



BASH CRASH COURSE

Mary-Ann Blätke, Jędrzej Jakub Szymański

Course Objective

1. **Understand Bash Basics:** Learn the fundamental concepts of the Bash command line, including navigation, file manipulation, and simple commands.
2. **Scripting Essentials:** Introduction to writing and executing basic Bash scripts to automate routine tasks.
3. **Practical Examples and Exercises:** Apply what you learn through hands-on exercises, simulating real-world scenarios like managing a "Superhero Project".



generated by DALL·E



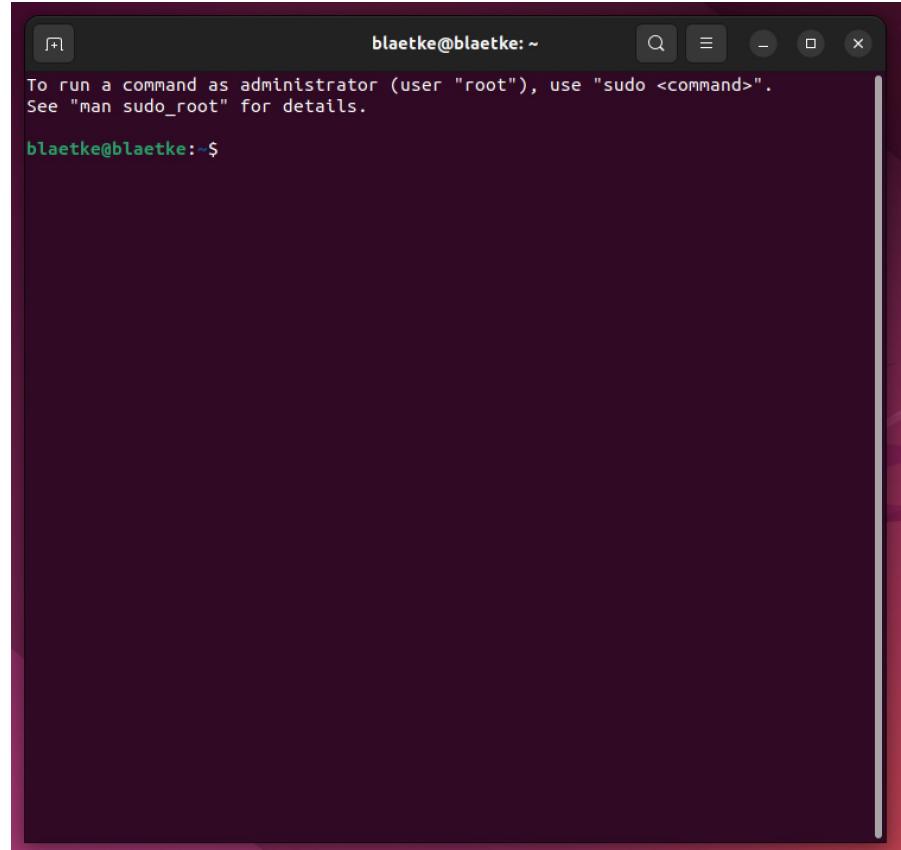
What is Bash?

- **Bash (Bourne Again SHell):** A powerful command-line interface (CLI) used in Unix and Linux environments, providing a platform for users to interact directly with the operating system.
- **Default Shell:** Often the default shell on Linux systems and macOS, integral for system administration, scripting, and data management tasks.

Linux Terminal (=Shell or Command Prompt)

Open Terminal Window on Linux

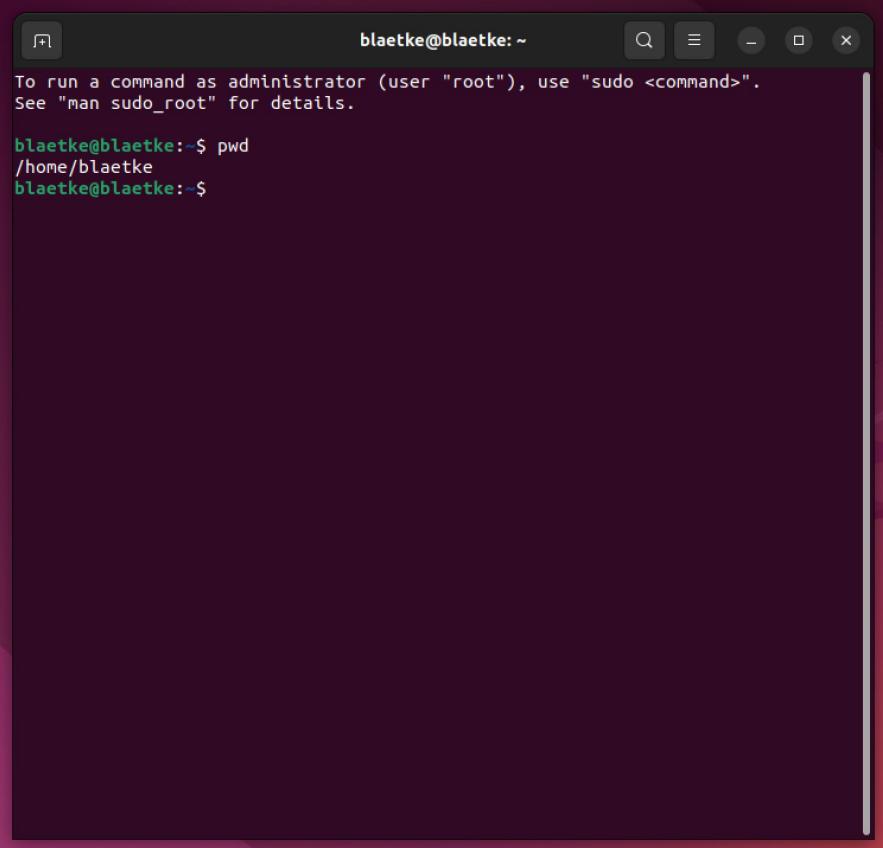
1. Press CTRL + ALT + T simultaneously to open the terminal.
2. Or open Show Application in the Dash Bar & find out Terminal Application.
3. Or execute the command gnome-terminal in the command dialogue opened using ALT + F2



What is the path of my working directory?

pwd [*options*]

Returns the *absolute path*, which is the complete path from the root directory to the current directory. This is helpful for confirming your exact location in the file system's hierarchy.



```
blaetke@blaetke: ~
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

blaetke@blaetke:~$ pwd
/home/blaetke
blaetke@blaetke:~$
```

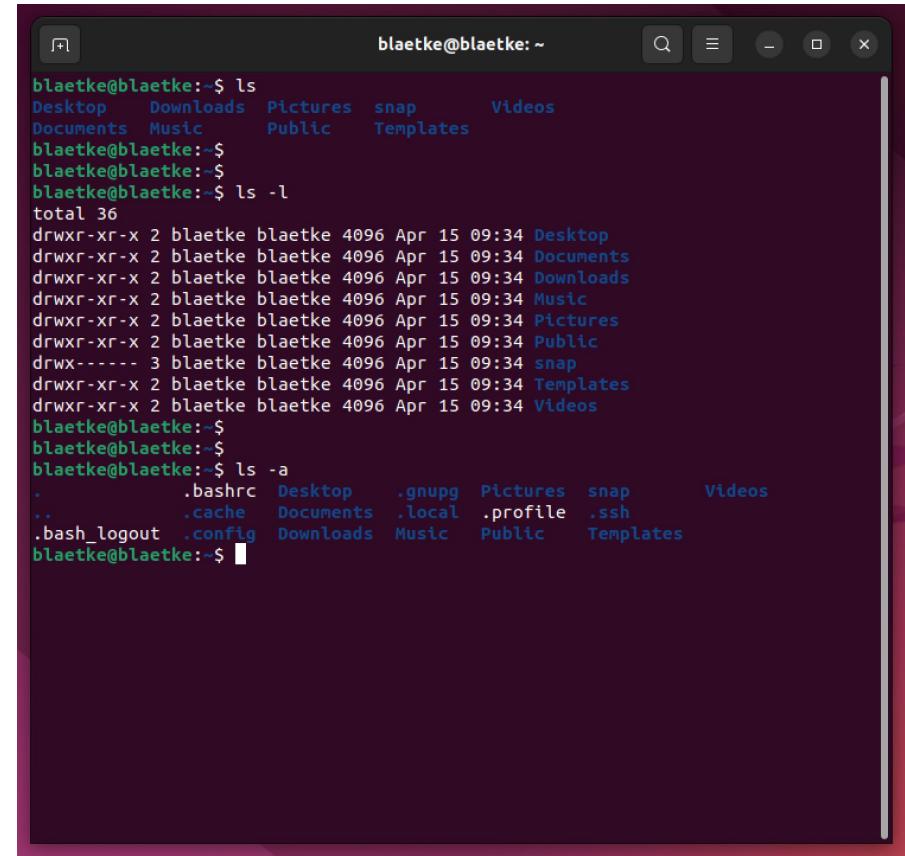
How to list everything inside a directory?

