



ICT 3020 (2020/21)

Fundamentals of Software Engineering

Module 10: Software Configuration Management

Lighton Phiri <lighton.phiri@unza.zm>
Department of Library & Information Science
University of Zambia
<https://bit.ly/3u15u5V>

Lecture Outline

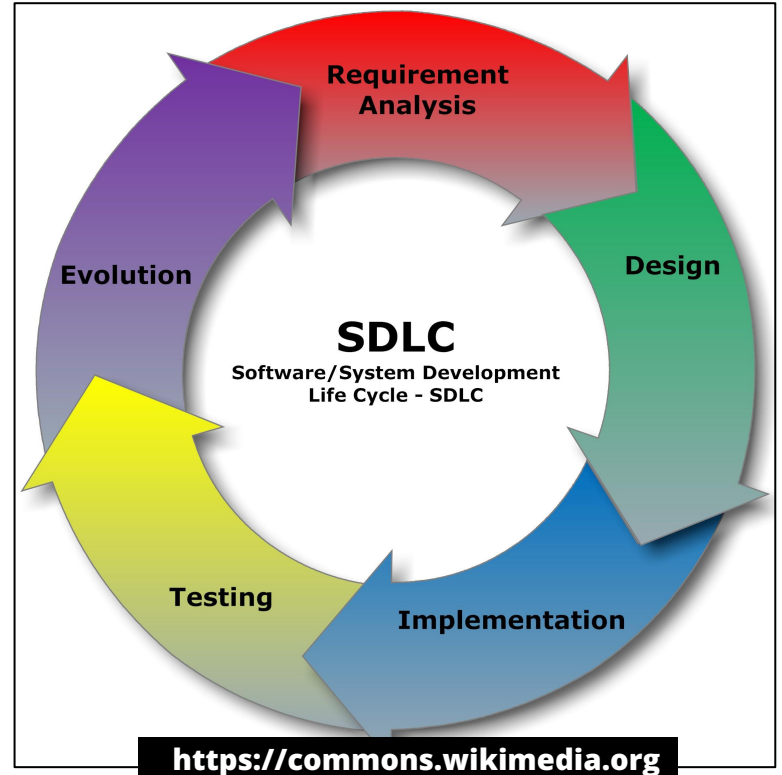
- Introduction
- Illustrating Distributed Version Control
- Software Configuration Management
- Version Control Systems
- Distributed Version Control
- Distributed Version Control Using Git

Lecture Outline

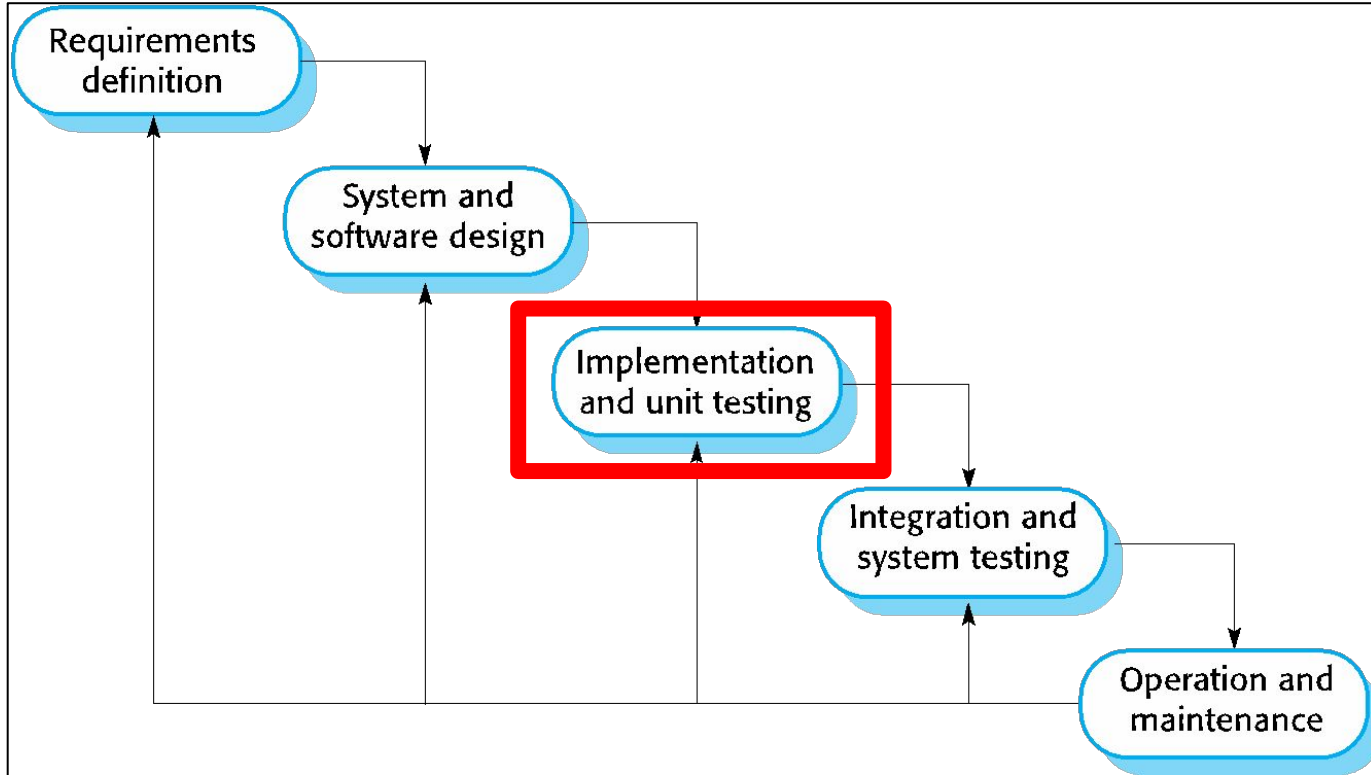
- Introduction
- Illustrating Distributed Version Control
- Software Configuration Management
- Version Control Systems
- Distributed Version Control
- Distributed Version Control Using Git

Introduction (1/4)

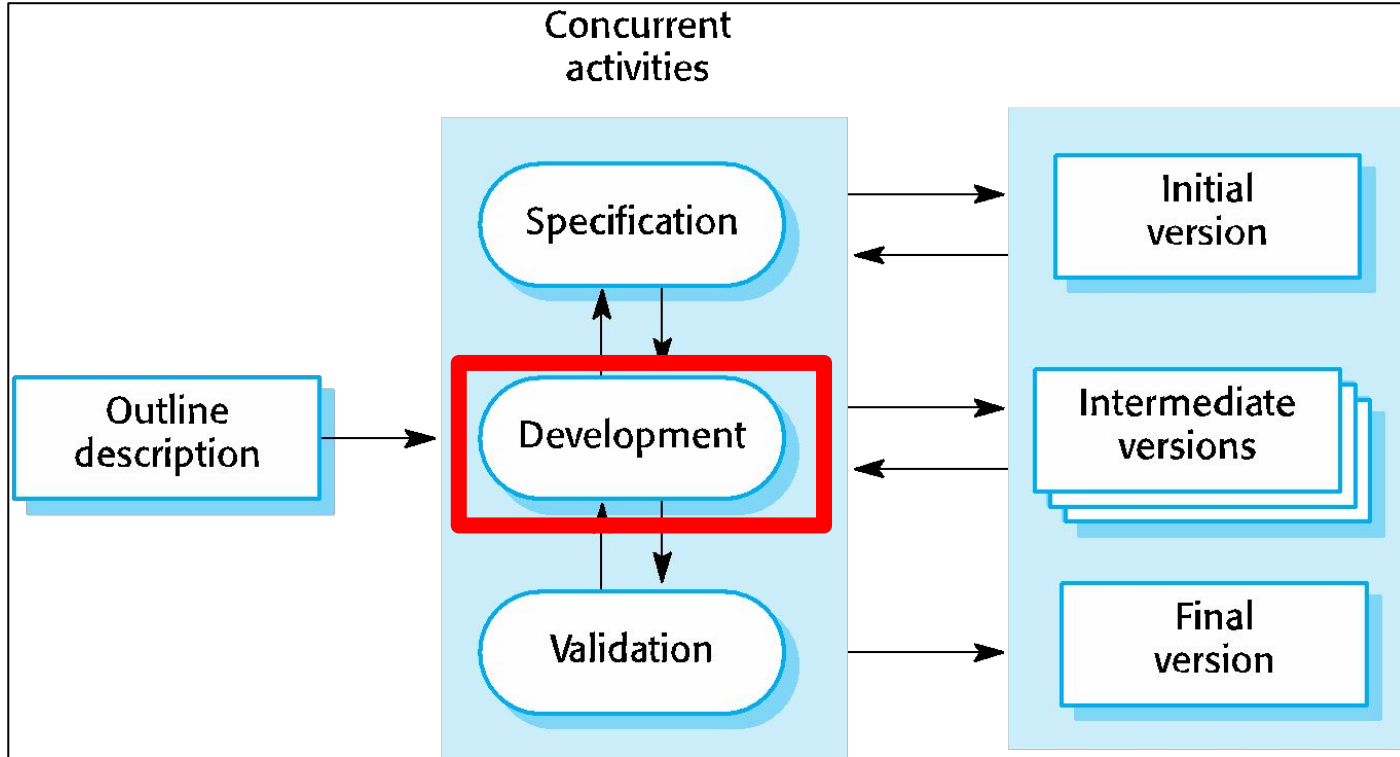
- Real software processes are interleaved sequences of technical, collaborative and managerial activities with the overall goal of specifying, designing, implementing and testing a software system.








