CREATIVE & DESIGN TOOLS

Vector Graphics

Anchor Points, Paths & the Pen Tool

Pathfinder, Layers & Typography

Illustrator & Inkscape, Side by Side

Capstone: Designing a Complete Logo

Philip Osztromok

Generated with Claude

Table of Contents

Vector Graphics — 10 Chapters

Raster Editing Fundamentals opened with pixels, resolution, and color modes — a fixed grid of colored dots that a photo or scanned image is fundamentally made of. This course starts from a genuinely different foundation. A vector image isn't a grid of dots at all; it's a set of mathematical instructions describing shapes, and that single difference changes almost everything about how these tools work.

What "Vector" Actually Means: Math, Not Pixels

A raster image stores a color value for every single pixel in its grid. A vector image instead stores a mathematical description of each shape — where a curve starts, where it ends, and how it bends in between — and the software recalculates exactly how that shape should look every time it's displayed or printed. There's no pixel grid sitting underneath a vector shape at all; it's recomputed fresh at whatever size and resolution is actually needed.

Resolution Independence: Why Scaling Never Degrades

Raster Editing Fundamentals' own Chapter 2 explained that enlarging a raster image stretches its existing, fixed pixels, producing visible blurriness or blockiness past a certain point. A vector shape has no fixed pixels to stretch — enlarging it just means recalculating the same mathematical description at a larger size, producing an identically crisp result whether it's rendered at business-card size or printed on a billboard. This property, called resolution independence, is the single biggest practical reason vector tools exist at all.

Anchor Points and Paths: The Building Blocks

Every vector shape is built from anchor points connected by paths — straight lines or curved segments defined by mathematical handles. A simple square might use four anchor points connected by straight paths; a smooth, flowing curve uses anchor points with curve handles that control exactly how the path bends between them. Chapter 3 goes hands-on with actually placing and adjusting these, but recognizing that every vector shape, however complex, ultimately reduces to anchor points and paths is the mental model this entire course builds on.

When Raster Still Wins

Vector graphics aren't a universal upgrade over raster — they solve a different problem. A photograph, with its continuous, irregular variation in color and tone, has no clean mathematical shape to describe; it genuinely needs a pixel grid to represent accurately, which is exactly why Raster Editing Fundamentals' own tools exist. Vector graphics excel specifically at clean, geometric, or illustrative work — logos, icons, typography, technical diagrams — where shapes have real, describable edges rather than photographic complexity.

What This Course Actually Teaches

This course covers Adobe Illustrator and Inkscape (free and open-source) side by side, the same comparative approach Raster Editing Fundamentals used for Photoshop and GIMP — two tools built around the same underlying discipline. Chapter 2 picks that framing up directly before this course moves into the actual drawing tools.

ASPECT	RASTER (PHOTOSHOP/GIMP)	VECTOR (ILLUSTRATOR/INKSCAPE)
Stored as	A fixed grid of pixel colors	Mathematical descriptions of shapes
Scaling up	Degrades — blur or blockiness	No degradation — recalculated fresh
Best suited to	Photographs, continuous tone	Logos, icons, typography, diagrams
Building block	The pixel	The anchor point and path

SAME DISCIPLINE, DIFFERENT FOUNDATION

Chapter 2 reapplies Raster Editing Fundamentals' own opening framing — two tools sharing a discipline, not a platform — to this course's own pairing of Illustrator and Inkscape, before moving into the Pen tool and shape tools that actually put anchor points and paths to work.

VECTOR TOOLS CAN'T MAGICALLY TURN A PHOTO INTO CLEAN SHAPES

It's tempting to assume a vector program can take any raster photo and produce a clean, scalable vector version automatically. Auto-tracing tools exist and can approximate simple, high-contrast images reasonably well, but a genuine photograph's continuous tonal detail has no clean mathematical shape to extract — the result is either a rough approximation or an enormous number of tiny paths trying to fake photographic detail, neither of which behaves like a real vector illustration.

Hands-On Exercises

EXERCISE 1

A colleague enlarges a company logo saved as a JPEG for a large banner print and notices it looks blurry and blocky. Using this chapter's own material, explain what's happening and what kind of file would have avoided the problem.

View solution

EXERCISE 2

A friend asks why designers don't just use vector tools for everything, since vector graphics never lose quality when scaled. Using this chapter's own material, explain why raster tools are still genuinely necessary.

View solution

EXERCISE 3

A beginner tries to auto-trace a detailed vacation photo into a vector file, expecting a clean, scalable illustration, and is disappointed by the result. Using this chapter's own warning box, explain why this didn't work the way they expected.

View solution

CHAPTER 1 QUICK REFERENCE

- Vector graphics store **mathematical descriptions of shapes**, not a grid of pixel colors
- **Resolution independence** means a vector shape never degrades when scaled — it's recalculated fresh each time
- Every vector shape reduces to **anchor points and paths** — the building blocks this entire course works with
- Vector graphics suit **clean, geometric, or illustrative** work; photographs still genuinely need raster tools
- This course compares **Illustrator and Inkscape**, the same comparative approach Raster Editing Fundamentals used for Photoshop and GIMP

Chapter 1 established the paradigm this whole course works from — paths and anchor points instead of pixels. This chapter introduces the two tools that put that paradigm into practice, and reapplies a framing Raster Editing Fundamentals' own Chapter 1 already used: two tools sharing a discipline, not a platform.

Two Tools, One Discipline

Raster Editing Fundamentals opened by treating Photoshop and GIMP as two applications built around the same underlying discipline — raster image editing — rather than two versions of the same platform. Illustrator and Inkscape deserve exactly the same framing here: both are vector drawing applications built around anchor points, paths, and resolution-independent shapes, even though they come from very different origins and have real, genuine differences worth naming honestly rather than smoothing over.