`ls [options][path/to/dir]`

Lists the contents of the current directory. Files and directories are displayed in a simple list format.

Common Usage and Options

- `ls -l`: Provides a detailed listing of files, including permissions, number of links, owner, group, size, and the last modified date.
- `ls -a`: Lists all entries including those starting with a dot (.), which are hidden by default.



```
blaetke@blaetke: $ ls
Desktop  Downloads  Pictures  snap      Videos
Documents  Music    Public    Templates

blaetke@blaetke:-
blaetke@blaetke:-
blaetke@blaetke: $ ls -l
total 36
drwxr-xr-x  2 blaetke blaetke 4096 Apr 15 09:34 Desktop
drwxr-xr-x  2 blaetke blaetke 4096 Apr 15 09:34 Documents
drwxr-xr-x  2 blaetke blaetke 4096 Apr 15 09:34 Downloads
drwxr-xr-x  2 blaetke blaetke 4096 Apr 15 09:34 Music
drwxr-xr-x  2 blaetke blaetke 4096 Apr 15 09:34 Pictures
drwxr-xr-x  2 blaetke blaetke 4096 Apr 15 09:34 Public
drwx----- 3 blaetke blaetke 4096 Apr 15 09:34 snap
drwxr-xr-x  2 blaetke blaetke 4096 Apr 15 09:34 Templates
drwxr-xr-x  2 blaetke blaetke 4096 Apr 15 09:34 Videos
blaetke@blaetke:-
blaetke@blaetke:-
blaetke@blaetke: $ ls -a
.          .bashrc  Desktop   .gnupg  Pictures  snap      Videos
..         .cache   Documents  .local   .profile  .ssh
.bash_logout .config  Downloads  Music    Public    Templates
blaetke@blaetke: $
```

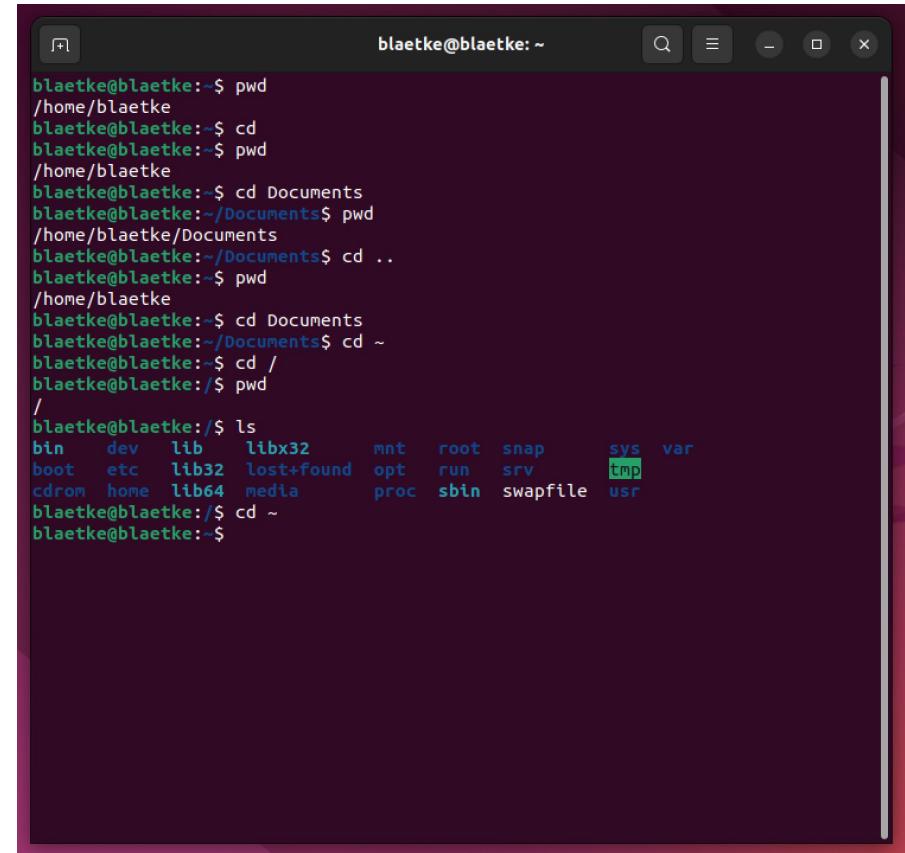
How to change the directory?

cd [path/to/dir]

Changes the current working directory to the specified directory. If no directory is specified, `cd` typically defaults to the user's home directory.

Common Usage and Options

- `cd ~`: *Moves to the home directory*. The tilde (~) is a shortcut for the home directory of the current user.
- `cd ..`: *Moves up one level* in the directory hierarchy. This command changes the working directory to the parent directory of the current directory.
- `cd /`: *Moves to the root directory*. The slash (/) represents the root of the filesystem.



A screenshot of a terminal window titled "blaetke@blaetke: ~". The terminal shows a session where the user runs several commands:

```
blaetke@blaetke:~$ pwd
/home/blaetke
blaetke@blaetke:~$ cd
blaetke@blaetke:~$ pwd
/home/blaetke
blaetke@blaetke:~$ cd Documents
blaetke@blaetke:/Documents$ pwd
/home/blaetke/Documents
blaetke@blaetke:/Documents$ cd ..
blaetke@blaetke:~$ pwd
/home/blaetke
blaetke@blaetke:~$ cd Documents
blaetke@blaetke:/Documents$ cd ~
blaetke@blaetke:~$ cd /
blaetke@blaetke:/$ pwd
/
blaetke@blaetke:/~$ ls
bin dev lib lib32 mnt root snap sys var
boot etc lib32 lost+found opt run srv tmp
cdrom home lib64 media proc sbin swapfile usr
blaetke@blaetke:/~$ cd ~
blaetke@blaetke:~$
```

Home vs. Root Directory

Root “/”

The root directory is the topmost level of the file system. Everything on your system resides under the root directory. This includes all the files used by the operating system, applications, and user-generated content. It generally contains critical system directories and files, such as:

- `/bin` - Essential user command binaries
- `/etc` - Configuration files
- `/lib` - System libraries
- `/usr` - Secondary hierarchy for user data, system programs, libraries, documentation, etc.
- `/var` - Variable files like logs and caches
- `/home` - Home directories of users

Home “/home/username” = “~”

The home directory is the personal storage space for a user. It's designated for storing personal files, user-specific configuration files, and other personal data. For each user on the system, there is typically a separate home directory, identified by their username, like `/home/john`.



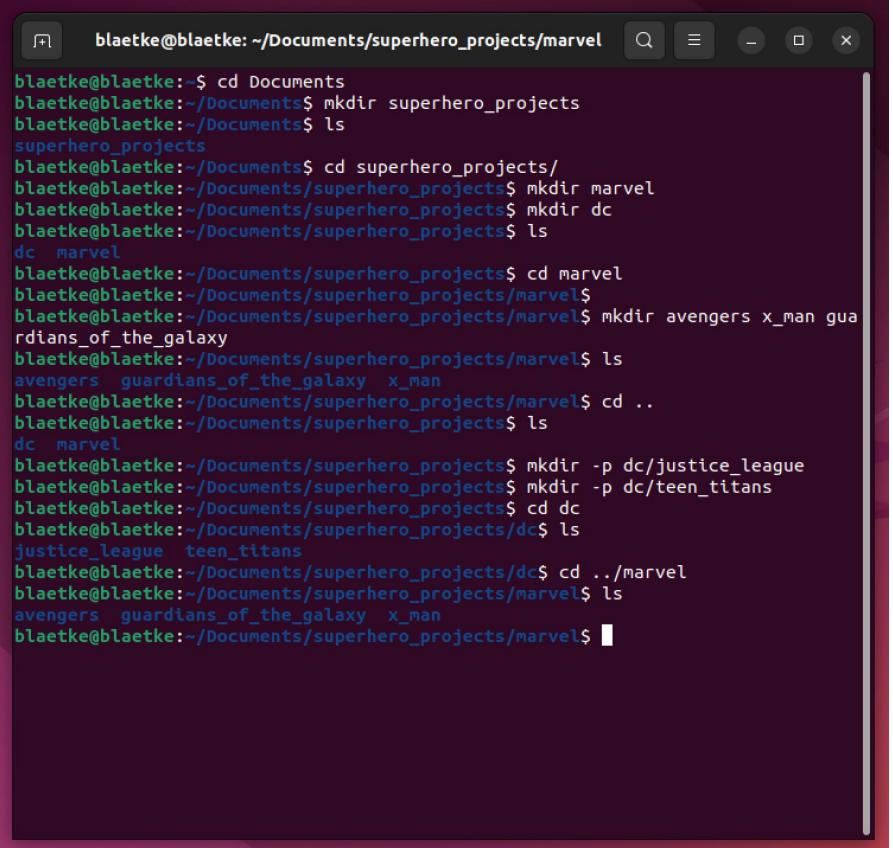
How to make a new directory?

