

java from scratch Knowledge Base

- Welcome
- Mastering Agile and Scrum: Video Training Series
- Program
- Introduction
- The architecture of an operating system
- The structure of files and directories
- Navigating through directories
 - Environment variables
 - Extracting archives
 - Installing the software
 - Monitoring the usage of system resources
 - Ending – control questions
- Software Installations
 - IntelliJ EduTools – installation
- Introduction
- A brief history of Java
- First program
- Types of data
- Operators
- Conditional statements

| java from scratch Knowledge Base

Navigating through directories

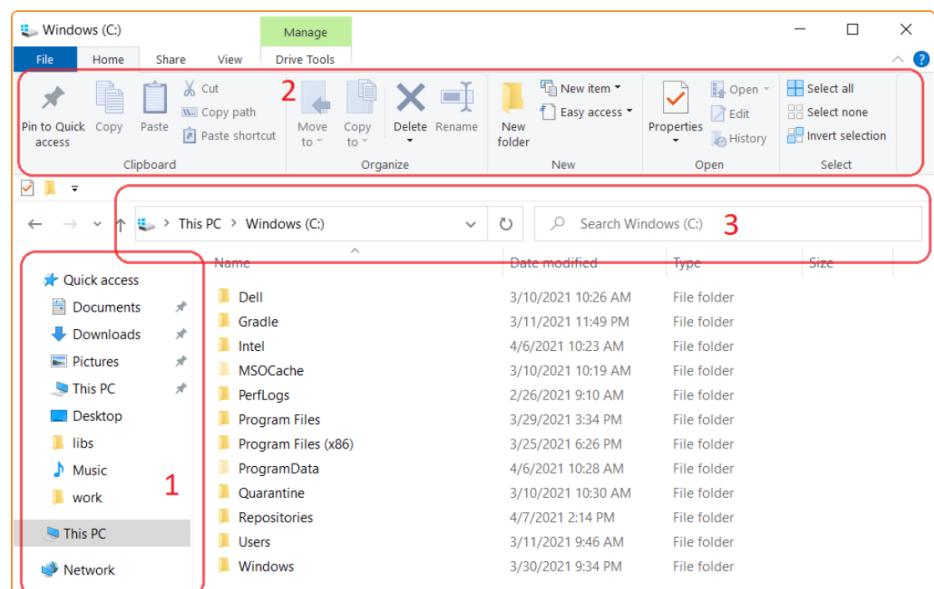
One of the most frequently performed operations in operating systems are the ones done on files and directories. Those actions may include creating, editing, or deleting files and directories. However, before we move on to file operations, you should navigate through directories in a relatively efficient way. There are two ways to accomplish this:

- via the graphical user interface
- via a text interface (by using a command line terminal)

Graphical User Interface

Navigating the graphical user interface is very intuitive. The user switches between directories using the keyboard or mouse and specialized auxiliary icons. In Windows, the user can:

- use shortcuts to the most frequently used directories (Quick Access) – 1
- use shortcuts in the toolbar (moving, copying, deleting, opening, editing, changing the properties of files and directories) – 2
- use the search engine – 3

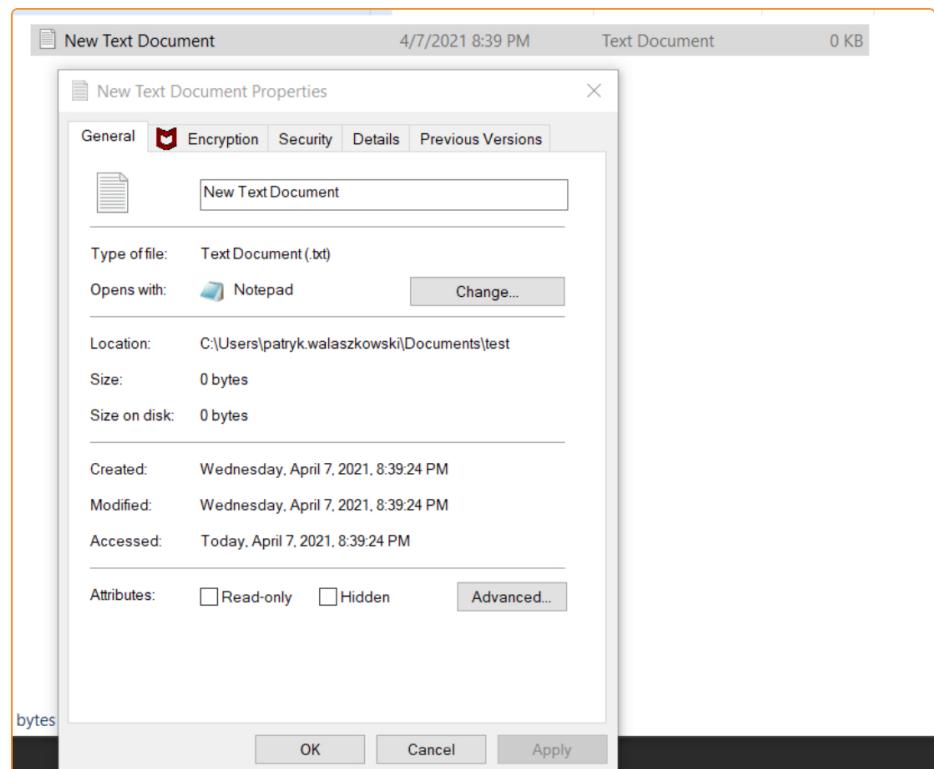


- Loops
- Arrays
- Object-oriented programming
- Conclusion
- Assignments
- Basics of GIT – video training
- HTTP basics – video training
- Design patterns and good practices video course
- Prework Primer: Essential Concepts in Programming
- Cybersecurity Essentials: Must-Watch Training Materials
- Java Developer – introduction
- Java Fundamentals – coursebook
- Java fundamentals slides
- Java fundamentals tasks
- Test 1st attempt | after the block: Java fundamentals
- Test 2nd attempt | after the block: Java fundamentals
- GIT version control system coursebook
- Java – Fundamentals: Coding slides
- Java fundamentals tasks
- Software Testing slides
- Software Testing Coursebook
- Software Testing tasks
- Test 1st attempt | after the block: Software testing
- Test 2nd attempt | after the block: Software testing
- Java – Advanced Features coursebook