Adobe Illustrator: The Industry Standard

Illustrator is the dominant professional tool for vector work — the default expectation in most design studios, agencies, and print production pipelines. It's subscription-only, part of Adobe's Creative Cloud, and deeply integrated with other Adobe tools covered elsewhere on this site (Photoshop and, by extension, the raster workflow Raster Editing Fundamentals already taught).

Inkscape: Free, Open-Source, Genuinely Capable

Inkscape is free, open-source, and genuinely capable of professional-quality vector work — logos, icons, illustrations, and technical diagrams all fall well within its real capability, not just simple hobbyist projects. It's built around SVG (Scalable Vector Graphics) as its native format, an open, web-standard format rather than a proprietary one.

Where They Genuinely Differ

Cost is the most obvious difference, but not the only one worth naming. Illustrator's native format is proprietary (.ai), while Inkscape's is the open .svg standard — a real, practical difference when a file needs to move between tools or onto the web. Illustrator also has deeper integration with the broader Adobe ecosystem and a larger commercial plugin marketplace, while Inkscape's own extension ecosystem is smaller but entirely free. Neither of these differences makes one tool strictly better than the other — they're trade-offs that matter differently depending on the actual job.

What This Course Actually Teaches

From here forward, this course teaches the same core skills — the Pen tool, shapes, fill and stroke, Pathfinder operations, layers, text, and export — in both tools side by side, exactly like Raster Editing Fundamentals did for Photoshop and GIMP. Differences get called out honestly wherever they're real, rather than treating one tool as simply a lesser version of the other.

ASPECT	ADOBE ILLUSTRATOR	INKSCAPE
Cost	Subscription-only (Creative Cloud)	Free, open-source
Native format	.ai (proprietary)	.svg (open, web-standard)
Ecosystem	Deep Adobe integration, large commercial plugin market	Smaller, entirely free extension ecosystem
Industry use	Dominant professional standard	Widely used by independent designers, web/open-source communities

SAME SHARED DISCIPLINE AS RASTER EDITING FUNDAMENTALS

Chapter 3 starts working hands-on with the Pen tool in both applications — the same "shared discipline, different tools" approach used throughout Raster Editing Fundamentals continues from here for the rest of this course.

"FREE" DOESN'T MEAN "LESS CAPABLE" HERE EITHER

It's tempting to assume Inkscape must be a stripped-down alternative simply because it costs nothing, the same assumption Video Editing Fundamentals' own Chapter 1 warned against for DaVinci Resolve. Inkscape's core vector toolset — paths, Pathfinder-equivalent boolean operations, layers, text — is genuinely professional-capable. The real differences lie in ecosystem size, file format, and Adobe-specific integrations, not in Inkscape being fundamentally unable to do the actual work.

Hands-On Exercises

EXERCISE 1

A student who took Raster Editing Fundamentals asks whether Illustrator and Inkscape's relationship is basically the same as Photoshop and GIMP's. Using this chapter's own material, explain what's genuinely the same and what's genuinely different.

View solution

EXERCISE 2

A colleague needs to send a logo file to a web developer who only needs an SVG, and separately to a print shop that expects a native Illustrator file. Using this chapter's own material, explain which tool's native format serves which need.

View solution

EXERCISE 3

A beginner assumes Inkscape must be missing important core features since it costs nothing. Using this chapter's own warning box, explain why this assumption doesn't hold up.

View solution

CHAPTER 2 QUICK REFERENCE

- Illustrator and Inkscape share the same **vector discipline** — anchor points, paths, resolution independence — not a shared platform
- **Illustrator** is the subscription-only industry standard with deep Adobe integration
- **Inkscape** is free, open-source, and genuinely professional-capable, built around the open SVG standard
- Real differences are **cost, native file format, and ecosystem size** — not core capability
- This course teaches the same skills in **both tools side by side**, honestly naming differences where they're real

Chapter 1 introduced anchor points and paths as vector graphics' own building blocks; Chapter 2 introduced the two tools that work with them. This chapter puts both of those ideas into practice — actually drawing custom paths with the Pen tool, the single most important instrument in either application.

The Pen Tool: Vector Drawing's Core Instrument

The Pen tool (identically named in Illustrator, and called the Bezier tool in Inkscape's own toolbox, though functionally the same instrument) places anchor points one click at a time, connecting each new point to the last with a path segment. Every custom shape that isn't a simple rectangle or ellipse ultimately gets drawn this way — it's the tool that turns Chapter 1's abstract "paths and anchor points" description into an actual drawing skill.

Placing Anchor Points: Corner Points vs. Smooth Points

A simple click places a corner point — the path meets at a sharp angle on either side, useful for straight edges and sharp corners. Clicking and dragging instead places a smooth point, pulling out a curve handle on both sides of the point that controls how the path flows through it, useful for rounded, flowing shapes. Choosing the right point type at each click is the difference between a shape that looks deliberately drawn and one that looks like it fought the tool the whole way.

Bezier Handles: Controlling Curve Shape

Each smooth point's curve handles can be dragged independently to reshape the curve on either side of that point — longer handles produce a broader, more gradual curve, shorter handles produce a tighter one, and the handle's angle controls the curve's direction as it enters and leaves the point. A small number of well-placed smooth points with carefully adjusted handles almost always produces a cleaner curve than many corner points trying to approximate the same shape.

Open vs. Closed Paths

A path is open if it has a distinct start and end point that never connect — useful for a line, an underline, or a decorative squiggle. A path is closed if its last anchor point connects back to its first, forming a continuous loop — necessary for any shape that needs a fillable interior, like a logo mark or an icon. Clicking back on a path's own starting point while drawing is what closes it in both applications.

Editing Existing Paths: Adding, Removing, and Adjusting Points

A path is rarely perfect on the first attempt. Both applications let existing anchor points be selected and repositioned, new points added along an existing segment, and unwanted points deleted — all without needing to redraw the whole path from scratch. Learning to edit an existing path confidently is just as important as drawing one in the first place, since real vector work is mostly refinement, not one perfect unbroken pass.

