



Google Developer Student Clubs

Introduction to **Git & GitHub**

GDSC Huddersfield

Thu 10 Feb 2022

```
filterByOrg = study.lead_organization === filterByOrg .  
filterStatus = filterByStatus ? study.status === filterByStatus : true  
const matchStatus) {  
    return true;  
}  
};  
  
function filterStudies({ studies, filterByOrg ,  
filterByStatus }) {  
    return studies.filter(study =>  
        filterByOrg ? study.lead_organization === filterByOrg : true  
        &amp;
```



Version Control

Method for Tracking Files

- Don't Keep Track
- Copy Files
- Use Git (VCS)



Version Control

What is Git?

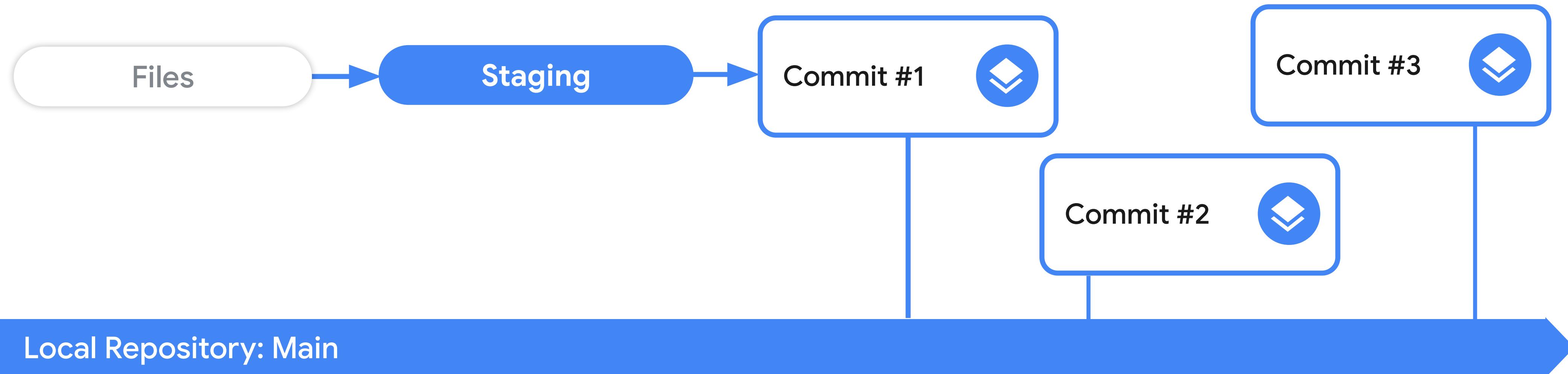
- Git is a version control system (VCS)
- VCS tracks changes to files over time
- VCS enable developer collaboration
- VCS maintains complete history (who, what, when)

