

Unit 4

UI Components

| What this unit covers

§	Topic	What you must be able to do
3.1	System menus & functions of menus	Define a menu; state and illustrate its four functions.
3.2	Formatting of menus	Apply all eight formatting aspects to a menu design.
3.3	Types of menus	Name, describe and choose between the six menu types.
3.4	Windows, components & presentation styles	Label a window's parts; compare tiled, overlapping and cascading.
3.5	Types of windows	Distinguish primary, secondary and dialog boxes.
3.6	Screen-based controls	Use operable, text-entry and selection controls correctly.
3.7	Other operable controls	Apply slider, tabs, date picker, tree view and scroll bars.
3.8	Selecting the proper controls	Justify a control choice from the task and the data.
3.9	Meaningful graphics, icons and images	Design and judge icons and images.

SYLLABUS 3.1

System Menus & Functions of Menus

3.1 · What is a system menu?

A **menu** is a set of options presented on screen, from which the user chooses one. The **system menu** is the complete set of such options through which the whole application is operated.

The key idea: a menu lets the user **recognise** a choice instead of having to **recall** a command. Recognition is far easier than recall, that is why menus replaced typed commands.

Why menus are used

- They **show what is possible** : the user need not know the system in advance.
- They **prevent errors**: only valid options are offered.
- They **reduce memory load**: nothing has to be memorised.
- They **structure the system**: related commands are collected together.

Command line

```
copy a.txt b.txt
```

The user must **recall** the exact command and its syntax. Powerful for experts, unusable for beginners.

Menu

Edit ▶ Copy

The user **recognises** the option from a list. Slower for experts, but learnable by anyone in seconds.

Trade-off: menus are slower for expert users. That is why good menus also show **keyboard shortcuts**.

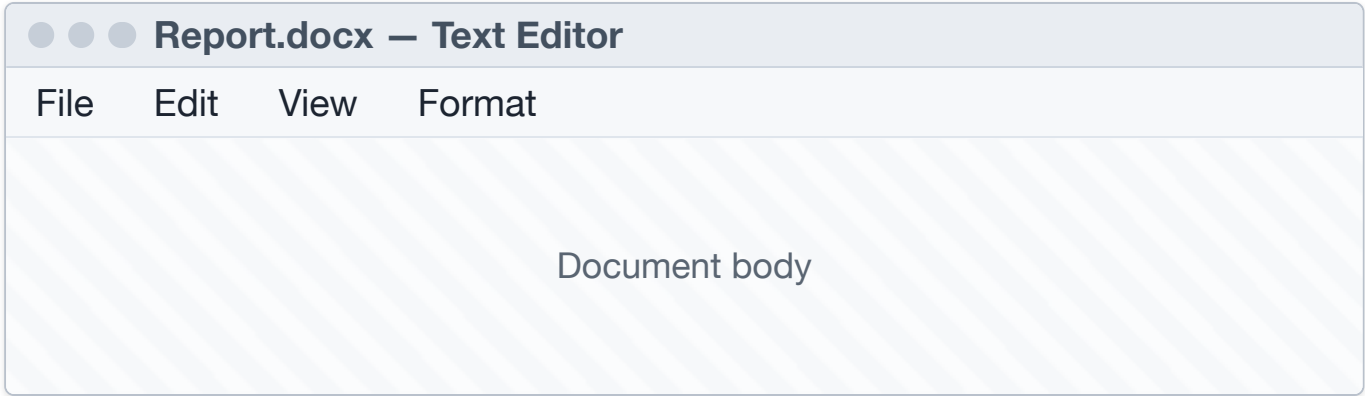
Beginners use the menu, experts graduate to `Ctrl+C`.

3.1 · Functions of menus

A menu is not only for running commands. It performs **four** distinct functions.

Function	Example
1 · Navigation move to another screen, view or level	View ▶ Full Screen File ▶ Export ▶ (opens a submenu)
2 · Execute a command perform an action immediately	File ▶ Print Edit ▶ Copy
3 · Display information show the current state or setting	Format ▶ Bold ✓ the tick says Bold is currently ON
4 · Data / parameter entry supply a value to the system	Format ▶ Font Size ▶ 14 pt the choice <i>is</i> the data

LIVE MENU BAR interactive — see live deck



Open a menu and choose an item.

SYLLABUS 3.2

Formatting of Menus

Consistency · display · presentation ·
organization · complexity · item arrangement ·
ordering · grouping

3.2 · Consistency, display & presentation

1 · Consistency

Menus must look and behave the **same everywhere** in the application: same position, same wording, same visual cues.

2 · Display

How the menu is **brought onto the screen**: it must appear quickly, near the point of use, without hiding what the user is working on.

3 · Presentation

How the menu **looks**: clearly separated from the background, legible type, adequate spacing, a visible border or shadow.

Rules to apply

- **Consistency**: if "File" is leftmost in one window, it is leftmost in every window. The same command keeps the same name — never "Delete" here and "Remove" there.
- **Display**: a pull-down opens directly below its title; a pop-up opens at the pointer. The menu must never cover the object being acted on.
- **Presentation**: enough contrast to be read at a glance, and enough padding that items are not mistaken for one another.

Poor

Menu in a different place on each screen · "Save" on one screen, "Store" on another · grey text on a grey background · items touching each other.

Good

Same menu bar position everywhere · one name per command · dark text on white with a clear border · comfortable row spacing.

3.2 · Organization & complexity

4 · Organization

Arrange menus so their **structure is obvious**. Keep the path visible, show the user where they are, and keep the main menu on screen while a submenu is open.

5 · Complexity

Offer a **simple menu** containing the common choices, with an **expanded menu** available for everything else. Do not force every user to face every option.