Introduction (2/4)




Introduction (3/4)



Introduction (4/4)

 .gitignore	Edited the login page and added the google auth2.0	last month
 README.md	Initial commit	7 months ago
 app.js	Edited the login page and added the google auth2.0	last month
 package-lock.json	Edited the login page and added the google auth2.0	last month
 package.json	Edited the login page and added the google auth2.0	last month

README.md 

About smartirdeposit




A CRUD application for depositing Electronic Theses and Dissertaions produced at The University of Zambia

No releases published
[Create a new release](#)


Packages




No packages published
[Publish your first package](#)

Contributors 3

-  **lightonphiri** Lighton Phiri
-  **Victor-Mwelwa** Victor Mwelwa
-  **ElijahChileshe01**






Languages




-  **EJS** 58.1%
-  **JavaScript** 37.6%
-  **CSS** 4.3%

<https://github.com/DataLab-UNZA/smartirdeposit>

Introduction (4/4)

 .gitignore	Edited the login page and added the google auth2.0	last month
 README.md	Initial commit	7 months ago
 app.js	Edited the login page and added the google auth2.0	last month
 package-lock.json	Edited the login page and added the google auth2.0	last month
 package.json	Edited the login page and added the google auth2.0	last month

README.md 

About smartirdeposit




A CRUD application for depositing Electronic Theses and Dissertaions produced at The University of Zambia

No releases published
[Create a new release](#)


Packages




No packages published
[Publish your first package](#)

Contributors 3

-  **lightonphiri** Lighton Phiri
-  **Victor-Mwelwa** Victor Mwelwa
-  **ElijahChileshe01**

Languages



-  **EJS** 58.1%
-  **JavaScript** 37.6%
-  **CSS** 4.3%

<https://github.com/DataLab-UNZA/smartirdeposit>

Introduction (5/4)

The screenshot shows the GitHub repository page for DSpace/DSpace. At the top, the repository name is displayed as 'DSpace / DSpace'. To the right, there are statistics for 'Used by' (263), 'Watch' (105), 'Unstar' (494), and 'Fork' (943). Below this, navigation tabs include 'Code', 'Pull requests' (274), 'Actions', 'Projects' (0), 'Security', and 'Insights'. The repository description is '(Official) The DSpace digital asset management system that powers your Institutional Repository' with a link to the wiki. Below the description are tags for 'dspace', 'repository', 'java', 'open-access', and 'open-source'. A summary bar shows '12,200 commits', '18 branches', '0 packages', '96 releases', and '134 contributors'. Below this, there are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. The commit history shows a recent merge by tdonohue and several other commits related to formatting, API updates, and development iterations.

DSpace / DSpace

Used by 263 Watch 105 Unstar 494 Fork 943

Code Pull requests 274 Actions Projects 0 Security Insights

(Official) The DSpace digital asset management system that powers your Institutional Repository <https://wiki.duraspace.org/display/DS...>

dspace repository java open-access open-source

12,200 commits 18 branches 0 packages 96 releases 134 contributors View license

Branch: master New pull request Create new file Upload files Find file Clone or download

tdonohue Merge pull request #2737 from tdonohue/DS-4476 Latest commit 1c514b8 1 hour ago

- .github Formatting changes & grammar correction 12 days ago
- dspace-api Merge pull request #2707 from Micheleboychuk/DS-4122_comcol_permissions 7 days ago
- dspace-oai Fix new checkstyle failures for "CustomImportOrder: Extra separation ..." 26 days ago
- dspace-rdf [maven-release-plugin] prepare for next development iteration last month

<https://github.com/DSpace/DSpace>

Introduction (5/4)

Contributing

DSpace is a community built and supported project. We do not have a centralized development or support team, but have a dedicated group of volunteers who help us improve the software, documentation, resources, etc.

We welcome contributions of any type. Here's a few basic guides that provide suggestions for contributing to DSpace:

- [How to Contribute to DSpace](#): How to contribute in general (via code, documentation, bug reports, expertise, etc)
- [Code Contribution Guidelines](#): How to give back code or contribute features, bug fixes, etc.
- [DSpace Community Advisory Team \(DCAT\)](#): If you are not a developer, we also have an interest group specifically for repository managers. The DCAT group meets virtually, once a month, and sends open invitations to join their meetings via the [DCAT mailing list](#).

We also encourage GitHub Pull Requests (PRs) at any time. Please see our [Development with Git](#) guide for more info.

In addition, a listing of all known contributors to DSpace software can be found online at:

<https://wiki.duraspace.org/display/DSPACE/DSpaceContributors>

Getting Help

<https://github.com/DSpace/DSpace>

Introduction (5/4)

Code Style Guide

Created by Tim Donohue, last modified by Marsa Haoua on Sep 03, 2018

- [DSpace Java Style Guide \(Adopted 2018, for DSpace 7.x and above\)](#)
- [Old DSpace Java Style Guide \(for versions 6.x and prior\)](#)
- [Resources](#)
 - [IDE Support](#)
 - [IntelliJ IDEA](#)
 - [Eclipse](#)
 - [NetBeans](#)
 - [Checking code style from commandline](#)
 - [Fixing existing code / PRs](#)
 - [Use IntelliJ to perform bulk cleanup](#)

DSpace Java Style Guide (Adopted 2018, for DSpace 7.x and above)

① [DSpace Java Style Guide is enforced on "master" branch](#)

As of Feb 2018, the below DSpace Java Style Guide is enforced on all Pull Requests to the "master" branch. Therefore, if a Pull Request to the "master" branch does not align with the below Style Guide, it will fail the build process within Travis CI.

- The enforcement is handled by [Checkstyle](#) (maven-checkstyle-plugin) using the Checkstyle configuration file in the root source directory:
<https://github.com/DSpace/DSpace/blob/master/checkstyle.xml>
- If you would like to pre-check your Pull Request for any Code Style issues, you can do so by running: `mvn checkstyle:check`

<https://wiki.lyrasis.org/display/DSPACE/How+to+Contribute+to+DSpace>

Introduction (6/4)

The screenshot shows the Islandora 8 documentation page. The header is red with the Islandora 8 logo and name on the left, a search bar in the center, and the GitHub logo with 88 stars and 58 forks on the right. The left sidebar contains a navigation menu with the following items: Islandora 8, About, Concepts, Installation, Docker (ISLE), Ansible Playbook, Manual Installation, Introduction (highlighted in red), Preparing a LAPP Webserver, Installing Composer, Drush, and Drupal, Installing Tomcat and Cantaloupe, Installing Fedora, Syn, and Blazegraph, Installing Solr, and Installing Crayfish. The main content area has the title "Introduction" and a "Notice" box with a blue pencil icon. The notice text reads: "The manual installation guide is not intended to describe *the* Islandora 8 installation but rather *an* Islandora 8 installation. The server created using this guide is not hardened, will not be easily scalable, and the components may not be configured in a way you consider easy to work with. A production instance of Islandora 8 should be installed and maintained by a professional with an understanding of Linux and server administration." Below the notice, the text states: "This guide will contain generalized steps on installation and configuration of the various components, but will contain specific example commands for executing these steps on an Ubuntu 18.04 Server." The right sidebar contains a "Table of contents" with the following items: Some Prerequisite Knowledge, Conventions Used in This Guide, Chronological Organization, Replacement Variables, Bash Commands, Editing Files In Place, Adding or Replacing Files, and Troubleshooting. At the bottom right, there is a black box with the URL <https://islandora.github.io/documentation> in white text.

Introduction (6/4)

 Installing Composer, Drush, and Drupal 🔍 Search GitHub
☆ 88 🗨 58

- Islandora 8
 - About
 - Concepts >
 - Installation >
 - Docker (ISLE) >
 - Ansible Playbook
 - Manual Installation >
 - Introduction
 - Preparing a LAPP Webserver
 - Installing Composer, Drush, and Drupal**
 - Installing Tomcat and Cantaloupe
 - Installing Fedora, Syn, and Blazegraph
 - Installing Solr
 - Installing Crayfish
 - Installing Karaf and Alpaca

Drush 9 and Drupal 8

Clone drupal-project and run composer install

Before we can fully install Drupal, we're going to need to clone `drupal-project` and provision it using Composer. We're going to install it into the `/opt` directory:

```
# Start by giving Drupal somewhere to live. The Drupal project is installed to
# an existing, empty folder.
sudo mkdir /opt/drupal
sudo chown www-data:www-data /opt/drupal
sudo chmod 775 /opt/drupal
# Clone drupal-project and build it in our newly-created folder.
git clone https://github.com/Islandora/drupal-project.git
cd drupal-project
# Expect this to take a little while, as this is grabbing the entire
# requirements set for Drupal.
sudo -u www-data composer create-project drupal-composer/drupal-project:8.x-dev /opt
```

Table of contents

In this section, we will install:

- Composer 1.x
 - [Download and install Composer](#)
- Drush 9 and Drupal 8
 - Clone drupal-project and run composer install
 - Make Drush accessible in \$PATH
 - Make the new webroot accessible in Apache
 - Prepare the PostgreSQL database
 - Run the Drupal installer with Drush

Make Drush accessible in \$PATH

<https://islandora.github.io/documentation>

Introduction (6/4)

 Installing Composer, Drush, and Drupal 🔍 Search GitHub
☆ 88 🗨 58

- Islandora 8
 - About
 - Concepts >
 - Installation >
 - Docker (ISLE) >
 - Ansible Playbook
 - Manual Installation >
 - Introduction
 - Preparing a LAPP Webserver
 - Installing Composer, Drush, and Drupal**
 - Installing Tomcat and Cantaloupe
 - Installing Fedora, Syn, and Blazegraph
 - Installing Solr
 - Installing Crayfish
 - Installing Karaf and Alpaca

Drush 9 and Drupal 8

Clone drupal-project and run composer install

Before we can fully install Drupal, we're going to need to clone `drupal-project` and provision it using Composer. We're going to install it into the `/opt` directory:

```
# Start by giving Drupal somewhere to live. The Drupal project is installed to
# an existing, empty folder.
sudo mkdir /opt/drupal
sudo chown www-data:www-data /opt/drupal
sudo chmod 775 /opt/drupal

# Clone drupal project and build it in our newly-created folder.
git clone https://github.com/Islandora/drupal-project.git
cd drupal-project
# Expect this to take a little while, as this is grabbing the entire
# requirements set for Drupal.
sudo -u www-data composer create-project drupal-composer/drupal-project:8.x-dev /opt
```

Table of contents

In this section, we will install:

- Composer 1.x
 - [Download and install Composer](#)
- Drush 9 and Drupal 8
 - Clone drupal-project and run composer install
 - Make Drush accessible in \$PATH
 - Make the new webroot accessible in Apache
 - Prepare the PostgreSQL database
 - Run the Drupal installer with Drush

Make Drush accessible in \$PATH

<https://islandora.github.io/documentation>

Introduction (6/4)

Installing Crayfish

Search

GitHub 88 58

Islandora 8

- About
- Concepts >
- Installation >
 - Docker (ISLE) >
 - Ansible Playbook
 - Manual Installation >
 - Introduction
 - Preparing a LAPP Webserver
 - Installing Composer, Drush, and Drupal
 - Installing Tomcat and Cantaloupe
 - Installing Fedora, Syn, and Blazegraph
 - Installing Solr
 - Installing Crayfish
 - Installing Karaf and Alpaca
 - Configuring Drupal
 - Installing Modules
 - Tutorials >

• As always, this should be a secure password of some kind, and not this default.

