

macOS Basics Course

Flavio Copes

Updated August 3, 2026

Contents

Meet macOS	2
What macOS is	2
Desktop, Dock, and menu bar	3
Windows and applications	3
System Settings and system information	4
Check your understanding: macOS foundations	5
Finder and files	5
Meet the Finder	5
Home, Library, and file paths	6
Hidden files and file extensions	7
Quick Look, tags, and search	7
Check your understanding: Finder and files	8
Apps and developer tools	8
Install and remove applications	8
Gatekeeper and application permissions	9
Install developer tools with Homebrew	10
Terminal and command-line tools	11
Check your understanding: apps and tools	12
Work faster	12
Keyboard shortcuts	12
Use Spotlight	13
Mission Control and Spaces	14
Screenshots and small automations	14
Check your understanding: working faster	15
Maintenance and security	15
Use Activity Monitor	15
Manage storage and updates	16
FileVault and account security	17
Time Machine and recovery	18
Check your understanding: maintenance and security	19

Learn the macOS desktop, Finder, apps, developer tools, shortcuts, backups, security, and the built-in utilities that keep a Mac working well.

What you'll learn: Set up and use a Mac confidently for everyday development, organize files and apps, diagnose common problems, and protect your work.

Meet macOS

Understand the desktop, menus, windows, settings, accounts, and system information.

What macOS is

Understand the operating system, its UNIX foundation, and the boundary between the graphical desktop and command-line tools.

macOS is the operating system that runs on a Mac. It manages the hardware, files, applications, accounts, networking, and the graphical desktop you use every day.

An operating system sits between your programs and the machine. When you save a document, macOS decides where the bytes land on disk. When two apps want the processor at the same time, macOS shares it between them. When you plug in a USB drive, macOS makes it appear in Finder.

A UNIX under the desktop

Under the desktop, macOS is a UNIX operating system. Apple builds the graphical interface on top of a core called **Darwin**. This is why many commands and development tools feel familiar when you later work on Linux.

You can see that foundation yourself. Open Terminal from **Applications → Utilities** and run:

```
uname
```

It prints **Darwin**, the name of the underlying system. Now ask for the name you see in marketing material:

```
sw_vers
```

This prints the product name (**macOS**), the version number, and the build identifier. Both commands describe the same machine from different angles.

Two views of the same system

The graphical interface and the command line are two views of the same system. A file you drop on the Desktop is immediately visible in Terminal:

```
ls ~/Desktop
```

Nothing is copied or synchronized. There is one filesystem, shown two ways. Use the one that makes the current task clearer. Renaming one file is faster in Finder. Renaming two hundred files is faster with a command.

Close to Linux, not identical

The shared UNIX heritage means the concepts transfer: files, permissions, processes, shells. The specific commands do not always transfer. A Linux tutorial that tells you to run **apt install something** fails on a Mac with **command not found**, because **apt** is a Linux package manager. When that happens, don't force the Linux instructions. Look for the macOS way to do the same job — later lessons in this course cover it.

Desktop, Dock, and menu bar

Use the three persistent parts of the macOS interface and understand why an application can stay open without a visible window.

The **Desktop** is a folder and a workspace. The **Dock** launches apps and shows running ones. The **menu bar** contains the active application menu and system controls. These three parts stay on screen while windows come and go.

The Desktop is a real folder

Everything you see on the desktop background lives in a folder inside your home directory. Prove it:

```
open ~/Desktop
```

A Finder window opens showing the same items. Files are never "on the screen" — the wallpaper area is one more view of the `~/Desktop` folder.

The Dock

The Dock has two zones separated by a thin divider: applications on one side, then minimized windows, folders, and the Trash on the other. A small dot under an icon means that application is running.

Right-click a Dock icon to see its options. **Options** → **Keep in Dock** pins an app you use often. Dragging an app out of the Dock removes the shortcut, not the application.

The menu bar

The menu bar changes when you switch apps. The bold name right after the Apple logo always tells you which application is active. Look there when you cannot find a command inside a window: Mac apps put their commands in the menu bar, not in the window itself.

The right side holds status menus, Control Center, and the clock. The Apple menu on the far left never changes, and contains system-wide actions like Sleep and Shut Down.