ACTION	ILLUSTRATOR	INKSCAPE
Tool name	Pen tool	Bezier tool (Pen tool)
Place a corner point	Click	Click
Place a smooth point	Click and drag	Click and drag
Close a path	Click back on the start point	Click back on the start point

A FASTER ALTERNATIVE IS COMING IN CHAPTER 4

Regular geometric shapes — rectangles, circles, polygons — don't need to be drawn point by point with the Pen tool at all. Chapter 4 introduces the dedicated shape tools that build these directly, reserving the Pen tool for genuinely custom, irregular shapes.

MORE ANCHOR POINTS ISN'T THE SAME AS A SMOOTHER CURVE

It's tempting, especially early on, to place many closely-spaced corner points to approximate a curve, clicking repeatedly along where a smooth arc should be. This usually produces a rough, faceted result rather than a smooth curve, and it makes the path much harder to edit later. A single smooth point with well-adjusted handles almost always produces a cleaner, more professional-looking curve than a dozen corner points trying to fake the same shape.

Hands-On Exercises

EXERCISE 1

A beginner clicks (without dragging) at several points around a curve they want to draw, and the result looks angular and rough instead of smooth. Using this chapter's own material, explain what they should have done differently.

View solution

EXERCISE 2

Someone draws a shape they intend to fill with color, but when they apply a fill, nothing appears to happen. Using this chapter's own material on open vs. closed paths, explain the likely cause.

View solution

EXERCISE 3

Explain, using this chapter's own material, why a beginner drawing a simple rectangle by hand with the Pen tool is taking the harder path compared to what's coming in Chapter 4.

View solution

CHAPTER 3 QUICK REFERENCE

- The **Pen tool** (Bezier tool in Inkscape) places anchor points and connects them with path segments
- **Click** places a corner point; **click and drag** places a smooth point with curve handles
- **Bezier handles** control a curve's breadth and direction on either side of a smooth point
- A path is **closed** (connects back to its start) to hold a fill, or **open** for a line with no interior
- Fewer, well-placed **smooth points** produce cleaner curves than many corner points trying to fake one

Chapter 3 closed with a tip pointing here directly: regular geometric shapes don't need to be drawn point by point with the Pen tool. This chapter introduces the dedicated shape tools built specifically for rectangles, ellipses, and polygons — faster, more precise, and far less error-prone than reaching for the Pen tool every time.

The Rectangle Tool

Clicking and dragging with the Rectangle tool draws a rectangle directly, with perfectly straight sides and precise right angles guaranteed by the tool itself, not by careful manual point placement. Both applications also support rounded corners on a rectangle — either dragged live from a small corner handle or set numerically — without needing to hand-edit anchor points into curves the way Chapter 3's Pen tool work would require.

The Ellipse Tool

The Ellipse tool works the same way for circles and ellipses — click and drag for a free-form ellipse, or hold a modifier key (Shift, in both applications) while dragging to constrain it to a perfect circle. A perfect circle drawn by hand with the Pen tool would require four smooth points with carefully matched handle lengths; the Ellipse tool produces the same mathematically exact result in one drag.

The Polygon Tool (and Star Tool)

The Polygon tool draws regular polygons — triangles, pentagons, hexagons, and beyond — with a numeric setting controlling exactly how many sides it has. A companion Star tool (present in both applications) works the same way for star shapes, with separate settings for the number of points and how deep the points cut inward. Both remove any need to manually calculate and place the points of a perfectly regular polygon or star by hand.

Precise Numeric Input: Dialog Boxes for Exact Dimensions

Rather than dragging by eye, both applications let a shape tool be used with a single click instead of a drag, opening a dialog box for exact numeric width, height, corner radius, or number of sides. This matters whenever a shape needs to be an exact size — a business card at precisely 3.5 by 2 inches, an icon at exactly 64 by 64 pixels — rather than whatever size a mouse drag happened to land on.

Shapes Are Still Paths: Converting to Editable Anchor Points

A shape drawn with these tools is still fundamentally made of anchor points and paths underneath, exactly as Chapter 1 described — the shape tools are simply a faster, more precise way of generating that same underlying structure. Both applications let a shape be converted into a fully editable path afterward, exposing its individual anchor points for the same kind of hand-editing Chapter 3 covered, useful once a rectangle or polygon needs to become something less regular than its starting form.

SHAPE	ILLUSTRATOR TOOL	INKSCAPE TOOL
Rectangle (with rounded corners)	Rectangle tool	Rectangle tool
Circle/ellipse (Shift = perfect circle)	Ellipse tool	Ellipse/Circle/Arc tool
Regular polygon	Polygon tool	Star tool (polygon mode)
Star	Star tool	Star tool (star mode)

THESE BASIC SHAPES FEED DIRECTLY INTO CHAPTER 6

Pathfinder's own boolean operations — union, subtract, intersect, exclude — work by combining exactly these kinds of simple shapes into more complex ones, rather than requiring every complex shape to be hand-drawn with the Pen tool from the start.

CONVERTING A SHAPE TO A PATH TRADES EDITABILITY FOR FLEXIBILITY

It's tempting to convert every shape to a fully editable path immediately, assuming more editing options are always better. Once converted, a rectangle loses its "live" rounded-corner handle and precise numeric width/height fields — it becomes a set of plain anchor points like any Pen-tool path, with no memory of having started as a rectangle at all. Keeping a shape in its original, editable "live" form for as long as possible preserves the ability to adjust its exact dimensions or corner radius numerically later.

Hands-On Exercises

EXERCISE 1

A designer needs an icon that's exactly 64 by 64 pixels with a precise 8-pixel corner radius. Using this chapter's own material, explain the fastest, most precise way to create this shape.

View solution

EXERCISE 2

A colleague converts a rounded rectangle to a fully editable path right after drawing it, then later needs to change the corner radius and finds there's no longer a simple way to do it. Using this chapter's own warning box, explain what happened.

