

The ArcGIS Cheat Sheet Pack

ON THIS PAGE

- [The ArcGIS Cheat Sheet Pack](#)
 - [Map Viewer at a Glance](#)
 - [Arcade Quick Reference](#)
 - [Analysis Tool Picker](#)
 - [Projections Quick Card](#)
 - [Layer and Format Picker](#)
 - [Sharing and Permissions Checklist](#)
 - [Field Apps Chooser and Offline Checklist](#)
 - [Schema Design Checklist](#)
 - [Styling Decision Guide](#)
 - [Troubleshooting Triage Card](#)

The ArcGIS Cheat Sheet Pack

Ten dense quick-reference cards distilled from the ArcGIS Compendium. Find any answer in under ten seconds; chapter references point into the eight volumes.

Map Viewer at a Glance

One-page reference for the ArcGIS Online Map Viewer. Left toolbar = what is in the map (Contents). Right toolbar = what you do to the selected layer or the map (Settings). Full treatment: Compendium Chapter 6 (Map Viewer Complete Reference).

Left toolbar (Contents)

ENTRY	ONE-LINE PURPOSE
Layers	The layer list: drag to reorder, eye icon to toggle visibility,  menu for per-layer actions. The Add button at the top of this panel brings in new layers.
Tables	Standalone (non-spatial) tables added to the map; view and filter records.
Basemap	Swap the basemap from your org's gallery.
Charts	Build charts from layer attributes; charts save with the map.
Legend	Read-only symbol key for visible layers. Nothing is configurable here — change symbols in Styles.
Bookmarks	Saved extents you can add, rename, and reorder. Saved with the map.
Save and open	Save, Save as, Open map. The only place your work gets written.
Map properties	Map-level options such as the map background color.
Share map	Set the map item's sharing level; prompts you about under-shared layers.
Print	Export a printable layout of the current view.
Create app	Launch an app builder (Instant Apps, Dashboards, Experience Builder, StoryMaps — Compendium Chapters 26-29) from this map.

Right toolbar (Settings)

Most panes act on the **currently selected layer** in the Layers panel — select first, then configure. Entries vary with layer type and your org privileges.

ENTRY	ONE-LINE PURPOSE
Properties	Visibility range, transparency, blend modes, and a symbology summary for the selected layer.
Styles	Smart mapping: choose attributes, choose a drawing style, adjust symbols (Compendium Chapter 7).
Filter	Attribute expressions that limit which features draw.
Effects	Visual effects (drop shadow, bloom, etc.), optionally split by a filter expression.
Aggregation	Clustering and binning for point layers, including aggregate pop-ups.
Pop-ups	Enable, disable, and compose pop-up content blocks (Compendium Chapter 9).
Fields	Map-level overrides for field aliases, decimal places, and date formats.
Labels	Label classes, expressions (Arcade — Compendium Chapter 8), style, and visible range.
Forms	Build the edit form used by Map Viewer editing and Field Maps (Compendium Chapter 30).
Edit	Create and modify features on editable layers. Edits commit to the layer immediately.
Analysis	Run spatial analysis tools; requires privileges and most tools consume credits (Compendium Chapters 16-20).
Search	Find places and addresses; also accepts typed coordinates.
Measurement	Distance and area; change units inside the tool's dropdown.

ENTRY	ONE-LINE PURPOSE
Directions	Point-to-point routing; requires network analysis privileges and consumes credits.
Sketch	Draw graphics on a sketch layer; saved with the map, not to any feature layer.

Save and share semantics

- Nothing autosaves. Closing the tab discards unsaved configuration. Save early via `Save and open > Save` .
- Saving writes **configuration** into the web map item: layer order, styles, filters, pop-ups, labels, bookmarks, charts, and the current extent (which becomes the map's opening extent).
- Feature **edits** are different: the Edit tool commits to the layer the moment you finish a feature. No map save required, and no map save will undo them.
- Styling in a map is an override. It never changes the layer item's defaults or other maps using that layer. To push a style back to a layer you own, use the layer's options menu (Save layer).
- `Save as` creates a new map item and leaves the original untouched. Other users opening your map must Save as — they cannot overwrite your item (the exception: members of a shared update group the item is shared to can save over it).
- Sharing is per item. A public map with private layers is broken for everyone but you. Accept the update-sharing prompt or fix each layer's sharing yourself.