Closing is not quitting

Closing the last window does not always quit a Mac app. The app keeps running with zero windows: the menu bar still shows its name and the Dock dot stays lit.

This is the most common point of confusion for people arriving from Windows. You close Safari's last window, then wonder why the menu bar still says Safari. It is still open. Use **Command-Q** when you want to stop the application itself, or right-click its Dock icon and choose **Quit**.

To verify the difference: close a window and watch the Dock dot stay. Press **Command-Q** and watch it disappear.

Windows and applications

Move between applications and windows without confusing hiding, minimizing, closing, and quitting.

macOS separates two ideas that Windows blends together: switching between **applications** and switching between **windows** of one application. Each has its own shortcut.

Command-Tab switches applications. Hold **Command**, tap **Tab** to move through the list of running apps, and release on the one you want. **Command-`** (the backtick key) cycles through the windows belonging to the current application.

Together they cover most movement: pick the app first, then the window.

The three window controls

The red control closes a window. The yellow control minimizes it into the right side of the Dock. The green control enters full screen — hover over it (or hold it) to see tiling options that place the window on half of the screen instead.

Closing a window does not quit the app, as you saw in the previous lesson.

Hiding versus minimizing

Use **Command-H** to hide an app and **Command-M** to minimize a window. Both preserve your work, but they behave differently when you come back:

Command-H	hides every window of the app; they all return the moment you switch back to it
Command-M	sends one window to the Dock; it stays there even when you Command-Tab back to the app

That second behavior causes the classic "where did my window go" moment. You minimize a window, later switch back to the app with **Command-Tab**, and nothing appears. The window is not gone — it is waiting as a thumbnail in the Dock. Click the thumbnail to restore it.

My advice: prefer **Command-H**. Hiding gets windows out of the way and brings all of them back with a normal app switch. Minimizing is better when you want to park one specific window while keeping others visible.

Try the difference now. Open **TextEdit** with two windows, minimize one, hide the app, then **Command-Tab** back. The hidden window returns on its own; the minimized one stays in the Dock until you click it.

System Settings and system information

Find the settings and hardware details you need before installing software or diagnosing a problem.

Open **System Settings** from the Apple menu. This is where you change how the Mac behaves: display, sound, network, accounts, privacy.

The sidebar lists dozens of sections, and options move between them across macOS versions. Do not browse — search. The search field at the top of the sidebar is faster than remembering which section contains an option. Type "resolution", "screenshot", or "Bluetooth" and jump straight to the setting.

What Mac is this?

Open **General** → **About** to see the macOS version, chip, memory, serial number, and storage summary.

You can read the same facts from Terminal:

```
system_profiler SPHardwareDataType
```

The output looks like this:

```
Model Name: MacBook Air
Chip: Apple M2
Total Number of Cores: 8
Memory: 16 GB
Serial Number (system): X0XXD4XXXX
```

For the full hardware and software inventory, hold the Option key and open the Apple menu: **About This Mac** becomes **System Information**, a separate app that lists every device, network interface, and installed extension.

Why this matters before asking for help

Record the exact macOS version and chip when asking for technical help or before installing software. Instructions can differ between Intel and Apple silicon Macs, and between macOS versions.

A typical failure looks like this: you follow a guide written for an Intel Mac on an Apple silicon machine, and an installer or fix does nothing or errors out. The guide was not wrong — it was written for different hardware. Checking **General** → **About** first tells you which instructions apply to you.

The same applies to version numbers. "Requires macOS 15 or later" is only meaningful once you know what you are running. Thirty seconds in About saves an hour of confusion later.

Check your understanding: macOS foundations

Check macOS, UNIX, applications, windows, the menu bar, System Settings, and system information.

Answer each question before continuing. Read the explanation after every answer, including the ones you get right.

[Take this interactive quiz online](#)

Finder and files

Navigate folders, paths, hidden files, Quick Look, tags, aliases, and search.

Meet the Finder

Use Finder as the main interface for browsing files, folders, disks, network locations, and connected devices.

Finder is the application you use to browse files, folders, disks, network locations, and connected devices. It is always running — you cannot quit it. Click its face icon in the Dock to open a window, or press **Command-N** inside Finder for another one.