View solution

EXERCISE 3

A beginner insists on drawing a perfect circle with the Pen tool from Chapter 3 rather than using the Ellipse tool. Using this chapter's own material, explain why this is unnecessarily difficult.

View solution

CHAPTER 4 QUICK REFERENCE

- The **Rectangle, Ellipse, and Polygon/Star tools** draw regular geometry directly, without manual anchor-point placement
- **Shift** while dragging constrains an ellipse to a perfect circle in both applications
- A single click (instead of a drag) opens a dialog for **exact numeric dimensions**
- Shapes are still made of **anchor points and paths** underneath — the tools just generate that structure faster
- **Converting a shape to a path** trades away its live, numerically-editable properties — do it only once the shape needs to become irregular

Chapter 5 · Fill, Stroke, Gradients & the Appearance/Style Panels

Chapters 3 and 4 covered drawing shapes — custom paths and regular geometry alike. Every one of those shapes has so far been an outline with no real appearance yet. This chapter gives them one: fill, stroke, and gradients, controlled through a panel that goes by a different name in each application but does the same underlying job.

Fill and Stroke: The Two Basic Appearance Properties

Fill is the color (or gradient, or pattern) applied to a shape's interior — which is exactly why Chapter 3 emphasized closed paths, since an open path has no bounded interior for a fill to occupy. Stroke is the color and width applied along the shape's own outline path itself. A single shape can have a fill, a stroke, both, or neither, and each is set independently.

Solid Colors, Color Models & Swatches

Both applications support the same basic color models — RGB for screen work, CMYK for print work — plus a swatches panel for saving and reusing specific colors consistently across a project, exactly the same practical need Raster Editing Fundamentals' own color-mode material addressed for raster work. Consistency matters even more in vector work, where the same logo or icon often needs to reuse an exact brand color across many separate shapes.

Stroke Weight, Cap, and Join

Stroke weight sets how thick the outline is. Cap controls how a stroke's open ends are finished — butt (a flat cut-off), round, or square (extended slightly past the endpoint). Join controls how a stroke behaves at a sharp corner — miter (a sharp point), round, or bevel (a flattened corner). These options matter more than they might first seem: a thick stroke with the wrong join can look noticeably different at a shape's corners than the same stroke with a different join setting.

Gradients: Fills That Transition Between Colors

A gradient fill transitions smoothly between two or more colors across a shape, rather than filling with one flat color. Linear gradients transition along a straight line; radial gradients transition outward from a center point. Both applications let a gradient's individual color stops be repositioned and recolored, and let the gradient's own angle or center point be adjusted directly on the shape rather than only through a dialog box.

Illustrator's Appearance Panel vs. Inkscape's Fill and Stroke Dialog

Illustrator manages all of this through its Appearance panel, which does something genuinely distinctive: it can stack multiple fills and multiple strokes on a single object, each one layered on top of the last, with the stacking order directly affecting the final look — the same "ordered stack, order matters" idea this site's own Audio & Video Production courses kept encountering in different forms (layers, Sources, timeline tracks), showing up here again inside a single object's own appearance. Inkscape's Fill and Stroke dialog covers the same fill/stroke/gradient properties but as a single fill and single stroke per object, without that same multi-layered stacking model.

ASPECT	ILLUSTRATOR (APPEARANCE PANEL)	INKSCAPE (FILL AND STROKE DIALOG)
Fills per object	Multiple, stacked	One
Strokes per object	Multiple, stacked	One
Stacking order matters?	Yes — directly affects final look	Not applicable (single fill/stroke)
Gradient editing	Live, on-object gradient handles	Live, on-object gradient handles

ORDER MATTERS AGAIN — THIS TIME INSIDE ONE OBJECT

Illustrator's Appearance panel stacking a shape's own fills and strokes is the same underlying idea this site's own Audio & Video Production courses ran into repeatedly at a larger scale — order matters whenever multiple things are layered together, whether that's Photoshop layers, OBS Sources, video timeline tracks, or now the fills and strokes stacked on a single vector object.

SCALING A SHAPE CAN CHANGE ITS STROKE WEIGHT UNEXPECTEDLY

It's tempting to resize a shape by dragging its corner handle and assume the stroke simply scales proportionally along with it. Depending on a setting (often labeled something like "scale strokes and effects"), resizing a shape can either scale its stroke weight along with the shape or leave the stroke weight fixed at its original thickness — producing a visibly thicker or thinner-looking stroke relative to the shape than intended if that setting isn't what was expected.

Hands-On Exercises

EXERCISE 1

A designer wants a logo mark with a bold outer stroke, a secondary thinner stroke just inside it, and a solid fill, all on the exact same shape. Using this chapter's own material, explain which application's own panel makes this straightforward, and why.

View solution

EXERCISE 2

A colleague resizes a shape smaller and is confused that its stroke now looks proportionally much thicker than before. Using this chapter's own warning box, explain what's likely going on.

View solution

EXERCISE 3

Explain, using this chapter's own material, how the stacking order inside Illustrator's Appearance panel relates to the "order matters" idea from this site's own Audio & Video Production courses.

View solution

CHAPTER 5 QUICK REFERENCE

- **Fill** colors a shape's interior; **stroke** colors and weights its outline — set independently
- **Stroke cap and join** control how a stroke's ends and corners are finished
- **Gradients** transition between colors, linear or radial, with live on-object handles in both applications
- Illustrator's **Appearance panel** stacks multiple fills/strokes on one object, order-dependent; Inkscape's Fill and Stroke dialog handles one of each
- Check the **"scale strokes and effects"** setting before resizing, or stroke weight may not scale as expected

Chapter 4's own tip box pointed here directly: the simple rectangles, circles, and polygons from that chapter become genuinely useful once they're combined. Pathfinder's boolean operations build complex, custom-looking shapes out of these simple building blocks, rather than requiring every complex shape to be hand-drawn with the Pen tool from scratch.

