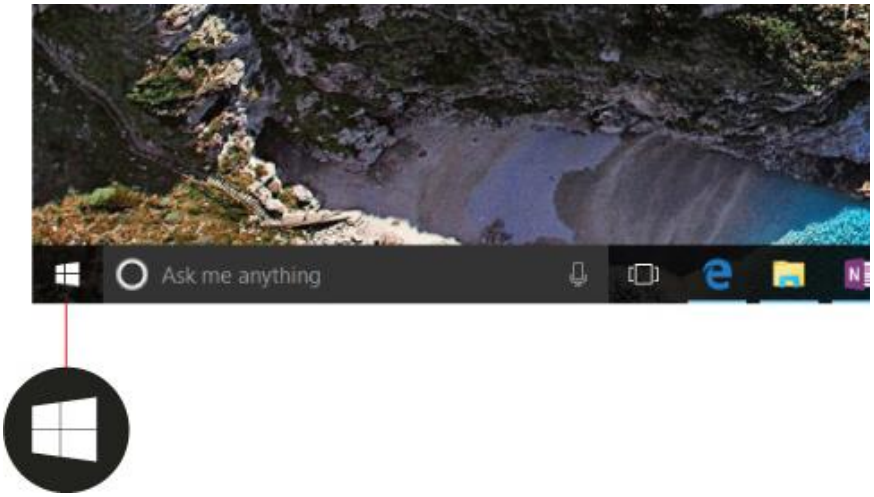
Working with the Command Prompt

Content

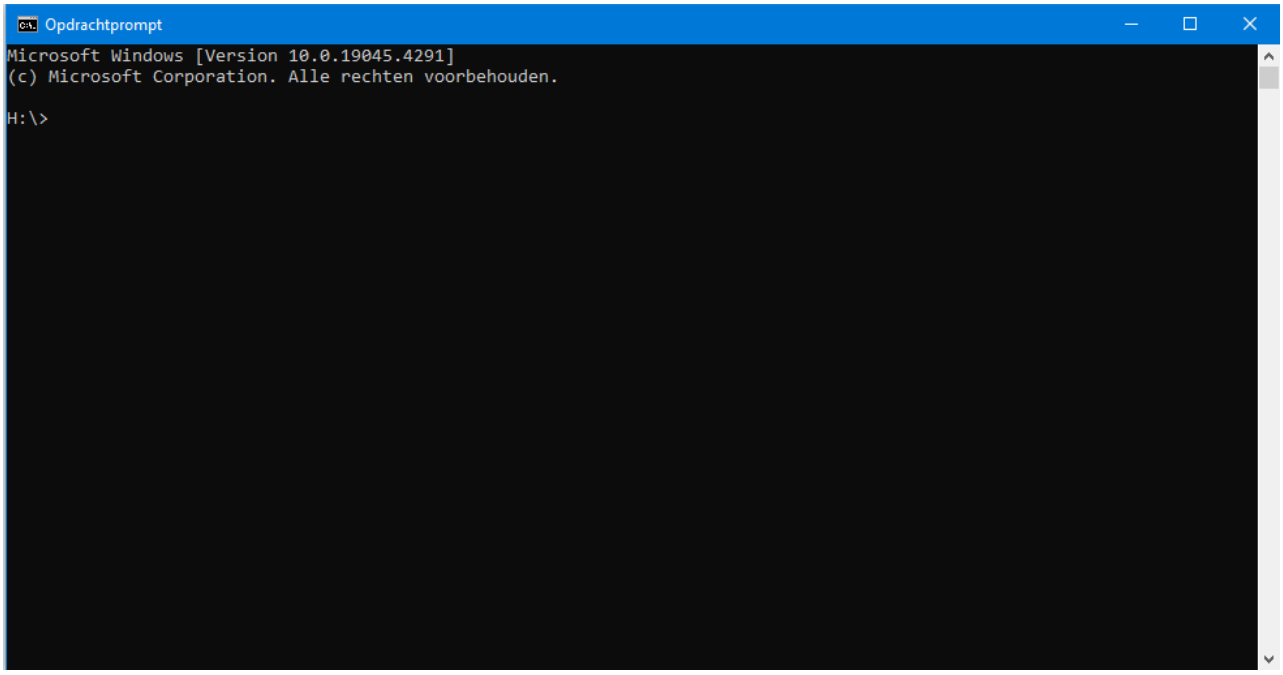
Disclaimer: for the guide and demo I have used the Windows Command Prompt. I have added a table at the end with each command in Windows/Linux/Mac.