Why complexity matters: about 80% of users use about 20% of the commands. A menu built for the remaining 20% punishes everyone else.

Simple vs expanded — a worked example

Simple menu (default)	Expanded menu (on request)
Print	Print
Print Preview	Print Preview
More options ▶	Print Range...
	Duplex Settings...
	Colour Profile...
	Print to File...

The beginner sees three items and is not frightened. The expert opens "More options" once and gets everything.

3.2 · Item arrangement & grouping

6 · Item arrangement

Lay the items out so the eye can scan them: **one item per line**, left-aligned text in a single column, shortcuts right-aligned in their own column, and markers in a fixed position.

8 · Grouping

Put **related items together** and separate the groups with a thin line. Keep each group to about **5 ± 2** items.

Why 5 ± 2: short-term memory holds only a handful of items at once. Fourteen ungrouped commands is a wall of text; the same fourteen in five labelled groups can be scanned in a second.

Markers also belong to arrangement: a dialog will open · a submenu follows · ☒ the setting is on · grey not available now.

SAME 14 COMMANDS, TWO WAYS [interactive — see live deck](#)

Show grouped

Print
Bold
New
Copy
Italic
Save
Paste
Open
Underline

Ungrouped — 14 items in random order. Every search is a scan of the whole list.

3.2 · Ordering

Aspect 7. The right order depends on the **nature of the data** and the **expertise of the user**.

Scheme	Use when...	Example
Frequency of use	a few items are used far more than the rest	Home at the top
Sequence of use	the task has a natural order	File → Edit → Print
Importance	one action matters most	Save above Exit
Alphabetical	the list is long and unrelated	Country names
Categorical	items fall into natural groups	Font / Paragraph / Styles
Numeric	values have a natural order	Font size 8, 9, 10...

Never reorder a menu automatically based on recent use. Users find items by *position* and muscle memory; moving them destroys that.

CHOOSE THE ORDERING SCHEME interactive — see live deck

A form asks for the user's country from all 195 options. Which ordering scheme?

Frequency of use

Alphabetical

Sequence of use

Importance

✗ Not quite. The list is long and the items have no workflow between them — the user already knows the answer and only needs to find it. Alphabetical makes finding predictable.

SYLLABUS 3.3

Types of Menus

Menu bar · pull-down menu · cascading menu ·
pop-up menu · tear-off menu · iconic menu

3.3 · The six types at a glance

Type	Description	Example	Best used for
Menu bar	A fixed horizontal row of menu titles across the top of a window.	File Edit View Format	The main entry point to an application.
Pull-down	A list that drops down when a menu-bar title is chosen.	File ▼ New, Open, Save	Grouping the commands of one topic.
Cascading	A submenu that slides out sideways from an item marked ▶.	File ▶ Export ▶ PDF	Choices that have sub-choices.
Pop-up	A menu that appears at the pointer, usually on right-click.	Right-click a file	Actions for one specific object.
Tear-off	A pull-down that can be detached and left floating on screen.	Photoshop tool panels	Commands used over and over.
Iconic	A menu of pictures instead of words, always visible.	A toolbar of icons	Frequent actions; multilingual and touch UIs.

How they relate: a *menu bar* holds *pull-down* menus; a pull-down item may open a *cascading* submenu, or be torn off into a *tear-off* panel. A *pop-up* is invoked at the pointer instead of from the bar, and an *iconic* menu replaces words with pictures.

3.3 · Menu bar & pull-down menu

Menu bar

A **horizontal row of menu titles** fixed at the top of a window. It is always visible, so it advertises what the application can do.

- Keep to about **7 titles** — the bar must not wrap to a second line.
- One-word titles wherever possible.
- Order by sequence of use: File, Edit, View... Help is always last.

Pull-down menu

The list that **drops down** when a menu-bar title is selected.

- Opens directly **below its title**, left edges aligned.
- Must not run off the bottom of the screen.
- Group related items and separate the groups with a line.

ILLUSTRATION



The **menu bar** is the row. The list hanging below "File" is the **pull-down menu**.

3.3 · Cascading menu

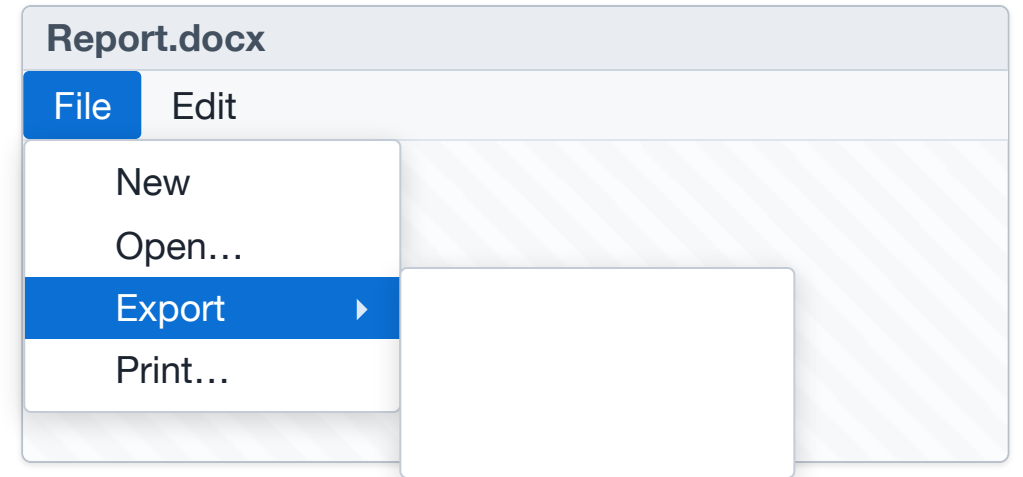
A **cascading menu** (or submenu) slides out **sideways** from a parent item, giving a second level of choices. The parent item carries a ▶ **cascade indicator** to promise that more is coming.