What Pathfinder Actually Does: Combining Simple Shapes Into Complex Ones

Boolean operations take two or more overlapping shapes and combine them according to a specific rule — merge them, cut one out of the other, keep only where they overlap, or keep only where they don't. A rounded rectangle with a circular notch cut into one edge, a crescent moon shape, a plus sign made from two overlapping rectangles — all of these are far easier to build from simple overlapping shapes than to draw directly with the Pen tool.

Union: Merging Shapes Into One

Union merges two or more overlapping shapes into a single new shape, combining their outlines into one continuous path and discarding the internal overlap. Two overlapping circles become one rounded, peanut-like shape; a rectangle overlapping a triangle becomes a single house-shaped outline.

Subtract (Minus Front): Cutting One Shape Out of Another

Subtract (often labeled "Minus Front" in Illustrator, since it removes whichever shape is on top from whichever is beneath) cuts the overlapping area of the front shape out of the back shape, like a cookie cutter. A circle placed over the corner of a square and subtracted produces a square with a rounded notch cut out of that corner.

Intersect: Keeping Only the Overlap

Intersect keeps only the area where all the selected shapes overlap, discarding everything else. Two overlapping circles intersected produce only the lens-shaped sliver where both circles cover the same space — a classic technique for a Venn-diagram-style shape or a stylized eye.

Exclude: Keeping Only the Non-Overlap

Exclude does the opposite of Intersect — it keeps everything except the overlapping area, effectively punching a hole wherever the shapes cover each other. Two overlapping circles excluded produce two separate crescent-like shapes where the overlap has been removed from each.

Illustrator's Pathfinder Panel vs. Inkscape's Path Menu

Illustrator groups these operations in its dedicated Pathfinder panel, with each operation represented by an icon button. Inkscape offers the same four core operations through its Path menu's own Boolean Operations submenu, under slightly different names — Union, Difference, Intersection, and Exclusion. The underlying logic is identical in both; only the panel location and exact wording differ.

OPERATION	ILLUSTRATOR (PATHFINDER PANEL)	INKSCAPE (PATH MENU)
Merge shapes into one	Unite	Union
Cut front shape out of back	Minus Front	Difference
Keep only the overlap	Intersect	Intersection
Keep only the non-overlap	Exclude	Exclusion

COMBINED SHAPES STILL NEED ORGANIZING

A logo built from several Pathfinder operations often ends up as multiple resulting shapes that need to stay together as the design is refined further. Chapter 7's own layers and grouping material is exactly what keeps a multi-shape composition like this organized as it grows more complex.

MOST PATHFINDER OPERATIONS ARE DESTRUCTIVE — THE ORIGINALS ARE GONE

It's tempting to apply Union, Subtract, Intersect, or Exclude freely, assuming the original separate shapes can always be recovered afterward. In both applications, these standard Pathfinder operations replace the original shapes with a single new resulting path — the original separate shapes are gone once the operation is applied, similar to Chapter 4's own warning about converting a live shape into a plain path. Keeping a duplicate of the original shapes before applying a boolean operation is the safest way to preserve the ability to start over if the result isn't right.

Hands-On Exercises

EXERCISE 1

A designer wants to create a rounded rectangle with a circular notch cut out of one corner, using only the basic shapes from Chapter 4. Using this chapter's own material, explain which Pathfinder operation achieves this and how.

View solution

EXERCISE 2

After applying Union to combine two circles, a colleague wants to go back and adjust just one of the original circles, but finds they're no longer separately selectable. Using this chapter's own warning box, explain what happened and how to avoid this in the future.

View solution

EXERCISE 3

Explain, using this chapter's own material, the difference between Intersect and Exclude when applied to the same two overlapping circles.

View solution

CHAPTER 6 QUICK REFERENCE

- **Union** merges overlapping shapes into one continuous outline
- **Subtract/Difference** cuts the front shape's overlap out of the back shape
- **Intersect/Intersection** keeps only the overlapping area; **Exclude/Exclusion** keeps only the non-overlapping area
- Illustrator's **Pathfinder panel** and Inkscape's **Path menu** offer the same four core operations under slightly different names
- These operations are usually **destructive** — duplicate the original shapes first if they might be needed again

Chapter 6 closed with a tip pointing here directly: a logo built from several Pathfinder operations often ends up as multiple resulting shapes that need to stay organized as a design grows more complex. This chapter covers exactly that — grouping, layers, and stacking order, the tools that keep a multi-shape composition manageable.

Grouping: Treating Multiple Objects as One

Grouping bundles several separate objects so they can be selected, moved, scaled, or rotated together as a single unit, without merging their underlying paths the way Chapter 6's Pathfinder operations do. A grouped set of shapes stays fully editable individually after ungrouping — grouping is purely an organizational convenience, not a destructive combination.

Layers: Organizing a Composition Into Independent Levels

A layer holds a collection of objects on its own independent level within the overall composition, which can be shown, hidden, locked, or reordered as a whole relative to other layers. A logo project might use one layer for background shapes, one for the main mark, and one for text — each editable in isolation without accidentally disturbing the others.

Object Stacking Order Within a Layer

Within a single layer, individual objects also have their own front-to-back stacking order, exactly like the layer order in Raster Editing Fundamentals or the fill/stroke stacking inside Chapter 5's own Appearance panel — whichever object is listed higher renders on top of whichever is listed lower. Both applications let an object be moved forward, backward, to the very front, or to the very back within its own layer.

Illustrator's Layers Panel vs. Inkscape's Objects Panel

Illustrator's Layers panel shows every layer, with each layer's own objects nested inside it and independently expandable. Inkscape's Objects panel covers the same ground — layers and the objects within them, shown/hidden and locked/unlocked from the same panel — under a different name reflecting Inkscape's own slightly different internal terminology, though the practical organizing behavior is the same in both.