How to start the command prompt?

To start the command prompt on Windows, you open the start menu on the left of your taskbar.



Once opened, you type ‘cmd’. The result is the command prompt which you can then click on to open. Once opened, it looks like this:

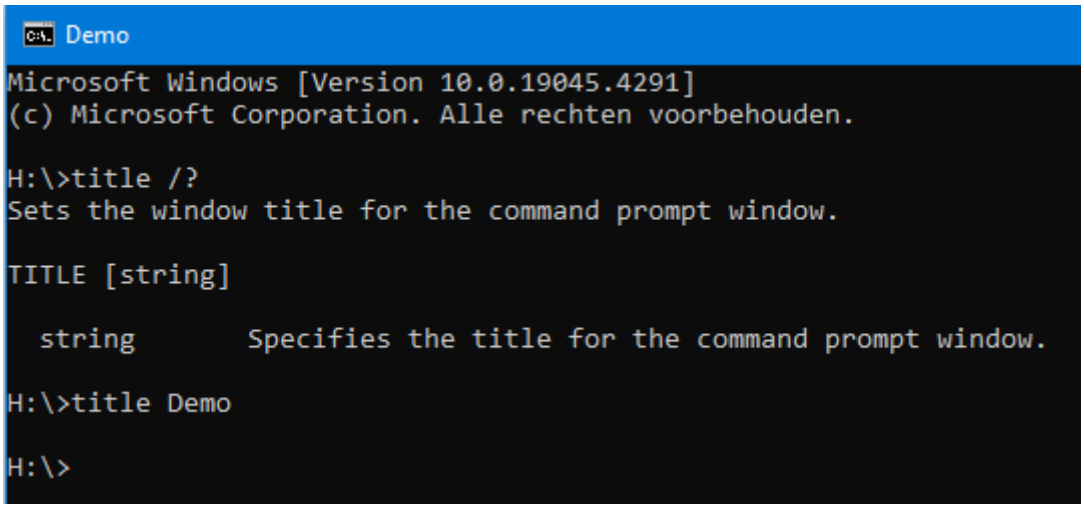


Changing the title and color

To start, you can try and change the title of the window to give it a more meaningful name. You do this using the `title` command. To know what a command does, you can type in: `title /?`. Do not forget to press ENTER after each command.

By doing this, you get an explanation of the command. Furthermore, you also get an example of how to use it. In the case of *title*, you get: `TITLE [string]`. The string specifies the title for the command prompt window.

To change the title of the window to Demo you type in the following: `title Demo`.



Next, we can change the color of the window and the letters by using the `color` command. We start again with using `/?` so we can get more guidance on how to use this command.

The color command sets the default console foreground and background colors. The use of it is: `COLOR [attr]`. `[attr]` is what specifies the color attribute of console output. These attributes are specified by two hex digits: the first corresponds to the background, the second to the foreground. Then a list is given with options.

To get a bright white background with red letters: `color fc`

To go back to a black background with white letters: `color 07`

```
H:\>color /?
Sets the default console foreground and background colors.

COLOR [attr]

  attr        Specifies color attribute of console output

Color attributes are specified by TWO hex digits -- the first
corresponds to the background; the second the foreground.  Each digit
can be any of the following values:

    0 = Black       8 = Gray
    1 = Blue        9 = Light Blue
    2 = Green       A = Light Green
    3 = Aqua        B = Light Aqua
    4 = Red         C = Light Red
    5 = Purple      D = Light Purple
    6 = Yellow      E = Light Yellow
    7 = White       F = Bright White

If no argument is given, this command restores the color to what it was
when CMD.EXE started.  This value either comes from the current console
window, the /T command line switch or from the DefaultColor registry
value.

The COLOR command sets ERRORLEVEL to 1 if an attempt is made to execute
the COLOR command with a foreground and background color that are the
same.

Example: "COLOR fc" produces light red on bright white
```

Echo

As the people following the Python track are familiar with, the first step is always to print the message “Hello World”. Instead of using the `print` function, we need to use the `echo` command in the command prompt.