Where commonly-hunted things hide

YOU WANT	WHERE IT IS
Rename a layer	<code>Layers > ... > Rename</code> — changes the map's label only, not the layer item
Attribute table	<code>Layers > ... > Show table</code>

YOU WANT	WHERE IT IS
Zoom to a layer	Layers > ... > Zoom to
Transparency / visible range	Select the layer, then the Properties pane
Turn a pop-up off	Select the layer, Pop-ups pane, toggle at the top
Field alias or decimals	Select the layer, Fields pane (map-level only)
Cluster on/off	Select a point layer, Aggregation pane
Rotate the map	Right-click and drag; reset north with the compass button
Refresh interval	The layer's options menu in the Layers panel
Undo a config change	There is no undo — reopen the map without saving

Five settings that live on the item page, not in the map

Open via `Content > (item) > Settings` unless noted.

SETTING	WHY IT MATTERS
Delete protection	Prevents the map (or layer) from being deleted accidentally. Turn it on for anything in production.
Offline use	Whether the map can be taken offline in Field Maps. Not configurable inside Map Viewer.
Extent override	Sets the item's default extent independently of what was saved in the viewer.
Item metadata	Title, summary, description, thumbnail, terms of use — on the Overview tab. This is what search, galleries, and many apps display.

SETTING	WHY IT MATTERS
Layer capabilities	Editing, sync, and export permissions live on each layer's item page, never in the map (Compendium Chapter 10).

Watch out: Sharing the map without sharing its layers. The map looks perfect to you because you own everything; everyone else opens a blank or partially broken map, and nobody tells you. When the share dialog warns that layers are shared more narrowly than the map, deal with it then — do not dismiss it.

Arcade Quick Reference

Compact reference for ArcGIS Arcade. The full tutorial is Compendium Chapter 8 (Arcade from Zero to Fluent); pop-ups and labels are Chapter 9, styling expressions Chapter 7, dashboards Chapter 28, forms Chapter 30, attribute rules Chapter 14.

Syntax essentials

CONSTRUCT	SYNTAX	NOTES
Variable	<code>var x = 10</code>	Semicolons optional
Field value	<code>\$feature.STATUS</code>	Field name , not alias
Field with spaces	<code>\$feature["Install Date"]</code>	Bracket form
Comment	<code>// line</code> or <code>/* block */</code>	
Two-way choice	<code>IIf(cond, a, b)</code>	Returns <code>a</code> or <code>b</code>

CONSTRUCT	SYNTAX	NOTES
Multi-branch	<code>When(c1, r1, c2, r2, default)</code>	First true condition wins
Block logic	<code>if (c) { } else if (c2) { } else { }</code>	Always use braces around bodies
Loop	<code>for (var i in arr) { arr[i] }</code>	<code>i</code> is the index
Function	<code>function f(x) { return x * 2 }</code>	Explicit <code>return</code> inside
Comparison	<code>== != > >= < <=</code>	
Logic	<code>&& !</code>	SQL strings inside <code>Filter()</code> use <code>AND / OR</code> instead
Join text	<code>"A" + "B"</code> or <code>Concatenate([a,b], ", ")</code>	
Template text	<code>`Total: \${x}`</code>	Backticks, current versions
Null guards	<code>IsEmpty(v)</code> , <code>DefaultValue(v, "n/a")</code>	Use constantly
Result	Last evaluated statement is returned	Or explicit <code>return</code>

Most-used functions by task

TASK	FUNCTIONS
Text	<code>Upper</code> <code>Lower</code> <code>Proper</code> <code>Left</code> <code>Right</code> <code>Mid</code> <code>Find</code> <code>Replace</code> <code>Split</code> <code>Trim</code> <code>Count</code> (length)
Formatting	<code>Text(value, "#,###")</code> for numbers; <code>Text(date, "MMM D, Y")</code> for dates
Math	<code>Round</code> <code>Floor</code> <code>Ceil</code> <code>Abs</code> <code>Min</code> <code>Max</code> <code>Pow</code> <code>Sqrt</code> <code>Number</code> (parse)