Locking and Hiding: Working Safely in a Complex File

Locking a layer or object prevents it from being accidentally selected or moved while working on something else nearby — genuinely useful once a background layer or a finished element needs to stay untouched. Hiding a layer or object removes it from view (and from accidental selection) entirely, without deleting it, useful for temporarily decluttering a busy composition while working on one specific part of it.

ASPECT	ILLUSTRATOR	INKSCAPE
Panel name	Layers panel	Objects panel
Grouping multiple objects	Group (non-destructive)	Group (non-destructive)
Reordering within a layer	Bring Forward/Backward/to Front/to Back	Raise/Lower/Raise to Top/Lower to Bottom
Lock/hide a layer	Toggle icons in Layers panel	Toggle icons in Objects panel

THE SAME "ORDER MATTERS" IDEA, AT YET ANOTHER SCALE

Object stacking order within a layer is the same underlying principle Chapter 5 encountered inside a single object's own fills and strokes, and the same principle Raster Editing Fundamentals encountered across whole layers — order determines the final visible result whenever multiple things share the same space, whether that's an entire layer, a single object, or a fill and stroke on one shape.

GROUPING IS NOT THE SAME AS LAYERING

It's tempting to use groups as a substitute for layers, or vice versa, since both involve bundling objects together. Grouping keeps objects together for selection and transformation but doesn't provide layers' own independent show/hide/lock control or a dedicated level in the overall stacking order. Using layers for genuinely separate parts of a composition (background, main artwork, text) and groups only for objects that should move and scale together keeps a file organized in a way that actually matches what each tool is for.

Hands-On Exercises

EXERCISE 1

A designer wants to move a logo's icon and its wordmark together as one unit while designing, but still needs each piece to stay fully editable afterward. Using this chapter's own material, explain the right approach.

View solution

EXERCISE 2

A colleague puts their background shapes and their main artwork all into one group instead of using separate layers, then struggles to hide just the background while working on the artwork. Using this chapter's own warning box, explain what went wrong.

View solution

EXERCISE 3

Explain, using this chapter's own material, how object stacking order within a layer relates to the fill/stroke stacking from Chapter 5's own Appearance panel.

View solution

CHAPTER 7 QUICK REFERENCE

- **Grouping** bundles objects for selection and transformation without merging their underlying paths
- **Layers** hold independent levels of a composition, each show/hide/lock-able as a whole
- Within a layer, objects have their own **front-to-back stacking order** — the same "order matters" idea from Chapter 5 and Raster Editing Fundamentals
- Illustrator's **Layers panel** and Inkscape's **Objects panel** cover the same organizing ground under different names
- Use **layers** for genuinely separate composition parts, **groups** only for objects that should move together

Chapter 7 organized shapes into layers and groups. This chapter introduces a genuinely distinct kind of object into that same composition — text — which behaves like a vector shape in some ways and like nothing else covered so far in others.

Text as a Special Kind of Vector Object

Text placed with the Type tool looks like any other vector content once rendered — made of the same curves and fills covered since Chapter 1 — but until it's specifically converted, it isn't stored as anchor points and paths at all. It's stored as actual characters, referencing a font file that supplies the actual shape of each letter on demand.

Live Text: Editable, Font-Dependent

While text stays "live," its actual words can be edited directly — fixing a typo, changing a word, adjusting the whole sentence — the same way editing text in any word processor works. This convenience comes with a real dependency: live text only displays correctly if the font it references is actually installed and available. Open the same file on a machine without that font installed, and the text either substitutes a different font or displays a warning, changing how the design actually looks.

Basic Typography Controls: Font, Size, Tracking, Leading

Font selects which typeface is used; size sets how large the characters render. Tracking adjusts the uniform spacing between all characters in a selection — tightened for a denser wordmark, loosened for a more spaced-out, airy feel. Leading (line height) sets the vertical spacing between lines of text in a multi-line block. These four controls cover most of the everyday typography adjustment either application actually needs.

Text on a Path

Both applications support attaching text to a custom path drawn with Chapter 3's own Pen tool, making the text follow that path's curve rather than sitting on a straight baseline — a badge-style logo with text curving around a circular border is a common real use of this feature.

Outlining Text: Converting Live Text to Editable Paths

Both applications offer a command (Create Outlines in Illustrator, Object to Path in Inkscape) that converts live text into ordinary vector paths — anchor points and curves, just like the shapes from Chapters 3 and 4. Once outlined, the actual words can no longer be edited by typing, but the font dependency is gone entirely: the shapes render identically on any machine, regardless of what fonts are or aren't installed there.

When to Outline: The Font-Availability Problem

A logo is a perfect case for outlining, since it's meant to look identical everywhere it's used — on a website, in a printed document, on someone else's machine that may not have the original font installed. Outlining trades away future text-editing convenience for guaranteed visual consistency, which is exactly the trade a finished logo needs, though it's not the right move for content still being actively drafted and revised.

ASPECT	ILLUSTRATOR	INKSCAPE
Outline command	Create Outlines	Object to Path
Result	Ordinary editable vector paths	Ordinary editable vector paths
Text editability after	Lost — no longer typeable	Lost — no longer typeable
Font dependency after	Removed entirely	Removed entirely

THIS DECISION MATTERS AGAIN AT EXPORT TIME

Chapter 9's own export material assumes a finished logo has already had its text outlined before export, exactly for the font-independence reason described here — an exported file with live, un-outlined text can still display incorrectly on a system without the original font.

OUTLINING IS THE THIRD ONE-WAY TRADE THIS COURSE HAS MADE

Chapter 4 warned that converting a live shape to a path trades away its numeric editability; Chapter 6 warned that most Pathfinder operations replace the original shapes permanently. Outlining text is the same kind of trade a third time — once outlined, the words themselves can no longer be edited by typing, only reshaped point by point like any other path. Keeping an un-outlined copy of the original text layer aside (the same precaution recommended in Chapters 4 and 6) preserves the ability to fix a typo or change the wording later without starting the whole design over.