`mkdir [options][path/to/dir]`

Creates a new directory with the name specified by **dirname** in the current working directory.

Common Usage and Options:

- **mkdir dir1 dir2 dir3** - create multiple directories at once by listing each directory name separated by a space
- **mkdir -p /path/to/directory/subdirectory** - creates parent directories as needed. If the specified directory already exists, it won't return an error. Instead, it will create any missing directories in the path



```
blaetke@blaetke:~/Documents/superhero_projects/marvel
blaetke@blaetke:~/Documents$ cd Documents
blaetke@blaetke:~/Documents$ mkdir superhero_projects
blaetke@blaetke:~/Documents$ ls
superhero_projects
blaetke@blaetke:~/Documents$ cd superhero_projects/
blaetke@blaetke:~/Documents/superhero_projects$ mkdir marvel
blaetke@blaetke:~/Documents/superhero_projects$ mkdir dc
blaetke@blaetke:~/Documents/superhero_projects$ ls
dc  marvel
blaetke@blaetke:~/Documents/superhero_projects$ cd marvel
blaetke@blaetke:~/Documents/superhero_projects/marvel$ 
blaetke@blaetke:~/Documents/superhero_projects/marvel$ mkdir avengers x_man guardians_of_the_galaxy
blaetke@blaetke:~/Documents/superhero_projects/marvel$ ls
avengers  guardians_of_the_galaxy  x_man
blaetke@blaetke:~/Documents/superhero_projects/marvel$ cd ..
blaetke@blaetke:~/Documents/superhero_projects$ ls
dc  marvel
blaetke@blaetke:~/Documents/superhero_projects$ mkdir -p dc/justice_league
blaetke@blaetke:~/Documents/superhero_projects$ mkdir -p dc/teen_titans
blaetke@blaetke:~/Documents/superhero_projects$ cd dc
blaetke@blaetke:~/Documents/superhero_projects/dc$ ls
justice_league  teen_titans
blaetke@blaetke:~/Documents/superhero_projects/dc$ cd ../marvel
blaetke@blaetke:~/Documents/superhero_projects/marvel$ ls
avengers  guardians_of_the_galaxy  x_man
blaetke@blaetke:~/Documents/superhero_projects/marvel$
```

How to make a new directory?

```
mkdir [options][path/to/dir]
```

~/Documents/superhero_project

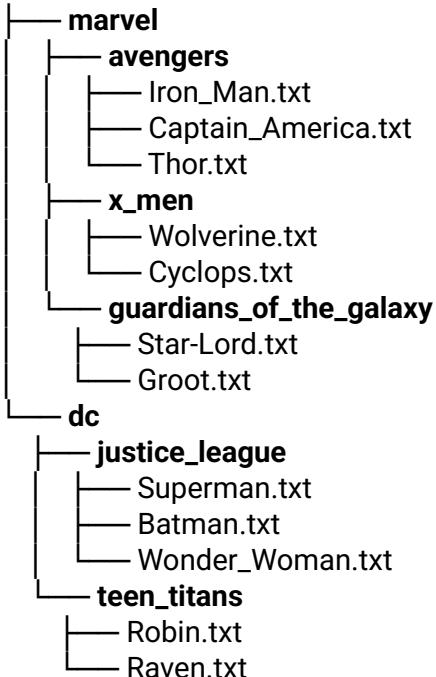
```
  └── marvel
      ├── avengers
      │   ├── Iron_Man.txt
      │   ├── Captain_America.txt
      │   └── Thor.txt
      ├── x_men
      │   ├── Wolverine.txt
      │   └── Cyclops.txt
      └── guardians_of_the_galaxy
          ├── Star-Lord.txt
          └── Groot.txt
  └── dc
      ├── justice_league
      │   ├── Superman.txt
      │   ├── Batman.txt
      │   └── Wonder_Woman.txt
      └── teen_titans
          ├── Robin.txt
          └── Raven.txt
```



How to make a new directory?

`mkdir [options][path/to/dir]`

~/Documents/superhero_project



```

blaetke@blaetke:~$ cd Documents
blaetke@blaetke:~/Documents$ mkdir superhero_projects
blaetke@blaetke:~/Documents$ ls
superhero_projects
blaetke@blaetke:~/Documents$ cd superhero_projects/
blaetke@blaetke:~/Documents/superhero_projects$ mkdir marvel
blaetke@blaetke:~/Documents/superhero_projects$ mkdir dc
blaetke@blaetke:~/Documents/superhero_projects$ ls
dc  marvel
blaetke@blaetke:~/Documents/superhero_projects$ cd marvel
blaetke@blaetke:~/Documents/superhero_projects/marvel$ 
blaetke@blaetke:~/Documents/superhero_projects/marvel$ mkdir avengers x_man gua
rdians_of_the_galaxy
blaetke@blaetke:~/Documents/superhero_projects/marvel$ ls
avengers  guardians_of_the_galaxy  x_man
blaetke@blaetke:~/Documents/superhero_projects/marvel$ cd ..
blaetke@blaetke:~/Documents/superhero_projects$ ls
dc  marvel
blaetke@blaetke:~/Documents/superhero_projects$ mkdir -p dc/justice_league
blaetke@blaetke:~/Documents/superhero_projects$ mkdir -p dc/teen_titans
blaetke@blaetke:~/Documents/superhero_projects$ cd dc
blaetke@blaetke:~/Documents/superhero_projects/dc$ ls
justice_league  teen_titans
blaetke@blaetke:~/Documents/superhero_projects/dc$ cd ../marvel
blaetke@blaetke:~/Documents/superhero_projects/marvel$ ls
avengers  guardians_of_the_galaxy  x_man
blaetke@blaetke:~/Documents/superhero_projects/marvel$ 
    
```

Absolute vs. Relative Paths

Absolute Path

Specifies the location of a file or directory from the root of the file system. It starts with a slash (/), indicating the root directory, and includes all subdirectories leading to the specified file or directory. Absolute paths always point to the same location, regardless of the current working directory.

```
/home/user/Documents/superhero_project/marvel/x_men/
```

This path starts from the root (/) and fully specifies the location of the file, making it clear and unambiguous independent of your current working directory.

Relative Path

Specifies the location of a file or directory relative to the current working directory. It does not begin with a slash; instead, it starts from the current directory and navigates the file system from there.

Relative path to `x_men` directory if the current working directory is `/superhero_project`:

```
marvel/x_men/
```

Relative path to `justice_league` directory if the current working directory is `/superhero_project/marvel`:

```
../dc/justice_league/
```

Exercise

Example

```
~/Documents/superhero_project
├── marvel
│   ├── avengers
│   │   ├── Iron_Man.txt
│   │   ├── Captain_America.txt
│   │   └── Thor.txt
│   ├── x_men
│   │   ├── Wolverine.txt
│   │   └── Cyclops.txt
│   └── guardians_of_the_galaxy
│       ├── Star-Lord.txt
│       └── Groot.txt
└── dc
    ├── justice_league
    │   ├── Superman.txt
    │   ├── Batman.txt
    │   └── Wonder_Woman.txt
    └── teen_titans
        ├── Robin.txt
        └── Raven.txt
```

Questions

What is the absolute path to **teen_titans**?

What is the relative path to **teen_titans** if your current working directory is **superhero_project**?

What is the relative path to the **avengers** directory, assuming your current working directory is **teen_titans**?

Check your answer in the terminal, use **pwd** and **cd** command!