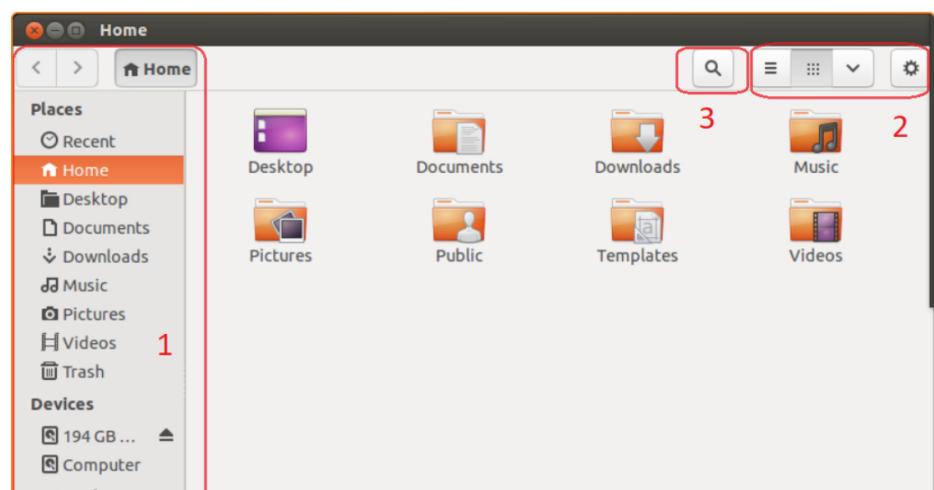


The user can view the properties of a file/directory through the context menu (Right Mouse Button → Properties or Properties – depending on the language version of the operating system), here you can preview such items as:

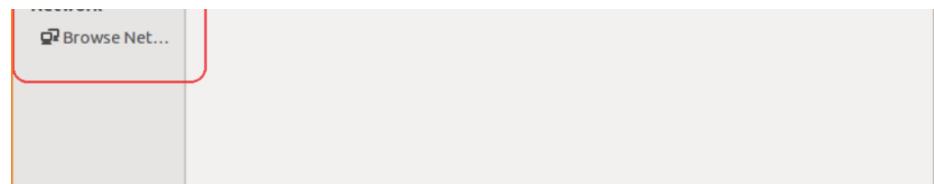
- the file type/extension – including which computer program supports it
- the file/directory location
- the file/directory size
- the date of creation/modification/last access
- the attributes – whether it's a read-only file or hidden file



Navigating the graphical interface in Linux is very similar (note: graphical interfaces in Linux systems may be different!)

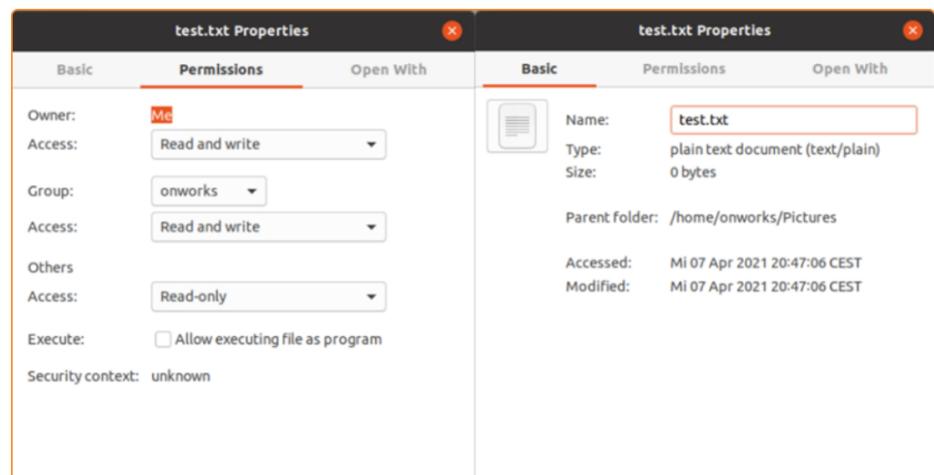


- Java – Advanced Features slides
- Java – Advanced Features tasks
- Test 1st attempt | after the block: Java Advanced Features
- Test 2nd attempt | after the block: Java Advanced Features
- Java – Advanced Features: Coding slides
- Java – Advanced Features: Coding tasks
- Test 1st attempt | after the block: Java Advanced Features coding
- Test 2nd attempt | after the block: Java Advanced Features coding
- Data bases SQL coursebook
- Databases SQL slides
- Databases – SQL tasks
- Coursebook: JDBC i Hibernate
- Exercises: JDBC & Hibernate
- Test 1st attempt | after the block: JDBC
- Test 2nd attempt | after the block: JDBC
- Design patterns and good practices
- Design patterns and good practices slides
- Design Patterns & Good Practices tasks
- Practical project coursebook
- Practical project slides
- HTML, CSS, JAVASCRIPT Coursebook
- HTML, CSS, JAVASCRIPT slides