Guidelines

- Always show the ▶ marker — otherwise the user expects an action and gets a submenu.
- Limit to **two levels**. Three levels of cascade is very hard to use.
- The submenu title must repeat the wording of the item that opened it.
- Keep the parent item highlighted while the submenu is open, so the path stays visible.

The weakness: the user must keep the pointer inside a narrow corridor to reach the submenu. Slip off and it closes. This is why cascading menus are avoided on touch screens entirely.

ILLUSTRATION — TWO LEVELS



File ▶ Export ▶ PDF Document — the second level **cascades** out to the right.

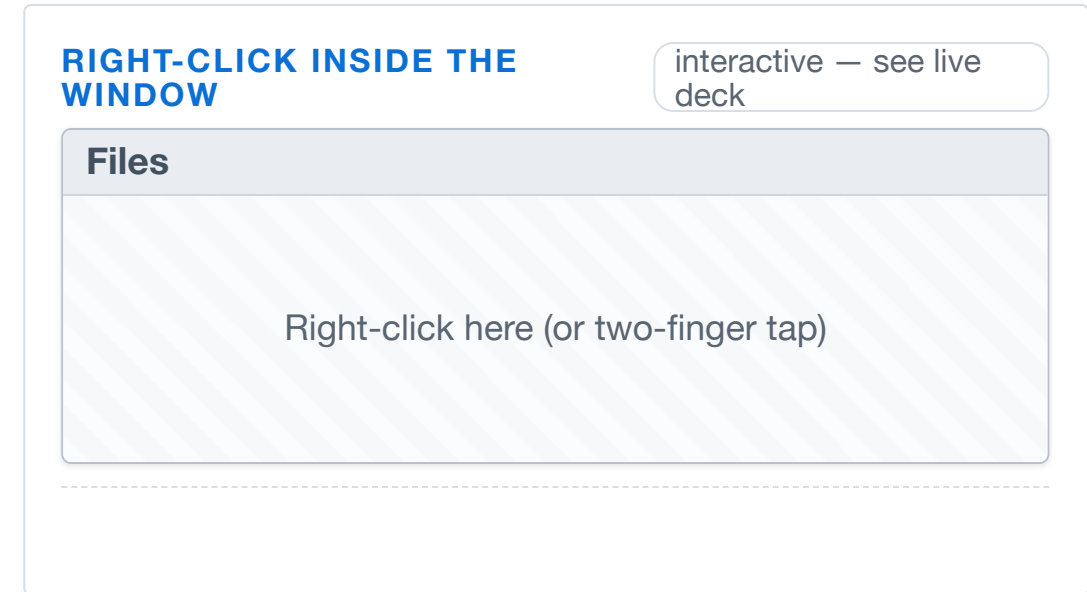
3.3 · Pop-up (context) menu

A **pop-up menu** appears **at the pointer**, usually on right-click. It is also called a **context menu** because it shows only the commands that make sense for the object that was clicked.

Advantages

- **Fast** — appears where the eye and hand already are; no travel to the menu bar.
- **Short and relevant** — only valid actions for that object are listed.
- **Saves screen space** — invisible until it is needed.

The one rule: a pop-up menu is ***hidden*** — a beginner may never discover it. Every command in it must also be reachable from a visible menu or toolbar. Never make right-click the only way to do something.



| 3.3 · Tear-off menu

A **tear-off menu** is a pull-down menu that the user can **detach** from the menu bar and leave floating on screen as a small window, so its commands stay available without reopening the menu each time.

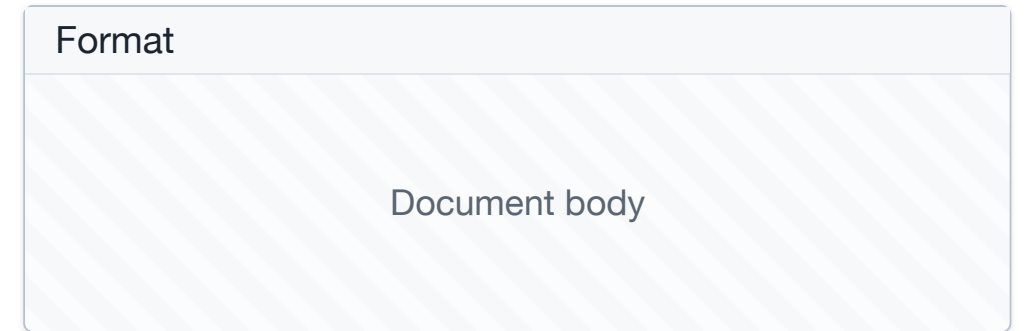
Why it exists

- Some commands are used **over and over** — formatting, drawing tools, colours.
- Without tearing off, each use costs three actions: open the bar, open the menu, choose.
- Torn off, it costs **one** click.

Advantage	Disadvantage
Repeated commands become one click; the panel can be placed near the work.	Takes up screen space and can clutter the workspace if several are torn off.

Real examples: Adobe Photoshop tool panels, GIMP dockable dialogs, older Microsoft Office toolbars.

TEAR THE MENU OFF interactive — see live deck



Open [Format](#), then click the dashed handle to tear the menu off.

3.3 · Iconic menu

An **iconic menu** presents its choices as **pictures instead of words**. The familiar toolbar is an iconic menu: it is always visible and each symbol stands for a command.