TASK	FUNCTIONS
Date	Now Today Date DateDiff DateAdd Year Month Day Weekday
Geometry	Geometry(\$feature) AreaGeodetic LengthGeodetic Distance Buffer Centroid Intersects Contains Within
FeatureSet	FeatureSetByName FeatureSetByPortalItem FeatureSetByRelationshipName Filter First Count OrderBy Top Sum Mean GroupBy Distinct
Domains / misc	DomainName Decode Console (debug) Push (append to array)

Where Arcade runs (profiles)

WHERE YOU WRITE IT	PROFILE	FEATURESET ACCESS	MUST RETURN
Pop-ups	Popup	Yes	Text, number, or rich-content dictionary
Styling attribute expressions	Visualization	No	Number or category text the style expects
Label expressions	Labeling	No	Text
Smart forms (calculated, constraint, visibility)	Form	Yes in calculations; limited elsewhere	Field-type value; constraint/visibility return true/false
Calculate field	Calculation	Yes	Value matching the field type
Attribute rules (Pro/Enterprise)	Attribute rule	Yes, plus \$datastore	Value, or dictionary describing edits

WHERE YOU WRITE IT	PROFILE	FEATURESET ACCESS	MUST RETURN
Dashboard data expressions	Data expression	Yes (portal items)	A FeatureSet

An expression that compiles in one profile can fail in another — test in the profile you will use.

Watch out: Calculate field on a hosted feature layer writes values immediately and permanently. There is no undo. A wrong expression silently overwrites every row it touches. Prove the expression in a pop-up or on a copy of the layer first, and calculate against a filtered selection before running it on everything.

Gotchas checklist

- ☐ `$feature` takes field names, not aliases. Check the item page field list.
- ☐ Null in any math or comparison poisons the result. Guard with `DefaultValue / IsEmpty`.
- ☐ `Month()` is zero-based (January = 0, December = 11); `Weekday()` starts at 0 = Sunday.
- ☐ `Filter()` conditions are SQL-like strings: single quotes for text, `@varName` to inject Arcade variables. Never concatenate values into the string.
- ☐ Request only needed fields in `FeatureSetByName(..., ["FIELD"])` — pop-ups get much faster.
- ☐ Broken pop-up expressions render blank, not as errors. Test in the expression editor.
- ☐ Avoid `for` loops over large FeatureSets; use `Filter`, `Sum`, `GroupBy` so the server does the work.

15 copy-paste snippets

1. Null-safe display — use when a field may be empty and you do not want "null" showing.

```
DefaultValue($feature.OWNER, "Not recorded")
```

2. Combine fields, skipping blanks — use when building one line from sometimes-empty parts.

```
var parts = [$feature.STREET, $feature.CITY, $feature.STATE];  
var out = [];  
for (var i in parts) {  
    if (!IsEmpty(parts[i])) { Push(out, parts[i]) }  
}  
Concatenate(out, ", ")
```

3. Thousands separators — use when raw numbers are unreadable.

```
Text($feature.POP2020, "#,###")
```

4. Readable date — use when showing a date field in a pop-up or label.

```
Text($feature.INSTALLED, "MMM D, Y")
```

5. Days since — use when flagging overdue inspections or stale records.

```
Round(DateDiff(Now(), $feature.LAST_INSP, "days"), 0)
```

6. Bucket a number — use when classifying a value for styling or display.

```
When($feature.SCORE >= 80, "Good", $feature.SCORE >= 50, "Fair", "Poor")
```

7. Translate codes without a domain — use when the layer stores codes but no domain exists.

```
Decode($feature.STATUS, 1, "Open", 2, "Closed", "Unknown")
```

8. Show the domain description — use when a coded value domain exists and you want its label.

```
DomainName($feature, "STATUS")
```

9. Safe percentage — use when dividing fields that can be zero or null.

```
IIf($feature.TOTAL > 0, Round($feature.PART / $feature.TOTAL * 100, 1) + "%",  
"n/a")
```