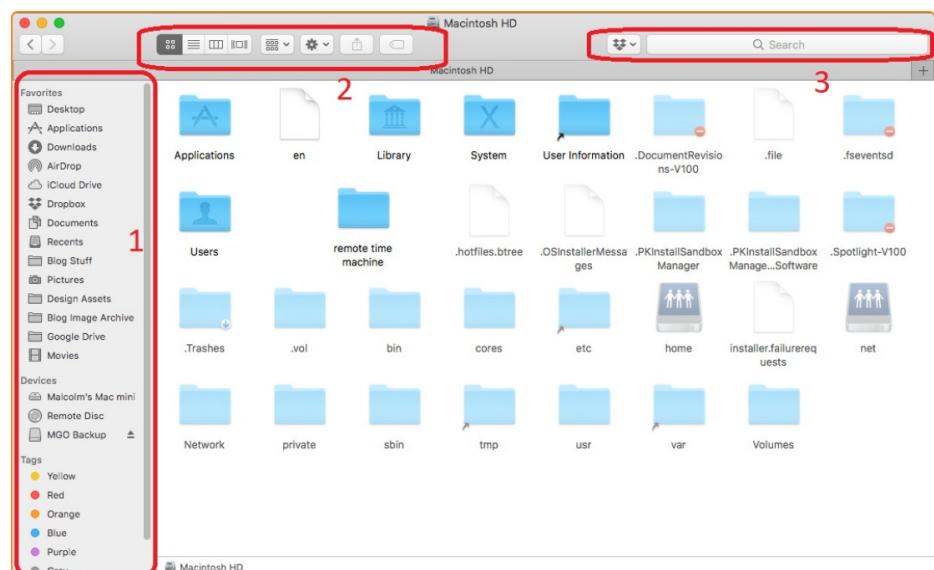


- use shortcuts to the most frequently used directories – 1
- use the shortcuts in the toolbar – 2
- use the search engine – 3

Context menu in Linux – access to properties. This is similar to how it's done in Windows.

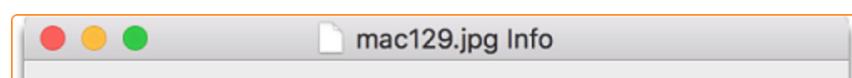


The macOS system differs from other systems mainly in the graphical interface, but the principle of operation remains the same:

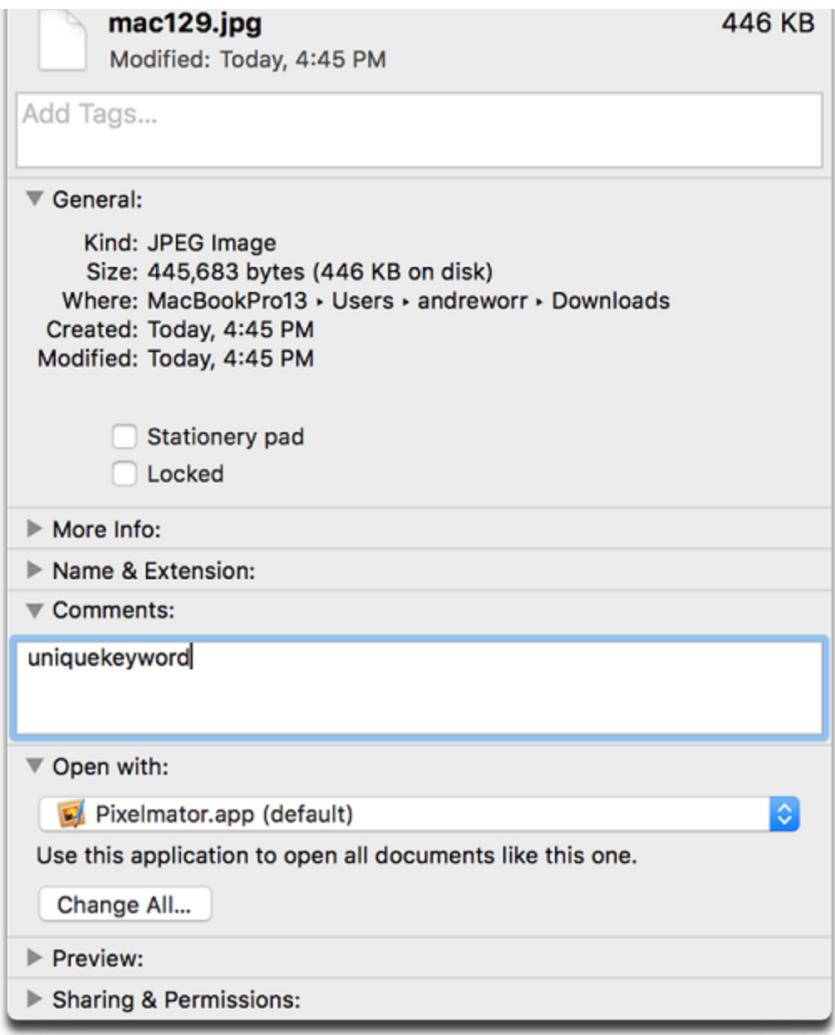


- use shortcuts to the most frequently used directories – 1
- use the shortcuts in the toolbar – 2
- use the search engine – 3

Context menu in macOS:



- HTML, CSS, JavaScript tasks
- Test 1st attempt | after the block: HTML,CSS,JS
- Test 2nd attempt | after the block: HTML,CSS,JS
- Frontend Technologies coursebook
- Frontend technologies slides
- Frontend Technologies tasks
- Test 1st attempt | after the block: FRONTEND TECHNOLOGIES (ANGULAR)
- Test 2nd attempt | after Frontend technologies
- Spring coursebook
- Spring slides
- Spring tasks
- Test 1st attempt | after the block: spring
- Test 2nd attempt | after the block: spring
- Mockito
- PowerMock
- Testing exceptions
- Parametrized tests
- Final project coursebook
- Final project slides
- Class assignments



Text interface

The second way to navigate through directories (and not only to navigate through directories, but also to manage the entire system) is through the text interface. This is where the user can issue commands to the operating system in text form. Current operating systems are graphics based, but for quite a long time a text interface was the only way to issue commands to the operating system and get feedback.