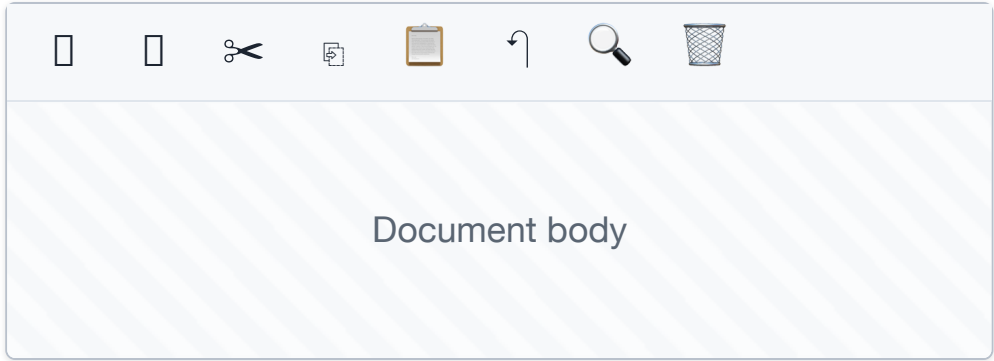
Advantage	Disadvantage
Compact — many commands in a small strip.	Meaning can be ambiguous.
Language-independent — no translation needed.	Must be learned if the symbol is arbitrary.
Recognised faster than text, once learned.	Small targets are hard to hit.
Essential on touch screens and small displays.	Needs a tooltip or label to be safe.

The safe rule: use an icon *and* a label for anything that is not universally understood. A tooltip on hover is the minimum.

ICONS WITH AND WITHOUT LABELS

interactive — see live deck

Show labels



Hover an icon for its tooltip. Which one is "Paste"?

SYLLABUS 3.4

Windows & Their Components

Selection of windows · components of a window
· presentation styles: tiled, overlapping,
cascading

| 3.4 · Windows and when to use them

A **window** is an independent rectangular area of the screen that displays information and accepts input. It behaves like a sheet of paper on a desk: it can be moved, resized, stacked and set aside.

Selecting windows — use another window when...

- The user must **compare** two things side by side.
- A task must continue while **another is referred to**.
- The content is genuinely **independent** of the current work.
- A **decision or extra input** is needed before a command can finish.

Do not open a window for every action. Each one must be positioned, read and closed — that is work the user did not ask for.

Advantages

- Several tasks visible at once — compare, copy, refer
- Reduces memory load: information stays on screen
- Access to more information than one screen holds
- Multiple contexts stay alive together

Disadvantages

- Windows must be managed — extra work for the user
- Screen clutter; windows get lost behind others
- Small windows show too little information
- The window system is itself something to learn

3.4 · Components of a window

CLICK ANY PART TO IDENTIFY IT

interactive — see live deck

Sales Report — Spreadsheet

— □ ×

File

Edit

View

B

I

U

≡

#

Work area — the content itself

Ready

Row 12, Col 4

▲

Click any part.

Component	Purpose
Frame / border	Bounds the window and separates it from what is behind. Drag an edge to resize.
Title bar	Names the window so it can be identified when stacked. Drag it to move the window.
Title-bar icon	Identifies the application and opens the window menu.
Sizing buttons	Minimise, maximise / restore and close.
Menu bar	All commands available in this window, in pull-down menus.
Toolbar	Icon shortcuts to the most frequent commands.
Work area	Where the user's content lives. Everything else serves this.
Scroll bar	Reveals content larger than the frame; the thumb shows how much is off-screen.
Status bar	Current state, progress and messages. Never interrupts.

Also found on many windows: a **split box** (divides the work area into panes), a **command area** (type a command directly) and a **size grip** (the resize handle in the bottom-right corner).

3.4 · Window presentation styles

1 · Tiled windows

The screen is **divided** between the windows; they never overlap. Enlarging one shrinks another.

2 · Overlapping windows

Each window has its **own size and position** and may cover the others, like sheets of paper on a desk.

3 · Cascading windows

A tidy overlapping arrangement, stepped **diagonally** so that every title bar remains visible.

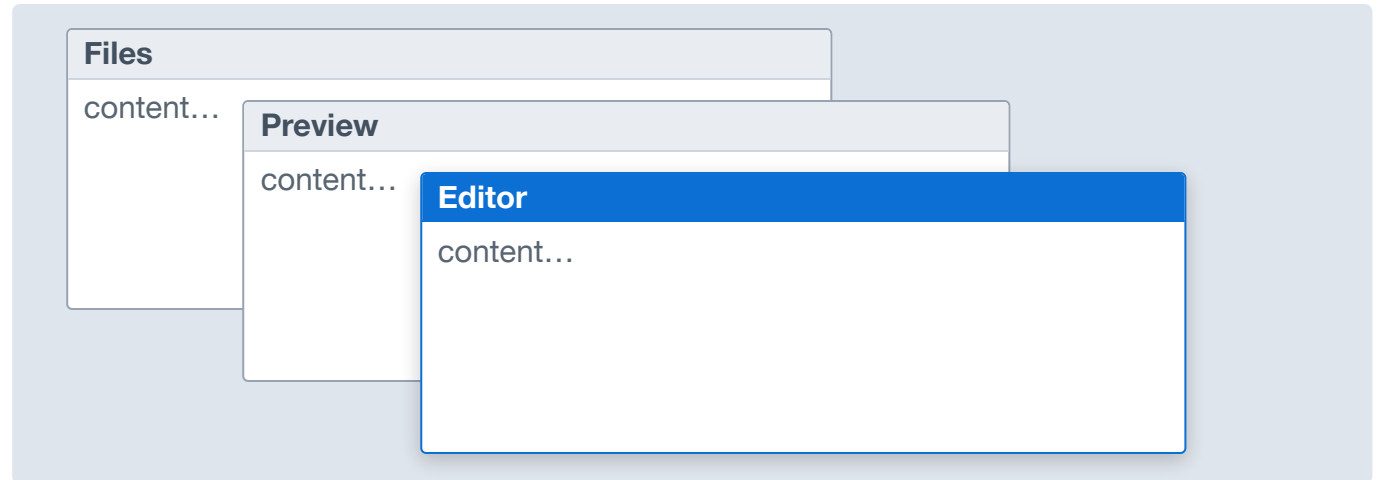
ARRANGE THE WINDOWS

[interactive](#) — [see live deck](#)

Tiled

Overlapping

Cascading



Stepped diagonally — every title bar stays visible.