10. Acres from geometry — use when there is no area field or it is stale.

```
Round(AreaGeodetic($feature, "acres"), 2)
```

11. Value from an intersecting layer — use when a pop-up needs another layer's attribute without a join. Use `if/else`, not `IIf`, so the attribute is never read from a null feature.

```
var zones = FeatureSetByName($map, "Zoning", ["ZONE_CODE"]);  
var z = First(Intersects(zones, $feature));  
if (IsEmpty(z)) { return "No zone" }  
return z.ZONE_CODE
```

12. Count related records — use when a parent pop-up should show how many children exist.

```
Count(FeatureSetByRelationshipName($feature, "Inspections"))
```

13. Sum a field across related records — use when totaling child values in the parent.

```
var insp = FeatureSetByRelationshipName($feature, "Inspections", ["COST"]);  
Sum(insp, "COST")
```

14. Two-line label — use when a label needs a name plus a stat.

```
$feature.NAME + TextFormatting.NewLine + Text($feature.CAPACITY, "#,###")
```

15. Dashboard data expression: grouped counts — use when a dashboard element needs statistics the layer does not expose. See Compendium Chapter 28 (Dashboards).

```
var fs = FeatureSetByPortalItem(Portal("https://www.arcgis.com"),  
    "<item-id>", 0, ["STATUS"], false);  
GroupBy(fs, "STATUS",  
    {name: "total", expression: "1", statistic: "COUNT"})
```

Analysis Tool Picker

Find your question, run the tool, apply the sanity check before trusting the output. Credit flags are qualitative — rates change; see Compendium Chapter 2 (The ArcGIS Ecosystem). Legend: **None** = Pro geoprocessing on local data; **Std** = standard Online analysis credits, scales with feature count; **High** = premium service (routing, GeoEnrichment, raster/elevation), scales fast. Tool names show Online first, then Pro where they differ.

Proximity

Depth: Compendium Chapter 16 (Proximity and Overlay).

YOU WANT TO KNOW	TOOL (ONLINE / PRO)	KEY INPUTS	CREDITS	SANITY CHECK
What lies within X distance of these features?	Create Buffers / Buffer (or Pairwise Buffer)	Layer, distance + units, dissolve on/off	Std / None	Measure one buffer edge with the measure tool; width matches your distance
Which feature is closest to each point, straight-line?	Find Closest, straight-line option (Find Nearest in Map Viewer Classic) / Near or Generate Near Table	Two layers, max distance, match count	Std / None	Spot-check one pair visually on the map
Which features are near others — selection only, no new layer?	(Pro) Select By Location, on the Map ribbon	Relationship, distance, selecting layer	None	Zoom to selection; count is plausible

Overlay and combine

YOU WANT TO KNOW	TOOL (ONLINE / PRO)	KEY INPUTS	CREDITS	SANITY CHECK
Where do two layers overlap?	Overlay Layers, Intersect mode / Intersect (or Pairwise Intersect)	Two layers, output geometry type	Std / None	Output area never exceeds the smaller input
What part of A is outside B?	Overlay Layers, Erase mode / Erase (or Pairwise Erase)	Input layer, erase layer	Std / None	Erased + intersected pieces roughly rebuild the original

YOU WANT TO KNOW	TOOL (ONLINE / PRO)	KEY INPUTS	CREDITS	SANITY CHECK
All pieces of A and B, attributes from both?	Overlay Layers, Union mode / Union	Two polygon layers	Std / None	Output attribute table carries fields from both inputs
Trim data to a study area?	Overlay Layers, Intersect with boundary / Clip (or Pairwise Clip)	Input layer, boundary polygon	Std / None	Nothing renders outside the boundary
Stack same-schema layers into one?	Merge Layers / Merge or Append	Layers with matching fields	Std / None	Row count equals the sum of inputs