A text interface (it is also graphical) is often called a shell because it is an intermediary between the user and the operating system or the applications. There are a very large number of text interfaces, but the most popular are:

- on Windows systems – Command Line / Command Line and Powershell, often called the blue Command Line (from the interface color)
- in Unix-based systems such as Linux or MacOS – Terminal

The commands issued in the text interface consists of the following elements:

- **Command** – that is the actual command that we want to execute, e.g. creating a directory
- **Option** – For example, a read-only directory
- **Argument** – For example, a directory name

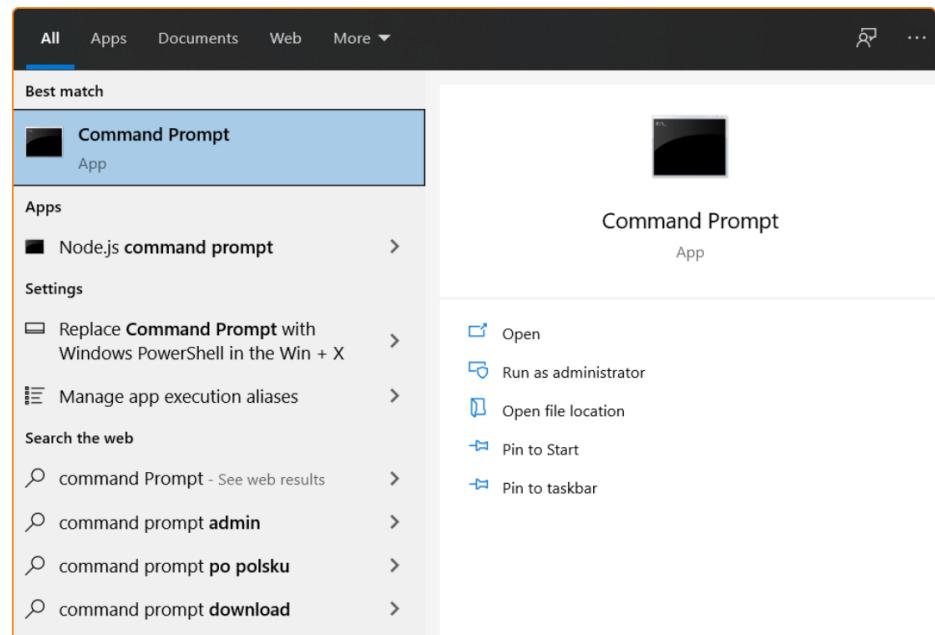
Example command on Linux:

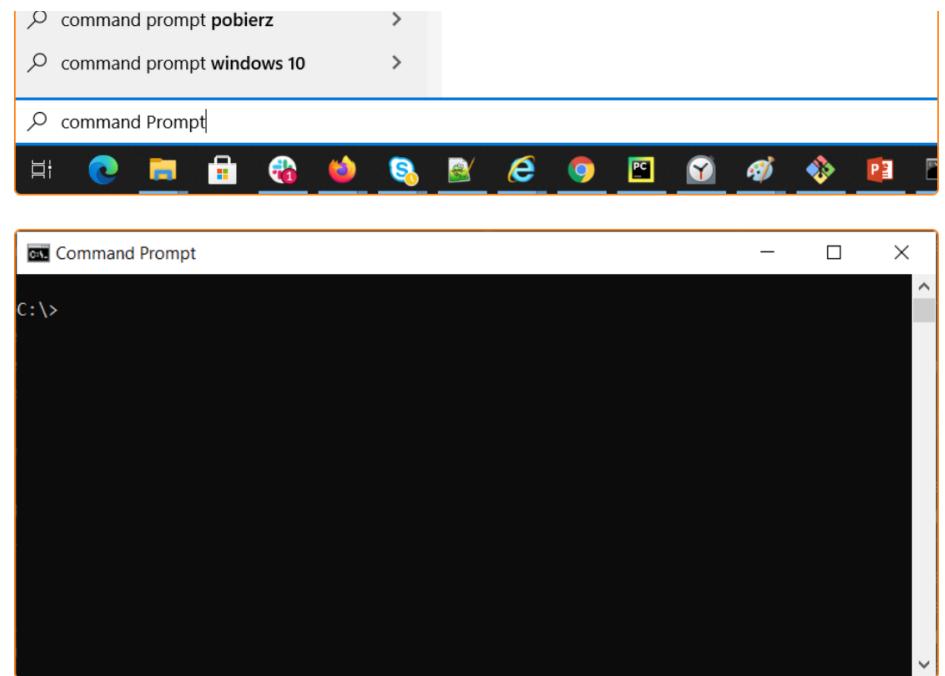
```
$ mkdir -m 777 first_directory 1
$ ls -la
total 17
drwxr-xr-x 6 user user 15 Apr 7 19:12 .
drwxr-xr-x 1 root root 4096 Apr 7 19:10 ..
-rw-r--r-- 1 user user 900 Apr 7 19:12 .Welcome to CoCalc.term-0.term'
-rw----- 1 user user 135 Apr 7 19:12 .bash_history
lrwxrwxrwx 1 user user 18 Apr 7 15:36 .bash_profile -> /home/user/.bashrc
-rw-r--r-- 1 user user 2355 Apr 7 15:36 .bashrc
-rw-r--r-- 1 user user 47 Apr 7 15:36 .gitconfig
-rw-r--r-- 1 user user 918 Apr 7 15:36 .gitexcludes
-rw-r--r-- 1 user user 8192 Apr 7 15:36 .jupyter-blobs-v0.db
drwxr-xr-x 2 user user 2 Apr 7 15:36 .sage
lwxrwxrwx 1 user user 12 Apr 7 19:10 .smc -> /tmp/.coCalc
dr-xr-xr-x 2 user user 2 Apr 7 19:10 .snapshots
drwxr-xr-x 2 user user 3 Apr 7 19:10 .ssh
-rw-r--r-- 1 user user 0 Apr 7 19:10 'Welcome to CoCalc.term'
drwxrwxrwx 2 user user 2 Apr 7 19:12 first_directory 2
$ mkdir --mode 777 second_directory 3
```

