

Basic Computing Tools

CS251 Assignment

Group 22

Indian Institute of Technology, Kanpur

April 9, 2016

Table of Contents

Bash

GREP
SED
AWK
MATLAB
compatibility

HTML

CSS
CSS Syntax

Git

BitBucket

- 1 Bash
 - GREP
 - SED
 - AWK
 - MATLAB compatibility

- 2 HTML
 - CSS
 - CSS Syntax

- 3 Git

- 4 BitBucket

Bash

GREP
SED
AWK
MATLAB
compatibility

HTML

CSS
CSS Syntax

Git

BitBucket

Bash is a Unix shell and command language written by Brian Fox for the GNU Project as a free software replacement for the Bourne shell. Released in 1989, it has been distributed widely as the shell for the GNU operating system and as a default shell on Linux and OS X. It was announced during the 2016 Build Conference that Windows 10 has added a Linux subsystem which fully supports Bash and other Ubuntu binaries running natively in Windows. In the past, and currently, it has also ported to Microsoft Windows and distributed with Cygwin and MinGW, to DOS by the DJGPP project, to Novell NetWare and to Android via various terminal emulation applications. In the late 1990s, Bash was a minor player among multiple commonly used shells; at present Bash has overwhelming favor.

Bash

GREP

SED

AWK

MATLAB

compatibility

HTML

CSS

CSS Syntax

Git

BitBucket

① For quick display of files: `$ cat helloworld.sh`

`grep` is a command-line utility for searching plain-text data sets for lines matching a **regular expression**. `Grep` was originally developed for the Unix operating system, but is available today for all Unix-like systems. Its name comes from the `ed` command `g/re/p` (**g**lobally search a **re**gular expression and **p**rint), which has the same effect: doing a global search with the regular expression and printing all matching lines.

Some basic grep commands are as follows:

- ① For basic string search: `$ grep "literal_string" filename`
- ② For case insensitive search: `$ grep -i "string" filename`
- ③ For regular expressions: `$ grep "REGEX" filename`
- ④ To display N lines after match: `$ grep -A N "string" filename`
- ⑤ To display N lines before match: `$ grep -B N "string" filename`
- ⑥ To display lines which do not contain match: `$ grep -v -e "pattern" -e "pattern" filename`
- ⑦ Counting number of matches: `$ grep -c "pattern" filename`
- ⑧ To display N lines before match: `$ grep -B N "string" filename`

Bash

GREP

SED

AWK

MATLAB
compatibility

HTML

CSS

CSS Syntax

Git

BitBucket

sed (stream editor) is a Unix utility that parses and transforms text, using a simple, compact programming language. **sed** was developed from 1973 to 1974 by Lee E. McMahon of Bell Labs, and is available today for most operating systems. sed was based on the scripting features of the interactive editor ed ("editor", 1971) and the earlier qed ("quick editor", 1965–66). sed was one of the earliest tools to support regular expressions, and remains in use for text processing, most notably with the substitution command. Other options for doing "stream editing" include AWK and Perl.

- ① To match files and replace: `sed -e 's/find expression/replace expression/' filename`
- ② To use the match as a part of replace string, we can use the following command: `sed -n -e 's/United States/& of America/p' country.txt` United States of America
- ③ To convert lower case letters to upper case: `sed 'y/ul/UL/' file.txt`

AWK is an interpreted programming language designed for text processing and typically used as a data extraction and reporting tool. It is a standard feature of most Unix-like operating systems.

The AWK language is a data-driven scripting language consisting of a set of actions to be taken against streams of textual data – either run directly on files or used as part of a pipeline – for purposes of extracting or transforming text, such as producing formatted reports. The language extensively uses the string datatype, associative arrays (that is, arrays indexed by key strings), and regular expressions. While AWK has a limited intended application domain and was especially designed to support one-liner programs, the language is Turing-complete, and even the early Bell Labs users of AWK often wrote well-structured large AWK programs.

Octave

GNU Octave is software featuring a high-level programming language, primarily intended for numerical computations. It provides a command-line interface for solving linear and nonlinear problems numerically, and for performing other numerical experiments using a language that is mostly compatible with MATLAB. It may also be used as a batch-oriented language.

Bash

GREP

SED

AWK

MATLAB

compatibility

HTML

CSS

CSS Syntax

Git

BitBucket

Octave is one of the major free alternatives to MATLAB, others being Julia and Scilab. These however put less emphasis on (bidirectional) syntactic compatibility with MATLAB than Octave does.

Octave has been built with MATLAB compatibility in mind, and shares many features with MATLAB:

- ① Matrices as fundamental data type.
- ② Built-in support for complex numbers.
- ③ Powerful built-in math functions and extensive function libraries.
- ④ Extensibility in the form of user-defined functions.

