UI EXPLORATION » REPORTING / USAGE
REPORTING
DEVICE USAGE
- Hours used
- Duration
- Avg. hours (within range)
- Peak usage time (window)
≡ MENU
PAIN REPORT
INTERF. REPORT
HOLD

Git For Everyday Development

ATOMIC OBJECT

Who We Are

We’re software developers and members of the Accelerator program at Atomic Object.

Atomic Object creates applications for web, mobile, desktop, and devices. We help companies innovate and grow with custom software products that are beautiful, reliable, and easy to use. Offices in Grand Rapids and Ann Arbor.

Workshop Goals:

- You'll be comfortable using basic Git commands.
- You'll be able to use Git for school and personal projects.
- You'll have a basic understanding and mental model of how Git works.
- You'll have a good idea of where you can go to learn more about Git.

What is Git?

Git is a version control system that keeps track of how files in a codebase have changed.

Git tracks things like what changed in a file, who changed it, and their reason for making that change.

What is Git?

Git also allows others to work in the same repository (project) as you, so you can share your progress with others.

Git is the most widely used version control system in the world, and the majority of software developers use it every day.

Setup

Setup

Clone the repository to your local machine from GitHub using your terminal:

```
$ git clone <github-url>
```

Navigate to the repository on the command line:

```
$ cd <path-to-repository>
```

Open the webpage and look around:

```
$ open index.html
```

Need a command line refresher? Reference the Git cheat sheet or raise your hand.

Making Your First Commit

ATOMIC OBJECT

Making Your First Commit

Check which files have been changed since your last commit:

```
$ git status
```

Check what changes have been made line by line:

```
$ git diff
```

Tip: Use `git diff <filename>` to see the changes for a single file.

Making Your First Commit

Stage the changed files that you would like to commit:

```
$ git add <file(s)>
```

or

```
$ git add --all
```

Commit all staged files with a commit message:

```
$ git commit -m “<message>”
```

Take a look at the reference manual for more information on commit message best practices.

Making Your First Commit

Push to a remote repository:

```
$ git push
```

Making Your First Commit

Open the `index.html` file in your chosen text editor.

Change the placeholders for name and school in the “About Me” section.

Use `git status` to check which files have changed.

Add your changes and commit them.

Push your commit up to GitHub.

Making Your First Commit

We made changes directly on develop, but this isn’t usually how you want to do things.

Each feature should have its own branch so that your changes don’t affect the main repository.

We’ll talk about branches and merging next.

Creating Branches

Creating Branches

Create a new branch:

```
$ git checkout -b <branch-name>
```

Checkout a branch:

```
$ git checkout <branch-name>
```

Go back to the previous branch:

```
$ git checkout -
```

Typically, branch names start with the type of work you're doing and a slash, like `feature/add-new-button`, or `bug/fix-login-issue.`

Creating Branches

Create a new branch off of `develop` for your first task.

Go back to `develop`.

Create another branch off `develop` for your second task.

Implementing a Feature

ATOMIC OBJECT

Implementing a Feature

Merging another branch into your working branch:

```
$ git merge <other-branch>
```

Implementing a Feature

Go to the branch for your first task.

Complete your first task.

Add, commit, and push the changes for your first task.

Checkout `develop` and merge the branch for your first task into it.

Implementing a Feature

Go to the branch for your second task.

Complete your second task.

Add, commit, and push the changes for your second task.

Handling Merge Conflicts

Handling Merge Conflicts

This is how you know there was a merge conflict:

COMMAND LINE PIC

This is what a merge conflict will look like in your code:

```
TODO: put a picture here
```

Handling Merge Conflicts

Go to your `develop` branch.

Pull `develop`.