In the example above, the command to create a directory was invoked, with permissions to read / modify / write the directory by all users, named `first_directory`. The complete `mkdir -m 777 first_directory` (1) command. Then the command `ls` was called, which is responsible for displaying the contents of the directory in which we are currently located. The result of the command can be observed on the screen (2). Finally, the same command was called as in the first example, only in a changed form. Arguments and options can be passed to the command in two ways: in short form (1) or in a long form (3). Both forms are correct, users can choose the method of entering the command they find most convenient.

The most important commands in text interfaces – Windows

To get to the command prompt in Windows, select the appropriate option from the Start Menu -> Command Prompt.





The commands will take the names of the actions they refer to, e.g. a change in the directory (`cd`), or creating a directory (make directory – `mkdir`). Knowing the commands will be useful, but it is not essential, there is always a possibility to consult the documentation – [link](#). The most important commands that will be useful at the beginning of your adventure with the IT industry are:

- the `cd` command – it will allow you to display the place / current directory in which you are located
 - `dir` command – it will display all files and directories from the directory in which we are currently located
 - the commands `echo 'Test'` – it will allow you to display the selected text on the screen. In our case it is the word 'Test' (the command `echo` will e.g. be needed to display an environment variable)
 - the `cd` command with an argument in the form of a file path, e.g. `cd Users` will change the current directory to the `Users` directory, provided that the directory exists (if the directory does not exist, the error `The system cannot find the path specified.` will be displayed)
 - the `mkdir` command with an argument in the form of a name, e.g. `mkdir test` allows you to create a new directory called `test` in the current location
 - deleting `rmdir` directories with an argument e.g. `rmdir test` will allow you to delete the indicated directory (provided that it is empty), similarly the command `del test.txt` will delete the selected file

Windows commands take arguments (for example, in the form of a file or directory name), and also allow you to use various command options. On Windows, the command option differs from Linux, it is passed as `rmdir /S -` where `/S` is an option that will remove the directory even if the directory is not empty.

If you need help, you can run the command with the `/?` Option, e.g. `rmdir /?`, So that the help is displayed on the screen with an example of using the command. Below is an example of how commands behave in a text interface.

```
C:\>cd
C:\
C:\>dir
Volume in drive C is Windows
Volume Serial Number is 18E3-C152

Directory of C:\

03/10/2021  11:26 AM    <DIR>        Dell
03/12/2021  12:49 AM    <DIR>        Gradle
04/06/2021  10:23 AM    <DIR>        Intel
02/26/2021  10:10 AM    <DIR>        PerfLogs
03/29/2021  03:34 PM    <DIR>        Program Files
03/25/2021  07:26 PM    <DIR>        Program Files (x86)
03/10/2021  11:30 AM    <DIR>        Quarantine
04/07/2021  02:14 PM    <DIR>        Repositories
03/11/2021  10:46 AM    <DIR>        Users
03/30/2021  09:34 PM    <DIR>        Windows
          0 File(s)           0 bytes
          10 Dir(s)  346,720,329,728 bytes free

C:\>echo "Test"
"Test"
C:\>cd Users
C:\Users>cd john.doe
C:\Users\john.doe>mkdir Test
C:\Users\john.doe>echo > test.txt
C:\Users\john.doe>del test.txt
C:\Users\john.doe>rmdir Test
C:\Users\john.doe>mkdir Test
C:\Users\john.doe>rmdir /S Test
C:\Users\john.doe>rmdir /?
Removes (deletes) a directory.

RMDIR [/S] [/Q] [drive:]path
RD [/S] [/Q] [drive:]path

/S      Removes all directories and files in the specified directory
       in addition to the directory itself. Used to remove a directory
       tree.

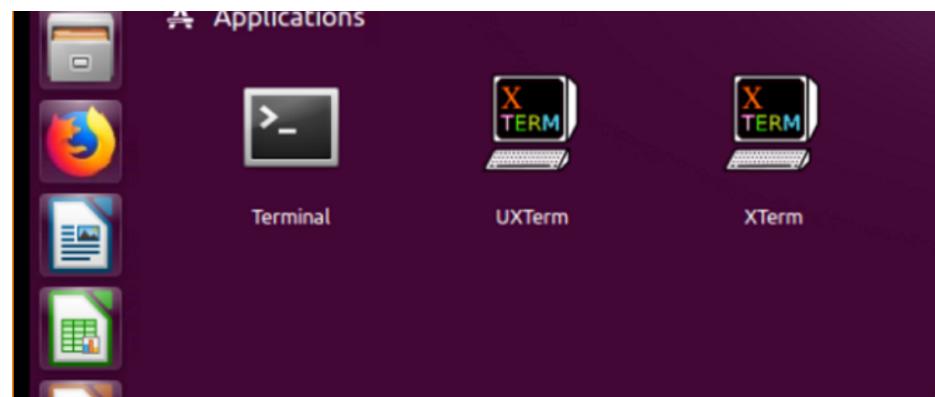
/Q      Quiet mode, do not ask if ok to remove a directory tree with /S

C:\Users\john.doe>
```