Due to this, it is easy to look for octave's documentation on Mathworks at <http://in.mathworks.com/>

Table of Contents

Bash

GREP

SED

AWK

MATLAB
compatibility

HTML

CSS

CSS Syntax

Git

BitBucket

1 Bash

GREP

SED

AWK

MATLAB compatibility

2 HTML

CSS

CSS Syntax

3 Git

4 BitBucket

HyperText Markup Language , commonly referred to as HTML, is the standard markup language used to create web pages. Along with CSS, and JavaScript, HTML is a cornerstone technology used to create web pages, as well as to create user interfaces for mobile and web applications.

HTML Introduction

Bash

GREP

SED

AWK

MATLAB
compatibility

HTML

CSS

CSS Syntax

Git

BitBucket

HTML elements form the building blocks of HTML pages. HTML allows images and other objects to be embedded and it can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as `<img/ >` and `<input/ >` introduce content into the page directly. Others such as `<p>...</p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

[Bash](#)[GREP](#)[SED](#)[AWK](#)[MATLAB](#)
[compatibility](#)[HTML](#)[CSS](#)[CSS Syntax](#)[Git](#)[BitBucket](#)

CSS is a stylesheet language that describes the presentation of an HTML (or XML) document.

CSS describes how elements must be rendered on screen, on paper, or in other media.

CSS Syntax

[Bash](#)[GREP](#)[SED](#)[AWK](#)[MATLAB](#)[compatibility](#)[HTML](#)[CSS](#)[CSS Syntax](#)[Git](#)[BitBucket](#)

A CSS rule-set consists of a selector and a declaration block. The selector points to the HTML element you want to style. The declaration block contains one or more declarations separated by semicolons. Each declaration includes a CSS property name and a value, separated by a colon. A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

Bash

GREP

SED

AWK

MATLAB
compatibility

HTML

CSS

CSS Syntax

Git

BitBucket

Table of Contents

1 Bash

- GREP
- SED
- AWK
- MATLAB compatibility

2 HTML

- CSS
- CSS Syntax

3 Git

4 BitBucket

Introduction

Bash

GREP

SED

AWK

MATLAB
compatibility

HTML

CSS

CSS Syntax

Git

BitBucket

A **version control system (VCS)** allows you to track the history of a collection of files. It supports creating different versions of this collection. Each version captures a snapshot of the files at a certain point in time and the VCS allows you to switch between these versions. These versions are stored in a specific place, typically called a repository. **Git** is a widely used source code management system for software development. It is a **distributed revision control system** with an emphasis on speed, data integrity, and support for distributed, non-linear workflows. In a distributed version control system each user has a complete local copy of a repository on his individual computer. The user can copy an existing repository. This copying process is typically called cloning and the resulting repository can be referred to as a clone.

Git terminology

Bash

GREP

SED

AWK

MATLAB

compatibility

HTML

CSS

CSS Syntax

Git

BitBucket

- **Cloning** : The process of copying an existing Git repository is called cloning. After cloning a repository the user has the complete repository with its history on his local machine.
- **Working Tree** : A local repository provides at least one collection of files which originate from a certain version of the repository. This collection of files is called the working tree.
- **Branching** : Git supports branching which means that you can work on different versions of your collection of files. A branch separates these different versions and allows the user to switch between these versions to work on them.
- **Repository** : A repository contains the history, the different versions over time and all different branches and tags. In Git each copy of the repository is a complete repository.

Setting up a Repository

Bash

GREP

SED

AWK

MATLAB
compatibility

HTML

CSS

CSS Syntax

Git

BitBucket

The `git init` command creates a new Git repository. It can be used to convert an existing, unversioned project to a Git repository or initialize a new empty repository. Executing `git init` creates a `.git` subdirectory in the project root, which contains all of the necessary metadata for the repo.

git init Transform the current directory into a Git repository. This adds a `.git` folder to the current directory and makes it possible to start recording revisions of the project.

The `git clone` command copies an existing Git repository.

git clone "repo" Clone the repository located at `repo` onto the local machine. The original repository can be located on the local filesystem or on a remote machine accessible via HTTP or SSH.

Saving changes

Bash

GREP

SED

AWK

MATLAB

compatibility

HTML

CSS

CSS Syntax

Git

BitBucket

The `git add` command adds a change in the working directory to the staging area. It tells Git that you want to include updates to a particular file in the next commit.

`git add "file"` Stage all changes in "file" for the next commit.

The `git commit` command commits the staged snapshot to the project history. Committed snapshots can be thought of as "safe" versions of a project—Git will never change them unless you explicitly ask it to.

`git commit` Commit the staged snapshot. This will launch a text editor prompting you for a commit message. After you've entered a message, save the file and close the editor to create the actual commit.