Cloning and Installing Crayfish

We're going to clone Crayfish to `/opt`, and individually run `composer install` against each of the microservice subdirectories.

```
cd /opt
sudo git clone https://github.com/Islandora/Crayfish.git crayfish
sudo chown -R www-data:www-data crayfish
sudo -u www-data composer install -d crayfish/Gemini
sudo -u www-data composer install -d crayfish/Homarus
sudo -u www-data composer install -d crayfish/Houdini
sudo -u www-data composer install -d crayfish/Hypercube
sudo -u www-data composer install -d crayfish/Milliner
sudo -u www-data composer install -d crayfish/Recast
```

Preparing Logging

Not much needs to happen here; Crayfish opts for a simple logging approach, with one `.log` file for each component. We'll create a folder where each logfile can live.

```
sudo mkdir /var/log/islandora
sudo chown www-data:www-data /var/log/islandora
```

Table of contents

In this section, we will install:

- Crayfish 1.1
 - Installing Prerequisites
 - Preparing a Gemini Database
 - Cloning and Installing Crayfish
 - Preparing Logging
 - Configuring Crayfish Components
 - Gemini
 - Homarus (Audio/Video derivatives)
 - Houdini (Image derivatives)
 - Hypercube (OCR)
 - Milliner (Fedora indexing)
 - Recast (Drupal to Fedora URI re-writing)
 - Installing the Gemini Database
 - Creating Apache Configurations for Crayfish Components
 - Enabling Each Crayfish

<https://islandora.github.io/documentation>

Introduction (6/4)

Installing Crayfish

Search

GitHub 88 58

- As always, this should be a secure password of some kind, and not this default.

Cloning and Installing Crayfish

We're going to clone Crayfish to `/opt`, and individually run `composer install` against each of the microservice subdirectories.

```
cd /opt
sudo git clone https://github.com/Islandora/Crayfish.git crayfish
sudo chown -R www-data:www-data crayfish
sudo -u www-data composer install -d crayfish/Gemini
sudo -u www-data composer install -d crayfish/Homarus
sudo -u www-data composer install -d crayfish/Houdini
sudo -u www-data composer install -d crayfish/Hypercube
sudo -u www-data composer install -d crayfish/Milliner
sudo -u www-data composer install -d crayfish/Recast
```

Preparing Logging

Not much needs to happen here; Crayfish opts for a simple logging approach, with one `.log` file for each component. We'll create a folder where each logfile can live.

```
sudo mkdir /var/log/islandora
sudo chown www-data:www-data /var/log/islandora
```

Table of contents

In this section, we will install:

- Crayfish 1.1
 - Installing Prerequisites
 - Preparing a Gemini Database
 - Cloning and Installing Crayfish
 - Preparing Logging
 - Configuring Crayfish Components
 - Gemini
 - Homarus (Audio/Video derivatives)
 - Houdini (Image derivatives)
 - Hypercube (OCR)
 - Milliner (Fedora indexing)
 - Recast (Drupal to Fedora URI re-writing)
 - Installing the Gemini Database
 - Creating Apache Configurations for Crayfish Components
 - Enabling Each Crayfish

<https://islandora.github.io/documentation>

Lecture Outline

- **Introduction**
- **Illustrating Distributed Version Control**
 - Git for Distributed Version Control
 - Github
 - Git Configuration
 - Sample “Git -> Github” Workflow
- **Software Configuration Management**
- **Version Control Systems**
- **Distributed Version Control**
- **Distributed Version Control Using Git**

Git Download and Installation (1/3)

The screenshot shows the Git website's Downloads page. At the top left is the Git logo and the tagline "--distributed-is-the-new-centralized". A search bar is located at the top right. On the left side, there is a navigation menu with links for "About", "Documentation", "Downloads" (highlighted in red), "GUI Clients", "Logos", and "Community". Below the menu is a promotional text box for the "Pro Git book". The main content area features a "Downloads" section with icons for Mac OS X, Windows, and Linux/Unix. A monitor graphic displays the latest source release "2.28.0" and a "Downloads for Linux" button. Below the monitor, there are sections for "GUI Clients" and "Logos", each with a "View" link. A black banner at the bottom right contains the URL "https://git-scm.com".

git --distributed-is-the-new-centralized

Search entire site...

About
Documentation
Downloads
GUI Clients
Logos
Community

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

Downloads

Mac OS X Windows
Linux/Unix

Latest source Release
2.28.0
[Release Notes \(2020-07-27\)](#)
Downloads for Linux

Older releases are available and the [Git source repository](#) is on GitHub.

GUI Clients

Git comes with built-in GUI tools ([git-gui](#), [gitk](#)), but there are several third-party tools for users looking for a platform-specific experience.
[View GUI Clients](#) →

Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.
[View Logos](#) →

<https://git-scm.com>

Git Download and Installation (2/3)

```
lightonphiri@lightonphiri-ThinkPad-T540p:~$ sudo apt-get install git
[sudo] password for lightonphiri:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libunity-gtk3-parser0 linux-headers-5.4.0-39 linux-headers-5.4.0-39-generic linux-image-5.4.0-39-generic linux-modules-5.4.0-39-generic
  linux-modules-extra-5.4.0-39-generic
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 234 not upgraded.
Need to get 5,464 kB of archives.
After this operation, 38.4 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://zm.archive.ubuntu.com/ubuntu focal/main amd64 liberror-perl all 0.17029-1 [26.5 kB]
Get:2 http://zm.archive.ubuntu.com/ubuntu focal/main amd64 git-man all 1:2.25.1-1ubuntu3 [884 kB]
Get:3 http://zm.archive.ubuntu.com/ubuntu focal/main amd64 git amd64 1:2.25.1-1ubuntu3 [4,554 kB]
Fetched 5,464 kB in 22s (253 kB/s)
Selecting previously unselected package liberror-perl.
[Reading database ... 90%
```

+ Shell

Ubuntu/Debian | sudo apt-get install git

Git Download and Installation (3/3)

```
lightonphiri@lightonphiri-Lenovo-ideapad-320-15IKB:~$ git --help
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
      [--exec-path[=<path>]] [--html-path] [--man-path] [--info-p
ath]
      [-p | --paginate | -P | --no-pager] [--no-replace-objects]
[--bare]
      [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>
]
      <command> [<args>]
```

These are common Git commands used in various situations:

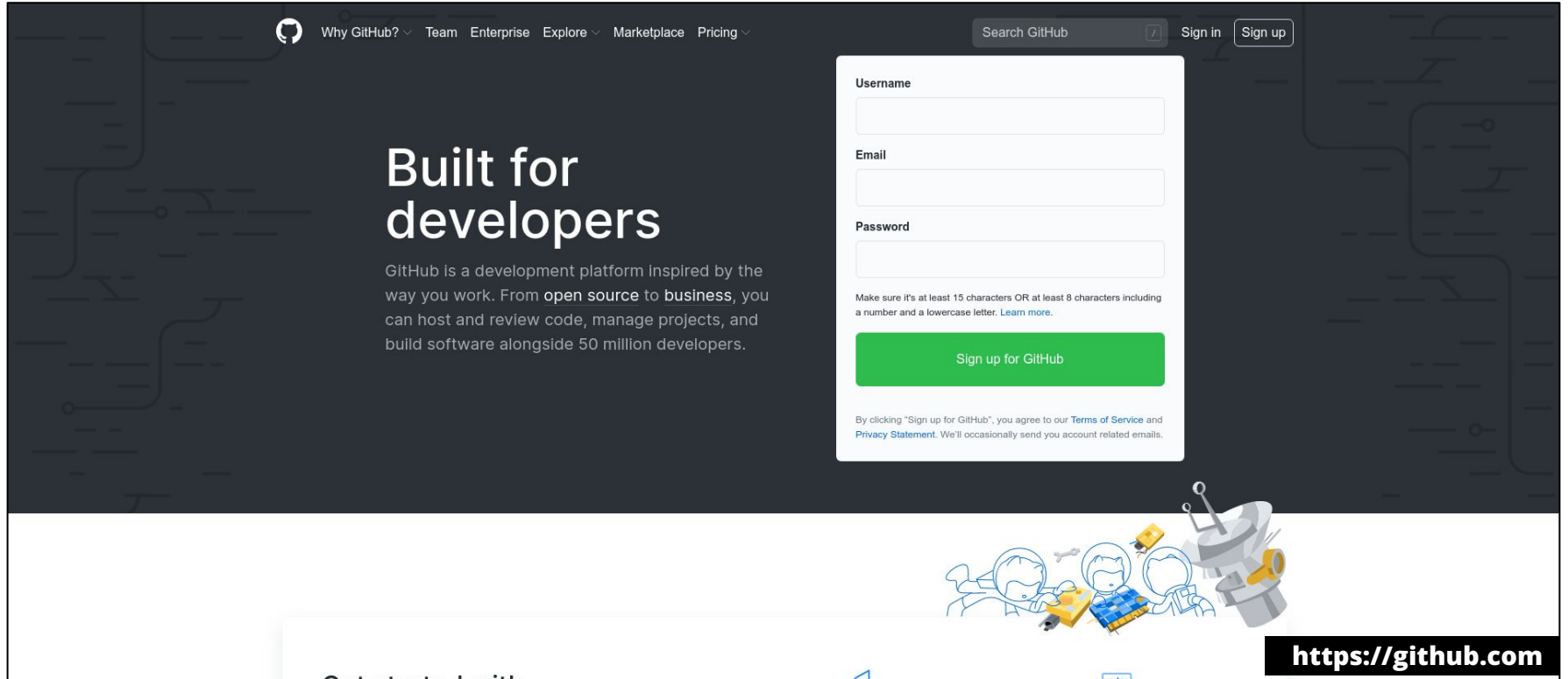
```
start a working area (see also: git help tutorial)
  clone          Clone a repository into a new directory
  init           Create an empty Git repository or reinitialize an
existing one
```