Hands-On Exercises

EXERCISE 1

A designer sends a finished logo file with live, un-outlined text to a client, who opens it on a machine without the original font installed and sees the design rendered incorrectly. Using this chapter's own material, explain what should have been done before sending the file.

View solution

EXERCISE 2

A colleague outlines their logo's text early in the design process, then later needs to fix a spelling mistake and finds they can no longer simply type the correction. Using this chapter's own warning box, explain what happened and how it could have been avoided.

View solution

EXERCISE 3

Explain, using this chapter's own material, why outlining text is described as the same kind of trade-off as converting a shape to a path (Chapter 4) or applying a Pathfinder operation (Chapter 6).

View solution

CHAPTER 8 QUICK REFERENCE

- **Live text** stays editable by typing but depends on the referenced font being installed to display correctly
- **Font, size, tracking, and leading** cover most everyday typography adjustments
- **Text on a path** makes text follow a custom curve rather than sitting on a straight baseline
- **Outlining** (Create Outlines / Object to Path) converts live text into ordinary editable paths, removing font dependency
- Outlining trades away typed editability permanently — the **third one-way trade** this course has made, after Chapters 4 and 6

Chapter 8 closed with text ready to be outlined for export. This chapter covers the two remaining pieces before a design actually leaves either application: artboards, which organize multiple compositions in one file, and export itself, where Chapter 1's own resolution independence finally pays off directly.

Artboards: Multiple Compositions in One File

An artboard is a defined canvas area within a single file — a logo project might use separate artboards for a full-color version, a single-color version, and a horizontal lockup, all in one file, each exportable independently. This avoids juggling several separate files for what's really one connected set of logo variations.

Illustrator's Artboards Panel vs. Inkscape's Multi-Page Support

Illustrator's Artboards panel manages any number of artboards within a single document natively, a long-standing core feature. Inkscape's own multi-page support is a more recently added feature, covering the same basic need — multiple compositions in one file — through its own Document Properties and Pages panel, with a smaller feature set than Illustrator's more mature artboard tooling.

Exporting for the Web: SVG

SVG is the native web format for vector graphics — resolution-independent, readable directly by web browsers, and small in file size compared to an equivalent raster image. Per Chapter 1's own resolution-independence material, an SVG logo displays crisply on any screen size or pixel density without needing separate image files for different resolutions, unlike the raster alternatives Raster Editing Fundamentals' own formats would require.

Exporting for Print: PDF and EPS

PDF is the most broadly compatible print-ready vector format, readable by virtually every design and print tool, and capable of preserving multiple pages or artboards in one file. EPS is an older vector format still requested by some print shops and legacy production workflows, though PDF has largely superseded it for new work. Both preserve a design's vector nature completely, so a printed piece scales to any size without the quality loss a raster file would suffer.

Raster Export When You Actually Need Pixels

Sometimes a raster file is genuinely what's needed — a PNG for a website favicon, a JPG thumbnail for a social media profile. Both applications can export a raster image at any chosen resolution directly from the same vector source, and because the underlying artwork has no fixed pixel grid until that export moment, it can be rendered cleanly at whatever size is actually needed — a small icon or a huge print-resolution image, from the exact same file.

Bringing It All Together: Resolution Independence at Export Time

This is the concrete, practical payoff of Chapter 1's opening idea. A single vector file, maintained once, can produce a crisp web SVG, a print-ready PDF, and raster exports at any resolution — all from the same underlying paths, with no separate "high-res" and "low-res" versions ever needing to be maintained separately the way raster work often requires.

FORMAT	BEST FOR	STAYS VECTOR?
SVG	Web use — browsers, responsive design	Yes
PDF	Print, broad compatibility, multi-page/artboard	Yes
EPS	Legacy print workflows	Yes
PNG/JPG	Web images needing an actual pixel grid (favicons, thumbnails)	No — rasterized at export

THE CAPSTONE PUTS EVERY FORMAT HERE TO USE

Chapter 10's capstone logo gets exported in several of these formats side by side — an SVG for the web, a PDF for print, and a PNG at a specific resolution — closing the loop from Chapter 1's own resolution-independence promise all the way through to real, usable deliverables.

EXPORT DOESN'T RETROACTIVELY OUTLINE FORGOTTEN LIVE TEXT

It's tempting to assume exporting to SVG or PDF automatically handles font dependency the way outlining does. If Chapter 8's own outlining step was skipped, live text in the export may still depend on the original font being installed wherever the file is opened or rendered — the export format alone doesn't solve the font-availability problem outlining was specifically meant to solve. Outlining text before export, not instead of it, is what actually guarantees the design looks identical everywhere.

Hands-On Exercises

EXERCISE 1

A designer needs a logo to display crisply on both a small mobile screen and a large desktop monitor, without maintaining separate image files for each. Using this chapter's own material, explain which export format solves this and why.

View solution

EXERCISE 2

A colleague exports their logo to PDF without outlining the live text first, and the client later reports the text looks wrong on their own machine. Using this chapter's own warning box, explain what went wrong.

View solution

EXERCISE 3

Explain, using this chapter's own material, why a single vector file can produce both a crisp web SVG and a high-resolution print-ready PDF without maintaining two separate source files.

View solution

CHAPTER 9 QUICK REFERENCE

- **Artboards** hold multiple related compositions in one file, each independently exportable
- **SVG** is the resolution-independent, browser-native web format
- **PDF and EPS** preserve vector data for print, with PDF now the more broadly used of the two
- **PNG/JPG export** rasterizes at a chosen resolution, exactly when a real pixel grid is actually needed
- **Outline text before export**, not instead of exporting carefully — the export format alone doesn't solve font dependency

Chapter 10 · Capstone: Designing a Complete Logo End to End in Both Tools

Every prior chapter introduced one concept at a time. This capstone designs a single real logo end to end, applying each of those concepts in sequence, with Illustrator's own workflow and Inkscape's own workflow shown side by side at every step — closing the course the way it began, treating both tools as equally real implementations of one shared discipline.