`git commit -m "message"` Commit the staged snapshot, but instead of launching a text editor, use "message" as the commit message.

Inspecting a repository

[Bash](#)[GREP](#)[SED](#)[AWK](#)[MATLAB](#)[compatibility](#)[HTML](#)[CSS](#)[CSS Syntax](#)[Git](#)[BitBucket](#)

The `git status` command displays the state of the working directory and the staging area. It lets you see which changes have been staged, which haven't, and which files aren't being tracked by Git. Status output does not show you any information regarding the committed project history.

git status List which files are staged, unstaged, and untracked.

Undoing Changes

Bash

GREP

SED

AWK

MATLAB
compatibility

HTML

CSS

CSS Syntax

Git

BitBucket

The `git revert` command undoes a committed snapshot. But, instead of removing the commit from the project history, it figures out how to undo the changes introduced by the commit and appends a new commit with the resulting content. This prevents Git from losing history.

`git revert "commit"` Generate a new commit that undoes all of the changes introduced in "commit", then apply it to the current branch.

`Git reset` can be used to remove committed snapshots, although it's more often used to undo changes in the staging area and the working directory. In either case, it should only be used to undo local changes—you should never reset snapshots that have been shared with other developers.

`git reset "file"` Remove the specified file from the staging area, but leave the working directory unchanged. This unstages a file without overwriting any changes.

Undoing Changes

Bash

GREP

SED

AWK

MATLAB
compatibility

HTML

CSS

CSS Syntax

Git

BitBucket

git reset Reset the staging area to match the most recent commit, but leave the working directory unchanged.

The `git clean` command removes untracked files from your working directory. This is really more of a convenience command, since it's trivial to see which files are untracked with `git status` and remove them manually. Like an ordinary `rm` command, `git clean` is not undoable, so make sure you really want to delete the untracked files before you run it.

git clean -n Perform a “dry run” of `git clean`. This will show you which files are going to be removed without actually doing it.

git clean -f Remove untracked files from the current directory. The `-f` (force) flag is required unless the `clean.requireForce` configuration option is set to false (it's true by default).

Bash

GREP

SED

AWK

MATLAB
compatibility

HTML

CSS

CSS Syntax

Git

BitBucket

Table of Contents

- 1 Bash
 - GREP
 - SED
 - AWK
 - MATLAB compatibility
- 2 HTML
 - CSS
 - CSS Syntax
- 3 Git
- 4 BitBucket

Introduction

Bash

GREP

SED

AWK

MATLAB
compatibility

HTML

CSS

CSS Syntax

Git

BitBucket

Bitbucket is a web-based hosting service for projects that use either the Mercurial (since launch) or Git (since October 2011) revision control systems. Bitbucket offers both commercial plans and free accounts. It offers free accounts with an unlimited number of private repositories. It is a system for hosting version control repositories owned by Atlassian. It is a competitor to github.

Creating a Git repository

Bash

GREP

SED

AWK

MATLAB

compatibility

HTML

CSS

CSS Syntax

Git

BitBucket

- 1 From Bitbucket, click Repositories and then Create repository button at the top of the page. The system displays the Create a new repository page.
- 2 Enter BitbucketStationLocations for the Name field. Bitbucket uses this Name in the URL of the repository. For example, if the user thebest has a repository called awesomerepo, the URL for that repository would be <https://bitbucket.org/thebest/awesomerepo>.
- 3 For Access level, leave the This is a private repository box checked. A private repository is only visible to you and those you give access to. If this box is unchecked, everyone can see your repository.
- 4 Pick Git for the Repository type.
- 5 Click Create repository. Bitbucket creates repository and displays its Overview page.

Cloning repository

Bash

GREP

SED

AWK

MATLAB
compatibility

HTML

CSS

CSS Syntax

Git

BitBucket

- 1 First make a directory on local machine.
- 2 From Bitbucket, go to BitbucketStationLocations repository. Click Clone. The system displays a pop-up clone dialog. By default, the clone dialog sets the protocol to HTTPS or SSH, depending on your settings. For the purposes of this tutorial, don't change your default protocol.
- 3 Copy the highlighted clone command. From terminal window, paste the command copied from Bitbucket and press Return.

Create a file in Bitbucket

Bash

GREP

SED

AWK

MATLAB
compatibility

HTML

CSS

CSS Syntax

Git

BitBucket

- 1 From BitbucketStationLocations repository, click Source to open the source directory
- 2 From the Source page, click New file in the top right corner. A page for creating the new file opens.
- 3 Enter filename in the filename field . Then select mode from the Syntax mode list. Finally add the following code into the text box.
- 4 Click Commit. The Commit message field appears with the message: "filename" created online with Bitbucket. Click Commit under the message field.