The sidebar

The sidebar gives quick access to your home folders, iCloud Drive, disks, and tags. Choose what appears there in **Finder** → **Settings** → **Sidebar**.

Adding a folder to the sidebar creates a shortcut. It does not move the folder. Drag any folder you use daily — a projects folder, a client folder — into the Favorites section and it is one click away

from every Finder window.

Watch where you drop, though. Dropping *between* sidebar items (you see a horizontal insertion line) adds a shortcut. Dropping *onto* a folder already in the sidebar moves your item into that folder. If a folder seems to vanish during this, check inside the sidebar item it was dropped on.

Four views

Switch between icon, list, column, and gallery views from the toolbar or with shortcuts:

```
Command-1    icon view
Command-2    list view
Command-3    column view
Command-4    gallery view
```

Each view answers a different question. List view is useful when you need to compare names, dates, kinds, or sizes — click a column header to sort by it, click again to reverse. Column view shows the path you took through nested folders, which makes deep project trees much easier to navigate. Gallery view gives large previews, good for photos and PDFs.

Try this now: open your Downloads folder, switch to list view with **Command-2**, and sort by Date Added. The file you downloaded most recently jumps to the top. That one habit replaces most “where did my download go” searches.

Home, Library, and file paths

Connect familiar Finder folders to their real paths and recognize which Library folder belongs to your account.

Your home folder contains Desktop, Documents, Downloads, Movies, Music, and Pictures. In a shell, its shortcut is `~`.

Open Terminal and compare these paths:

```
printf '%s\n' "$HOME"
pwd
```

`$HOME` contains the full home path. `~` is shell syntax that expands to it in an unquoted command. A path beginning with `/` starts at the filesystem root; a path without `/` is relative to the current directory.

macOS has several Library folders

- `~/Library` contains settings, caches, containers, and data for your account.
- `/Library` contains resources shared by users on the Mac.
- `/System/Library` belongs to macOS and is protected.

These folders are not interchangeable. Deleting a preference under your user Library can reset one app. Changing a shared or system location can affect every account or fail because macOS protects it.

Do not use a Library folder for your own documents. Store personal files in your home, Documents, or another intentional location.

Open an exact path in Finder

Choose **Go** → **Go to Folder** or press **Command-Shift-G**, then enter a path such as:

```
~/Library/Application Support
```

This is safer than clicking through similarly named folders. Before moving or deleting application data, quit the app, confirm the path, and keep a recoverable copy.

Try this: open your home folder and user Library in Finder. Explain why `~/Library` and `/Library` can contain folders with the same application name but different scopes.

Hidden files and file extensions

Reveal hidden items when needed and keep file extensions visible enough to understand what a file really is.

Files whose names begin with a dot are hidden by Unix convention. Development tools use them for configuration, including `.git`, `.env`, and `.ssh`.

Press **Command-Shift-.** in Finder to show or hide hidden items. In Terminal, use:

```
ls -la
```

Hidden does not mean protected. A hidden file can still be read by an application or another user with permission. Secrets need correct access controls and must stay out of repositories, screenshots, and shared archives.

Show filename extensions

An extension such as `.jpg`, `.txt`, or `.js` helps macOS and applications identify a file. Finder can hide extensions, which makes names such as these easy to confuse:

```
invoice.pdf
invoice.pdf.command
config.json.txt
```

Enable **Finder** → **Settings** → **Advanced** → **Show all filename extensions** when you work with source files or downloaded artifacts.

Changing an extension does not convert the file. Renaming `photo.png` to `photo.jpg` changes only the name. Use Preview or another application to export the actual format.

Inspect before opening

Select a downloaded file and press **Command-I** to view **Get Info**. Check its full name, kind, source, and opening application.

Do not reveal hidden files and then delete unfamiliar dotfiles to “clean” your Mac. A shell, editor, Git repository, or application may depend on them.

Try this: create `notes.txt`, hide its extension in Get Info, then enable all extensions in Finder. Observe which setting wins and restore the file afterward.

Quick Look, tags, and search

Inspect and organize files without opening every document in its full application.

Select a file and press the Space bar to open **Quick Look**. It previews many images, PDFs, text files, videos, and development files without launching the full application.