The Scenario

A logo needs three things before delivery: a custom icon mark, a wordmark set in the brand's chosen typeface, and final export covering a web SVG, a print-ready PDF, and a fixed-resolution PNG for a social media profile.

Step 1 — Sketching the Icon's Custom Shape (Chapter 3)

Use the Pen tool (Bezier tool in Inkscape) to draw the icon's own unique curved silhouette — a shape distinctive enough that Chapter 4's regular shape tools alone couldn't produce it, placing smooth points with carefully adjusted handles rather than many small corner points.

Step 2 — Building the Icon's Regular Elements (Chapter 4)

Draw the icon's simpler, regular elements — a circular badge outline, a rectangular accent bar — with the dedicated Rectangle and Ellipse tools, using precise numeric dimensions rather than dragging by eye, since these elements need to align exactly with the custom shape from Step 1.

Step 3 — Combining the Icon's Shapes with Pathfinder (Chapter 6)

Duplicate all the icon's component shapes first, then apply Union to merge the custom silhouette with the regular elements into one continuous icon shape, and Subtract to cut a small negative-space detail out of it — keeping the duplicated originals aside in case the combined result needs revisiting.

Step 4 — Applying Fill, Stroke & a Gradient to the Finished Icon (Chapter 5)

Apply the brand's solid fill color to the combined icon shape, add a subtle stroke for definition, and use Illustrator's Appearance panel to stack a second, lighter fill beneath the first for a subtle depth effect — recreated in Inkscape with a layered duplicate object beneath the main shape, since Inkscape's own Fill and Stroke dialog handles one fill per object.

Step 5 — Organizing the File with Layers and Groups (Chapter 7)

Group the icon's several component shapes together so they move and scale as one unit, then place that group on its own dedicated layer, separate from the wordmark's own layer still to come — keeping each major piece of the logo independently manageable as the file grows.

Step 6 — Adding and Styling the Wordmark (Chapter 8)

Set the brand name as live text on its own layer alongside the icon, adjusting font, size, and tracking until it visually balances against the finished icon mark — kept live and typeable at this stage, since the exact wording and spacing may still need adjustment.

Step 7 — Outlining the Wordmark Before Export (Chapter 8)

Once the wordmark's wording and styling are finalized, duplicate the live text layer and keep it hidden aside, then outline the visible copy (Create Outlines / Object to Path) — removing the font dependency entirely so the logo renders identically on any machine, while preserving the hidden original in case a future revision is ever needed.

Step 8 — Setting Up Artboards and Exporting for Web and Print (Chapter 9)

Create separate artboards for the full-color lockup and a single-color variant, then export: an SVG from the full-color artboard for the website, a PDF preserving both artboards for the print vendor, and a PNG at a fixed pixel size for the social media profile — all generated from the exact same underlying vector file.

CAPSTONE STEP	CHAPTER IT DRAWS FROM
Step 1 — Custom icon silhouette	Chapter 3
Step 2 — Regular icon elements	Chapter 4
Step 3 — Combining with Pathfinder	Chapter 6
Step 4 — Fill, stroke & layered depth	Chapter 5
Step 5 — Layers and grouping	Chapter 7
Step 6 — Live wordmark styling	Chapter 8
Step 7 — Outlining before export	Chapter 8
Step 8 — Artboards and export	Chapter 9

THE SAME SHAPE AS RASTER EDITING FUNDAMENTALS' OWN CAPSTONE

One real design worked end to end, with both tools' own equivalent workflow shown side by side at each step, is the exact same capstone shape Raster Editing Fundamentals used for its own portrait-photo capstone — fitting for a second comparative, two-tool course rather than the narrative-session shape this site's Audio & Video Production courses used for their own single-tool workflows.

EVERY SAFETY HABIT FROM THIS COURSE GETS USED HERE, NOT JUST MENTIONED

Duplicating shapes before a destructive Pathfinder operation (Step 3), keeping a hidden original before outlining text (Step 7), and using live, numerically precise shapes before converting anything (Steps 1–2) all reappear in this capstone exactly as this course originally introduced them — the same recurring "keep the original before converting" caution from Chapters 4, 6, and 8, applied together in one real project rather than left as three separate warnings.

Hands-On Exercises

EXERCISE 1

In this capstone's own Step 7, why is the wordmark outlined only after Step 6's styling is finalized, rather than immediately when the text was first placed?

View solution

EXERCISE 2

A client later asks for the icon's fill color to be adjusted slightly, after Step 4 has already been completed. Explain why this request is straightforward given how this capstone's own workflow was structured, referencing the specific step and chapter that made this possible.

View solution

EXERCISE 3

Explain why this capstone's own shape — one logo, worked through step by step with both tools' equivalent techniques shown side by side — matches Raster Editing Fundamentals' own capstone shape rather than the narrative-session shape used across this site's Audio & Video Production courses, and why that fits this specific course.

View solution

COURSE COMPLETE

Vector Graphics — 10 of 10 chapters complete. Advanced Compositing & Retouching, Figma, and Blender remain outstanding in the wider Creative & Design Tools subject.

CHAPTER 10 QUICK REFERENCE

- One real logo, worked **end to end**, with both Illustrator's and Inkscape's own workflows shown side by side at each step
- Every step draws directly from a specific earlier chapter — **nothing new is introduced** in this capstone itself
- The **"keep the original before converting"** caution from Chapters 4, 6, and 8 is applied together throughout, not just mentioned once
- The finished file exports to **SVG, PDF, and PNG** from one single vector source — Chapter 1's own resolution independence, fully realized