3.4 · Comparing the three styles

Style	How it works	Advantages	Disadvantages
Tiled	Screen space is divided; no window ever covers another.	Nothing is hidden · predictable layout · easy for beginners · no management needed.	Few windows fit · each is small and cramped · wastes space when content sizes differ.
Overlapping	Windows are sized and placed freely and may cover one another.	Larger windows · familiar desk metaphor · user controls size and position · more windows at once.	Windows hide each other and get lost · more clicks to manage · can look cluttered.
Cascading	Overlapping windows stepped diagonally from the top-left.	Every title bar stays visible · tidy and orderly · easy to pick any window.	Only the front window's content is visible · a tidy-up command, not a working layout.

Which to use: choose ***tiled*** when the user must see everything at once (a dashboard, a comparison). Choose ***overlapping*** when the user needs one large window at a time and will switch between tasks. ***Cascading*** is not a working layout — it is a way of tidying overlapping windows so none is lost.

SYLLABUS 3.5

Types of Windows

Primary windows · secondary windows · dialog
boxes

3.5 · Primary & secondary windows

Primary window

The **main window** of the application, where the principal work is done. It is **independent** — it does not depend on any other window.

- Contains the menu bar, toolbar and the main work area
- Can be minimised, maximised and resized freely
- Closing it usually ends the task

Examples: the MS Word document window, a browser window.

Secondary window

Dependent on a primary window. It performs a supporting task related to the work in the primary window.

- Always associated with a parent window
- Closes automatically when its parent closes
- Usually smaller, and often cannot be maximised

Examples: Find and Replace, Options / Preferences, the Font settings window.

The test: can this window exist on its own? If yes it is **primary**; if it dies with its parent it is **secondary**. The user's main work must never happen in a secondary window.

3.5 · Dialog boxes

A **dialog box** is a secondary window that **carries on a dialogue** with the user: it asks a question, reports something, or collects the information needed to finish a command.

Kind	Purpose & example
Modal	Blocks the parent window until answered. "Save changes before closing?"
Modeless	Leaves the parent usable while it stays open. Find and Replace.
Message box	Delivers information, a warning or an error. "File not found."
Property / settings	Shows and edits the settings of an object. Paragraph settings, often in tabs.

Use modal sparingly. A modal dialog takes control away from the user. Use it only when work cannot correctly continue without an answer — a confirmation before deleting, or required input. Overused, users dismiss it without reading.

FEEL THE DIFFERENCE interactive — see live deck

Open modal

Open modeless

Try typing here while each one is open:

Type something...

SYLLABUS 3.6

Screen-Based Controls

Operable controls: buttons, toolbars · text entry /
read-only controls · selection controls: radio
buttons, checkboxes, palettes, list boxes, list
view controls, drop-down / pop-up list boxes

3.6 · What is a screen-based control?

A **screen-based control** (also called a ***widget***) is an object on the screen that the user operates to give the system a command or a piece of information. Buttons, text boxes, checkboxes and lists are all controls.

Every control makes a promise. Its shape tells the user what it will do: a round radio button promises "only one", a square checkbox promises "any number", a raised button promises "something happens when you press me". Break that promise and the interface lies to the user.

1 · Operable controls

Operated to **cause an action**.

Buttons · toolbars

2 · Text entry / read-only

Accept typed information, or display it without allowing changes.

Single-line and multiple-line text boxes · read-only fields

3 · Selection controls

Let the user **choose from a fixed set** of values.

Radio buttons · checkboxes · palettes · list boxes · list view controls · drop-down / pop-up list boxes

3.6 · Operable controls — buttons & toolbars

Control	Purpose and guidelines
Command button	Causes an action <i>immediately</i> . Label it with a verb that says what will happen — "Save", "Delete", "Search" — not "OK" when a real verb exists.
Default button	The one activated by <code>Enter</code> . Visually emphasised. It must be the safest choice, never the destructive one.
Toggled button	Stays pressed to show an ON state — Bold, Italic. The pressed appearance must be unmistakable.
Toolbar	A row of buttons for the most frequently used commands, always visible. Every toolbar button needs a tooltip and an equivalent in the menus.

Button order and spacing. Keep affirmative before dismissive (**Save · Cancel**), place buttons in the same position on every screen, and leave enough space between a confirm button and a destructive one that a slip cannot be fatal.

COMMAND VS TOGGLED BUTTONS

interactive — see live
deck



Press `Tab` to move between them — a control that cannot be reached from the keyboard is broken.

Click [Save](#), then click [B](#). Notice the difference.

3.6 · Text entry & read-only controls

Control	Use it for
Single-line text box	Short data of a known kind — name, email, phone, PIN.
Multiple-line text box	Longer or unpredictable text — address, comments, description. Wraps and scrolls.
Read-only control	A value the user may see but not change — account number, total, generated ID. Shown greyed and not editable.

Guidelines

- **Size the box to the data.** A narrow box for a PIN, a wide one for an email. The width is a silent hint about what is expected.
- **Always give a visible label.** Placeholder text is not a label — it disappears the moment the user starts typing.
- **Show the format before the error** — `DD/MM/YYYY`.
- **Say how to fix it.** "Invalid input" tells the user nothing.