Quick Look is useful for confirming that you selected the correct file before attaching, moving, or deleting it. It is a preview, so test the file in its real application when formatting or behavior matters.

Organize without moving files

Finder tags add searchable labels while the file stays in its current folder. You might tag active contracts, reference images, or files waiting for review.

An alias is different. It is a shortcut to the original item. Moving or deleting the alias does not move or delete the original. If the original moves, macOS can often keep the alias connected, but verify it before relying on the alias in an automated workflow.

Finder search and Spotlight

Use Finder search when you want filters such as kind, date, name, or folder. Begin in the intended folder, then confirm whether Finder is searching that folder or the entire Mac.

Use Spotlight with **Command-Space** when you want to find and launch an app, file, setting, or calculation quickly.

A **Smart Folder** stores search criteria, not copies of matching files. New files appear automatically when they match the rules.

Search depends on the Spotlight index. If a folder is excluded in Spotlight privacy settings, its files may not appear. Do not move or duplicate everything just because one search returns no result; first check the scope and filters.

Try this: tag two disposable files, find them through the tag, then create a Finder search that matches only one file type. Confirm that the originals never moved.

Check your understanding: Finder and files

Check Finder views, paths, Library folders, hidden files, extensions, Quick Look, aliases, tags, and search.

Answer each question before continuing. Read the explanation after every answer, including the ones you get right.

[Take this interactive quiz online](#)

Apps and developer tools

Install, update, remove, and trust apps, then prepare a practical developer setup.

Install and remove applications

Recognize App Store, disk image, and installer-package workflows, then remove an app without deleting unrelated personal files.

Mac applications arrive through several installation paths. The path tells you something about where files may be placed and how removal works.

App Store and disk images

App Store applications install through your Apple account and update through the store.

Many applications downloaded from a developer arrive in a `.dmg` disk image. Open the image, drag the application into **Applications**, then eject the image. Launch the copy in Applications, not the temporary copy inside the mounted image.

A disk image is a container. Dragging the app usually copies one application bundle, but the app can create settings and data later when it runs.

Installer packages

A `.pkg` installer can place files in several system locations and may request an administrator password. Use it only when you trust the publisher and expected the package.

Before continuing, read the installer steps and destination. For sensitive software such as drivers, VPN clients, or security tools, check the developer's current removal instructions too.

Remove according to the installation method

For a simple app bundle:

1. Quit the application.
2. Move it from Applications to the Trash.
3. Empty the Trash only when you are sure you no longer need it.

This may leave preferences, caches, and documents. Those files are not necessarily harmful, and deleting every file with a similar name can remove user data you wanted to keep.

Use the vendor's uninstaller when the app installed login items, background services, system extensions, or drivers. Confirm those components stopped afterward.

Try this: choose one installed app and identify whether it came from the App Store, a disk image, or a package. Find its official uninstall guidance before changing anything.

Gatekeeper and application permissions

Understand the checks macOS performs and grant sensitive permissions only when an app genuinely needs them.

Gatekeeper checks downloaded software before its first launch. It verifies signing and notarization information, checks for known malicious content, and asks you to confirm that you intended to open the download.

Prefer the App Store or software downloaded directly from a developer you trust. A valid signature identifies the signer and detects later changes; it does not prove that every behavior in the app is safe for your data.

Treat warnings as a review point

If macOS blocks an application, confirm:

- you downloaded the intended file from the official source
- the developer documents the current macOS version
- the checksum or signature matches when the publisher provides one

- you understand why the app is not passing the normal check

Do not disable Gatekeeper globally to open one application. Do not follow an installation guide that asks you to remove security attributes or run an unexplained command as administrator.

Application permissions have different blast radii

Applications may request access to the camera, microphone, files, screen recording, accessibility controls, input monitoring, or Full Disk Access.

Accessibility and input-monitoring permissions can allow an app to control or observe other applications. Full Disk Access can expose data that normal sandbox rules would protect. Grant only the permission required for a feature you intentionally use.

Review permissions later in **System Settings → Privacy & Security**. Remove access you no longer need, then verify the application still performs the intended job.