Version Control

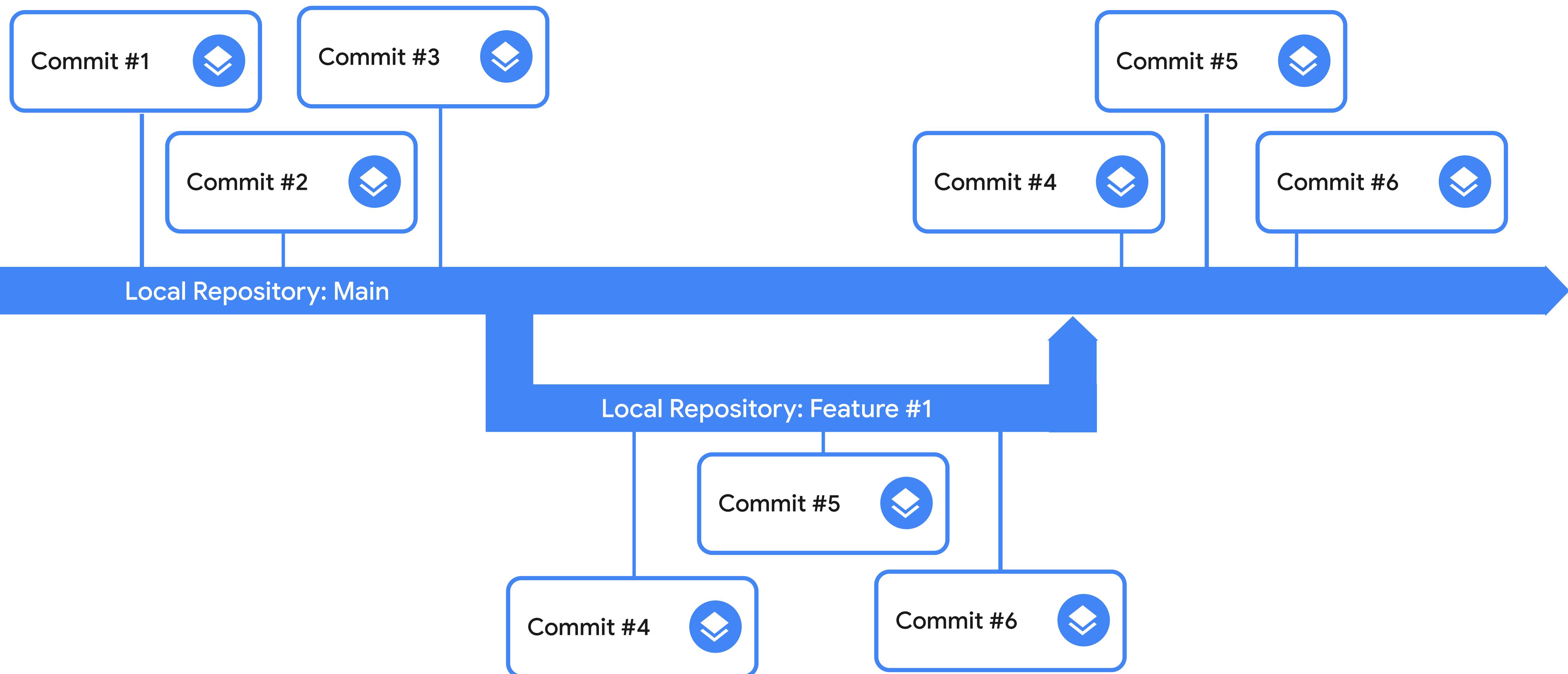
Git Terminologies

- **Working Directory**: Folder on the local file system containing code files
- **Staging Area**: File changes ready for next commit
- **Commit**: Snapshot of currently staged changes
- **Branch**: A separate version of code in repository
- **Local Repository**: Contains commits history for all branches
- **Remote Repository**: Git Repository hosted online for collaboration

