer





## Exercise 6

# Bisecting bug



# Bisecting bug

## Exercise 6

Which commit introduced typo in Section 4?  
Write a script to test it and try to use it with bisect.



```
1  #!/bin/bash
2
3  file_path="./GitBisect/Page.html"
4  search_value="Secction"
5
6  # Use grep to search for the value in the file
7  if grep -q "$search_value" "$file_path"; then
8      # echo "Value found"
9      exit 1
10 else
11     # echo "Value not found"
12     exit 0
13 fi
```

```
$ git bisect start
status: waiting for both good and bad commits
```

```
$ git bisect bad
status: waiting for good commit(s), bad commit known
```

```
$ git bisect good first-commit
Bisecting: 13 revisions left to test after this (roughly 4 steps)
[69c7d85e310a91625efd5d1407353965640e88ab] Merge pull request #1 from galkatie/git-bisect
```

```
$ git bisect run ./bisect_typo.sh
running './bisect_typo.sh'
Bisecting: 6 revisions left to test after this (roughly 3 steps)
[4cfa3b0dc2ac504ced53a3d1c8b7b5f91f0098fb] [EMERGENCY] Add border to sections
running './bisect_typo.sh'
Bisecting: 2 revisions left to test after this (roughly 2 steps)
[f4932c01800ae3b0965523c3c76aa3153fefb1] [EMERGENCY] Stick footer to the bottom of the page
running './bisect_typo.sh'
Bisecting: 0 revisions left to test after this (roughly 1 step)
[aee314d87527b3a15c2abcfdbfa2bf6127bb5b8d] [EMERGENCY] Add new css file
running './bisect_typo.sh'
4cfa3b0dc2ac504ced53a3d1c8b7b5f91f0098fb is the first bad commit
commit 4cfa3b0dc2ac504ced53a3d1c8b7b5f91f0098fb
Author: galkatie <galwa.katarzyna@gmail.com>
Date: Sun May 5 21:47:46 2024 +0200
```

```
[EMERGENCY] Add border to sections
```

```
GitBisect/Page.html | 2 +-
GitBisect/grid.css | 4 ++++
2 files changed, 5 insertions(+), 1 deletion(-)
bisect found first bad commit
```





**Exit Quiz**

A photograph of Earth's horizon from space, showing a bright blue curve against a black background. The horizon line is curved, and the blue color transitions from a deep blue to a lighter, almost white glow at the top of the curve. The background is a solid black, representing the vacuum of space.

**Thank you**