Permission prompts can be triggered by a legitimate feature, but the prompt itself does not explain the whole data flow. Check whether captured data stays on the Mac or is sent to a service.

Try this: review one category in Privacy & Security. For each listed app, explain which feature needs that permission and remove nothing until you can verify the impact.

Install developer tools with Homebrew

Use Homebrew for command-line tools while understanding what it changes and how to inspect an installation first.

Homebrew is a package manager commonly used by Mac developers. It installs command-line tools and applications that Apple does not bundle with macOS.

On Apple Silicon, the supported default prefix is `/opt/homebrew`. On Intel Macs it is `/usr/local`. Homebrew installs versioned packages inside its prefix and links the active commands into a directory on your shell path.

Install Homebrew only from brew.sh or its official package. Read the installer output before confirming. After installation, follow the printed `brew shellenv` step so your shell can find `brew`.

Inspect before installing

```
brew search tree
brew info tree
brew install tree
```

`brew info` shows the source, version, dependencies, installation status, and caveats. A **formula** generally installs command-line software. A **cask** installs a macOS application or other packaged artifact.

Verify the result:

```
command -v tree
brew list --versions tree
```

Homebrew should not need `sudo` for ordinary package installation in its supported prefix. Stop if a random formula asks you to make broad permission changes.

Updates are separate steps

```
brew update
brew outdated
brew upgrade
```

`brew update` refreshes Homebrew metadata. `brew upgrade` changes installed packages. Review outdated packages and service impact before a large upgrade.

Use `brew uninstall tree` to remove the formula. `brew autoremove --dry-run` can preview dependencies no longer required.

Third-party taps add another software source. Trusting one gives its package definitions meaningful influence over installations, so add only sources you have reviewed.

Try this: run `brew info` for one installed formula. Identify its version, dependencies, prefix, and whether an upgrade is available without changing anything.

Terminal and command-line tools

Open Terminal, recognize the shell prompt, install Apple command-line tools, and continue into the dedicated shell course.

Open Terminal with Spotlight or from **Applications** → **Utilities**. Terminal is the application window; the **shell** is the program reading your commands inside it.

The default interactive shell on current macOS installations is Zsh. Confirm the current shell and working directory:

```
printf '%s\n' "$SHELL"
pwd
```

The prompt often shows your account, machine, and current folder. Do not paste a command merely because a website says to “open Terminal.” Read the program, arguments, paths, redirections, and use of `sudo` first.

Apple Command Line Tools

Git, compilers, headers, and build utilities may require Apple's Command Line Tools. Request their installer with:

```
xcode-select --install
```

If they are already installed, macOS will tell you. Verify the selected developer directory and Git:

```
xcode-select -p
git --version
```

The complete Xcode application is much larger and is not required for every command-line workflow. Install it when your project needs the full Apple platform SDKs or IDE.

Know where a command comes from

```
command -v git
type git
```

These checks help distinguish an Apple-provided command, a Homebrew command, an alias, and a shell function.

Continue with the [Shell Commands Course](#) for navigation, files, permissions, pipes, processes, and scripting.

Try this: open a fresh Terminal window. Identify its shell, current directory, and the path of `git` without changing the machine.

Check your understanding: apps and tools

Check application installation, removal, Gatekeeper, privacy permissions, Homebrew, Terminal, Zsh, and developer tools.

Answer each question before continuing. Read the explanation after every answer, including the ones you get right.

[Take this interactive quiz online](#)

Work faster

Use shortcuts, Spotlight, spaces, screenshots, and small automations.

Keyboard shortcuts

Learn a small reusable set of shortcuts and discover the correct shortcut from application menus.

Start with `Command-C`, `Command-V`, `Command-Z`, `Command-S`, `Command-W`, and `Command-Q`. These work in most Mac apps: copy, paste, undo, save, close window, quit.

Six shortcuts sounds small. It is deliberate. A small set you actually use beats a long list you half-remember.

Read shortcuts from the menus

Menus display the shortcut beside each command. Use the menu as a live reference instead of memorizing a giant list.

Menus write shortcuts with symbols, so learn to decode them:

Command
Shift
Option
Control

So `⌘ S` next to a menu item means Shift-Command-S.