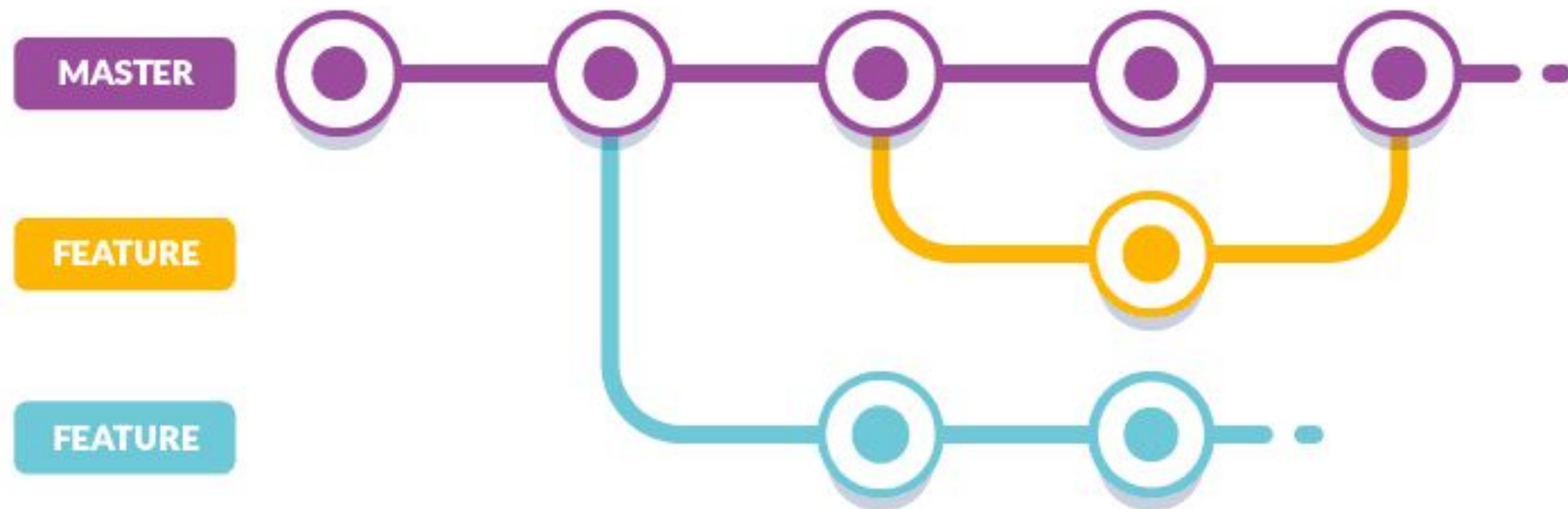
Staging & Commits



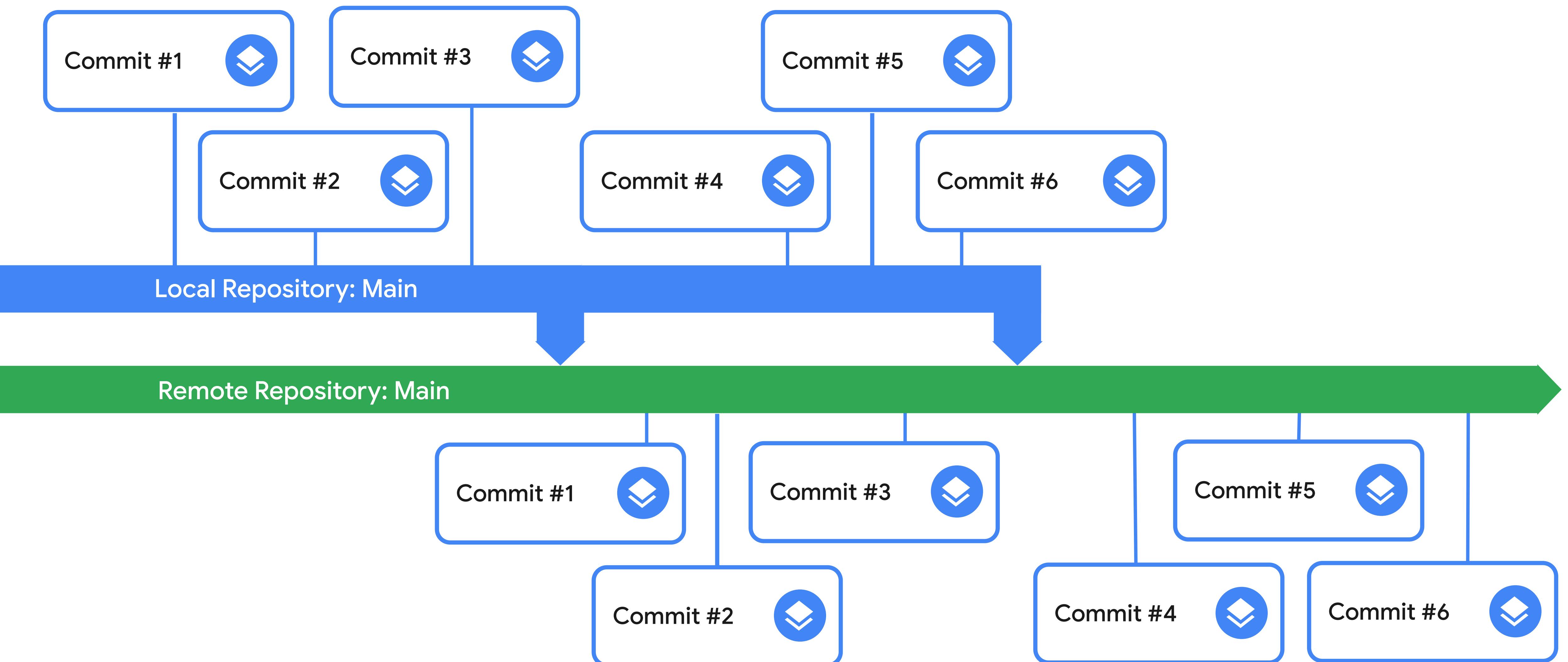
Local Branches



Local Workflow



Remote Branches



Remote Workflow



Commands

`git config --global user.name [name]`

`git config --global user.email [email]`

`git init`

- Create empty Repository

`git clone [url] [folder]`

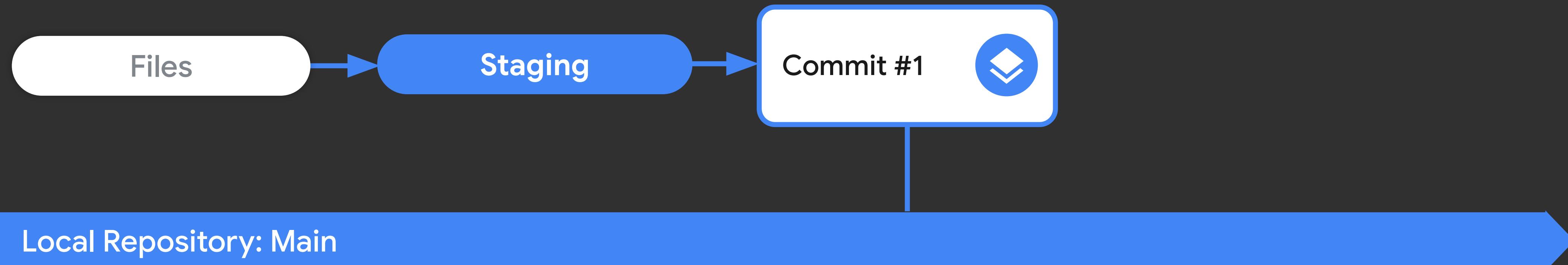
- Clone a remote repository locally

`git add [file]`

- Add [file] to Staging Area

`git commit -m "message"`

- Create commit of the Staged files



Commands

git status

- Show files in the Staging Area

git log

- Show commits history of the branch

git reset [file]

- Show commits history of the branch

git reset --hard [commit]

- Discards all changes and changes back to specified commit

```
commit 742a27add5306b33e0f3042b9595c35cd383b618
Author: ByWaleed <waleedi825@gmail.com>
Date:   Tue Jan 11 17:33:58 2022 +0000
```

```
    API project delete endpoint implemented
```

```
commit 588055b70185439bbd9bbe9b62c58c6a8ab05170
Author: ByWaleed <waleedi825@gmail.com>
Date:   Tue Jan 11 17:28:23 2022 +0000
```

```
    API project update endpoint implemented
```

```
commit 08ffc69dc3fc8f325e023d6a7a4c2084d4228e41
Author: ByWaleed <waleedi825@gmail.com>
Date:   Tue Jan 11 16:44:47 2022 +0000
```

```
:
```