Ubuntu/Debian | git --help

Lecture Outline

- **Introduction**
- **Illustrating Distributed Version Control**
 - Git for Distributed Version Control
 - Github
 - Git Configuration
 - Sample “Git -> Github” Workflow
- **Software Configuration Management**
- **Version Control Systems**
- **Distributed Version Control**
- **Distributed Version Control Using Git**

GitHub Account Registration (1/5)

A screenshot of the GitHub registration page. The page has a dark background with a faint circuit-like pattern. At the top, there is a navigation bar with links: 'Why GitHub?', 'Team', 'Enterprise', 'Explore', 'Marketplace', and 'Pricing'. On the right side of the navigation bar, there is a search bar labeled 'Search GitHub', a 'Sign in' button, and a 'Sign up' button. The main content area features the text 'Built for developers' in a large, white font. Below this, a paragraph describes GitHub as a development platform. To the right, there is a white registration form with three input fields: 'Username', 'Email', and 'Password'. Below the 'Password' field, there is a note about password requirements and a link to 'Learn more'. A green button labeled 'Sign up for GitHub' is positioned below the form. At the bottom of the form, there is a disclaimer about the Terms of Service and Privacy Statement. In the bottom right corner of the page, there is a black box with the URL 'https://github.com' in white text. At the bottom of the page, there is a white banner with a cartoon illustration of three people working on a computer and a satellite dish, and the date 'October 3, 2023' on the left and 'ICT 3020 (2022/23) M10 - 22' on the right.

GitHub Account Registration (2/5)

Back to GitHub.com


GitHub Support Contact GitHub

Education

Students Teachers Schools Events [Get benefits](#)


Real-world tools, engaged students

GitHub Education helps students, teachers, and schools access the tools and events they need to shape the next generation of software development.




GitHub Student Developer Pack

The best developer tools, free for students




GitHub Campus Experts

Training to enrich the technology community at your school



GitHub Teacher Toolbox

The best developer tools for teaching, free for academic use



GitHub Campus Advisors

Teacher training to master GitHub and GitHub

<https://education.github.com>

GitHub Account Registration (3/5)

Back to GitHub.com GitHub Support Contact GitHub


Education Students Teachers Schools Events [Get benefits](#)

[Home](#) / [Students](#)


With GitHub Education, your work will speak for itself.

Build your portfolio, grow your network, and level up your skills.

[Get benefits for students](#)



GitHub Student Developer Pack



GitHub Campus Expert

<https://education.github.com>

Github Account Registration (4/5)

The screenshot shows the GitHub Education website. At the top, there is a dark navigation bar with "Back to GitHub.com" on the left and "GitHub Support" and "Contact GitHub" on the right. Below this is a white header with the GitHub logo and "Education" on the left, and navigation links for "Students", "Teachers", "Schools", "Events", and a "Get benefits" button on the right. The main content area has a breadcrumb trail: "Home / Students / GitHub Student Developer Pack". The title "GitHub Student Developer Pack" is prominently displayed, followed by a blue "Get your pack" button. A paragraph of text explains the pack's purpose: "Learn to ship software like a pro. There's no substitute for hands-on experience. But for most students, real world tools can be cost-prohibitive. That's why we created the GitHub Student Developer Pack with some of our partners and friends: to give students free access to the best developer tools in one place so they can learn by doing." Below this is a social sharing section with the text "Love the pack? Spread the word" and a "Tweet" button. At the bottom, there is a "Browse by category:" section with various tags like "All", "Cloud", "Design", "Developer tools", "Domains", "Game development", "Infrastructure & APIs", "Internet of Things", "Learn", "Marketing", "Mobile", "Productivity", and "Security & analytics". A black banner at the bottom right contains the URL "https://education.github.com".

Back to GitHub.com GitHub Support Contact GitHub

Education Students Teachers Schools Events [Get benefits](#)

Home / Students / GitHub Student Developer Pack

GitHub Student Developer Pack [Get your pack](#)

Learn to ship software like a pro. There's no substitute for hands-on experience. But for most students, real world tools can be cost-prohibitive. That's why we created the GitHub Student Developer Pack with some of our partners and friends: to give students free access to the best developer tools in one place so they can learn by doing.

Love the pack? Spread the word [Tweet](#)

Browse by category:

- All
- Cloud
- Design
- Developer tools
- Domains
- Game development
- Infrastructure & APIs
- Internet of Things
- Learn
- Marketing
- Mobile
- Productivity
- Security & analytics

<https://education.github.com>

Github Account Registration (5/5)

The screenshot shows the GitHub repository page for 'lightonphiri / etd_autoclassifier'. At the top, there are navigation tabs for Code, Issues, Pull requests (9), Actions, Projects, Wiki, Security (11), Insights, and Settings. A yellow alert banner states: 'We found potential security vulnerabilities in your dependencies. Only the owner of this repository can see this message.' with a 'See Dependabot alerts' button. Below the alert, there are repository statistics: 'master' branch, '10 branches', and '1 tag'. Action buttons include 'Go to file', 'Add file', and 'Code'. A commit history table is visible, showing a commit by 'lightonphiri' with the message 'Added documentation' and a list of files: docs, models, Procfile, README.md, app.py, nltk.txt, and requirements.txt. The right sidebar contains sections for 'About' (describing it as a Flask API), 'Releases' (v1.0: Added documentation), 'Packages' (No packages published), and 'Languages'. A black banner at the bottom right contains the URL 'https://github.com'.

lightonphiri / etd_autoclassifier

<> Code Issues Pull requests 9 Actions Projects Wiki Security 11 Insights Settings

⚠ We found potential security vulnerabilities in your dependencies.
Only the owner of this repository can see this message. [See Dependabot alerts](#)

🔗 master 10 branches 1 tag [Go to file](#) [Add file](#) [Code](#)

lightonphiri Added documentation ... b2d4071 on Jan 2 10 commits

docs	Added documentation	9 months ago
models	Added saved ETD type models	9 months ago
Procfile	Heroku configuration changes	9 months ago
README.md	Added documentation	9 months ago
app.py	Project configuration changes	9 months ago
nltk.txt	Fixing Heroku deployment errors	9 months ago
requirements.txt	Heroku configuration changes	9 months ago

About [Settings](#)
A Flask API that processes POST requests to classify ETDs
[Readme](#)

Releases 1
[v1.0: Added documentation](#) [Latest](#)
on Jun 28

Packages
No packages published
[Publish your first package](#)

Languages

<https://github.com>

Lecture Outline

- **Introduction**
- **Illustrating Distributed Version Control**
 - Git for Distributed Version Control
 - Github
 - Git Configuration
 - Sample “Git -> Github” Workflow
- **Software Configuration Management**
- **Version Control Systems**
- **Distributed Version Control**
- **Distributed Version Control Using Git**

Sample Git -> Github Workflow

- **Creating a new repository**
 - Configuring Git for initial use
 - Initialising a new repository
- **Cloning an existing repository**
- **Working with an existing repository**
 - Committing changes
 - Pushing changes
 - Pulling changes

Creating a New Repository

- **In-class example [...]**
 - Git init
 - Git config
 - Github synchronisation
 - Bitbucket synchronisation

Cloning an Existing Repository

DataLab-UNZA / **smartirdeposit** lines 6.6k Private

Unwatch 1 Unstar 1

<> Code Issues Pull requests Actions Projects Security Insights Settings

master 2 branches 0 tags

Go to file Add file Code

ElijahChileshe01 edited the register page and added the google auth2.0 404c048 on 4 Aug 22 commits

config	Edited the login page and added the google auth2.0	last month
models	Edited the login page and added the google auth2.0	last month
public	updated the ingestion module	2 months ago
routes	edited the register page and added the google auth2.0	last month
views	edited the register page and added the google auth2.0	last month
.gitignore	Edited the login page and added the google auth2.0	last month
README.md	Initial commit	7 months ago
app.js	Edited the login page and added the google auth2.0	last month
package-lock.json	Edited the login page and added the google au	

About

A CRUD application for depositing Electronic Theses and Dissertations produced at The University of Zambia.

datalab.unza.zm

Readme






Releases


No releases published
[Create a new release](#)

Packages

<https://github.com/DataLab-UNZA/smartirdeposit>

Working With an Existing Repository

 .gitignore	Edited the login page and added the google auth2.0	last month
 README.md	Initial commit	7 months ago
 app.js	Edited the login page and added the google auth2.0	last month
 package-lock.json	Edited the login page and added the google auth2.0	last month
 package.json	Edited the login page and added the google auth2.0	last month

README.md 

About smartirdeposit




A CRUD application for depositing Electronic Theses and Dissertaions produced at The University of Zambia

No releases published
[Create a new release](#)


Packages




No packages published
[Publish your first package](#)

Contributors 3

-  **lightonphiri** Lighton Phiri
-  **Victor-Mwelwa** Victor Mwelwa
-  **ElijahChileshe01**

Languages



-  **EJS** 58.1%
-  **JavaScript** 37.6%
-  **CSS** 4.3%

<https://github.com/DataLab-UNZA/smartirdeposit>

Lecture Outline

- Introduction
- Illustrating Distributed Version Control
- Software Configuration Management
 - Software Configuration Management
- Version Control Systems
- Distributed Version Control
- Distributed Version Control Using Git

Significance of Configuration Management (1/2)

- **Software projects involve multiple people working on deliverables that are constantly changing**
- **Multiple individuals could potentially be working on different versions of the same deliverable**
 - Project management documents, e.g. SRS document
 - Source code ***
 - Software releases
- **Software configuration management manages the evolution of software systems being developed and deployed**


Significance of Configuration Management (2/2)

- **Software projects involve multiple people working on deliverables that are constantly changing**
- **Multiple individuals could potentially be working on different versions of the same deliverable**
 - Project management documents, e.g. SRS document
 - Source code ***
 - Software releases
- **Software configuration management manages the evolution of software systems being developed and deployed**

Lecture Outline

- **Introduction**
- **Illustrating Distributed Version Control**
- **Software Configuration Management**
- **Version Control Systems**
 - Introduction
 - Centralised Version Control
 - Distributed Version Control
 - Tooling for Version Control
- **Distributed Version Control Using Git**
- **Markdown 101**

Version Control Systems (1/5)

 **git** --distributed-even-if-your-workflow-isnt

About

Documentation

- Reference
- Book**
- Videos
- External Links

Downloads

Community

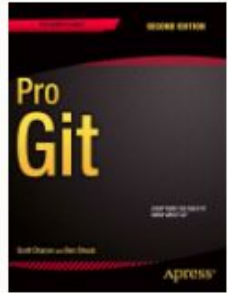
This book is available in [English](#).