To get Hello World, type: `echo Hello World`

```
H:\>echo /?
Displays messages, or turns command-echoing on or off.

  ECHO [ON | OFF]
  ECHO [message]

Type ECHO without parameters to display the current echo setting.

H:\>echo Hello World
Hello World
```

Knowing where you are

dir command

In the command prompt, you can go into different folders (also called directories) and navigate your way around. As you can see in the previous screenshots, I am currently in my H drive (H:\). I want to go into a directory, but I am not sure which directories there are in my H drive. To find out, I use the `dir` command. This command can be used to display a list of files and subdirectories. When you type in `dir /?` you get an overview of all the different options available to use with this command. The following are useful:

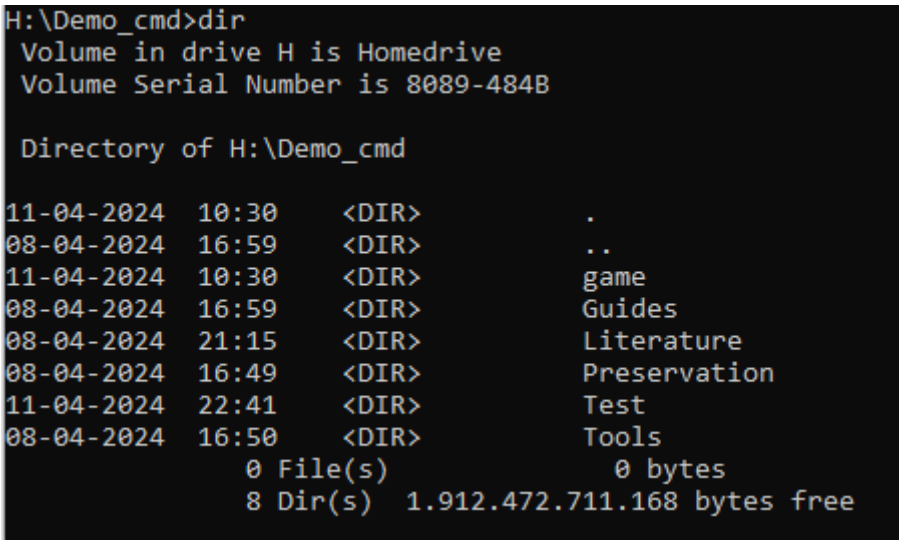
Command	Explanation
dir	Displays the list of files and subdirectories in a directory.
dir /B	Uses bare format (no heading information or summary).
dir /S	Displays files in specified directory and all subdirectories.
dir /Q	Display the owner of the file.

```
H:\>dir /?
Displays a list of files and subdirectories in a directory.

DIR [drive:][path][filename] [/A[[:]attributes]] [/B] [/C] [/D] [/L] [/N]
  [/O[[:]sortorder]] [/P] [/Q] [/R] [/S] [/T[[:]timefield]] [/W] [/X] [/4]

  [drive:][path][filename]
              Specifies drive, directory, and/or files to list.

  /A          Displays files with specified attributes.
  attributes   D  Directories                R  Read-only files
               H  Hidden files               A  Files ready for archiving
               S  System files               I  Not content indexed files
               L  Reparse Points             O  Offline files
               -  Prefix meaning not
  /B          Uses bare format (no heading information or summary).
  /C          Display the thousand separator in file sizes.  This is the
              default.  Use /-C to disable display of separator.
  /D          Same as wide but files are list sorted by column.
  /L          Uses lowercase.
  /N          New long list format where filenames are on the far right.
  /O          List by files in sorted order.
  sortorder    N  By name (alphabetic)       S  By size (smallest first)
               E  By extension (alphabetic)  D  By date/time (oldest first)
               G  Group directories first    -  Prefix to reverse order
  /P          Pauses after each screenful of information.
  /Q          Display the owner of the file.
  /R          Display alternate data streams of the file.
  /S          Displays files in specified directory and all subdirectories.
  /T          Controls which time field displayed or used for sorting
  timefield   C  Creation
              A  Last Access
              W  Last Written
  /W          Uses wide list format.
  /X          This displays the short names generated for non-8dot3 file
              names.  The format is that of /N with the short name inserted
              before the long name. If no short name is present, blanks are
              displayed in its place.
  /4          Displays four-digit years

Switches may be preset in the DIRCMD environment variable.  Override
preset switches by prefixing any switch with - (hyphen)--for example, /-W.

H:\>
```