ALL THREE KINDS [interactive](#) — [see live deck](#)

Email — single-line text box

Address — multiple-line text box

Account number — read-only

Type an email with no "@", then click away.

3.6 · Selection controls — radio buttons & checkboxes

	Radio button	Checkbox
Choose	<i>Exactly one</i> of a set	<i>Zero, one or many</i>
Relationship	Mutually exclusive	Independent of each other
Shape	Round ○	Square □
Can it be cleared?	No — one stays selected	Yes — click again to clear
How many options	2–8, all visible	Any number
Example	Gender · Payment method	Hobbies · Terms accepted

The classic mistake: a *single* radio button. One radio can never be unselected, so the user is trapped. A single on/off choice is always a **checkbox**.

TRY TO BREAK THEM interactive — see live deck

Shipping speed — radio (pick one)

- ☒ Standard (5 days)
- ☐ Express (2 days)
- ☐ Overnight

Extras — checkbox (pick any)

- ☐ Gift wrap
- ☐ Insurance
- ☐ Signature on delivery

Speed: **Standard (5 days)** — always one. Extras: **0** selected.

3.6 · List boxes & drop-down / pop-up list boxes

Control	Description	Choose it when...
List box	A permanently visible, scrollable list of choices.	Options are consulted often and there is room to show them.
Drop-down / pop-up list box	Shows only the current value until it is opened.	Screen space is tight and the value is rarely changed.
Multiple-selection list box	A list box that allows several items to be selected at once.	Many independent choices, where a long checkbox list would not fit.

How many options?

2–8 → **radio buttons** (all visible, one click)

8–15 → **drop-down list box**

15+ → drop-down with type-ahead, or a search field

Order the list sensibly — alphabetical for long lists, most-likely-first for short ones. Never hide a critical choice inside a closed drop-down.

THREE LIST CONTROLS interactive — see live deck

List box (always visible)

Kathmandu
Pokhara
Lalitpur
Bhaktapur
Biratnagar
Bharatpur

Drop-down list box (opens on click)

Kathmandu ▼

Multiple-selection list box

Nepali
English
Maithili
Newari
Bhojpuri

Try each one. Which shows you the most information?

| 3.6 · Palettes

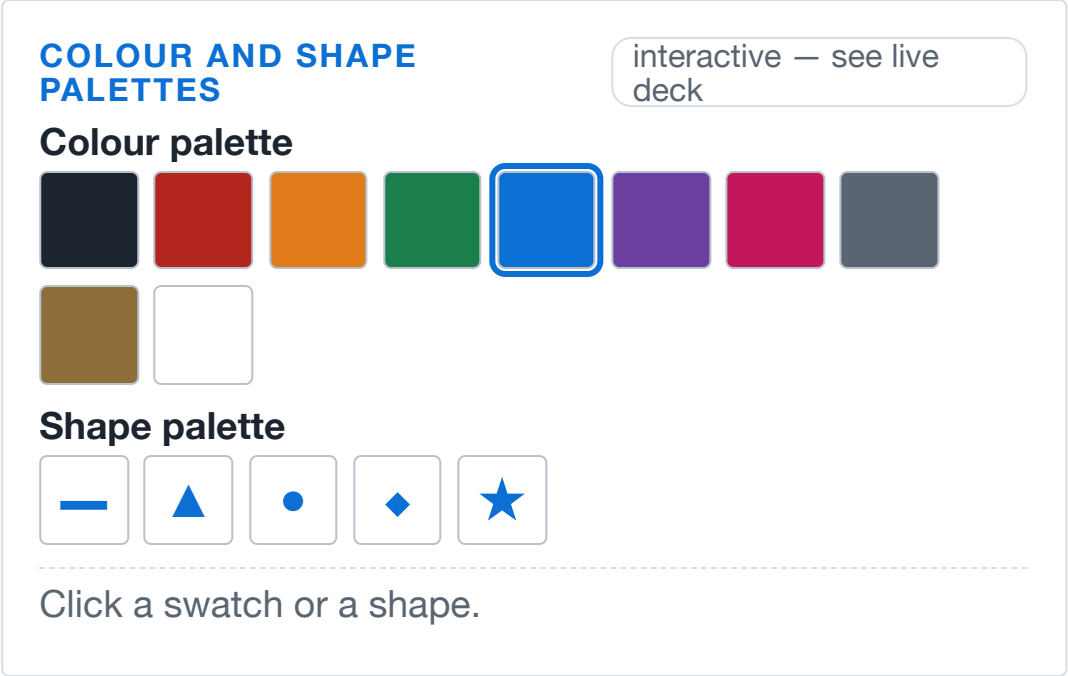
A **palette** is a selection control that presents its choices **graphically** — as a grid of colours, patterns, shapes or tools — rather than as words.

When to use a palette

- The choices are **naturally visual** — a colour, a brush, a border style.
- The user would **recognise** the option instantly but could not name it.
- Showing the choice is **clearer than describing** it: "#0b6fd4" means nothing, but the blue square is obvious.

Advantage	Disadvantage
Instant recognition · compact · no language needed · shows the result directly.	Needs screen space · unusable for choices that are not visual · small swatches are hard to hit and hard to tell apart.

Examples: the colour picker in MS Word, the tool palette in Photoshop, border and shading choosers.