Full translation available in

Book

The entire Pro Git book, written by Scott Chacon and Ben Straub and published by Apress, is available here. All content is licensed under the [Creative Commons Attribution Non Commercial Share Alike 3.0 license](#). Print versions of the book are available on [Amazon.com](#).

- 1. Getting Started**
 - 1.1 [About Version Control](#)
 - 1.2 [A Short History of Git](#)
 - 1.3 [What is Git?](#)
 - 1.4 [The Command Line](#)



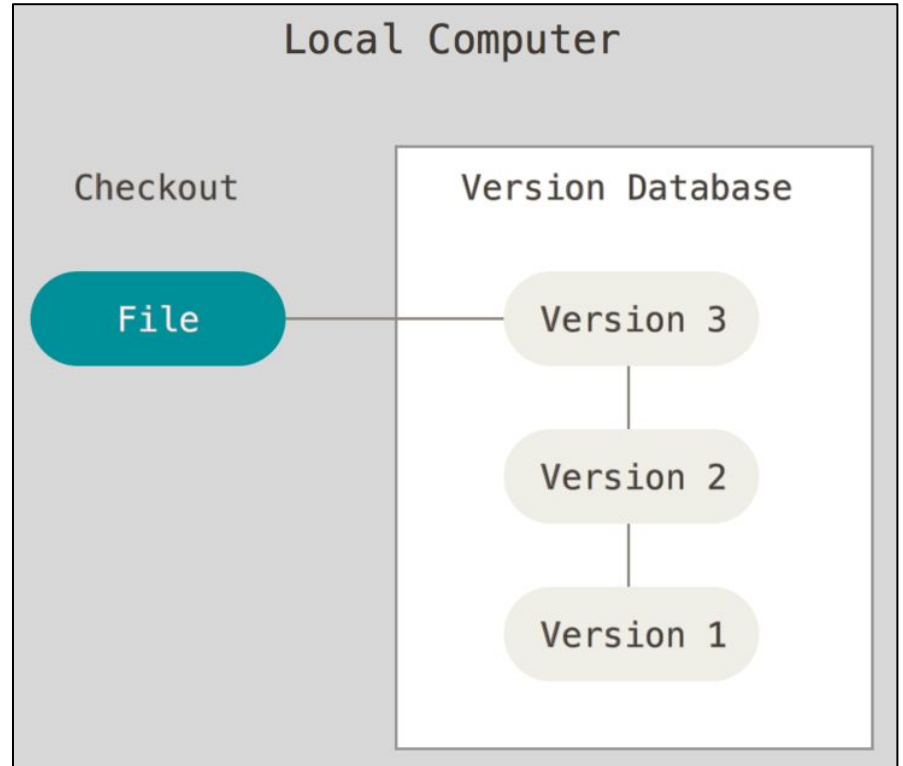
2nd Edition (2014)

Download Ebook

<https://git-scm.com/book>

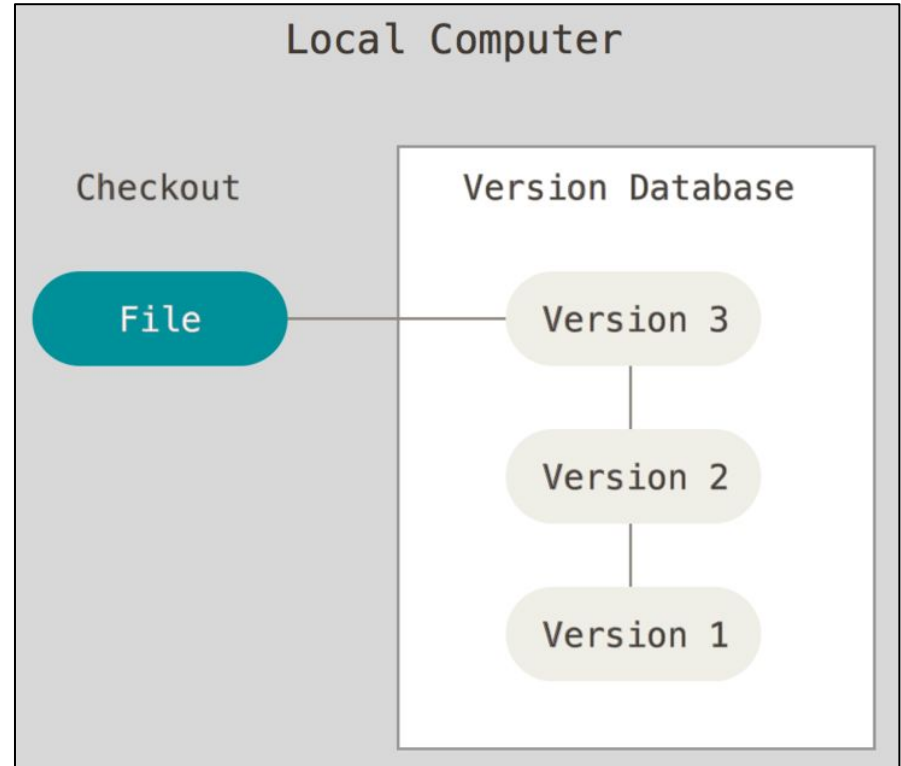
Version Control Systems (2/5)

- **Systems for keeping track of changes to a file or set of files over time**
 - In order to a record of specific versions later



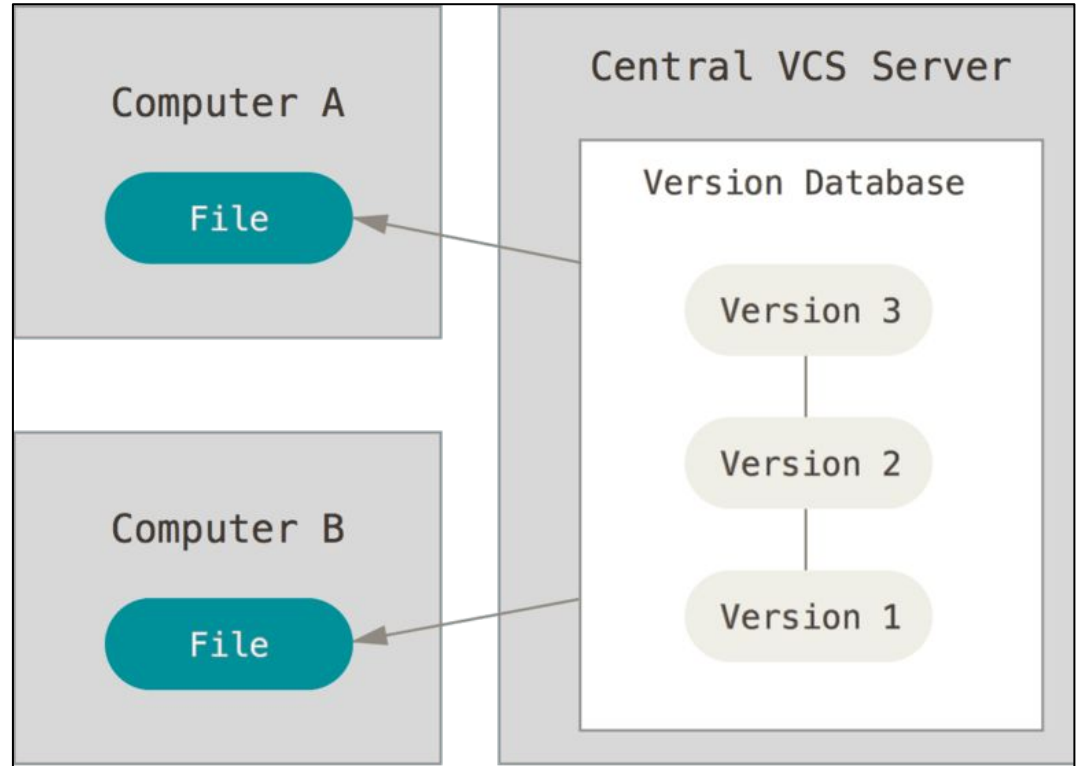
Version Control Systems (3/5)

- **Features of VCS**
 - Versioning much more than source code
 - Reliably keep versions around for as long as necessary
 - Individually versioning multiple project files
 - Revision history availability
 - Version comparison
 - Revert to older version