Lets see how they all work. For this demo, I have created a directory with several subdirectories in it: demo_cmd. Lets try some options of the `dir` command here.

dir

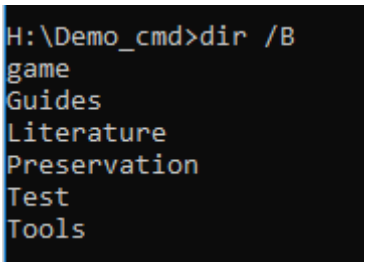
We start by just using the `dir` command.



```
H:\Demo_cmd>dir
 Volume in drive H is Homedrive
 Volume Serial Number is 8089-484B

 Directory of H:\Demo_cmd

11-04-2024  10:30    <DIR>          .
08-04-2024  16:59    <DIR>          ..
11-04-2024  10:30    <DIR>          game
08-04-2024  16:59    <DIR>          Guides
08-04-2024  21:15    <DIR>          Literature
08-04-2024  16:49    <DIR>          Preservation
11-04-2024  22:41    <DIR>          Test
08-04-2024  16:50    <DIR>          Tools
               0 File(s)              0 bytes
               8 Dir(s)  1.912.472.711.168 bytes free
```

dir /B

dir /B gives you the bare format without heading information or summary.



```
H:\Demo_cmd>dir /B
game
Guides
Literature
Preservation
Test
Tools
```

dir /S

```
H:\Demo_cmd>dir /S
 Volume in drive H is Homedrive
 Volume Serial Number is 8089-484B

 Directory of H:\Demo_cmd

11-04-2024  10:30    <DIR>          .
08-04-2024  16:59    <DIR>          ..
11-04-2024  10:30    <DIR>          game
08-04-2024  16:59    <DIR>          Guides
08-04-2024  21:15    <DIR>          Literature
08-04-2024  16:49    <DIR>          Preservation
11-04-2024  22:41    <DIR>          Test
08-04-2024  16:50    <DIR>          Tools
               0 File(s)              0 bytes

 Directory of H:\Demo_cmd\game

11-04-2024  10:30    <DIR>          .
11-04-2024  10:30    <DIR>          ..
               0 File(s)              0 bytes

 Directory of H:\Demo_cmd\Guides

08-04-2024  16:59    <DIR>          .
11-04-2024  10:30    <DIR>          ..
15-03-2024  12:43           933.344 HTML-guide.pdf
15-03-2024  12:19         1.479.126 Python-guide.pdf
               2 File(s)      2.412.470 bytes
```

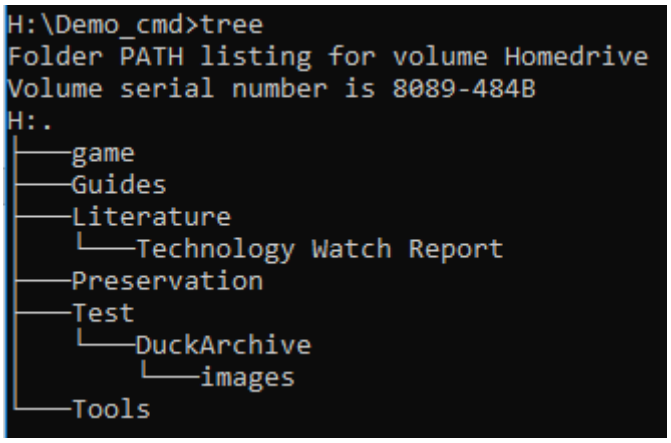
Note that this is not the entire result, just a part of it. As you can see, you get an overview of the subdirectories and the files in it, but it also checks those subdirectories and gives you an overview of that.

dir /Q

dir /Q also gives you the owner. In my case, this is not very interesting because I created everything. But as archivists, we receive collections from others all the time, so this command could be handy.