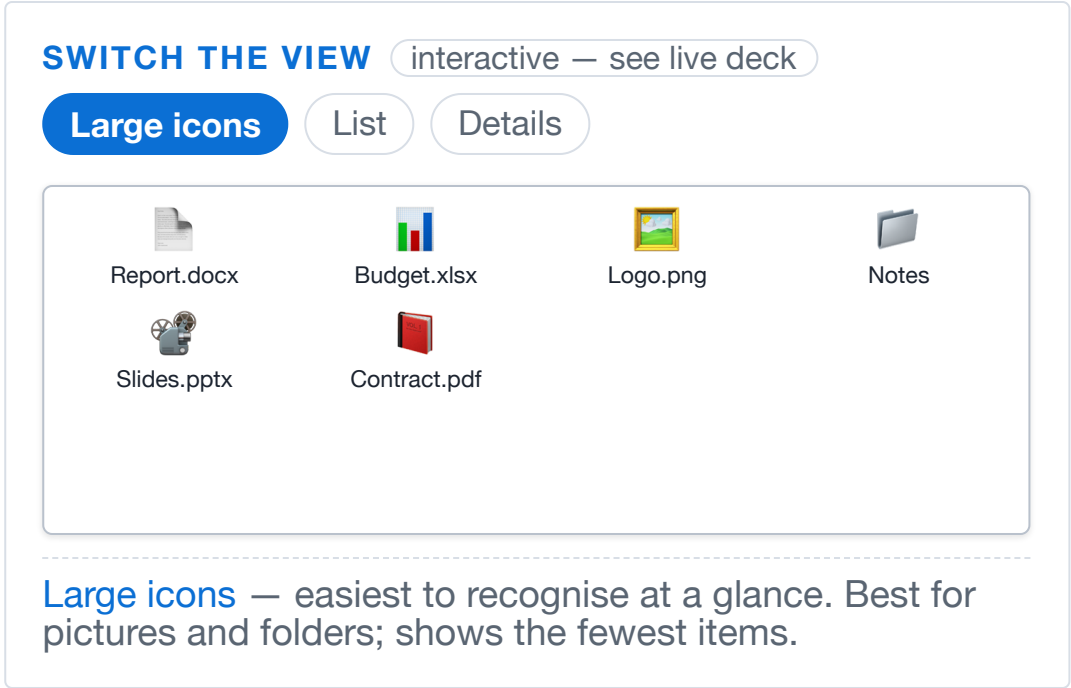


3.6 · List view controls

A **list view control** displays a collection of items — files, records, products — and lets the user **switch how that same collection is shown**. Windows Explorer is the standard example.

View	Shows	Best for
Large icons	A big picture with the name beneath.	Photographs and folders — recognising by appearance.
List	Small icon and name, in compact columns.	Finding a known name in a long list.
Details	One row per item with sortable columns.	Comparing size, type or date; sorting.

Why it matters: the **same data** serves different tasks. "Which photo is this?" needs large icons. "Which file is biggest?" needs details. A list view control lets the user choose, instead of the designer guessing.



SYLLABUS 3.7

Other Operable Controls

Slider · tabs · date picker · tree view · scroll bars

3.7 · Slider, tabs, date picker, tree view, scroll bars

Control	Use it for	Guideline
Slider	Choosing a value in a continuous range — volume, brightness, price.	Always display the current value; a slider with no number is guesswork.
Tabs	Dividing one window into related sections shown one at a time.	One row of tabs only; never nest tab bars; 1–2 word labels.
Date picker	Choosing a date or a date range from a calendar.	Prevents impossible dates like 31/02 — but typing must still be allowed.
Tree view	Showing hierarchical data — folders, categories, an organisation chart.	Indent each level clearly, mark expandable nodes, remember what was expanded.
Scroll bar	Moving through content larger than its window.	The thumb's size must show how much content is off-screen.

Slider or text box? Use a *slider* when the approximate feel matters and the exact number does not — volume. Use a *text box* when the precise value matters — quantity = 3.

ALL FIVE, LIVE interactive — see live deck

Slider — volume 60%



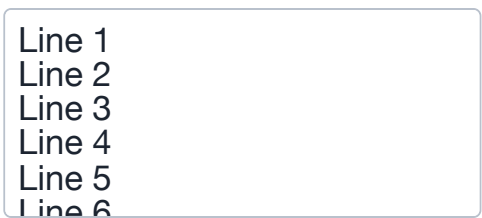
Details Shipping Reviews

Tabs — product details panel.

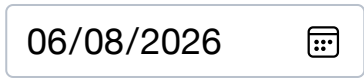
Tree view



Scroll bar



Date picker



Tree view — shows hierarchy. Indent clearly and remember what was expanded.

SYLLABUS 3.8

Selecting the Proper Controls

3.8 · Choosing the proper control

If the user must...	Use	Because
Cause an action immediately	Command button	Immediate and unambiguous; labelled with a verb.
Choose exactly one of 2–8 options	Radio buttons	All options visible; exclusivity is obvious from the shape.
Choose exactly one of many (8+)	Drop-down list box	Saves space and still allows only one choice.
Choose any number of options	Checkboxes	Each is independent and can be cleared.
Switch one setting on or off	Single checkbox	A lone radio button can never be unselected.
Choose several from a long list	Multiple-selection list box	Compact; a long checkbox list would not fit.
Choose something visual	Palette	Recognised instantly; cannot easily be named in words.
Enter short known text	Single-line text box	Its width signals the expected length.
Enter long or unpredictable text	Multiple-line text box	Wraps and scrolls without hiding the input.
Set an approximate value in a range	Slider	Direct and visual; the exact number is not needed.
Move between sections of one window	Tabs	Keeps one context; avoids opening another window.
Browse a hierarchy	Tree view	Shows parent–child relationships directly.

PICK THE RIGHT CONTROL [interactive — see live deck](#)

1 / 3 · A settings screen has one option: "Send me email notifications" — on or off. Which control?