Summarize and join

YOU WANT TO KNOW	TOOL (ONLINE / PRO)	KEY INPUTS	CREDITS	SANITY CHECK
How many points fall in each polygon?	Aggregate Points / Summarize Within or Spatial Join	Points, polygons, stats fields	Std / None	With non-overlapping polygons, sum of counts $\leq$ total points (points outside all polygons drop)
Sum or average a field inside each polygon?	Summarize Within (same name in both)	Polygons, layer to summarize, statistic	Std / None	Grand total matches the source table's total
What surrounds each feature	Summarize Nearby (same name in	Features, travel mode + time, stats	High	Urban vs rural features show

YOU WANT TO KNOW	TOOL (ONLINE / PRO)	KEY INPUTS	CREDITS	SANITY CHECK
within a drive time?	both; uses the routing service)			very different totals
Attach fields by a shared ID?	Join Features (attribute option) / Add Join or Join Field	Key field each side, join type	Std / None	Check the unmatched-record count before using results
Attach attributes by location?	Join Features (spatial option) / Spatial Join	Spatial relationship, one-to-one vs one-to-many	Std / None	One-to-many multiplies rows — compare counts before and after
Add demographics around my features?	Enrich Layer / Enrich (GeoEnrichment service behind both)	Variables, buffer or drive-time area	High	Check one known area's population against a census figure

Patterns and statistics

Depth: Compendium Chapter 17 (Statistical and Pattern Analysis).

YOU WANT TO KNOW	TOOL (ONLINE / PRO)	KEY INPUTS	CREDITS	SANITY CHECK
Significant clusters of high/low values?	Find Hot Spots / Hot Spot Analysis (Getis-Ord Gi*)	Analysis field or point counts, aggregation scale	Std / None	Hot spots persist at a different aggregation scale
Features unlike their neighbors?	Find Outliers / Cluster and Outlier Analysis	Analysis field	Std / None	Inspect one flagged outlier —

YOU WANT TO KNOW	TOOL (ONLINE / PRO)	KEY INPUTS	CREDITS	SANITY CHECK
				the contrast should be obvious
Where are points densest?	Calculate Density / Kernel Density (Spatial Analyst)	Points, search radius, cell size	Std / None	Peaks sit where you visibly see the most points
Natural groupings of points, no polygons?	Find Point Clusters / Density-based Clustering	Min features per cluster, search distance	Std / None	Expect unassigned "noise" points — zero noise suggests bad parameters
Predict values between sample points?	Interpolate Points / IDW or Kriging (Spatial Analyst)	Sample points, value field	Std-High / None	IDW output range stays within the measured min/max

Network

Depth: Compendium Chapter 18 (Network Analysis). All of these use street-network routing, not straight lines.

YOU WANT TO KNOW	TOOL (ONLINE / PRO)	KEY INPUTS	CREDITS	SANITY CHECK
Area reachable within X minutes' drive?	Generate Travel Areas / Service Area (Network Analyst)	Facilities, travel mode, time/distance breaks	High	Shape hugs the roads — a neat circle means you got straight-line by mistake
Best route through several stops?	Plan Routes / Route	Stops, travel mode, optimize order on/off	High	Compare one route's time against a mapping app you trust

YOU WANT TO KNOW	TOOL (ONLINE / PRO)	KEY INPUTS	CREDITS	SANITY CHECK
Closest facility by travel time?	Find Closest, travel-mode option / Closest Facility	Incidents, facilities, cutoff	High	Nearest-by-road can differ from nearest-by-air — verify one case

Raster and zonal

Depth: Compendium Chapter 19 (Terrain and Raster Analysis) and Chapter 15 (Imagery and Rasters).

YOU WANT TO KNOW	TOOL (ONLINE / PRO)	KEY INPUTS	CREDITS	SANITY CHECK
Average raster value per polygon (rainfall per district)?	Zonal Statistics as Table (Pro, Spatial Analyst; Online raster analysis needs imagery privileges)	Zone polygons, value raster, statistic	High / None	Every zone mean falls inside the raster's min/max
Slope, aspect, hillshade from elevation?	Slope, Aspect, Hillshade (Pro surface tools); in Online, Living Atlas terrain layers cover most display needs, and computing your own requires raster analysis with imagery privileges	Elevation raster or terrain service	High / None	Steepest slopes align with tight contour spacing
What is visible from a point?	Create Viewshed (ready-to-use elevation service) / Viewshed (Spatial Analyst)	Observer points, height, distance	High / None	Ridgelines block visibility behind them