Commands: Branches

`git branch`

- List all branches in current repository

`git branch [name]`

- Create a new branch with the given name

`git branch -d [name]`

- Delete [name] branch

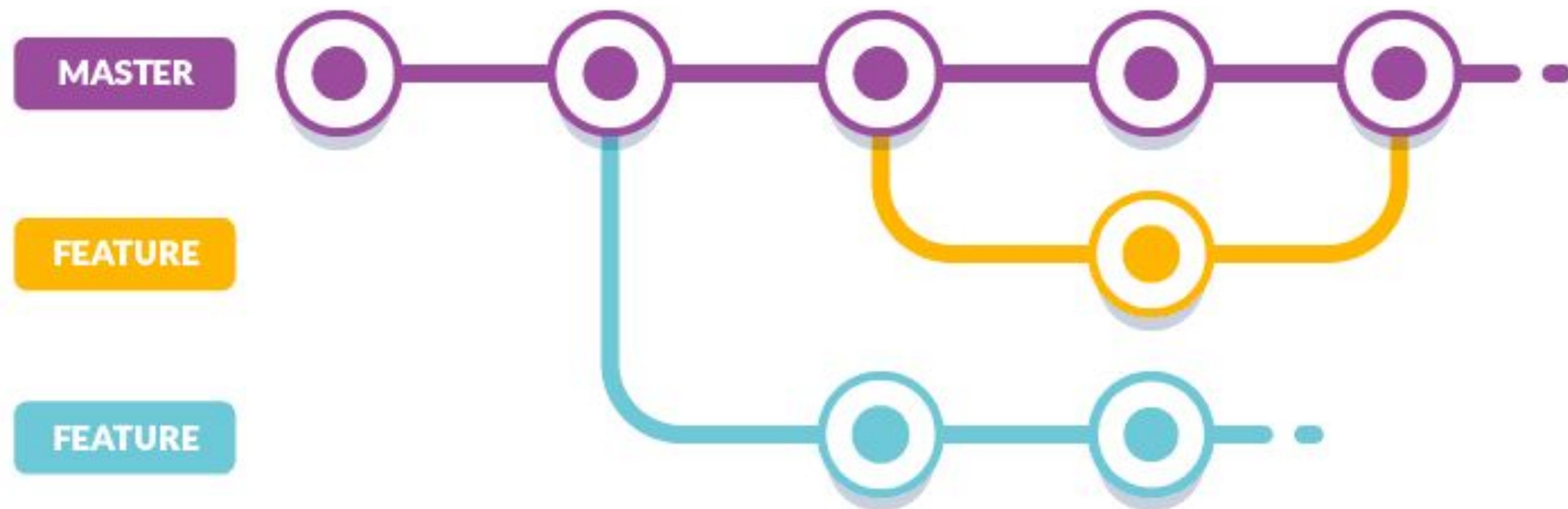
`git checkout [name]`

- Switch to [name] branch & update working directory

`git merge [name]`

- Combines [name] branch's history into the current branch

Local Branches



Commands: Remotes

`git remote -v`

- List all remote git repositories

`git remote add [name] [url]` - Create a new connection to a remote repository

`git push [r-name] [r-branch]` - Send commits in Local Repository to Remote Repository

`git pull [r-name] [r-branch]` - Fetch commits from Remote Repository to Local Repository

Remote Repository on GitHub

Quick setup — if you've done this kind of thing before

 Set up in Desktop

or

HTTPS

SSH

<https://github.com/GDSC-Huddersfield/Introduction-to-Git-and-GitHub.git>



Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# Introduction-to-Git-and-GitHub" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/GDSC-Huddersfield/Introduction-to-Git-and-GitHub.git
git push -u origin main
```



...or push an existing repository from the command line

```
git remote add origin https://github.com/GDSC-Huddersfield/Introduction-to-Git-and-GitHub.git
git branch -M main
git push -u origin main
```



README & Markdown