How to output the value of a string of variable?

```
echo [option(s)][string(s)]
```

Outputs the strings it is given to standard output, which is typically the terminal

Common Usage and Options

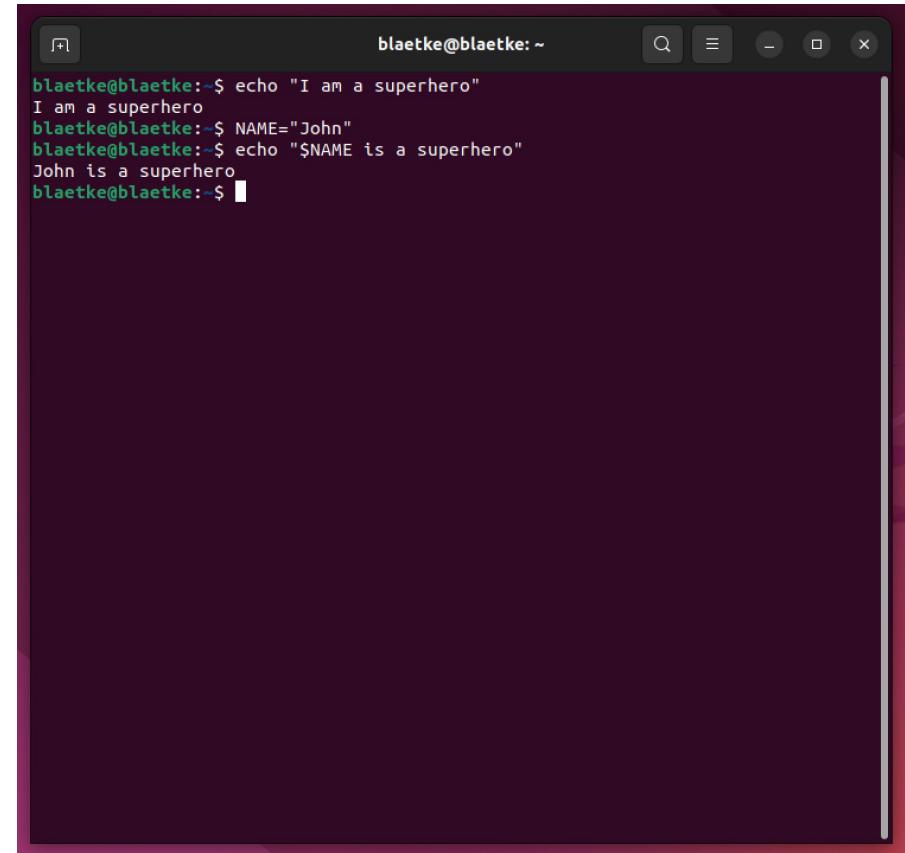
- Display variables:

```
NAME="John"
```

```
echo "Hello, $NAME"
```

- Write to file:

```
echo "Some data for the  
file" > file.txt
```



A screenshot of a terminal window titled 'blaetke@blaetke: ~'. The window shows the following session:

```
blaetke@blaetke: ~ echo "I am a superhero"  
I am a superhero  
blaetke@blaetke: ~ NAME="John"  
blaetke@blaetke: ~ echo "$NAME is a superhero"  
John is a superhero  
blaetke@blaetke: ~
```

How to output the value of a string of variable?

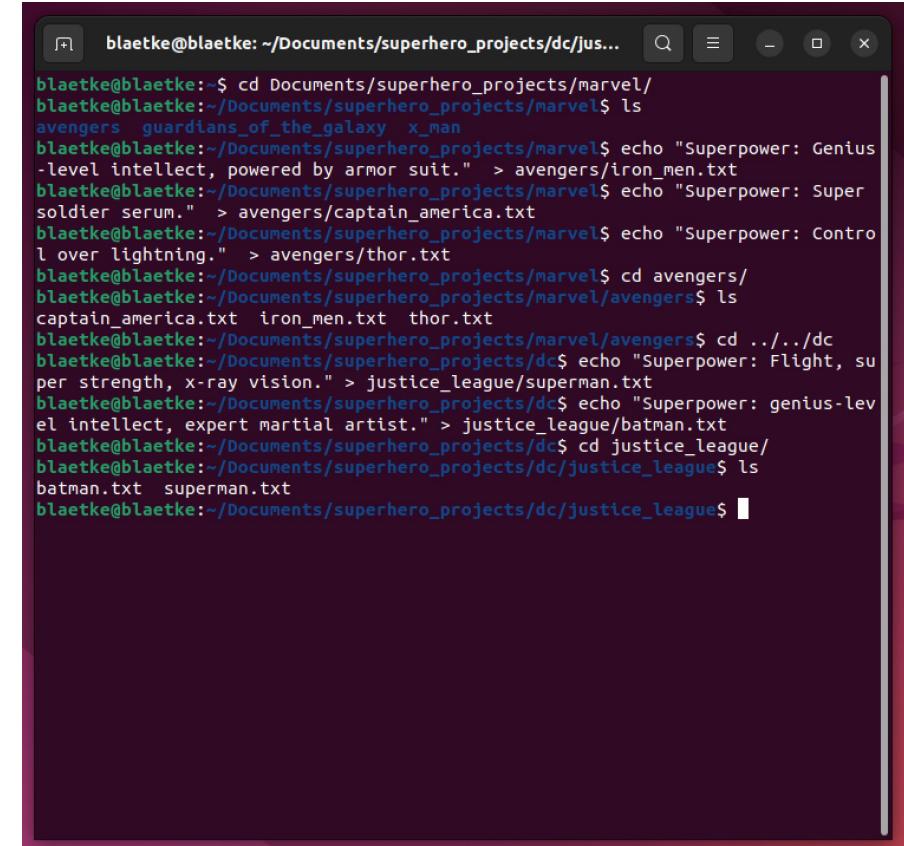
`echo [option(s)][string(s)]`

~/Documents/superhero_project

```

  └── marvel
      ├── avengers
      │   ├── Iron_Man.txt
      │   ├── Captain_America.txt
      │   └── Thor.txt
      ├── x_men
      │   ├── Wolverine.txt
      │   └── Cyclops.txt
      └── guardians_of_the_galaxy
          ├── Star-Lord.txt
          └── Groot.txt
  └── dc
      ├── justice_league
      │   ├── Superman.txt
      │   ├── Batman.txt
      │   └── Wonder_Woman.txt
      └── teen_titans
          ├── Robin.txt
          └── Raven.txt

```



```

blaetke@blaetke:~/Documents/superhero_projects/dc/jus... Q _ x
blaetke@blaetke:~$ cd Documents/superhero_projects/marvel/
blaetke@blaetke:~/Documents/superhero_projects/marvel$ ls
avengers  guardians_of_the_galaxy  x_men
blaetke@blaetke:~/Documents/superhero_projects/marvel$ echo "Superpower: Genius
-level intellect, powered by armor suit." > avengers/iron_men.txt
blaetke@blaetke:~/Documents/superhero_projects/marvel$ echo "Superpower: Super
soldier serum." > avengers/captain_america.txt
blaetke@blaetke:~/Documents/superhero_projects/marvel$ echo "Superpower: Contro
l over lightning." > avengers/thor.txt
blaetke@blaetke:~/Documents/superhero_projects/marvel$ cd avengers/
blaetke@blaetke:~/Documents/superhero_projects/marvel/avengers$ ls
captain_america.txt  iron_men.txt  thor.txt
blaetke@blaetke:~/Documents/superhero_projects/marvel/avengers$ cd ../../dc
blaetke@blaetke:~/Documents/superhero_projects/dc$ echo "Superpower: Flight, su
per strength, x-ray vision." > justice_league/superman.txt
blaetke@blaetke:~/Documents/superhero_projects/dc$ echo "Superpower: genius-lev
el intellect, expert martial artist." > justice_league/batman.txt
blaetke@blaetke:~/Documents/superhero_projects/dc$ cd justice_league/
blaetke@blaetke:~/Documents/superhero_projects/dc/justice_league$ ls
batman.txt  superman.txt
blaetke@blaetke:~/Documents/superhero_projects/dc/justice_league$ █

```

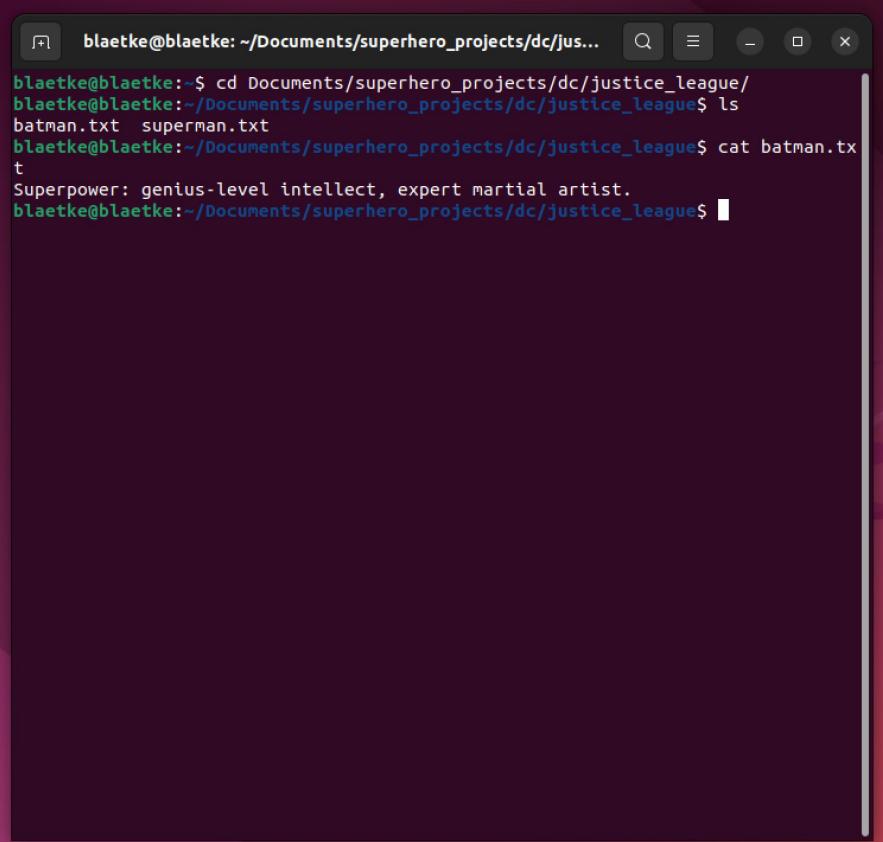
How to display the content of a file?

cat [option][file...]