Tree command

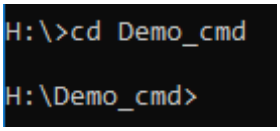
You could also want a more visual view of the subdirectories and files. Then you can use the tree command.



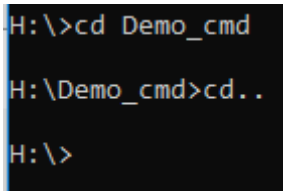
However, this does not show the files. It just shows the directories and subdirectories.

Change directories

As you can see in the previous screenshot I am currently in my H drive in the subdirectory Demo_cmd. How did I get here? By using the change directories command: `cd [directory you want to go in]`. Cd stands for changing directories.



I can also go back by using the `cd` command, but with the addition of two dots: `cd..`



You can also change drives. I am currently in my H drive, but I also have a C drive. To change drive, you simple type in `[DRIVE]:`. For me, this is: `C:`

```
H:\>C:

C:\>
```

Renaming a file

In H:\Demo_cmd\Test I have a file named B.txt.

```
H:\Demo_cmd\Test>dir
 Volume in drive H is Homedrive
 Volume Serial Number is 8089-484B

 Directory of H:\Demo_cmd\Test

11-04-2024  22:41    <DIR>          .
11-04-2024  10:30    <DIR>          ..
08-04-2024  18:03                10 B.txt
11-04-2024  10:54                95 checksum_script.bat
08-04-2024  21:22                11 deletethis.txt
08-04-2024  22:05    <DIR>          DuckArchive
04-04-2024  23:49             1.227 Jokes.py
               4 File(s)          1.343 bytes
               3 Dir(s)  1.850.389.676.032 bytes free
```

I want to change B.txt to A.txt. I need to rename it. I can do this using the `ren` command.

```
H:\Demo_cmd\Test>ren /?
Renames a file or files.

RENAME [drive:][path]filename1 filename2.
REN [drive:][path]filename1 filename2.

Note that you cannot specify a new drive or path for your destination file.
```

I can change the name by typing in the following: `ren [old name file] [new name file]`

In this case: `ren B.txt A.txt`

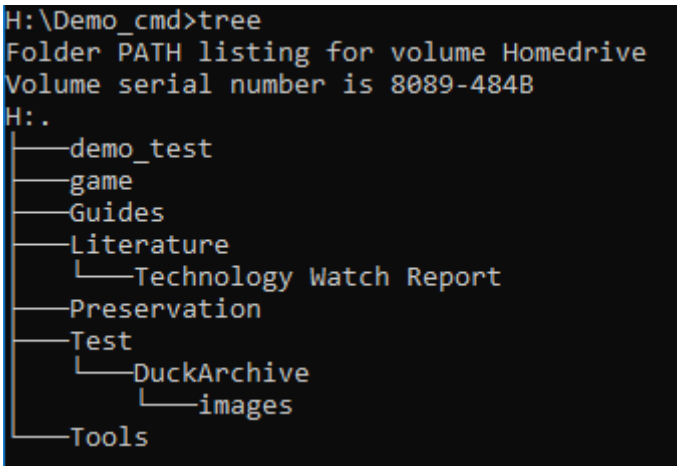
I do not need to specify the file path here because it is all happening where I currently am: H:\Demo_cmd\Test

However, here it is with the file paths included: `ren  H:\Demo_cmd\Test\B.txt H:\Demo_cmd\Test\A.txt`