The most important commands in text interfaces – Linux/MacOS

Linux and macOS commands are identical. They both come from the Unix family of systems. They have many similar elements including the way to issue commands using a text interface. On Linux/MacOS, the text interface is called a "Terminal". It can be accessed in a similar way as in Windows. Look for Terminal in system applications:





Knowing the commands will be useful, but it is not essential, there is always a possibility to consult the documentation – [link](#). The most important commands that will be useful at the beginning of your adventure with the IT industry are:

- **pwd** command – it will allow you to display the place / current directory in which you are located
- **ls** command – it will display all files and directories from the directory in which we are currently located
- the commands **echo 'Test'** – allows you to display the selected text on the screen. In our case it is the word **Test** (the **echo** command might be needed to display an environment variable)
- the command **cd** with an argument in the form of a file path, e.g. **cd first_directory/** will change the current directory to the directory **first_directory** provided that the directory exists (if the directory does not exist, the error **No such file or directory.** will be provided)
- the **mkdir** command with an argument in the form of a name, e.g. **mkdir Test** allows you to create a new directory called **test** in the current location
- deleting **rmdir** directories with an argument, e.g. **rmdir Test** will allow you to delete the selected directory (provided that it is empty), similarly the command **rm test.txt** will delete the selected file

```

~$ pwd
/home/user
~$ ls
'Welcome to CoCalc.term'  first_directory  second_directory
~$ echo "Test"
Test
~$ cd directory_which_not_exists
bash: cd: directory_which_not_exists: No such file or directory
~$ cd first_directory/
~/first_directory$ mkdir Test
~/first_directory$ echo > test.txt
~/first_directory$ rm test.txt
~/first_directory$ rm file_which_not_exists.txt
rm: cannot remove 'file_which_not_exists.txt': No such file or directory
~/first_directory$ rmdir Test
~/first_directory$ rmdir --help
Usage: rmdir [OPTION]... DIRECTORY...
Remove the DIRECTORY(ies), if they are empty.

--ignore-fail-on-non-empty
      ignore each failure that is solely because a directory

```

```

      is non-empty
-p, --parents  remove DIRECTORY and its ancestors; e.g., 'rmdir -p a/b/c' is
                 similar to 'rmdir a/b/c a/b'
-v, --verbose   output a diagnostic for every directory processed
--help        display this help and exit
--version     output version information and exit

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Report rmdir translation bugs to <https://translationproject.org/team/>
Full documentation at: <https://www.gnu.org/software/coreutils/rmdir>
or available locally via: info '(coreutils) rmdir invocation'
~/first_directory$
```

The most important difference between Windows and Unix systems (Linux/macOS) is primarily the way of passing options to commands. In the case of Unix systems, options are passed in short form, e.g. `rm -f` or `rm --force`. Some commands will have the same name, but some

The most important difference between Windows and Unix systems (Linux/macOS) is primarily the way of passing options to commands. In the case of Unix systems, options are passed in short form, e.g. `rm -f` or `rm --force`. Some commands will have the same name, but some commands will be different, such as `del` and `rm`.

Relative/absolute path

Directories and files that exist in the operating system are represented by the appropriate path, e.g.:

- Windows – C:\Program Files\Google\Chrome\Application\chrome.exe
- Linux – /opt/google/chrome/default_apps/gmail.crx

The first fundamental difference that appears in defining the path is the way of saving – in the case of Windows systems we use the backslash to separate specific directories, in the case of Unix systems we use the standard / slash. The second difference is the lack of a drive letter on Unix systems, it is due to how the operating system is built.

An *absolute* path describes the location of an item on the disk (e.g. a file or a directory) unambiguously. It always starts with the topmost element, the root. Examples:

- Windows – C:\Users\john.doe\Documents\file.txt
- Linux/Mac – /Users/john.doe/Documents/file.txt

A *relative* path points to a file or folder relative to the current location. Examples:

- Windows – .\Documents\file.txt
- Linux/Mac – ./john.doe/Documents\file.txt

To understand the differences, we will use a more advanced example using a terminal. Create the following file and directory structure.

```
# As a user, I go to the previously created directory in the terminal by using the
C:\Users\john.doe>cd C:\Users\john.smith\Documents\test
C:\Users\john.doe\Documents\test>
# The given path was an absolute path, even though I was currently in the username
```

```

# Using the relative path I can change the location
C:\Users\john.doe\Documents\test>cd ..\..\..\john.smith
C:\Users\john.doe>
# ... the dots instead of the directory name indicate that we are moving up the di
# Using a relative path I go back to the test directory and display its contents
C:\Users\john.doe>cd Documents\test
C:\Users\john.doe\Documents\test>dir
Directory of C:\Users\john.doe\Documents\test
04/07/2021 08:39 PM <DIR> .
04/07/2021 08:39 PM <DIR> ..
04/07/2021 08:39 PM 0 New Text Document.txt
1 File(s) 0 bytes
2 Dir(s) 350,798,442,496 bytes free

```

What else should be considered when naming files and directories:

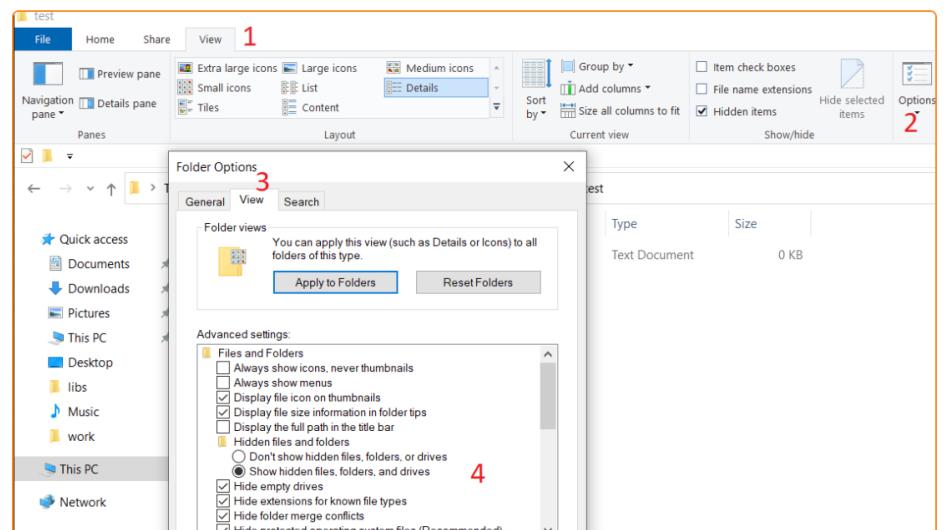
- the names of files and directories allow you to put spaces in the name, the path with a space is put in quotation marks, thanks to which we will avoid a situation that our terminal will misinterpret the name of our file or directory:
- Windows – cd "C:Usersjohn.doetext file.txt"
- Linux – cd ".john.doeDocumentstext file.txt"

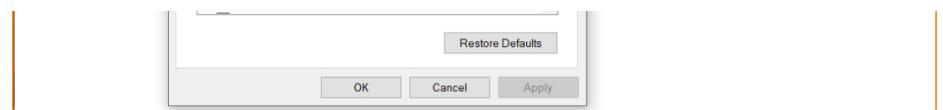
If we replace a space with a period or an underscore, there is no need to use quotation marks. When creating files or directories, you need to remember to create an easily identifiable name, with no Polish diacritics. In the case of files, it is also worth taking care of the appropriate file extensions. The operating system will automatically associate the created file with the appropriate application software.

Hidden files and directories

Operating systems have hidden files and directories. They are not visible to the common user for security reasons. They usually contain information necessary for the proper operation of the system and software configuration. Modifying them on your own, without proper knowledge, may cause a system failure.

In Windows, hidden files and directories are invisible. To be able to view hidden files or directories, go to the context menu view (1) in the windows explorer window. Then select the option (2). In the new window select the view tab (3), and then change the settings for the visibility of hidden files and directories.

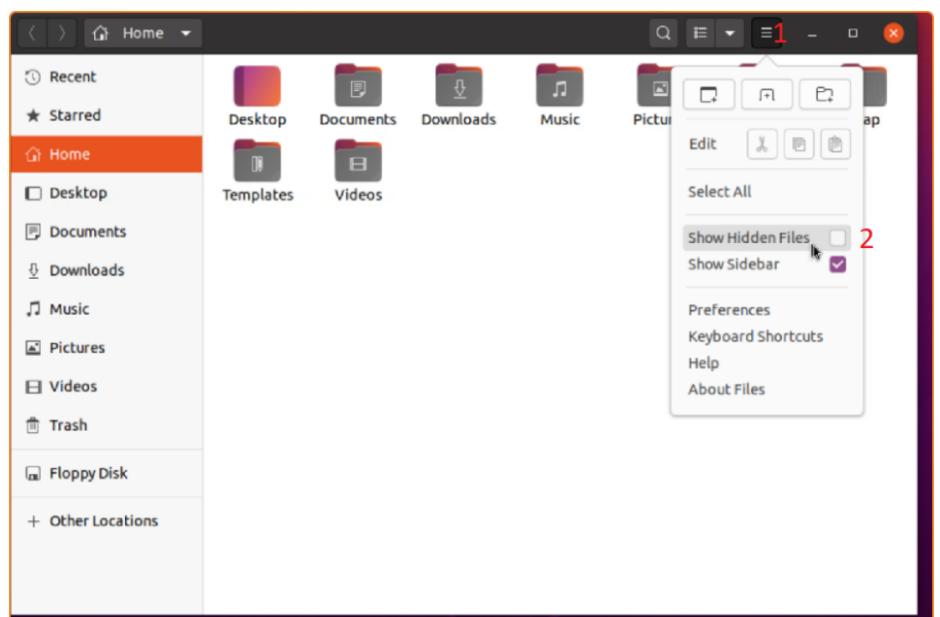




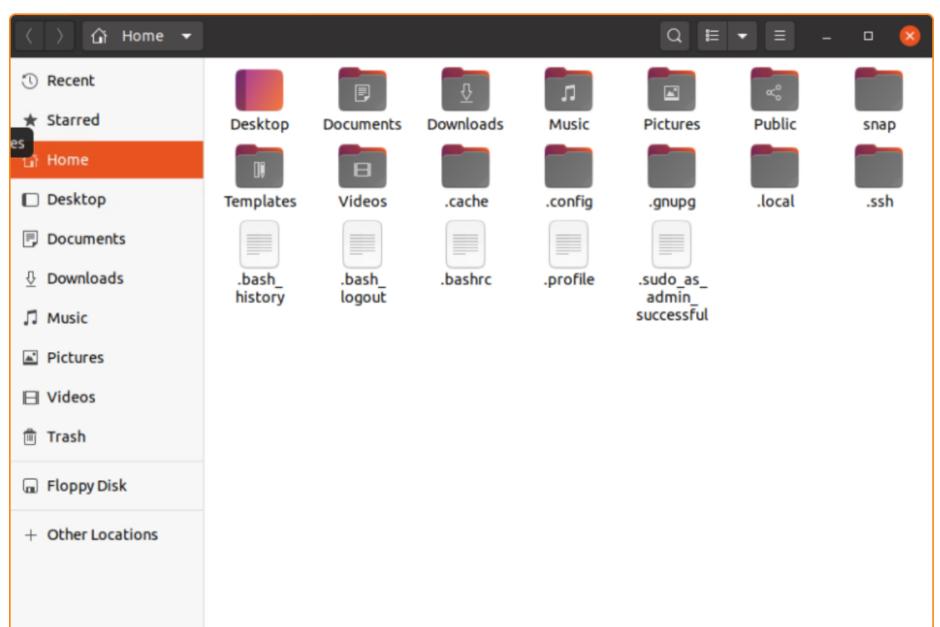
After this operation, hidden files or directories will be visible to the user.