Read one or more files and output their contents to the standard output, which is typically the terminal.

Common Usage and Options

- Displaying File Contents:
`cat filename.txt`
- Concatenating Multiple Files:
`cat file1.txt file2.txt > mergedfile.txt`
- Creating a New File:
`cat > newfile.txt`



A screenshot of a terminal window titled "blaetke@blaetke: ~/Documents/superhero_projects/dc/justice_league". The terminal shows the following command sequence:

```
blaetke@blaetke:~$ cd Documents/superhero_projects/dc/justice_league/
blaetke@blaetke:~/Documents/superhero_projects/dc/justice_league$ ls
batman.txt superman.txt
blaetke@blaetke:~/Documents/superhero_projects/dc/justice_league$ cat batman.txt
Superpower: genius-level intellect, expert martial artist.
blaetke@blaetke:~/Documents/superhero_projects/dc/justice_league$
```

Exercise

Example

```
~/Documents/superhero_project
├── marvel
│   ├── avengers
│   │   ├── Iron_Man.txt
│   │   ├── Captain_America.txt
│   │   └── Thor.txt
│   └── x_men
│       ├── Wolverine.txt (regenerative healing, retractable claws)
│       └── Cyclops.txt (powerful energy blasts through eyes)
└── guardians_of_the_galaxy
    ├── Star-Lord.txt (expert in combat and tactics, high-tech gadgets)
    └── Groot.txt (regenerate and manipulate tree-like body)

└── dc
    ├── justice_league
    │   ├── Superman.txt
    │   ├── Batman.txt
    │   └── Wonder_Woman.txt (super strength, agility, magic weapons)
    └── teen_titans
        ├── Robin.txt (expert in combat and acrobat, tactical intelligence)
        └── Raven.txt (teleporting and manipulating shadows and emotions)
```

Questions

Add text files for the remaining superheroes. Use the `cd` command to switch between directories, use the `echo` command to create files and the `cat` command to check the content of the files.

How to move files/directories?

`mv [options] old new`

Moves the file or directory from the source path to the destination path. If the destination is a directory, the source file or directory is moved into it. If the destination is a filename and it does not exist, the source is renamed to the destination name.

Common Usage and Options

- Renaming Files

```
mv oldname.txt newname.txt
```

- Moving Files and Folders

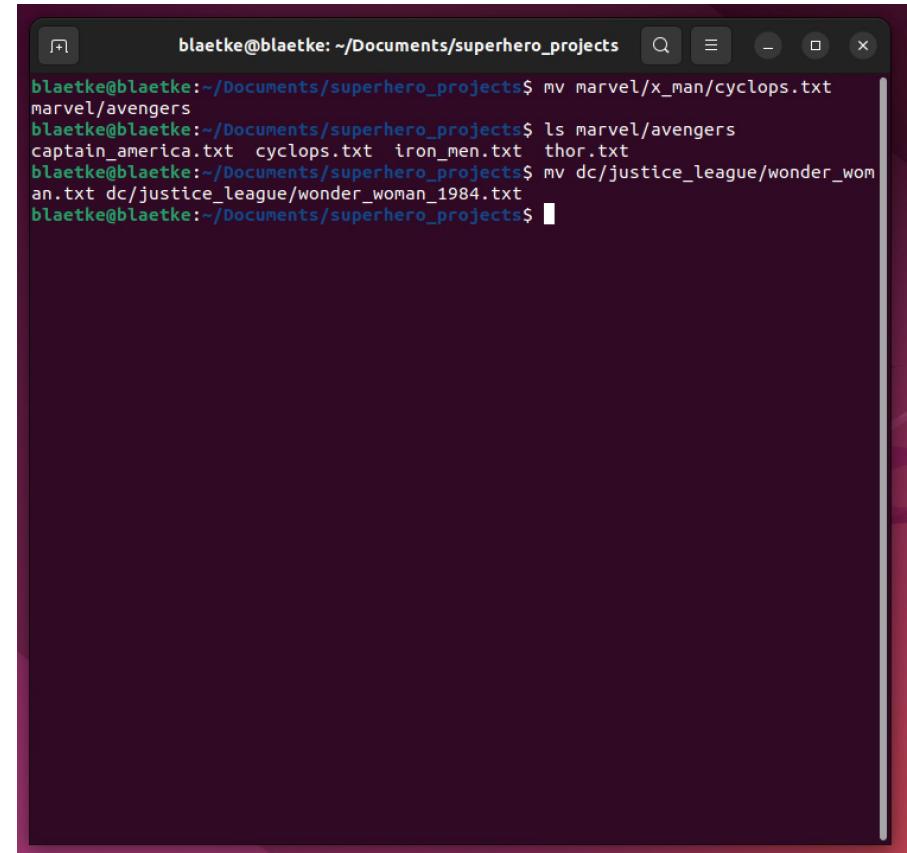
```
mv file1.txt /home/user/Documents
```

```
mv folder1 /home/user/Documents
```

- Prevent Overwriting

```
mv -i file1.txt
```

```
/home/user/Documents
```



A screenshot of a terminal window titled 'Terminal' showing the command-line interface. The terminal shows a user's session on a machine named 'blaetke'. The user is navigating through their home directory and executing the 'mv' command to move files between 'marvel' and 'dc' subdirectories under 'superhero_projects'. The terminal also displays the results of an 'ls' command to show the contents of each directory after the moves.

```
blaetke@blaetke: ~/Documents/superhero_projects$ mv marvel/x_man/cyclops.txt marvel/avengers
blaetke@blaetke: ~/Documents/superhero_projects$ ls marvel/avengers
captain_america.txt cyclops.txt iron_men.txt thor.txt
blaetke@blaetke: ~/Documents/superhero_projects$ mv dc/justice_league/wonder_woman.txt dc/justice_league/wonder_woman_1984.txt
blaetke@blaetke: ~/Documents/superhero_projects$
```

Exercise

Example

```
~/Documents/superhero_project
├── marvel
│   ├── avengers
│   │   ├── Iron_Man.txt
│   │   ├── Captain_America.txt
│   │   └── Thor.txt
│   ├── x_men
│   │   ├── Wolverine.txt
│   │   └── Cyclops.txt
│   └── guardians_of_the_galaxy
│       ├── Star-Lord.txt
│       └── Groot.txt
└── dc
    ├── justice_league
    │   ├── Superman.txt
    │   ├── Batman.txt
    │   └── Wonder_Woman.txt
    └── teen_titans
        ├── Robin.txt
        └── Raven.txt
```

Questions

Using `mkdir` create a new folder called `legends` in the `dc` directory!

Change into the directory `teen_titans`.

Move the files `robin.txt` and `raven.txt` into the new directory `legends`!

Check your answer in the terminal using `ls`!