```
# File: README.md
```

```
# Project Title
```

```
## Features
```

```
• • •
```

```
## How To Use
```

```
• • •
```

```
## Development
```

```
• • •
```

```
## Credits
```

```
• • •
```

.gitignore

```
# Ignore all files or file extensions specified
```

```
--pycache__/
```

```
lib/
```

```
.idea/
```

```
.env
```

[Code](#)[Pull requests 1.6k](#)[Actions](#)[Security](#)[Insights](#)[main](#)[8 branches](#)[497 tags](#)[Go to file](#)[Add file](#)[Code](#)

About

The Python programming language

www.python.org/

[Readme](#)

[View license](#)

[Code of conduct](#)

42.7k stars

1.4k watching

21.8k forks

Releases

497 tags

Sponsor this project

python Python



[https://www.python.org/psf/donations/...](https://www.python.org/psf/donations/)

[Learn more about GitHub Sponsors](#)

Packages

No packages published

ericsnowcurrently	bpo-46541: Replace core use of _Py_IDENTIFIER() with statically...	81c7204 1 hour ago	112,272 commits
.azure-pipelines	bpo-33125: Enables building traditional installer for Windows ARM64 (G...	11 days ago	
.github	bpo-46541: Replace core use of _Py_IDENTIFIER() with statically initi...	1 hour ago	
Doc	bpo-46475: Add typing.Never and typing.assert_never (GH-30842)	3 hours ago	
Grammar	bpo-46272: Fix two heading comments in python.gram (GH-30499)	last month	
Include	bpo-46541: Replace core use of _Py_IDENTIFIER() with statically initi...	1 hour ago	
Lib	bpo-46475: Add typing.Never and typing.assert_never (GH-30842)	3 hours ago	
Mac	bpo-45925: Update macOS installer to SQLite 3.37.2 (GH-30921)	7 days ago	
Misc	bpo-46475: Add typing.Never and typing.assert_never (GH-30842)	3 hours ago	
Modules	bpo-46541: Replace core use of _Py_IDENTIFIER() with statically initi...	1 hour ago	
Objects	bpo-46541: Replace core use of _Py_IDENTIFIER() with statically initi...	1 hour ago	
PC	bpo-46638: Makes registry virtualisation setting stable when building...	yesterday	
PCbuild	bpo-33125: Enables building traditional installer for Windows ARM64 (G...	11 days ago	
Parser	bpo-46541: Replace core use of _Py_IDENTIFIER() with statically initi...	1 hour ago	
Programs	bpo-46541: Replace core use of _Py_IDENTIFIER() with statically initi...	1 hour ago	
Python	bpo-46541: Replace core use of _Py_IDENTIFIER() with statically initi...	1 hour ago	
Tools	bpo-46541: Replace core use of _Py_IDENTIFIER() with statically initi...	1 hour ago	

[Code](#)[Pull requests 1.6k](#)[Actions](#)[Security](#)[Insights](#)

main

Commits on Feb 8, 2022

[bpo-46556: emit DeprecationWarning from pathlib.Path.__enter__\(\)](#) (GH-31209)

barneygale and brettcannon committed 1 hour ago

Verified

06e1701

[bpo-46541: Replace core use of _Py_IDENTIFIER\(\) with statically initia...](#)

ericsnowcurrently committed 1 hour ago

Verified

81c7204

[bpo-45952: Get the C analyzer tool working again. \(gh-31220\)](#)

ericsnowcurrently committed 2 hours ago

Verified

c018d30

[bpo-46475: Add typing.Never and typing.assert_never \(GH-30842\)](#)

JelleZijlstra committed 3 hours ago

Verified

243436f

[bpo-45952: Get the C analyzer tool working again. \(gh-31219\)](#)

ericsnowcurrently committed 3 hours ago

Verified

1e6214d

[bpo-46323: Use PyObject_Vectorcall while calling ctypes callback func...](#)

corona10 committed 9 hours ago

Verified

b552768

[bpo-46521: Fix codeop to use a new partial-input mode of the parser \(GH-31219\)](#)

pablogsal committed 10 hours ago

Verified

69e1097

[bpo-46675: Allow object value arrays and split key dictionaries large...](#)

markshannon committed 10 hours ago

Verified

25db2b3

[Print summary stats for overall success of specialization. \(GH-31211\)](#)

markshannon committed 10 hours ago

Verified

328fe3f

[bpo-46676: Make ParamSpec args and kwargs equal to themselves \(GH-31203\)](#)

GBeauregard committed 14 hours ago

Verified

c8b62bb

[bpo-46323 Fix ref leak if ctypes.CFuncPtr raises an error. \(GH-31209\)](#)

Verified

e959dd9

Allow the parser to avoid nested processing of invalid rules (GH-31252)

[Browse files](#)[main \(#31252\)](#)[pablogsal committed 20 minutes ago](#) Verified

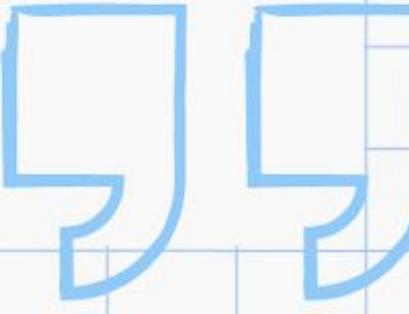
1 parent 2cea8c2 commit 390459de6db1e68b79c0897cc88c0d562693ec5c

[Showing 6 changed files with 1,085 additions and 1,084 deletions.](#)[Split](#)[Unified](#)