Making a new directory

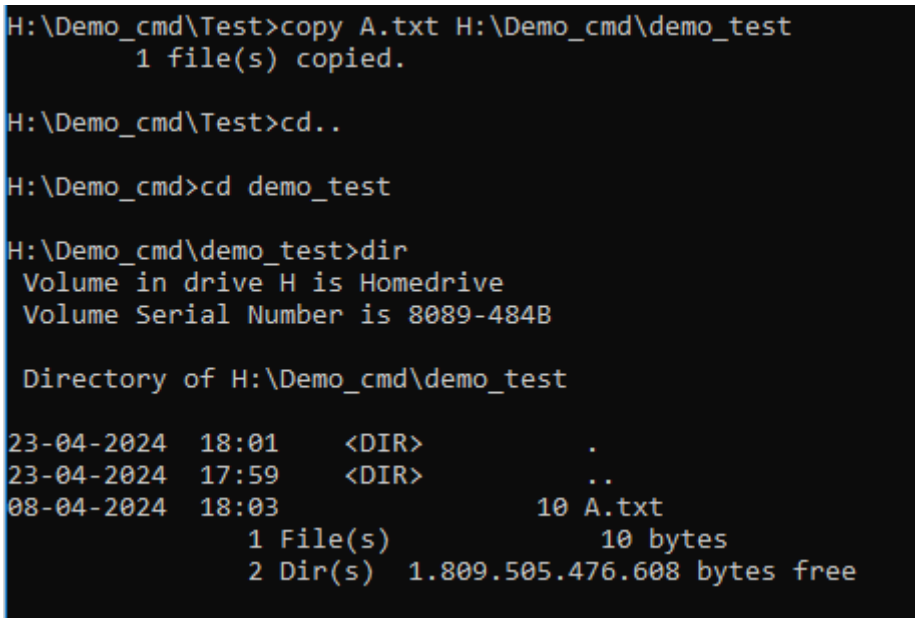
If I want to create a new directory in my demo_cmd I can do this using the mkdir command. To create a new directory I type: `mkdir demo_test`

To see if it worked, you can use the dir or tree command again.



Copy a file

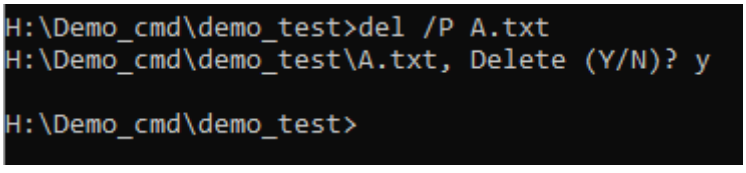
If I want to copy A.txt (which is in my Test directory) to the new demo_test directory, I can use the `copy` command. This leads to the following: `copy A.txt H:\Demo_cmd\demo_test`.



Deleting a file

To delete a file, you can use the `del` command. However, it is recommended to also add `/P` to your command. `/P` gives you a prompt for confirmation before deleting each file. This is useful so you have an extra check built in.

To delete the file I just copied: `del /P A.txt`

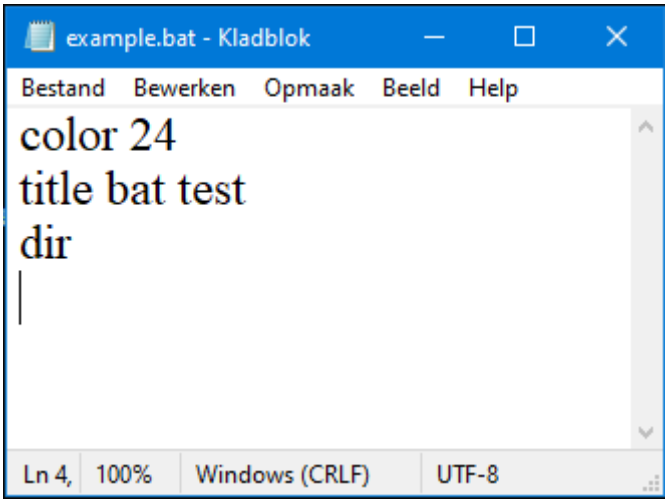


In the screenshot you can see the prompt, where I need to type in Y or N for it to do it, or stop it.

Executing a batch script

A batch script is a series of command to be executed by the command-line interpreter, stored in a plain text file. This is useful if you have a workflow for example. If I want to change the title, change the color, copy a file, and delete a file all in one go without typing these in all the time.

To create a batch script, you simply open your notepad and type in the command (one on each line). Then you save it as a .bat file.



After creating this script, I go into the directory where the bat script resides and type in the name of the file: `example.bat`

This leads to the command prompt doing these command one after the other without me having to type them in individually.