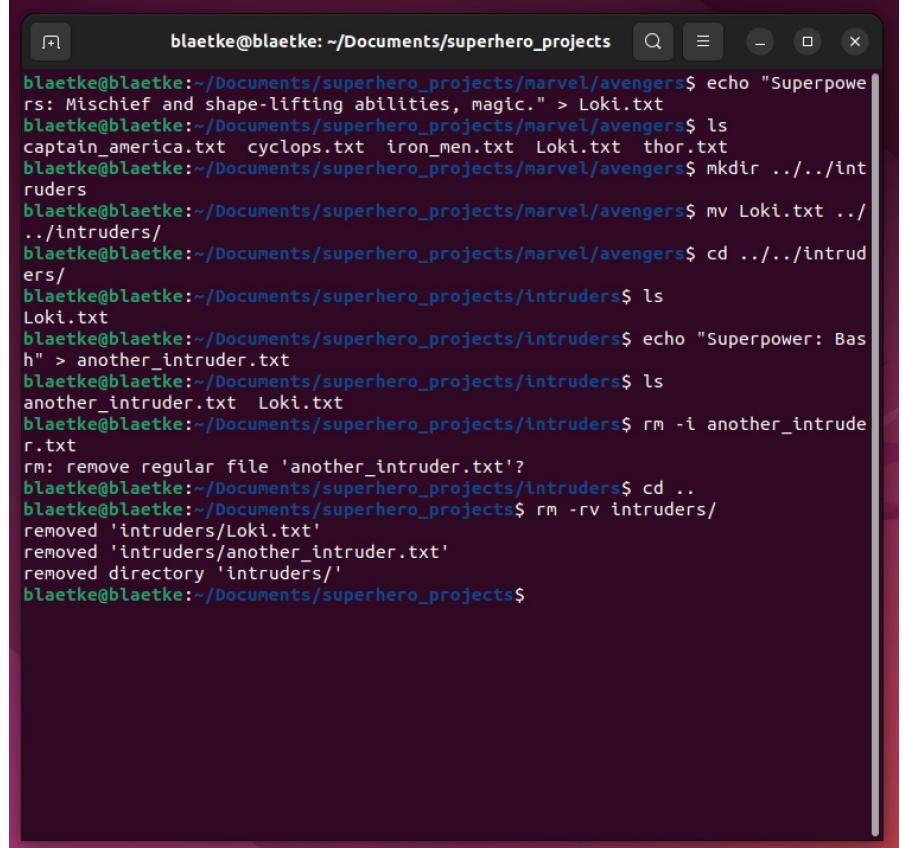
How to remove files and directories?

`rm [options] file1 file2`

Removes the files specified as arguments. It does not remove directories by default. The command should be used with **caution**, as deleting files or directories is typically irreversible, especially in environments without a trash or recycle bin system.

Common Usage and Options:

- Deleting a single file
`rm file1.txt`
- Deleting files with confirmation
`rm -i file1.txt file2.txt`
- Deleting a directory and showing detailed information
`rm -rv directory_name`
(all contents, including subdirectories and files will be deleted)

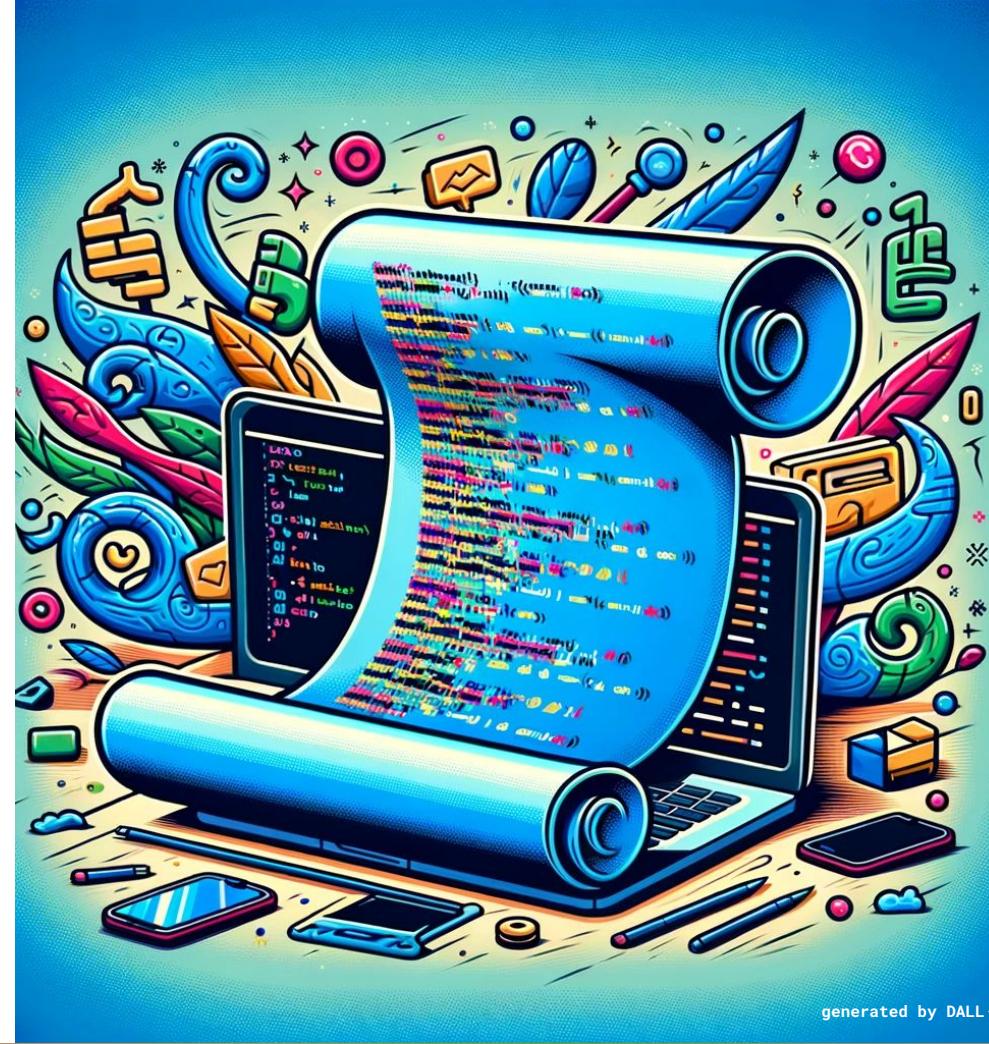


```
blaetke@blaetke: ~/Documents/superhero_projects$ echo "Superpowers: Mischief and shape-lifting abilities, magic." > Loki.txt
blaetke@blaetke: ~/Documents/superhero_projects/marvel/avengers$ ls captain_america.txt cyclops.txt iron_men.txt Loki.txt thor.txt
blaetke@blaetke: ~/Documents/superhero_projects/marvel/avengers$ mkdir ../../intruders
blaetke@blaetke: ~/Documents/superhero_projects/marvel/avengers$ mv Loki.txt ../../intruders/
blaetke@blaetke: ~/Documents/superhero_projects/marvel/avengers$ cd ../../intruders/
blaetke@blaetke: ~/Documents/superhero_projects/intruders$ ls Loki.txt
blaetke@blaetke: ~/Documents/superhero_projects/intruders$ echo "Superpower: Bash" > another_intruder.txt
blaetke@blaetke: ~/Documents/superhero_projects/intruders$ ls another_intruder.txt Loki.txt
blaetke@blaetke: ~/Documents/superhero_projects/intruders$ rm -i another_intruder.txt
rm: remove regular file 'another_intruder.txt'?
blaetke@blaetke: ~/Documents/superhero_projects/intruders$ cd ..
blaetke@blaetke: ~/Documents/superhero_projects$ rm -rv intruders/
removed 'intruders/Loki.txt'
removed 'intruders/another_intruder.txt'
removed directory 'intruders/'
blaetke@blaetke: ~/Documents/superhero_projects$
```

Bash Scripts

- **Definition:** Bash scripts are files containing a series of commands that the shell executes. They allow for automating repetitive tasks, managing system operations, and more.
- **Advantages:**
 - Automation of repetitive tasks
 - Consistency and Accuracy
 - Efficient System Management
 - Scheduling and Background Tasks
 - Customization and Flexibility

Helping streamline operations and enhance the functionality and manageability of systems.



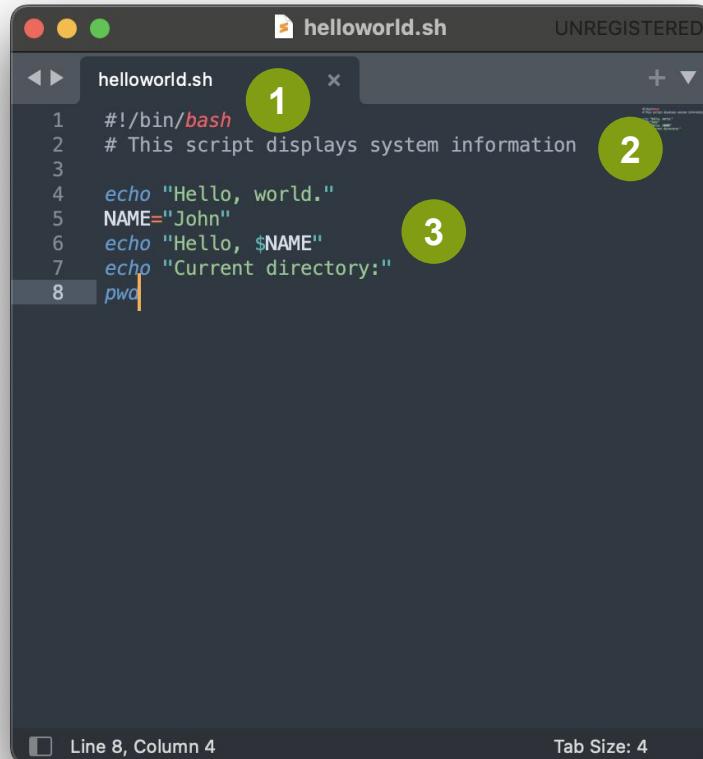
generated by DALL·E

How to write a bash script?