Naming traps

- **Overlay Layers** (Online) is one tool with a mode dropdown; Pro splits it into Intersect, Union, and Erase.
- **Find Closest** (Online; called Find Nearest in Map Viewer Classic) offers straight-line or travel mode — the travel option bills like a network tool.
- **Join Features** (Online) covers both attribute and spatial joins; Pro separates Add Join from Spatial Join.
- **Pairwise** tool variants in Pro (Buffer, Clip, Erase, Intersect) are faster equivalents for most everyday uses.

Before you run anything

- ☐ Suitable projected coordinate system, or geodesic distance option enabled — Compendium Chapter 3 (Coordinate Systems and Projections)
- ☐ Filter to only the features you need — faster runs, fewer credits
- ☐ Check the "use current map extent" toggle (Online); it silently includes or excludes data
- ☐ Field types match the job: numeric stats need numeric fields, joins need matching key types
- ☐ In Online, read the estimated-credits dialog before clicking Run
- ☐ Name outputs so you can tell run 3 from run 4 next week

Watch out: The costliest mistake here is running a credit-hungry tool (Enrich Layer, Generate Travel Areas, Interpolate Points, raster analysis) against an entire large layer when you meant a filtered subset — with the extent toggle set wrong. Credits are spent even when the result is useless. Filter first, confirm the extent toggle, and read the credit estimate before every run.

Projections Quick Card

One-page lookup for choosing, checking, and fixing coordinate systems. Full treatment: Compendium Chapter 3 (Coordinate Systems and Projections).

Pick a system by task

YOUR TASK	USE THIS	WHY
Web map display	Web Mercator (EPSG:3857)	Matches every standard basemap and tile service; tiles align, pan/zoom is fast
Comparing areas or densities	An equal-area projection (Albers for continental US, Equal Earth or Mollweide for world)	Only equal-area projections preserve relative size; Web Mercator wildly inflates high latitudes
Regional engineering, surveying, utilities (US)	State Plane zone for your area	Designed for very low distortion inside its zone; matches surveyor and CAD data
Regional work outside the US, or multi-state corridors	UTM zone for your area	Worldwide zone grid, metric units, low distortion within one zone
Global data exchange, storage, GPS coordinates	WGS84 geographic (EPSG:4326)	Universal lingua franca; every system reads lat/long on WGS84
Navigation with constant compass bearings	Mercator	Straight rhumb lines; its one legitimate specialty
Polar mapping	A polar stereographic projection	Web Mercator cannot even show the poles

Decision shortcut: display on the web = Web Mercator, measure or analyze = a local projected system, store and share = WGS84, compare sizes = equal-area.

The four systems you will actually meet

SYSTEM	KIND	UNITS	BUILT FOR	DO NOT USE FOR
WGS84 (EPSG:4326)	Geographic (datum + lat/long, not a projection)	Decimal degrees	GPS output, data storage and exchange, API defaults	Planar measurement of any kind; a "degree" is not a fixed ground distance
Web Mercator (EPSG:3857)	Projected, conformal- ish	Meters (nominal, badly distorted with latitude)	Web basemaps, tiled services, ArcGIS Online display	Measuring distance or area, comparing region sizes, anything legal or engineering
UTM (60 north/south zone pairs, each 6 degrees wide)	Projected, transverse Mercator	Meters	Regional analysis and fieldwork anywhere on Earth	Data spanning multiple zones without reprojecting to one
State Plane (US only, per-state zones)	Projected, zone-specific	Feet or meters — always check which	County and state engineering, parcels, public works	Anywhere outside the designed zone; assuming units without checking

Checklist before any analysis:

- ☐ All inputs share one coordinate system (mixed inputs cause misalignment and slivers — Compendium Chapter 16)
- ☐ That system is projected, local, and suited to the measure (equal-area for area, conformal for shape)
- ☐ Units confirmed (feet vs meters is a classic State Plane bug)

