Terminal Commands/Cheat-Sheet

This cheat-sheet provides a comprehensive overview of essential terminal commands, ranging from basic command-line operations to more advanced tools like Docker, Git, and Rails. Whether you're navigating directories, managing version control, or running a Rails application, these commands are vital for efficient workflow management. The document breaks down each command, explaining its functionality and purpose, ensuring that developers can utilize it to streamline development tasks across different environments.

Command line basics:

- **pwd**: Prints the current working directory.
- **ls**: Lists files and directories in the current directory.
- **cd [directory]**: Changes the current directory to the specified one.
- **mkdir [directory]**: Creates a new directory.
- **rm [file]**: Deletes a file.
- **cp [source] [destination]**: Copies files or directories.
- **mv [source] [destination]**: Moves or renames files or directories.
- **touch [file]**: Creates a new, empty file.
- **cat [file]**: Displays the content of a file.

Docker commands:

- **docker pull**: Downloads a docker image from a registry to your local machine.
- **docker run**: Creates and starts a new container from a specified image.
- **docker ps**: Lists currently running containers.
- **docker stop**: Stops a running container.
- **docker restart**: Restarts a running or stopped container.

- Creating a Docker container
    - [Windows PowerShell] **docker run -it -p 3000:3000 -v ${PWD}:/workspace Docker_USERNAME/IMAGENAME**
    - [Windows Command Prompt] **docker run -it -p 3000:3000 -v %cd%:/workspace Docker_USERNAME/IMAGENAME**
    - [Windows WSL 2, Mac/Linux Terminal or Git Bash] **docker run -it -p 3000:3000 -v $(pwd):/workspace Docker_USERNAME/IMAGENAME**

- Breakdown:
    - **docker run**: Start a new container from an image.
    - **-it**: Interactive terminal for interacting with the container.
    - **-p 3000:3000**: Exposes port 3000 on the host and maps it to port 3000 in the container.
    - **-v $(pwd):/workspace**: Mounts the current directory from your host machine to /workspace in the container, allowing file synchronization.
    - **Docker_USERNAME/IMAGENAME**: The Docker image to use.
- **/workspace#** is where you can run commands in your Linux environment. When you create folders, they will show up in your local folder.

Git commands:

- **git status**: Shows the status of the working directory and staging area, indicating any changes, untracked files, or files to be committed.
- **git log**: Displays the commit history of the repository, showing a list of past commits.
- **git clone**: Creates a local copy of a remote repository.
- **git add**: Stages changes (new files or modifications) for the next commit.
- **git commit**: Records the staged changes in the local repository with a message describing the commit.
- **git push**: Uploads local commits to a remote repository.
- **git pull**: Fetches and merges changes from a remote repository into the current branch.
- **git tag**: Creates a tag, which is a specific point in the commit history often used for releases.
- **git branch**: Lists, creates, or deletes branches in the repository.
- **git checkout**: Switches to a different branch or commit.

Rails commands:

- Database commands:
    - **rails db:create**: Creates the database for the current Rails environment.
    - **rails db:drop**: Drops (deletes) the database for the current Rails environment.
    - **rails db:migrate**: Applies pending migrations to the database and alters the database structure to match the latest version of the schema.
    - **rails db:reset**: Drops the database, recreates it, and then runs all migrations from scratch.
    - **rails db:rollback**: Reverts the most recent migration applied to the database.
- **rails new portfolio_app --skip -bundle**

- Breakdown:
    - **rails new portfolio_app**: Creates a new default directory, configuration files, and initial dependencies under the name "portfolio_app".
    - **--skip -bundle**: Skips running the "bundle install" command that installs all the gems listed in a "Gemfile", which are dependencies required by the application. This allows you to control when or how gems are installed.
- **ruby --version**: Checks/prints the version of Ruby installed on the system.
- **rails --version**: Checks/prints the version of Rails installed on the system.
- **rails server -b 0.0.0.0**:
- Breakdown:
    - **rails server**: Starts the Rails application server in the default web browser. By default the server will start on "localhost" (127.0.0.1) and bind it to port 3000.
    - **-b 0.0.0.0**: Binds the server to the specific IP address 0.0.0.0.
- **rails generate**: Generates different types of resources or components for the application.
- **rails generate migration**: Generates a migration file that contains instructions to modify the database schema.
- **rails destroy**: Reverses the changes made by a corresponding generate command.