```
 11 Grammar/python.gram
@@ -1078,6 +1078,7 @@ invalid_kwarg:
1078     RAISE_SYNTAX_ERROR_KNOWN_RANGE(
1079         a, b, "expression cannot contain assignment, perhaps you meant \"==\"?\"")
1080 +
1081     expression_without_invalid[expr_ty]:
1082         | a=disjunction 'if' b=disjunction 'else' c=expression { _PyAST_IfExp(b, a, c, EXTRA) }
1083         | disjunction
@@ -1095,16 +1096,14 @@ invalid_expression:
1095     RAISE_SYNTAX_ERROR_KNOWN_RANGE(a, b, "invalid syntax. Perhaps you forgot a comma?")
1096     | a=disjunction 'if' b=disjunction !('else'|':') { RAISE_SYNTAX_ERROR_KNOWN_RANGE(a, b,
1097         "expected 'else' after 'if' expression") }
1098 - invalid_left_assignment_prefixes(memo): list|tuple|genexp|'True'|'None'|'False'
1099 -
1100 - invalid_named_expression:
1101     | a=expression ':=' expression {
1102         RAISE_SYNTAX_ERROR_KNOWN_LOCATION(
1103             a, "cannot use assignment expressions with %s", _PyPegen_get_expr_name(a))
1104         | a=NAME '=' b=bitwise_or !(='|':=')
1105 -     p->in_raw_rule ? NULL : RAISE_SYNTAX_ERROR_KNOWN_RANGE(a, b, "invalid syntax. Maybe you
1106         meant '==' or ':=' instead of '='?")
1107 -     | !invalid_left_assignment_prefixes a=bitwise_or b='=' bitwise_or !(='|':=')
1108 -     p->in_raw_rule ? NULL : RAISE_SYNTAX_ERROR_KNOWN_LOCATION(a, "cannot assign to %s here.
1109         Maybe you meant '==' instead of '='?", ...
1078     RAISE_SYNTAX_ERROR_KNOWN_RANGE(
1079         a, b, "expression cannot contain assignment, perhaps you meant \"==\"?\"")
1080 +
1081 + # IMPORTANT: Note that the "_without_invalid" suffix causes the rule to not call invalid rules
1082     expression_without_invalid[expr_ty]:
1083         | a=disjunction 'if' b=disjunction 'else' c=expression { _PyAST_IfExp(b, a, c, EXTRA) }
1084         | disjunction
1096     RAISE_SYNTAX_ERROR_KNOWN_RANGE(a, b, "invalid syntax. Perhaps you forgot a comma?")
1097     | a=disjunction 'if' b=disjunction !('else'|':') { RAISE_SYNTAX_ERROR_KNOWN_RANGE(a, b,
1098         "expected 'else' after 'if' expression") }
1099 + invalid_named_expression(memo):
1100     | a=expression ':=' expression {
1101         RAISE_SYNTAX_ERROR_KNOWN_LOCATION(
1102             a, "cannot use assignment expressions with %s", _PyPegen_get_expr_name(a))
1103         | a=NAME '=' b=bitwise_or !(='|':=')
1104 +     RAISE_SYNTAX_ERROR_KNOWN_RANGE(a, b, "invalid syntax. Maybe you meant '==' or ':='
1105             instead of '='?")
1106 +     | !(list|tuple|genexp|'True'|'None'|'False') a=bitwise_or b='=' bitwise_or !(='|':=')
1107 +     RAISE_SYNTAX_ERROR_KNOWN_LOCATION(a, "cannot assign to %s here. Maybe you meant '=='
1108             instead of '='?", ...)
```

Git Cheat Sheet

[education.github.com/
git-cheat-sheet-education.pdf](https://education.github.com/git-cheat-sheet-education.pdf)



Thank you for attending!

Join Society & Follow on Social Media



<https://bit.ly/gdsc-hudd-links>