Executing a Python or HTML script

Executing a Python script is almost the same. You simply go to where the file is and type in the name of the file.

```
H:\Demo_cmd\Test>jokes.py
Q: Where do cows who love history go?
```

You can also execute the file from somewhere else, just do not forget to type in the complete path.

```
H:\>H:\Demo_cmd\Test\jokes.py
Q: Where do cows who love history go?
```

For HTML this all works the same. Just type in the file (example: DuckArchive.HTML). With HTML files, this will start up in your browser which automatically gets booted up for you after entering the command.

Stopping a task

When starting the Jokes.py program or when executing the tree command on your complete H drive (may take it a while), you might want to stop the task. You can of course close the window, but this is of course not the proper way. The proper way is to use `ctrl + C`.

Start Wordpad

To start up Wordpad using your command prompt, you simple type `write`.

Start Word

For Word, I had to go into my C drive since that is where the program is. To change Drives, just type in the Drive you want to go to and it works. For example, I want to go from the H: Drive to the C: Drive:

```
H:\>C:

C:\>
```

I need to access the file Winword.exe. However, just typing this in like we did with Python and HTML files will not suffice. You also need to write start.

To start windows: `start winword.exe`

Calculating a checksum

Checksums are values that are generated from transmitted data before and after transmission. They are a sort of fingerprint for your file. If the checksum stays the same, you know the file is the same. If anything has changed, the checksum too will have changed. This can indicate bit rot or an error that happened during migration.

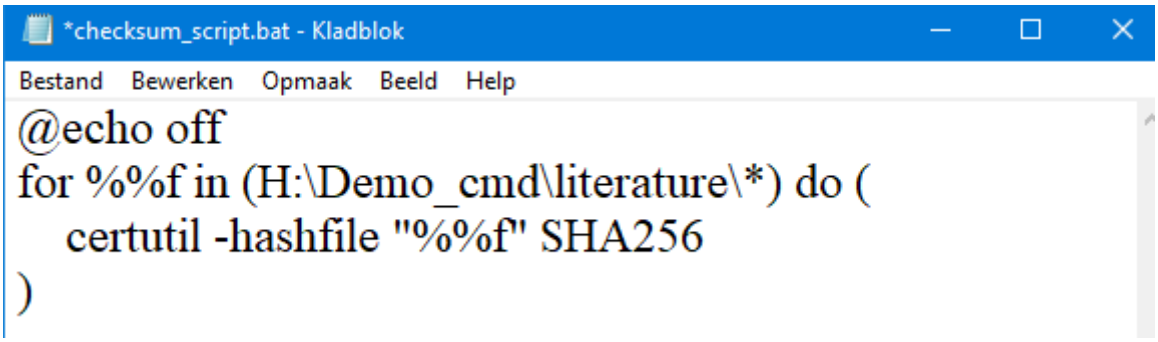
The command to calculate a checksum is certutil -hashfile

Calculate an MD5 checksum: `certutil -hashfile [file name] MD5`

Calculate a SHA512 checksum: `certutil -hashfile [file name] MD5`

```
H:\Demo_cmd\Test>certutil -hashfile A.txt MD5
MD5 hash of A.txt:
cd18203adcdc4404664fea34541d8717
CertUtil: -hashfile command completed successfully.
```

You can also calculate a checksum on an entire folder using a batch script:



Here is a breakdown of the script:

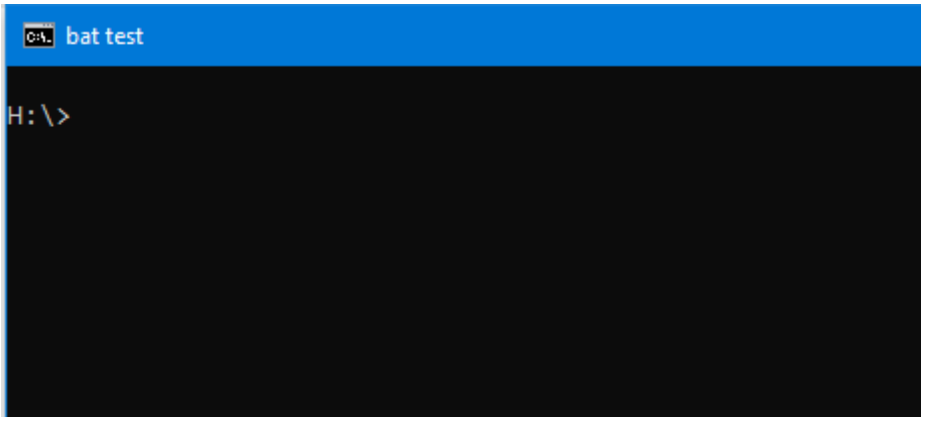
- **@echo off**
  This line turns off the command echoing in the batch script, meaning that commands themselves won't be printed to the console when they are executed.
- **for %%f in (H:\Demo_cmd\literature*) do (...)**
  This line sets up a loop that iterates over each file (%%f) in the directory specified (H:\Demo_cmd\literature\). The * is a wildcard character that matches any file in that directory.
- **certutil -hashfile "%%f" SHA256**
  Within the loop, this command calculates the SHA256 hash value for each file (%%f) found in the directory specified. certutil is a command-line utility that performs various cryptographic operations, including hashing files. -hashfile is an option that tells certutil to hash a specified file. "%%f" is the file being hashed, and SHA256 specifies that the hash algorithm used should be SHA256.

In short, this script goes through each file in the specified directory and calculates its SHA256 hash value using the `certutil` command.

Note that the certutil command can do a lot more. To see all the options, type `certutil /?`

Cleaning the window

If you enter multiple commands, your window can get full and a bit cluttered. To get a clean state, you can use the clean screen command: `cls`



Closing the command prompt

To close the command prompt, you can close the window by clicking the x on the top right. But the more correct way is to use the `exit` command. Simply type `exit`, press ENTER, and the window will close.

Overview commands

Here are the commands in order of usage in this document.

Command	Explanation
title	Sets the window title for the command prompt window.
color	The color command sets the default console foreground and background colors.
echo	Displays messages.
dir	Displays the list of files and subdirectories in a directory.
dir /B	Uses bare format (no heading information or summary).
dir /S	Displays files in specified directory and all subdirectories.
dir /Q	Display the owner of the file.
tree	Graphically displays the folder structure of a drive or path.
ren	Renames a file or files.
mkdir	Creates a directory.
copy	Copies one or more files to another location.
del	Deletes one or more files.
write	Open Wordpad
cls	Clears the screen.
exit	Closes the window.

Overview Windows/Linux/macOS

Windows	Linux	macOS
title	There isn't a direct equivalent, but you can use terminal titles by changing the PS1 variable in your shell configuration file (like .bashrc or .zshrc). For example, you can set export PS1="\[\033]0;Your Title Here\007\]\u@\h:\w\$ " in your .bashrc for Bash.	
color	The terminal colors are typically controlled by the terminal emulator rather than by shell commands. You can change terminal colors within the settings of your terminal emulator.	
echo	echo	echo
dir	ls	ls
dir /B	Use ls with the -1 (dash one) option to list files and directories in bare format, each on a new line.	
dir /S	Use ls with the -1 (dash one) option to list files and directories in bare format, each on a new line.	
dir /Q	There isn't a direct equivalent for displaying ownership information with ls alone. You can use ls -l to display permissions and ownership details.	
tree	You need to install the tree command if it's not already available. On macOS, you can use Homebrew to install it (brew install tree). Once installed, you can use tree to display directory trees.	
ren	Use the mv command to rename files and directories	
mkdir	mkdir	mkdir
copy	Use the cp command to copy files and directories.	
del	Use the rm command to delete files and directories.	
write	In Windows, the write command is used to write directly to a serial port. If you're looking for a similar functionality in macOS/Linux, you can use the echo command in combination with the appropriate device file representing the serial port. For example, if you want to write to the serial port /dev/ttyS0 (COM1 in Windows), you can use echo like this: echo "Your message here" > /dev/ttyS0 Replace /dev/ttyS0 with the appropriate device file for your serial port. You might need appropriate permissions to write to the serial port device file, so you may need to use sudo or adjust the permissions accordingly.	

cls	clear	clear
exit	exit	exit