- ☐ Datum transformation set when converting between NAD27, NAD83, and WGS84 — ignoring the prompt shifts every feature by a systematic offset (locally uniform, varies by region)

Geodesic vs planar measurement

	PLANAR	GEODESIC
How it works	Straight-line math on the flattened map	Computed on the curved ellipsoid
Correct when	Data is in a projection designed for that area and that measure	Always, in any coordinate system
Wrong when	Data is in Web Mercator or WGS84, or spans a large extent	Rarely; slightly slower

The rule: **if the data is in Web Mercator or WGS84, or covers more than one zone's worth of ground, measure geodesically.** Map Viewer's measure tools work geodesically. In Pro, pick the geodesic option in the Measure tool, in `right-click field > Calculate Geometry`, and in tools like Buffer (choose the geodesic method). Arcade has geodesic geometry functions for pop-ups and calculated fields (Compendium Chapter 8).

Buffers deserve special mention: a planar buffer on Web Mercator data is not a circle on the ground and its size error grows with latitude. Use geodesic buffers (Compendium Chapter 16).

Project vs Define Projection

Two tools in `Data Management Tools > Projections and Transformations` that sound alike and do opposite things:

TOOL	WHAT IT CHANGES	USE WHEN
Project	The coordinates themselves — data is mathematically converted to the new system and stays in the right place	You want the data in a different coordinate system
Define Projection	Only the metadata label — coordinates are untouched	The label is missing or you know for certain it is wrong (e.g., "Unknown coordinate system")

Watch out: Running Define Projection to "fix" a layer that draws in the wrong place is the single most damaging projection mistake. It relabels the coordinates without converting them, so the data lands anywhere from a few hundred meters off to a different continent — and the file now lies about what it is, which poisons every later workflow. Define Projection is for correcting a wrong or missing label only. If the data needs to be in a different system, use Project.

Fast diagnostics

SYMPTOM	LIKELY CAUSE	FIX
Layer draws in the ocean near the equator/prime meridian (0,0)	Coordinates are lat/long degrees but labeled as a projected system, or the label is missing	Define Projection with the true source system, then Project if needed
Layer is offset by a consistent amount (tens to hundreds of meters)	Datum mismatch — usually NAD27 vs NAD83/WGS84 with no transformation applied	Reproject with the correct geographic transformation
Areas or distances look inflated toward the poles	Measuring planar in Web Mercator	Switch to geodesic measurement, or project to a local system

SYMPTOM	LIKELY CAUSE	FIX
Two "identical" layers do not line up at high zoom	Different datums or a sloppy define somewhere upstream	Check each layer's coordinate system in its properties; fix at the source
Web layer publishes fine but sits in the wrong spot	Wrong define before publishing	Fix the source data and republish (Compendium Chapter 10)

Defaults worth setting in Pro

- A new map adopts the coordinate system of the first layer added — and since new maps usually open with a basemap already in them, that means Web Mercator. Set it deliberately: **right-click the map in Contents > Properties > Coordinate Systems** .
- Set the geoprocessing environment **Output Coordinate System** when a whole workflow should stay in one system (Compendium Chapter 22).
- When Pro prompts for a geographic transformation, choose one — do not dismiss the warning.
- Check a layer's system anytime via the layer's **Properties > Source** in Pro; for a hosted layer in ArcGIS Online, the spatial reference is listed on its service (REST) page rather than the item details page (Compendium Chapter 32).

Layer and Format Picker

Use this sheet when you are staring at "Add layer" or "New item" and need to know what a layer type can do, or what a file will turn into after upload. Deep treatment of each layer type is in Compendium Chapter 10 (Hosted Feature Layers), Chapter 15 (Imagery and Rasters), and Chapter 35 (ArcGIS Enterprise).