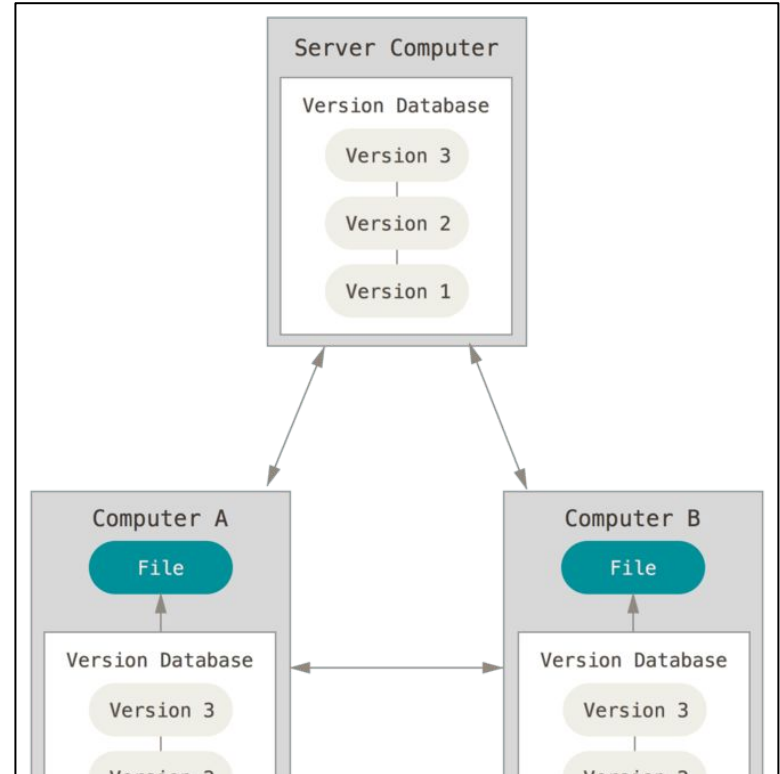
Version Control Systems (4/5)

- **Centralised version control systems make use of a single server**
 - Server contains all the versioned files
 - Clients that check out files from the server



Version Control Systems (5/5)

- **Distributed version control systems ensure clients mirror/replicate the central repository**
 - Clients fully mirror the repository as opposed to checking out the latest snapshot of the files



Lecture Outline

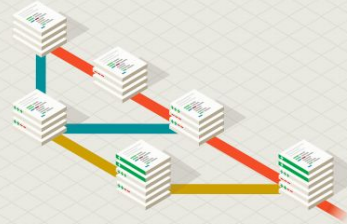
- **Introduction**
- **Illustrating Distributed Version Control**
- **Software Configuration Management**
- **Version Control Systems**
 - Introduction
 - Centralised Version Control
 - Distributed Version Control
 - Tooling for Version Control
- **Distributed Version Control Using Git**
- **Markdown 101**

Potential Version Control Software (1/4)



Git is a [free and open source](#) distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is [easy to learn](#) and has a [tiny footprint with lightning fast performance](#). It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like [cheap local branching](#), convenient [staging areas](#), and [multiple workflows](#).



About

The advantages of Git compared to other source control systems.



Documentation

Command reference pages, Pro Git book content, videos and other material.



Downloads

GUI clients and binary releases for all major platforms.



Community

Get involved! Bug reporting, mailing list, chat, development and more.



Pro Git by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).



Linux GUIs



Tarballs



Mae Build




Source Code

Companies & Projects Using Git

<https://git-scm.com>

Potential Version Control Software (2/4)



Not Logged In
Login
New User

This Page
Language
Clean Reload
Printer Version


Search

in **Projects**

Hosted Projects
Hosting requirements
Register New Project
Full List
Contributors Wanted
Statistics

Site Help
User Docs: Cookbook
User Docs: In Depth Guide
Get Support
Contact Savannah
User Docs: FAQ

GNU Project
Help GNU
All GNU Packages
Dev Resources
License List
GNU Mirrors



FREE SOFTWARE FOUNDATION
Help us protect your freedom and the rights of computer users everywhere by becoming a member

Concurrent Versions System - Summary





Group [Main](#) [Homepage](#) [Download](#) [Mailing lists](#) [Source code](#) [Bugs](#) [Tasks](#) [Patches](#) [News](#)


This project is not part of the GNU Project.


CVS is a version control system, an important component of Source Configuration Management (SCM). Using it, you can record the history of sources files, and documents. It fills a similar role to the free software RCS, PRCS, and Aegis packages.

Please see the ["http://www.nongnu.org/cvs/\"](http://www.nongnu.org/cvs/) project home page for more information.

Registration Date: Wed 06 Jul 2005 08:30:35 PM UTC
License: [GNU General Public License v2 or later](#)
Development Status: 5 - Production/Stable

Quick Overview
 Project Homepage
 Download Area
 Project Memberlist (10 members)
 Project Member GPG Keyring


Communication Tools
 Mailing Lists (7 public mailing lists)

Development Tools
 CVS Repository

Membership Info
Project Admins: - Larry Jones - Mark D. Baushke - Derek Robert Price 10 active members [View Members]



Group Identification
Id: #7788 System Name: cvs Name: Concurrent Versions System Group Type: non-GNU software and documentation

Search in this Group
<input type="text"/> in Cookbook <input type="button" value="Search"/>

Latest News 
Stable CVS Version 1.11.23 Released! <i>posted by dprice, Thu 08 May 2008 11:56:21 AM UTC - 0 replies</i>
Stable CVS 1.11.23 has been released. Stable releases contain only bug fixes from previous versions of CVS. This version includes an efficiency fix that reduces checkouts of very old revisions from an O(n^2) operation to an O(n) one, as well as ...
[Read more]
Stable CVS Version 1.11.22 Released! <i>posted by dprice, Fri 09 Jun 2006 05:59:28 PM UTC - 0 replies</i>
Stable CVS 1.11.22 has been released. Stable releases contain only bug fixes from previous versions of CVS. This version fixes a serious client problem that could cause CVS to report files with conflicts as unmodified when talking to old servers ...

<https://www.nongnu.org/cvs>

Potential Version Control Software (3/4)



Apache® Subversion®
"Enterprise-class centralized version control for the masses"

Welcome to **subversion.apache.org**, the online home of the Apache® Subversion® software project. Subversion is an open source version control system. Founded in 2000 by CollabNet, Inc., the Subversion project and software have seen incredible success over the past decade. Subversion has enjoyed and continues to enjoy widespread adoption in both the open source arena and the corporate world.

Subversion is developed as a project of the [Apache Software Foundation](#), and as such is part of a rich community of developers and users. We're always in need of individuals with a wide range of skills, and we invite you to participate in the development of Apache Subversion. Here's [how to get started](#).

For helpful hints about how to get the most out of your visit to this site, see the [About This Site](#) section below.

Our Vision

Subversion exists to be universally recognized and adopted as an open-source, centralized version control system characterized by its reliability as a safe haven for valuable data; the simplicity of its model and usage; and its ability to support the needs of a wide variety of users and projects, from individuals to large-scale enterprise operations.

News

2020-05-27 — Apache Subversion 1.14.0 Released

We are pleased to announce the release of Apache Subversion 1.14.0. This is the most complete Subversion release to date, and we encourage users of Subversion to upgrade as soon as reasonable. Please see the [release announcement](#), [press release](#), and the [release notes](#) for more information about this release.

To get this release from the nearest mirror, please visit our [download page](#).


2020-02-27 — Celebrating 20 Years of Subversion

The very first CVS commit in what would become Apache Subversion was recorded on a very special Leap Day, 29 February 2000.

This year, celebrate with us as we mark Subversion's 20th Anniversary. Read the [Press Release](#) and [join our community](#) in writing the next chapter of the Subversion story.

2019-07-31 — Apache Subversion Security Advisory

Read the official Subversion documentation [online](#) ²¹

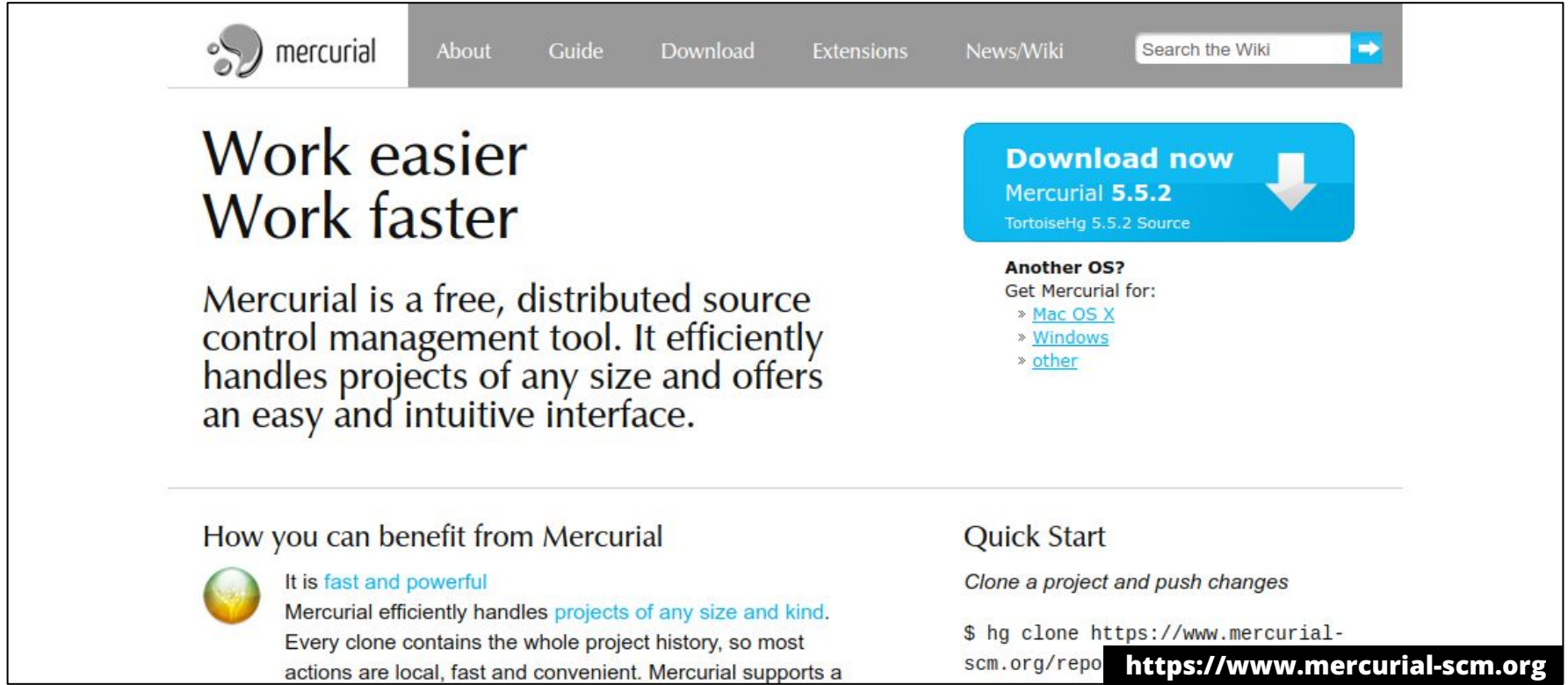


APACHE CON @Home
Sep 29 - Oct 1, 2020
Streaming live to your living room!