This gives you a repeatable way to learn any shortcut: use the menu command normally a few times, notice the symbols printed next to it, then try the keys the next time. After a few repetitions the shortcut sticks and you stop opening the menu.

To confirm a shortcut fired, watch the menu bar — the menu containing that command flashes briefly when its shortcut runs.

Changing shortcuts

You can change system and application shortcuts in Keyboard settings. Open **System Settings** → **Keyboard** → **Keyboard Shortcuts**. The **App Shortcuts** section even lets you add a shortcut

to any menu command: pick the app, type the menu item's exact name, and press the keys you want.

One rule when you do this: avoid reusing a combination that already belongs to another command. macOS does not warn you about conflicts. If your new shortcut does nothing, or triggers something unexpected, a collision is the usual cause — the app's own binding wins, or another command fires instead. Pick combinations with extra modifiers, like Control-Option-Command, which are rarely taken.

Verify a custom shortcut the same way as a built-in one: open the menu and check that your key combination now appears next to the command.

Use Spotlight

Launch apps, find files, reveal locations, calculate values, and convert units without reaching for Finder first.

Press **Command-Space** and begin typing. Spotlight can launch applications, find documents, and show settings. It is often the fastest route from an idea to the right application or file.

Spotlight exists so you never have to remember *where* something is — only *what* it is called. Instead of clicking through Applications or digging through folders, you type a few letters and press Return.

Launching and finding

Type the first letters of an app name:

`term`

Terminal appears as the Top Hit. Press Return to open it. This works for any application, and after a while it replaces the Dock and Launchpad for everything you do not use hourly.

The same flow works for documents and settings. Type part of a file name or a settings keyword like "wallpaper" and Spotlight lists the matches.

Select a result and hold **Command** to reveal its location. Press Return to open it. The reveal trick matters when you need the enclosing folder, not the file itself.

Calculations and conversions

Spotlight also handles calculations and conversions. Type an expression directly:

`1499 * 0.22`

`250 usd in eur`

`72 f in c`

The answer appears instantly. No calculator app, no browser tab.

When something does not show up

The usual failure: you search for a file you know exists and Spotlight returns nothing. Two common causes.

First, the folder may be excluded from indexing. Check **System Settings → Siri & Spotlight → Spotlight Privacy** — anything listed there is invisible to Spotlight, and it is easy to forget an exclusion added months ago.

Second, on a machine that just restored from backup or migrated, the index may still be rebuilding. Give it time; results improve as indexing completes.

Try this now: press **Command-Space**, type the name of an app you use daily, and launch it with Return. Then search a file, hold Command to see its path, and open its folder.

Mission Control and Spaces

Organize windows across desktops and move between focused work areas without losing applications.

Mission Control shows every open window and Space at once. When your screen fills up with overlapping windows, it is the fastest way to see what is actually open and jump to the window you need.

Open it with the configured trackpad gesture or keyboard shortcut. The defaults:

Control-Up	Mission Control (all windows and Spaces)
Control-Down	windows of the current app only
Control-Left	previous Space
Control-Right	next Space

On a trackpad, swipe up with three fingers for Mission Control and sideways with three fingers to move between Spaces.

Spaces

A **Space** is an extra desktop. In Mission Control, move the pointer to the top of the screen and click the + button in the Spaces bar to create one. Drag windows between the thumbnails to reorganize.

Create separate Spaces for different kinds of work, such as coding, communication, and reference material. Switching with **Control-Right** then feels like turning your head to a second desk instead of shuffling windows.

A full-screen application receives its own Space automatically. That is why swiping sideways sometimes reveals a full-screen app you forgot about — it lives in its own slot in the Spaces bar.

You can pin an app to a Space permanently: right-click its Dock icon and choose **Options → Assign To → This Desktop**. From then on the app always opens there.

Keep it predictable

One setting undermines the whole system: **Automatically rearrange Spaces based on most recent use**, in Desktop & Dock settings. With it on, macOS reorders your Spaces behind your back, and "coding is one swipe left" stops being true. If Spaces keep swapping positions on you, this setting is the reason. Turn it off.

Keep the setup small enough that you still know where each window lives. Two or three Spaces with clear jobs beat seven you have to search through — searching through Spaces is the problem Mission Control was meant to solve, not recreate.