Create Script

- Open a basic text/code editor like sublime.
- 1** Shebang (#!): This line starts with #! followed by the path to the Bash interpreter (/bin/bash), telling the system this is a script and specifying the interpreter to use.
- 2** Comments - Start with # and explain what the script or parts of the script do. Helpful for documentation and maintenance
- 3** Actual Code
- Save file with “helloworld.sh” extension in e.g. your “Documents” home directory (~/Documents)

Simple Script



```

1 #!/bin/bash
2 # This script displays system information
3
4 echo "Hello, world."
5 NAME="John"
6 echo "Hello, $NAME"
7 echo "Current directory:"
8 pwd

```

Line 8, Column 4 Tab Size: 4

How to run a bash script? - Permission

chmod [permission][file]

Modify file or folder permissions:

Permissions:

- r - Read permission.
- w - Write permission.
- x - Execute permission.

User groups:

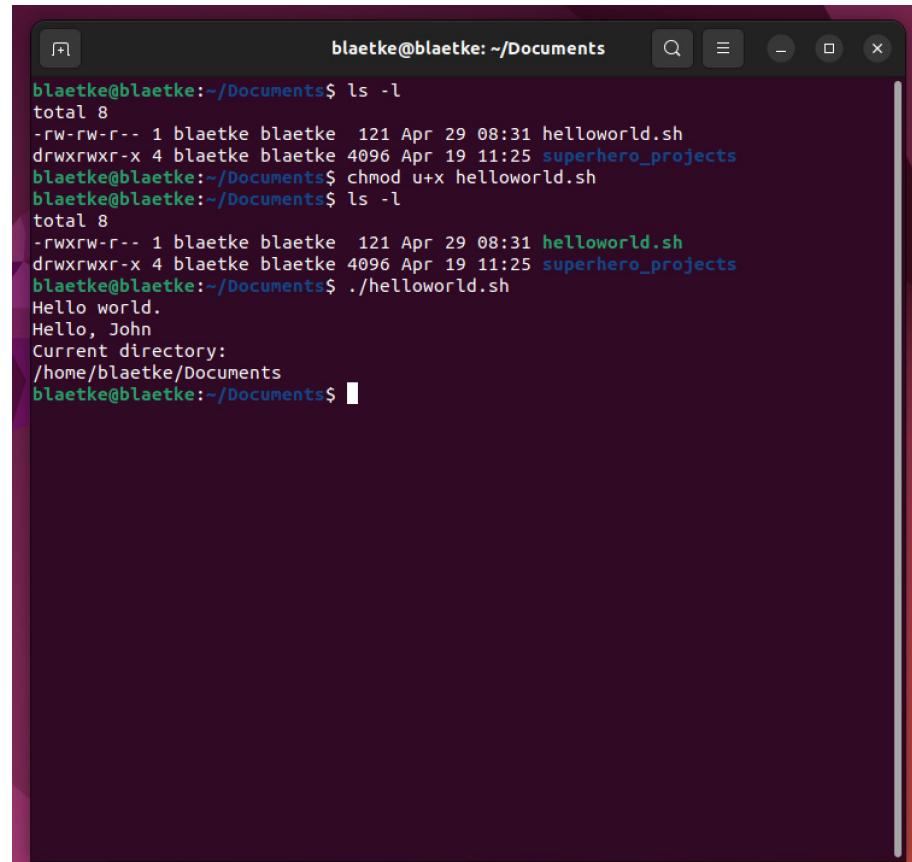
- u - User (owner of the file).
- g - Group (users who are in the file's group).
- o - Others (everyone else).
- a - All (all of the above: user, group, and others).

Combine file, permissions and user groups using symbolic links: + (add), - (remove), = (set exact)

`chmod u+x file.txt` - execute permission for user

`chmod o+w file.txt` - write permission for other users

`chmod g=rx file.txt` - read, execute permission for group



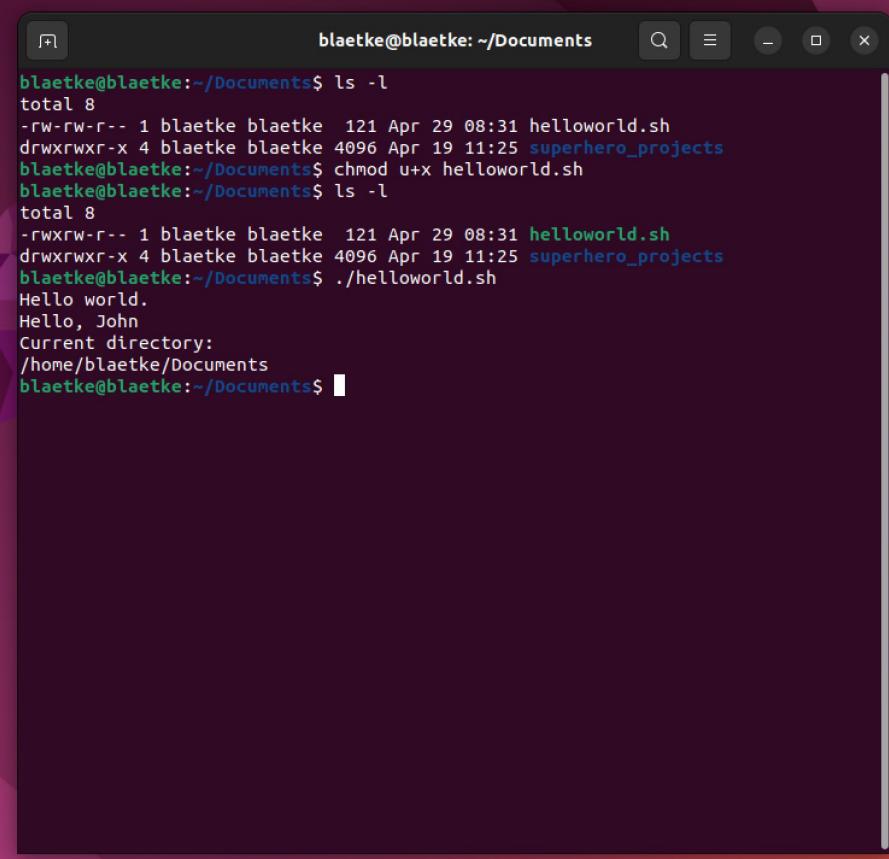
A screenshot of a terminal window titled "blaetke@blaetke: ~/Documents". The terminal shows the following session:

```
blaetke@blaetke:~/Documents$ ls -l
total 8
-rw-rw-r-- 1 blaetke blaetke 121 Apr 29 08:31 helloworld.sh
drwxrwxr-x 4 blaetke blaetke 4096 Apr 19 11:25 superhero_projects
blaetke@blaetke:~/Documents$ chmod u+x helloworld.sh
blaetke@blaetke:~/Documents$ ls -l
total 8
-rwxrwxr-x 1 blaetke blaetke 121 Apr 29 08:31 helloworld.sh
drwxrwxr-x 4 blaetke blaetke 4096 Apr 19 11:25 superhero_projects
blaetke@blaetke:~/Documents$ ./helloworld.sh
Hello world.
Hello, John
Current directory:
/home/blaetke/Documents
blaetke@blaetke:~/Documents$
```