Search...

<https://subversion.apache.org>

Potential Version Control Software (4/4)



The screenshot shows the Mercurial website homepage. At the top, there is a navigation bar with the Mercurial logo and links for 'About', 'Guide', 'Download', 'Extensions', and 'News/Wiki'. A search bar is also present. The main content area features a large heading 'Work easier Work faster' and a paragraph describing Mercurial as a free, distributed source control management tool. A prominent blue button labeled 'Download now' with a downward arrow is visible, along with a list of links for 'Mac OS X', 'Windows', and 'other'. Below this, there are sections for 'How you can benefit from Mercurial' and 'Quick Start', including a terminal command and the website URL.

mercurial

About Guide Download Extensions News/Wiki Search the Wiki


Work easier Work faster

Mercurial is a free, distributed source control management tool. It efficiently handles projects of any size and offers an easy and intuitive interface.

Download now
Mercurial 5.5.2
TortoiseHg 5.5.2 Source

Another OS?
Get Mercurial for:
» [Mac OS X](#)
» [Windows](#)
» [other](#)

How you can benefit from Mercurial

 It is **fast and powerful**
Mercurial efficiently handles **projects of any size and kind**.
Every clone contains the whole project history, so most actions are local, fast and convenient. Mercurial supports a

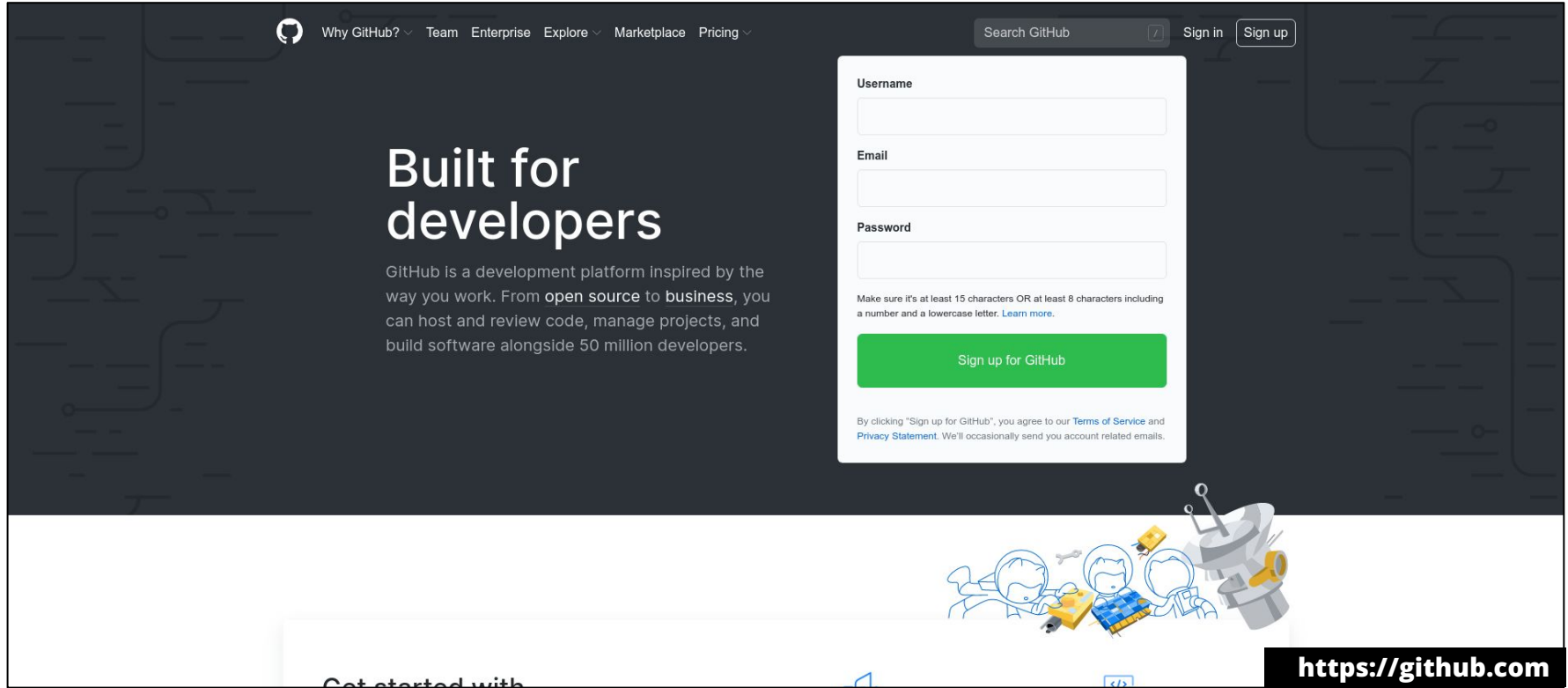
Quick Start

Clone a project and push changes

```
$ hg clone https://www.mercurial-scm.org/repo
```

<https://www.mercurial-scm.org>

Git Cloud-Based Hosting Services (1/3)



The image shows a screenshot of the GitHub website's sign-up page. The background is dark with a faint circuit-like pattern. On the left, the text "Built for developers" is prominently displayed in white. Below it, a paragraph describes GitHub as a development platform for open source and business. On the right, there is a white sign-up form with fields for Username, Email, and Password. A green "Sign up for GitHub" button is at the bottom of the form. Below the button, there is a small disclaimer about terms of service and privacy. At the bottom of the page, there is a white banner with a cartoon illustration of three astronauts and a satellite, and a black box containing the URL "https://github.com".

Why GitHub? Team Enterprise Explore Marketplace Pricing Search GitHub Sign in Sign up

Built for developers

GitHub is a development platform inspired by the way you work. From [open source](#) to [business](#), you can host and review code, manage projects, and build software alongside 50 million developers.

Username



Email

Password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails.

Get started with  

<https://github.com>

Git Cloud-Based Hosting Services (2/3)

ATLASSIAN
Bitbucket

Why Bitbucket ▾ Product Guide ▾ Self-Hosted Pricing

Log in [Get it free](#)

12+ new features for fewer meetings and less context switching. Get more time back to ship better code, faster. [Learn more.](#)

Built for professional teams

Bitbucket is more than just Git code management. Bitbucket gives teams one place to plan projects, collaborate on code, test, and deploy.

[Get it free](#)

Or host it yourself with [Bitbucket Data Center](#) →

bugfix/BUG-123 remove extra padding

bugfix/BUG-123 → master **OPEN**
Created 11 hours ago • Last updated one hour ago

+3 Approve Merge ...

I made a few changes to tidy up the code. Please let me know if all looks good! It should work on desktop browser and mobile browsers. I've run the tests and they pass locally. I'll check to make sure the pipelines tests pass as well.

```
js / core.js
@@ -17,22 +17,17 @@
...
19 19 # Only use spaces to indent your .yaml configuration.
20 20 # -----
21 21 # You can specify a custom docker image from Docker Hub at your build environment.
```

<https://bitbucket.org>

Git Cloud-Based Hosting Services (3/3)

The screenshot shows the GitHub interface for the repository 'lightonphiri / etd_autoclassifier'. At the top, there is a search bar and navigation links for Pull requests, Issues, Marketplace, and Explore. The repository name is displayed with options to Unwatch (1), Unstar (2), and Fork (0). Below the repository name, there are tabs for Code, Issues, Pull requests (9), Actions, Projects, Wiki, Security (11), Insights, and Settings. A yellow warning banner indicates 'We found potential security vulnerabilities in your dependencies.' with a 'See Dependabot alerts' button. The main content area shows the file tree with a commit by 'lightonphiri' on Jan 2, 10 commits ago. The files listed are docs, models, Procfile, README.md, app.py, nltk.bt, and requirements.txt. The right sidebar contains sections for About (A Flask API that processes POST requests to classify ETDs), Releases (v1.0: Added documentation), Packages (No packages published), and Languages (Python 100.0%). The README.md content is partially visible at the bottom, starting with 'About ETD Autoclassifier'.