Screenshots and small automations

Capture the right part of the screen and turn repeated visual tasks into small built-in workflows.

Press **Command-Shift-3** for the full screen, **Command-Shift-4** for a selection, or **Command-Shift-5** for capture and recording controls.

Two refinements make these far more useful. After pressing **Command-Shift-4**, tap the Space bar: the crosshair becomes a camera and you capture a whole window, cleanly, with its shadow. And hold Control with any screenshot shortcut to copy the result to the clipboard instead of saving a file — perfect when the screenshot's destination is a chat message or document, not your Desktop.

Command-Shift-5 opens a control bar with the extra options: screen recording, a timer, and the save destination. Saved screenshots land on the Desktop by default, named with the date and time. A thumbnail appears in the corner for a few seconds; click it to crop or annotate before the file is written.

Small automations with Shortcuts

Use the **Shortcuts** app for repeated tasks such as resizing images or organizing selected files. A shortcut is a chain of actions: take an input, transform it, put the result somewhere.

A practical example: converting images to JPG. Build a shortcut from three actions — "Get Selected Files in Finder", "Convert Image", and "Move File" to send the result to the Desktop. In the shortcut's settings, enable **Use as Quick Action** for Finder. Now right-clicking any image shows **Quick Actions → Convert to JPG**.

One detail decides whether this feels polished or broken: in the first action, unselect every file type except images. Skip that, and the Quick Action appears when you right-click folders and PDFs too, where it does nothing useful.

You can run shortcuts from Terminal as well:

```
shortcuts list
shortcuts run "Convert to JPG" -i photo.png
```

`shortcuts list` prints every shortcut you have; `shortcuts run` executes one, with `-i` naming the input file.

Start with one action you already repeat often. One reliable three-step shortcut you use weekly is worth more than ten clever ones you forget exist.

Check your understanding: working faster

Check common shortcuts, Spotlight, file locations, Mission Control, Spaces, screenshots, recording, and small automations.

Answer each question before continuing. Read the explanation after every answer, including the ones you get right.

[Take this interactive quiz online](#)

Maintenance and security

Monitor resources, manage storage, update safely, encrypt data, and restore backups.

Use Activity Monitor

Find the application or process consuming CPU, memory, energy, disk, or network resources before forcing anything to quit.

Open Activity Monitor from **Applications** → **Utilities** or Spotlight. It shows processes using CPU, memory, energy, disk, and network resources.

Start from the symptom. If the Mac is hot and the fans are running, inspect CPU and Energy. If applications pause while switching, inspect Memory. If storage activity is constant, inspect Disk.

Sort the relevant column, then watch it for more than one moment. A short CPU spike while an application exports a video can be normal. A process that stays at the top while idle deserves investigation.

Read memory pressure, not only free memory

macOS uses spare memory for caches. The **Memory Pressure** graph combines several signals and is more useful than one free-memory number.

Swap use is not automatically a failure. Look for sustained yellow or red pressure, growing swap, slow application behavior, and a process whose memory continues to increase.

Inspect before quitting

Double-click a process to inspect it. Confirm the name, user, open files, and whether it belongs to an application or macOS service.

Try **Quit** first. It asks the process to stop cleanly. **Force Quit** ends it without normal cleanup and can lose unsaved work.

Do not force quit an unfamiliar system process just because its name is near the top. It may restart, or other functions may depend on it.

After quitting a problem application, verify whether CPU, memory pressure, or responsiveness improved. Then find the cause: an extension, damaged file, runaway task, or repeatable application bug.

Try this: open one application that performs visible work, observe its CPU and memory during and after the task, and explain why the peak alone is not enough to call it a problem.

Manage storage and updates

Find what consumes disk space and install operating-system updates without deleting unknown system data by hand.

Open **System Settings** → **General** → **Storage** to see broad categories and recommendations. Treat the category sizes as a starting point, not a precise map of files you can safely delete.

Begin with data you recognize:

- old downloads and disk images
- applications you no longer use
- videos or archives you created
- old device backups
- build artifacts that your tools can recreate

Move items to the Trash first. Confirm your applications and projects still work before emptying it.