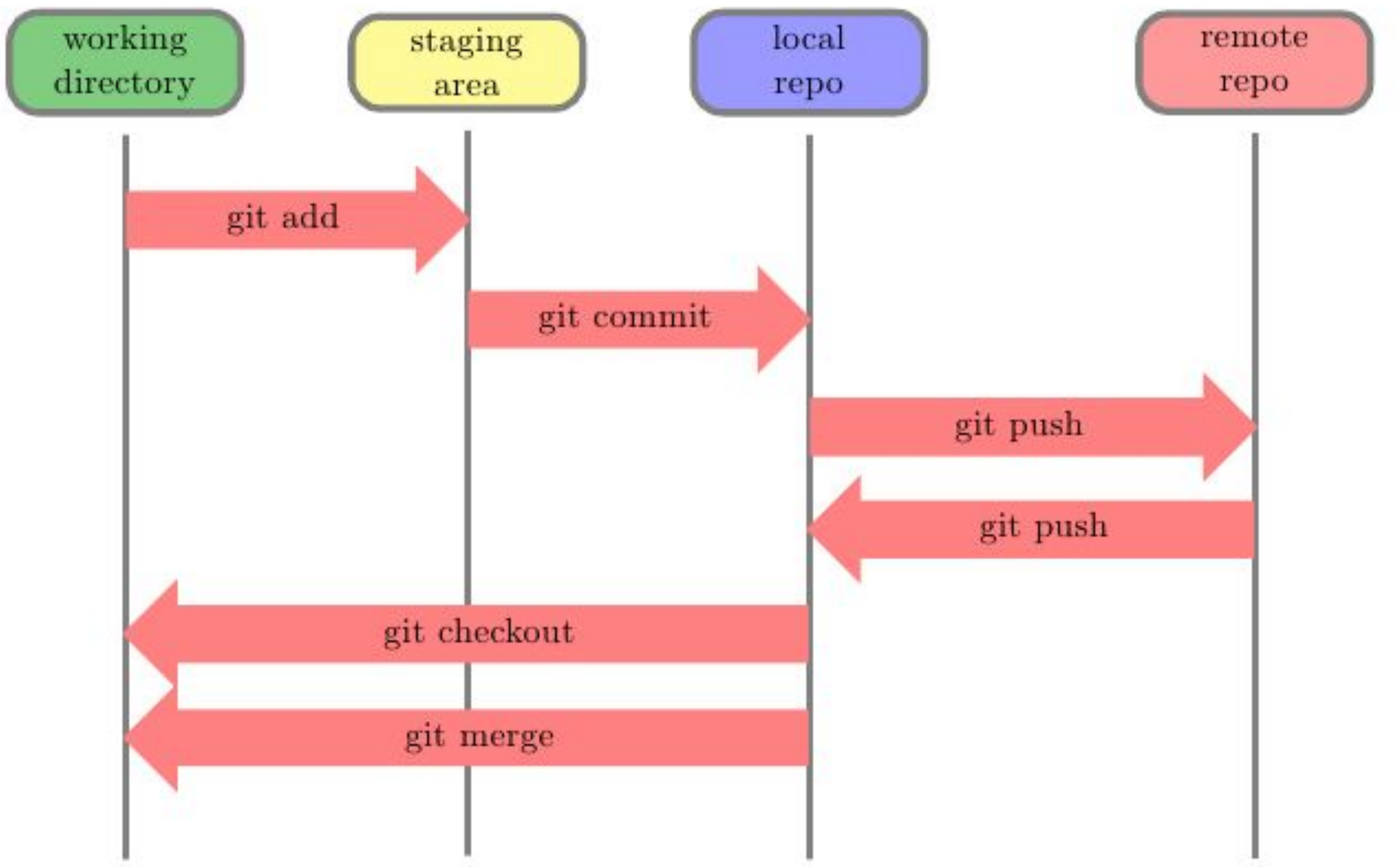
Go to the branch for your second task.

Merge `develop` into your branch.

Merge your branch into `develop`.

The Git Model

The Git Model



A remote repository is the actual main copy of the repository that exists wherever the repo is hosted (GitHub, in this instance).

ATOMIC OBJECT

Sources

Git Workflow Diagram:

https://tex.stackexchange.com/questions/70320/workflow-diagram?rq=1

Sample HTML Page:

https://html5-templates.com/preview/bootstrap-scrolling-sticky-menu.html

ATOMIC OBJECT