Search or jump to... Pull requests Issues Marketplace Explore

lightonphiri / etd_autoclassifier Unwatch 1 Unstar 2 Fork 0

<> Code Issues Pull requests 9 Actions Projects Wiki Security 11 Insights Settings

⚠ We found potential security vulnerabilities in your dependencies.
Only the owner of this repository can see this message. See Dependabot alerts

master 10 branches 1 tag Go to file Add file Code

lightonphiri Added documentation b2d4071 on Jan 2 10 commits

docs	Added documentation	9 months ago
models	Added saved ETD type models	9 months ago
Procfile	Heroku configuration changes	9 months ago
README.md	Added documentation	9 months ago
app.py	Project configuration changes	9 months ago
nltk.bt	Fixing Heroku deployment errors	9 months ago
requirements.txt	Heroku configuration changes	9 months ago

README.md

About ETD Autoclassifier

A Flask API for classifying Electronic Theses and Dissertations prepared by students at The University of Zambia

About A Flask API that processes POST requests to classify ETDs

Releases 1 v1.0: Added documentation Latest on Jun 28

Packages No packages published Publish your first package

Languages Python 100.0%

<https://github.com>

Lecture Outline

- Introduction
- Illustrating Distributed Version Control
- Software Configuration Management
- Version Control Systems
- Distributed Version Control Using Git
 - Initial Configuration
 - Creating a New Repository
 - Cloning an Existing Repository
 - Committing, Pushing and Pulling
 - Branching and Merging
- Markdown 101

Git Initial Configuration

```
GIT-CONFIG(1)                               Git Manual                               GIT-CONFIG(1)

NAME
    git-config - Get and set repository or global options

SYNOPSIS
    git config [<file-option>] [--type=<type>] [--show-origin] [--s
how-scope] [-z|--null] name [value [value_regex]]
    git config [<file-option>] [--type=<type>] --add name value
    git config [<file-option>] [--type=<type>] --replace-all name v
alue [value_regex]
    git config [<file-option>] [--type=<type>] [--show-origin] [--s
how-scope] [-z|--null] --get name [value_regex]
    git config [<file-option>] [--type=<type>] [--show-origin] [--s
how-scope] [-z|--null] --get-all name [value_regex]
    git config [<file-option>] [--type=<type>]

Ubuntu/Debian | git config --help
```

Creating a New Git Repository

```
GIT-INIT(1)                               Git Manual                               GIT-INIT(1)

NAME
    git-init - Create an empty Git repository or reinitialize an
    existing one

SYNOPSIS
    git init [-q | --quiet] [--bare] [--template=<template_directory>]
    [--separate-git-dir <git dir>] [--object-format=<format>]
    [--shared[=<permissions>]] [directory]

DESCRIPTION
    This command creates an empty Git repository - basically a
    git directory with subdirectories for objects, refs, and config.

    Ubuntu/Debian | git init --help
```

Cloning an Existing Git Repository

GIT-CLONE(1)

Git Manual

GIT-CLONE(1)

NAME

`git-clone` - Clone a repository into a new directory

SYNOPSIS

```
git clone [--template=<template_directory>]  
[-l] [-s] [--no-hardlinks] [-q] [-n] [--bare] [--mirr  
or]  
[-o <name>] [-b <name>] [-u <upload-pack>] [--referen  
ce <repository>]  
[--dissociate] [--separate-git-dir <git dir>]  
[--depth <depth>] [--[no-]single-branch] [--no-tags]  
[--recurse-submodules[=<pathspec>]] [--[no-]shallow-s  
ubmodules]  
[--[no-]remote-submodules] [--jobs <r
```

Ubuntu/Debian | `git --help`

Git Repositories: Committing, Pushing and Pulling (1/3)

```
GIT-COMMIT(1)                               Git Manual                               GIT-COMMIT(1)
NAME
  git-commit - Record changes to the repository

SYNOPSIS
  git commit [-a | --interactive | --patch] [-s] [-v] [-u<mode>]
  [--amend]
  [--dry-run] [(-c | -C | --fixup | --squash) <commit>]
  [-F <file> | -m <msg>] [--reset-author] [--allow-empty-
  message] [--allow-empty-message] [--no-verify] [-e] [--author
  =<author>]
  [--date=<date>] [--cleanup=<mode>] [--[no-]status]
  [-i | -o] [--pathspec-from-file] Ubuntu/Debian | git commit --help
```

Git Repositories: Committing, Pushing and Pulling (2/3)

GIT-PUSH(1)

Git Manual

GIT-PUSH(1)

NAME

`git-push` - Update remote refs along with associated objects

SYNOPSIS

```
git push [--all | --mirror | --tags] [--follow-tags] [--atomic]
[-n | --dry-run] [--receive-pack=<git-receive-pack>]
                [--repo=<repository>] [-f | --force] [-d | --delete]
[--prune] [-v | --verbose]
                [-u | --set-upstream] [-o <string> | --push-option=<
string>]
                [--[no-]signed | --signed=(true|false|if-asked)]
                [--force-with-lease[=<refname>[:<expect>]]]
                [--no-verify] [<repository> [<refspec>...]]
```

Ubuntu/Debian | `git push --help`

Git Repositories: Committing, Pushing and Pulling (3/3)

```
GIT-PULL(1)                               Git Manual                               GIT-PULL(1)

NAME
    git-pull - Fetch from and integrate with another repository
    or a local branch

SYNOPSIS
    git pull [<options>] [<repository>] [<refspec>...]

DESCRIPTION
    Incorporates changes from a remote repository into the
    current branch. In its default mode, git pull is shorthand
    for git fetch followed by git merge FETCH_HEAD.

    More precisely, git pull runs git fetch with the given
    parameters and calls git merge to merge the
    Ubuntu/Debian | git pull --help
```

Git Repositories: Branching

```
GIT-BRANCH(1)                               Git Manual                               GIT-BRANCH(1)

NAME
    git-branch - List, create, or delete branches

SYNOPSIS
    git branch [--color[=<when>] | --no-color] [--show-current]
               [-v [--abbrev=<length> | --no-abbrev]]
               [--column[=<options>] | --no-column] [--sort=<key>]
               [(--merged | --no-merged) [<commit>]]
               [--contains [<commit>] [--no-contains [<commit>]]]
               [--points-at <object>] [--format=<format>]
               [(-r | --remotes) | (-a | --all)]
               [--list] [<pattern>...]
    git branch [--track | --no-track] [-f] <branchname> [start-point]

Ubuntu/Debian | git branch --help
```

Git Repositories: Merging

```
GIT-MERGE(1)                               Git Manual                               GIT-MERGE(1)

NAME
    git-merge - Join two or more development histories together

SYNOPSIS
    git merge [-n] [--stat] [--no-commit] [--squash] [--[no-]edit]
              [--no-verify] [-s <strategy>] [-X <strategy-option>] [-S[<keyid>]]
              [--[no-]allow-unrelated-histories]
              [--[no-]rerere-autoupdate] [-m <msg>] [-F <file>] [<commit>...]
    git merge (--continue | --abort | --quit)

DESCRIPTION
    Incorporates changes from the named commit.
```

Ubuntu/Debian | [git merge --help](#)

Lecture Outline

- Introduction
- Illustrating Distributed Version Control
- Software Configuration Management
- Version Control Systems
- Distributed Version Control Using Git
- Markdown 101
 - Common Markdown tags

Markdown 101 (1/4)

```
1 # Introduction
```

```
2
```

```
3 During these "hands-on" activities, we will explore and experiment the following:
```

```
4 1. Jupyter Notebooks---Quick walkthrough of Jupyter Notebooks
```

```
5 2. Python 3---Crash crouse introduction to Python 3
```

```
6 3. Core Python Modules---Quick walkthrough of some core Python modules that will be used in the course.
```

```
7
```

```
8 In all instances, you are encouraged to make reference to online documentation for the various tools.
```

```
Additionally, you can exploit tools like \[Zeal Offline Documentation Browser\]\(https://zealdocs.org\) to  
download and search through offline documentation. You are also encouraged to look up and explore other  
libraries, especially as you work towards the Mini Projects.
```

- **Textual content is primarily specified using the markup language “Markdown”**
 - You essentially specify the structure of your text, similar to HTML

Markdown 101 (2/4)

```
1 # Introduction
2
3 During these "hands-on" activities, we will explore and experiment the following:
4 1. Jupyter Notebooks---Quick walkthrough of Jupyter Notebooks
5 2. Python 3---Crash crouse introduction to Python 3
6 3. Core Python Modules---Quick walkthrough of some core Python modules that will be used in the course.
7
8 In all instances, you are encouraged to make reference to online documentation for the various tools.
   Additionally, you can exploit tools like \[Zeal Offline Documentation Browser\]\(https://zealdocs.org\) to
   download and search through offline documentation. You are also encouraged to look up and explore other
   libraries, especially as you work towards the Mini Projects.
```

- **Common markup: Headings**

- # h1
- ## h2
- ### h3
- #### h4

Markdown 101 (3/4)

```
1 # Introduction
2
3 During these "hands-on" activities, we will explore and experiment the following:
4 1. Jupyter Notebooks---Quick walkthrough of Jupyter Notebooks
5 2. Python 3---Crash crouse introduction to Python 3
6 3. Core Python Modules---Quick walkthrough of some core Python modules that will be used in the course.
7
8 In all instances, you are encouraged to make reference to online documentation for the various tools.
   Additionally, you can exploit tools like \[Zeal Offline Documentation Browser\]\(https://zealdocs.org\) to
   download and search through offline documentation. You are also encouraged to look up and explore other
   libraries, especially as you work towards the Mini Projects.
```

- **Common markup: Lists—Unordered**
 - * Jupyter Notebooks
 - * Python 3
 - * Core Python Libraries

Markdown 101 (4/4)

```
1 # Introduction
```

```
2
```

```
3 During these "hands-on" activities, we will explore and experiment the following:
```

```
4 1. Jupyter Notebooks---Quick walkthrough of Jupyter Notebooks
```

```
5 2. Python 3---Crash crouse introduction to Python 3
```

```
6 3. Core Python Modules---Quick walkthrough of some core Python modules that will be used in the course.
```

```
7
```

```
8 In all instances, you are encouraged to make reference to online documentation for the various tools.
```

```
Additionally, you can exploit tools like \[Zeal Offline Documentation Browser\]\(https://zealdocs.org\) to  
download and search through offline documentation. You are also encouraged to look up and explore other  
libraries, especially as you work towards the Mini Projects.
```

- **Common markup: Lists—Unordered**

- 1. Jupyter Notebooks
- 2. Python 3
- 3. Core Python Libraries

Q & A Session

- **Comments, concerns and complaints?**



ict3020@unza.zm



<http://bit.ly/39oHRDY>



<http://bit.ly/2kK2ZkA>

Bibliography

[1] **Git**

<https://git-scm.com>

[2] **Chacon, S., & Straub, B. (2014). Pro Git. Apress.**

<https://git-scm.com/book>

[3] **Sommerville, I. (2011). Software engineering 9th Edition. ISBN-10, 137035152.**

[4] **Daring Fireball: Markdown**

<https://daringfireball.net/projects/markdown>

[5] **Markdown Cheatsheet**

<https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>



ICT 3020 (2020/21)

Fundamentals of Software Engineering

Module 10: Software Configuration Management

Lighton Phiri <lighton.phiri@unza.zm>
Department of Library & Information Science
University of Zambia
<https://bit.ly/3u15u5V>