Do not delete unknown Library data blindly

`~/Library` contains application settings, containers, caches, mail, and other account data. `/Library` contains shared resources.

Some caches are disposable, but deleting a random folder can sign you out, remove offline data, or break an application. Use the application's cleanup controls or current vendor guidance when available.

A cleanup tool also needs review. Full Disk Access gives it broad reach; a large “system data” number does not justify deleting files you cannot identify.

Prepare for macOS updates

Install security updates promptly, but treat a major macOS version upgrade as a project:

1. Complete and verify a backup.
2. Check free space and power.
3. Confirm critical applications, drivers, package managers, and development tools support the new version.
4. Record any tool versions you may need to reinstall.
5. Choose a time when recovery will not interrupt important work.

After updating, verify login, networking, backups, command-line tools, and the projects you depend on.

If storage remains unexpectedly full, use Finder or a trusted disk-usage tool to locate real files. Do not bypass macOS protections to delete system-managed data.

Try this: identify 1 GB of storage you could recover from files you recognize, but stop before deleting it. Write the verification you would perform afterward.

FileVault and account security

Protect the startup disk, keep a recoverable account, and understand the consequence of losing every recovery method.

FileVault encrypts the startup volume so someone without an authorized login or recovery method cannot read its data by removing or directly accessing the disk.

Turn it on from **System Settings** → **Privacy & Security** → **FileVault**. Keep the Mac connected to power while initial encryption completes.

Recovery is part of encryption

macOS offers a recovery method. Store it somewhere separate from the Mac and test that the responsible person knows where to find it.

If every authorized password and recovery method is lost, the encryption is doing its job: the data may be unrecoverable. Do not enable FileVault on a shared or business Mac without deciding who owns recovery.

Before changing account passwords or removing an account, confirm that another authorized recovery path remains.

Protect the unlocked session

Disk encryption mainly protects data when the Mac is locked or powered off. It cannot protect files from someone using an already unlocked session.

Use a strong, unique login password. Require the password promptly after sleep or the screen saver. Do not enable automatic login on a Mac containing sensitive data.

Keep separate accounts for separate people. An administrator account can change more of the system, so use a standard account for everyday work when that fits your setup.

FileVault does not replace backups. An encrypted disk can still fail, and encrypted files can still be deleted or corrupted while you are signed in.

Try this: without changing settings, review whether FileVault is enabled, which accounts can unlock the disk, where recovery is stored, and whether the Mac locks automatically.

Time Machine and recovery

Create an automatic backup, verify that it completes, and restore one file before you need recovery for real.

Time Machine can automatically back up applications, documents, photos, mail, and other files to an external or supported network disk.

Choose a backup destination with enough capacity and keep it separate from the Mac's internal disk. Local snapshots can help recover a recent file, but they are stored on the same disk and cannot protect you from that disk failing or the Mac being lost.

Sync and backup solve different problems

A sync service keeps files available across devices. If you delete or corrupt a synchronized file, that change can propagate.

A backup keeps recoverable versions according to its retention policy. For important work, use both intentional synchronization and a separate backup.

Verify more than the green status

After configuring Time Machine:

1. Wait for a completed backup.
2. Check the timestamp and destination.
3. Create a disposable file and let it back up.
4. Delete the local copy.
5. Restore it through Time Machine.
6. Open the restored file and compare its contents.

This small test verifies the path you will need during a real recovery.

Know the recovery options

Restoring one file is different from moving to a new Mac or recovering after disk failure. Keep a current recovery contact, account credentials, FileVault recovery method, and any encryption password needed for the backup.

Do not discover during an emergency that the backup disk was disconnected for months or that nobody knows its password.

Review backup exclusions too. Large virtual machines, external disks, or project folders may be omitted intentionally or accidentally.

Try this: restore one disposable file from your backup. Record how old the restored version is and which credential the recovery required.

Check your understanding: maintenance and security

Check Activity Monitor, memory pressure, storage, updates, FileVault, screen locking, Time Machine, and recovery.

Answer each question before continuing. Read the explanation after every answer, including the ones you get right.

[Take this interactive quiz online](#)

The interactive version of this course is available at <https://flaviocopes.com/courses/macOS-basics/>.