Two radio buttons: On / Off

A single checkbox

A drop-down with Yes / No

A slider from 0 to 1

SYLLABUS 3.9

Creating Meaningful Graphics, Icons & Images

| 3.9 · Meaningful graphics and icons

An **icon** is a small picture that represents an object, an action or a concept. Icons are **visual shorthand**: the eye recognises a shape faster than it reads a word, and a picture crosses language barriers.

What icons represent

- **Objects** — a file, a folder, a printer
- **Actions** — save, delete, share, print
- **States** — battery low, muted, syncing
- **Navigation** — home, back, settings

Four kinds of icon

Icon	Kind	How it carries meaning	Example
	Resemblance	Looks like the thing itself.	Scissors = Cut
	Symbolic	Shows a typical example of the idea.	Padlock = Secure / Locked
	Arbitrary	No natural link — it must be learned.	Three lines = Menu

Which kind is safest? **Resemblance** icons need no teaching at all. **Arbitrary** icons must be learned, so they always need a label, a tooltip and repetition before they mean anything.

The floppy disk problem. The Save icon is a floppy disk — an object most students have never held. It survives only because it has been **universally learned**. A new icon you invent will not get that luck: test it, or label it.

Graphics beyond icons

Diagrams, charts and illustrations are meaningful graphics when they carry information the words cannot — a process, a relationship, a proportion. If a graphic can be removed without losing meaning, it is decoration, and decoration costs loading time and attention.



Analogy

Something with a similar quality.

Trolley =
Shopping cart

3.9 · Principles of icon design

Principle	What it means
Familiarity	Use a metaphor the user already knows from the real world.
Clarity	Readable at the size it will actually appear — often 16–24 px.
Simplicity	The fewest elements that still carry the meaning. No fine detail.
Consistency	One visual style across the whole set — same weight, corners, perspective.
Directness	The link between the picture and its meaning needs no explanation.
Discriminability	Each icon clearly different from its neighbours at a glance.
Efficiency	One icon serves several related situations without redrawing.
Context	Meaning can change with surroundings — test the icon in place.

The five-second test. Show the icon to someone who has never used the product. If they cannot say what it does in five seconds, add a label. **Icon + label beats icon alone** for everything except the most universal symbols.

ICON RECOGNITION interactive — see live deck

Click the icon that means Delete.



3.9 · Using images meaningfully

When an image earns its place

- It shows something words cannot describe efficiently — a product, a place, a process.
- It helps the user **decide** — the photograph of the item being bought.
- It shows **how to do** something — a step in an instruction.
- It conveys a **relationship or proportion** — a chart or a diagram.

Decoration is not free. An image that carries no information still costs loading time, screen space and the user's attention. If removing it loses nothing, remove it.

Guidelines

- **Relevance** — every image must relate to the content beside it.
- **Quality and size** — sharp enough to be useful, compressed enough to load fast.
- **Consistency** — one style, crop and proportion across a set.
- **Alternative text** — describe every meaningful image in words, for users who cannot see it and for when it fails to load.
- **Caption where needed** — say what the user is looking at.
- **Never put important text inside an image** — it cannot be searched, selected, translated or read aloud.

The test for any graphic, icon or image: does it help the user *understand*, *decide* or *act*? If not, it is decoration.

Unit 4 — summary

§	Topic	The points to remember
3.1	System menus & functions	Recognition over recall. Four functions: navigation · execute a command · display information · data entry.
3.2	Formatting of menus	Consistency · display · presentation · organization · complexity · item arrangement · ordering · grouping. Groups of 5 ± 2 .
3.3	Types of menus	Menu bar · pull-down · cascading (►, two levels) · pop-up (at the pointer) · tear-off (detachable) · iconic (pictures).
3.4	Windows & components	Nine components. Tiled (no overlap) · overlapping (may cover) · cascading (all title bars visible).
3.5	Types of windows	Primary is independent · secondary depends on a parent · dialog boxes ask, and modal ones block.
3.6	Screen-based controls	Operable (buttons, toolbars) · text entry / read-only · selection (radio = one, checkbox = many, palettes, list boxes, list views, drop-downs).
3.7	Other operable controls	Slider · tabs · date picker · tree view · scroll bars — each matched to the shape of its data.
3.8	Selecting the proper controls	Ask: how many may be chosen? how many options are there? is the choice visual?
3.9	Graphics, icons and images	Four kinds of icon · eight design principles · icon + label beats icon alone · decoration is not free.

Likely exam questions

1 · State the functions of menus with examples. (3.1)

Show answer

2 · Explain any four aspects of menu formatting. (3.2)

Show answer

3 · Differentiate cascading and pop-up menus. (3.3)

Show answer

4 · Explain the three window presentation styles. (3.4)

Show answer

5 · Differentiate radio buttons and checkboxes. (3.6)

Show answer

6 · What is a list view control? (3.6)

Show answer

Check yourself

FIVE QUESTIONS ACROSS THE WHOLE UNIT

interactive — see live deck

1 / 5 · Which menu type appears at the pointer, usually on right-click?

Cascading menu

Pop-up menu

Tear-off menu

Menu bar

References

Books

- Galitz, W. O. — ***The Essential Guide to User Interface Design***, Wiley. The primary text for this unit; most exam wording follows it.
- Shneiderman, B. & Plaisant, C. — ***Designing the User Interface***.
- Norman, D. — ***The Design of Everyday Things***.

Guidelines

- **Microsoft Windows UX Guidelines** — menus, windows, common controls.
- **Apple Human Interface Guidelines** — controls and window behaviour.
- **Material Design** — components and their states.

Note: every diagram and widget in this deck is drawn in HTML/CSS within the file itself — there are no external images, so it works offline and prints cleanly.

END OF UNIT 4

Questions?

Match the component to the task, the data and the user.