Name	Date modified	Type	Size
hidden file	4/9/2021 8:05 PM	Text Document	0 KB
New Text Document	4/7/2021 8:39 PM	Text Document	0 KB

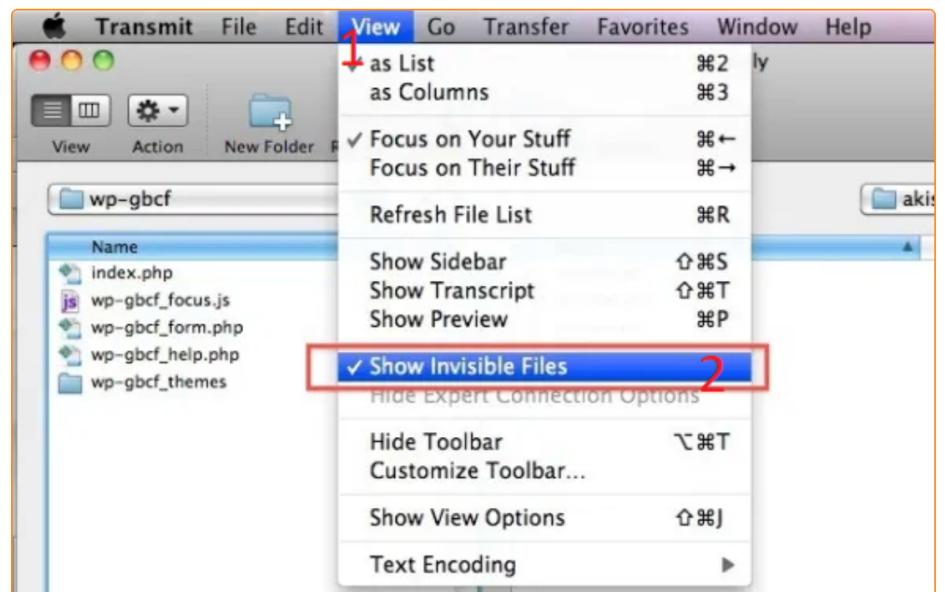
On Linux and Mac, hidden directories and files are displayed just like regular directories and files, but have names that start with a period. In Linux, the display of hidden files and directories can be activated in the file explorer by expanding the context menu (1) and selecting the checkbox for hidden files and directories (2).



After this operation, hidden files or directories will be visible to the user.



In MacOS, select Show Invisible Files (2) from the context menu (1).



An additional place to view hidden items is the command line / terminal. The following commands are responsible for displaying directories and hidden files:

- Windows – command line – `dir /AH` – the dir command with the /A option and an H argument

```
C:\Users\john.doe\Documents\test>dir /AH
Volume in drive C is Windows
Volume Serial Number is 18E3-C152

Directory of C:\Users\john.doe\Documents\test

04/09/2021  08:05 PM              0 hidden file.txt
               1 File(s)           0 bytes
               0 Dir(s)  350,742,519,808 bytes free
```

- Linux/macOS – terminal – the `ls -la` command with arguments -l (list) -a (all) will display all files in a given location as a list with all parameters

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ ls -la
total 108
drwxr-xr-x  16 onworks  onworks  4096 Aug 30  2019 .
drwxr-xr-x  3 root     root     4096 Mai  5  2019 ..
-rw-----  1 onworks  onworks   94 Mai 31  2019 .bash_history
-rw-r--r--  1 onworks  onworks  220 Mai  5  2019 .bash_logout
-rw-r--r--  1 onworks  onworks 3771 Mai  5  2019 .bashrc
drwx----- 11 onworks  onworks  4096 Mai  5  2019 .cache
drwx-----  3 onworks  onworks  4096 Mai  5  2019 .compiz
drwx----- 14 onworks  onworks  4096 Mai  5  2019 .config
drwxr-xr-x  2 onworks  onworks  4096 Mai  5  2019 Desktop
drwxr-xr-x  2 onworks  onworks  4096 Mai  5  2019 Documents
drwxr-xr-x  2 onworks  onworks  4096 Mai  5  2019 Downloads
-rw-r--r--  1 onworks  onworks 8980 Mai  5  2019 examples.desktop
drwx-----  2 onworks  onworks  4096 Mai  5  2019 .gconf
drwx-----  3 onworks  onworks  4096 Aug 30  2019 .gnupg
-rw-----  1 onworks  onworks 3496 Aug 30  2019 .ICEauthority
drwx-----  3 onworks  onworks  4096 Mai  5  2019 .local
drwxr-xr-x  2 onworks  onworks  4096 Mai  5  2019 Music
drwx-----  2 onworks  onworks  4096 Mai  5  2019 Pictures
```

```
drwxr-xr-x 2 onworks onworks 4096 Mai 5 2019 Pictures
-rw-r--r-- 1 onworks onworks 655 Mai 5 2019 .profile
drwxr-xr-x 2 onworks onworks 4096 Mai 5 2019 Public
-rw-r--r-- 1 onworks onworks 0 Mai 5 2019 .sudo_as_admin_successful
drwxr-xr-x 2 onworks onworks 4096 Mai 5 2019 Templates
```

Which of the following describes a way to navigate through directories on a computer?

- We can only navigate through directories with VS Code.
- Through a graphical or text based user interface
- By adjusting the brightness of the screen
- By creating new directories.

Submit

Complete Lesson