How to run a bash script? - Run

./script.sh

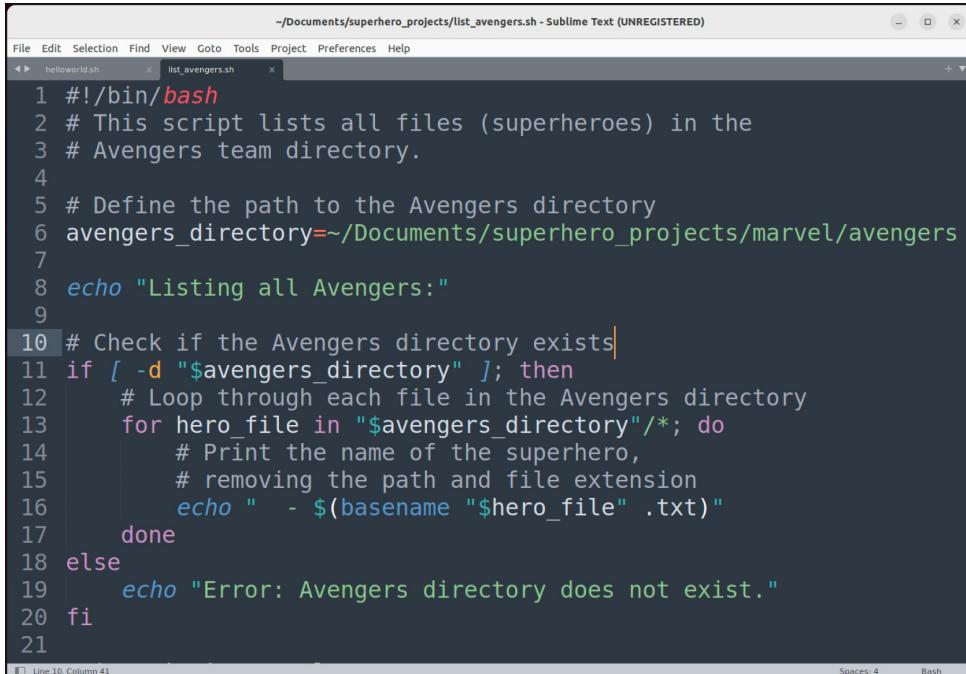
- The system looks in certain predefined locations to find and execute that command. These locations are specified in an environment variable called **PATH**.
- Current Directory is by default not in **PATH** as a security measure to prevent accidental or malicious execution of scripts or programs.
- Using **./** tells the system to look in the current directory "Execute the script that's right here in front of me, not anywhere else."
- Using **./** before a script ensures you are running the script you intend to, directly from the location you are in. This adds clarity and security to your command executions.



```
blaetke@blaetke:~/Documents$ ls -l
total 8
-rw-rw-r-- 1 blaetke blaetke 121 Apr 29 08:31 helloworld.sh
drwxrwxr-x 4 blaetke blaetke 4096 Apr 19 11:25 superhero_projects
blaetke@blaetke:~/Documents$ chmod u+x helloworld.sh
blaetke@blaetke:~/Documents$ ls -l
total 8
-rwxrwxr-- 1 blaetke blaetke 121 Apr 29 08:31 helloworld.sh
drwxrwxr-x 4 blaetke blaetke 4096 Apr 19 11:25 superhero_projects
blaetke@blaetke:~/Documents$ ./helloworld.sh
Hello world.
Hello, John
Current directory:
/home/blaetke/Documents
blaetke@blaetke:~/Documents$
```

Exercise

Example



The screenshot shows a Sublime Text window with two tabs: "helelloworld.sh" and "list_avengers.sh". The "list_avengers.sh" tab is active and contains the following bash script:

```
#!/bin/bash
# This script lists all files (superheroes) in the
# Avengers team directory.
#
# Define the path to the Avengers directory
avengers_directory=~/Documents/superhero_projects/marvel/avengers
echo "Listing all Avengers:"
#
# Check if the Avengers directory exists
if [ -d "$avengers_directory" ]; then
    # Loop through each file in the Avengers directory
    for hero_file in "$avengers_directory"/*; do
        # Print the name of the superhero,
        # removing the path and file extension
        echo "- $(basename "$hero_file" .txt)"
    done
else
    echo "Error: Avengers directory does not exist."
fi
```

Line 10, Column 41

Spaces: 4 Bash

Questions

Open a text editor like sublime and write the bash code on the left. Try to understand what it is doing.

Save the bash script in the **superhero_projects** directory as **list_avengers.sh**.

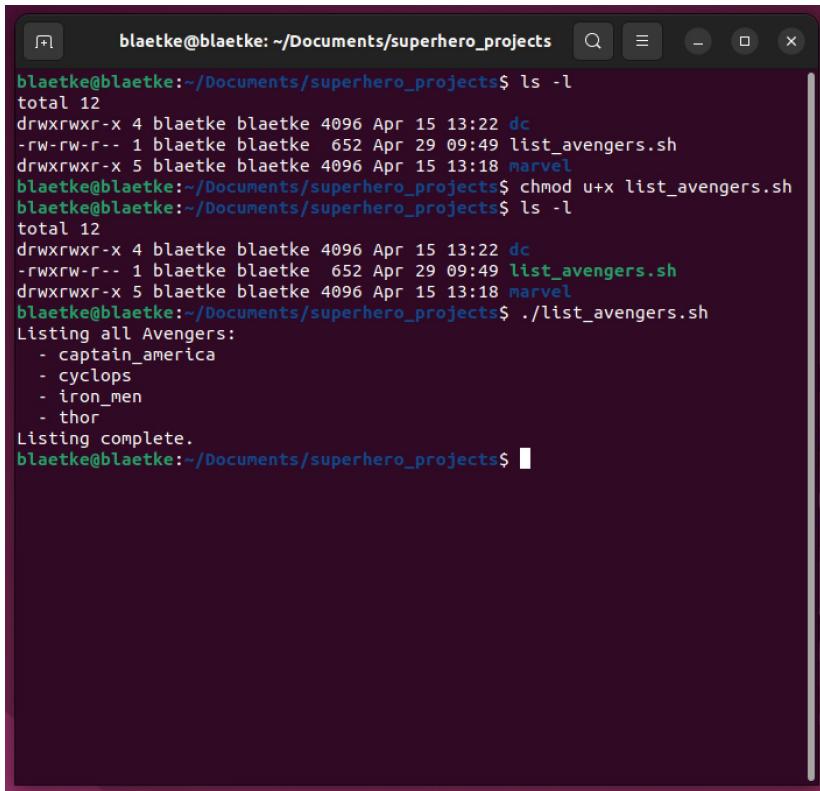
Open the terminal, go to the **superhero_projects** directory and change the permission for the "user" to "execute" the file, use the **chmod** command.

Use **ls -l** to check if the file permissions.

Type **./list_avengers.sh** to execute the script.

Exercise

Example



```
blaetke@blaetke: ~/Documents/superhero_projects$ ls -l
total 12
drwxrwxr-x 4 blaetke blaetke 4096 Apr 15 13:22 dc
-rw-rw-r-- 1 blaetke blaetke 652 Apr 29 09:49 list_avengers.sh
drwxrwxr-x 5 blaetke blaetke 4096 Apr 15 13:18 marvel
blaetke@blaetke:~/Documents/superhero_projects$ chmod u+x list_avengers.sh
blaetke@blaetke:~/Documents/superhero_projects$ ls -l
total 12
drwxrwxr-x 4 blaetke blaetke 4096 Apr 15 13:22 dc
-rwxrwxr-- 1 blaetke blaetke 652 Apr 29 09:49 list_avengers.sh
drwxrwxr-x 5 blaetke blaetke 4096 Apr 15 13:18 marvel
blaetke@blaetke:~/Documents/superhero_projects$ ./list_avengers.sh
Listing all Avengers:
- captain_america
- cyclops
- iron_men
- thor
Listing complete.
blaetke@blaetke:~/Documents/superhero_projects$
```

Questions

Open a text editor like sublime and write the bash code on the left. Try to understand what it is doing.

Save the bash script in the `superhero_projects` directory as `list_avengers.sh`.

Open the terminal, go to the `superhero_projects` directory and change the permission for the “user” to “execute” the file, use the `chmod` command.

Use `ls -l` to check if the file permissions.

Type `./list_avengers.sh` to execute the script.

Key Takeaways

1. **Bash Basics Mastered:** You've learned the core functionalities of the Bash shell, including file management, navigation, and basic scripting.
2. **Automation Skills:** You can now automate repetitive tasks, enhancing productivity and ensuring accuracy.
3. **Scripting Proficiency:** You've gained the skills to write and execute your own Bash scripts, enabling you to handle complex tasks efficiently.
4. **Practical Experience:** Through hands-on examples and exercises, like managing our Superhero project, you've applied what you've learned in real-world scenarios.



generated by DALL·E

Links

[Explainshell.com](https://explainshell.com)

[GitHub - Introduction to Bash Scripting](https://github.com/gitignore/Bash-Scripting-Ignore-File)

[GitHub - Bash Handbook](https://github.com/koenighaus/bash-handbook)

[GitHub - Bash Bible](https://github.com/koenighaus/bash-bible)

[GitHub - Awesome Bash](https://github.com/koenighaus/awesome-bash)