Layer type matrix

LAYER TYPE	CLICKABLE (POP-UPS)?	EDITABLE?	RESTYLEABLE?	BEST FOR
Feature	Yes	Yes, when hosted with editing enabled	Yes, fully (smart mapping, Arcade)	The default. Anything you query, filter, edit, analyze, or style yourself
Vector tile	No attribute pop-ups	No	Yes — colors, fonts, visibility via the style editor	Basemaps; huge datasets that must draw fast and never change
Tile (raster cache)	No	No	No — pixels are pre-rendered	Static reference maps, cached scans, legacy basemaps
Imagery	Yes — pixel values and attributes via identify	No	Rendering only (stretch, band combos, colormaps)	Satellite and aerial imagery, elevation, analysis rasters — Chapter 15
Map image	Yes (queries go to the server)	Not directly; edits go through a companion feature endpoint	Limited; the server draws it	Many-layer maps published from ArcGIS Enterprise — Chapter 35
Scene	Yes for point and 3D object scenes (attributes are cached with the layer); integrated mesh and point cloud	No; editing goes through an associated feature layer where one exists	Limited (3D symbology)	3D buildings, meshes, point clouds in Scene Viewer

LAYER TYPE	CLICKABLE (POP-UPS)?	EDITABLE?	RESTYLABLE?	BEST FOR
	carry little or none			
Table	No map presence at all	Yes, when hosted	n/a	Related records, joins, standalone lists — Chapter 12
Group	Inherits from children	n/a	Children styled individually	Organizing the layer list; one visibility toggle for a set
Media	No	Placement only (position, transparency)	No	Georeferenced images: floor plans, scanned historical maps

Quick rules of thumb:

- Need pop-ups, filters, or editing → it must be (or be backed by) a **feature layer**.
- Need speed on data that never changes → **tile or vector tile**.
- Someone else's server, many sublayers → **map image**; you consume it, they maintain it.
- If a layer refuses to restyle in Map Viewer, check its type first — tiles and media are pre-rendered by design, not broken.

Import format table

All of these upload from `Content > New item`, or straight into a map from the Add layer menu in Map Viewer. Full walkthroughs are in Compendium Chapter 11 (Creating Data).

FORMAT	PREP BEFORE UPLOAD	WHAT IT BECOMES	GOTCHAS
CSV / Excel	One header row; latitude/longitude columns, or clean single-purpose address columns; no merged cells or formulas in headers	Hosted feature layer (points) or hosted table	Address geocoding consumes credits — Chapter 2. Lat/lon columns cost nothing. Check field types after upload; ZIPs love to become numbers
Shapefile	Zip the whole set together — .shp, .shx, .dbf, and .prj at minimum, no nested folder	Hosted feature layer	Missing .prj = data lands in the wrong place (Chapter 3). Field names truncate at ten characters; long text and dates can be lossy
GeoJSON	Coordinates should be WGS84 (the spec requires it); one file	Hosted feature layer	No field aliases or domains come through; very large files upload slowly
KML / KMZ	None — upload as-is	A linked KML layer with styling baked in	Effectively read-only: not editable, not restylable, weak pop-ups. Convert to a feature layer if you need to do real work with it
GPX	Export straight from the device or app	Hosted feature layer (waypoints and tracks)	Attributes are minimal; timestamps usually survive, custom fields usually do not

FORMAT	PREP BEFORE UPLOAD	WHAT IT BECOMES	GOTCHAS
File geodatabase	Zip the .gdb folder itself	Hosted feature layer(s) — keeps domains, aliases, long field names, multiple feature classes	The highest-fidelity path from ArcGIS Pro. Prefer it over shapefile every time — Chapters 11 and 12

Format decision in one line: **have Pro → file geodatabase; have coordinates in a spreadsheet → CSV; handed a shapefile → zip it; handed KML → convert it.**

Hosted layer vs file dropped into a map

Small files can be added straight into a single map instead of becoming a hosted layer item. The convenience is real; so is the trap.

	HOSTED FEATURE LAYER	FILE ADDED DIRECTLY TO ONE MAP
Where the data lives	Its own item in Content	Embedded inside that one map
Reusable across maps/apps	Yes — one source, many maps	No — re-upload per map, copies drift apart
Editable later	Yes, with editor tracking and views	No — the embedded copy is a static snapshot; to change it, re-upload
Size and performance	Scales; indexed and queryable — Chapter 10	Small files only; the whole thing loads with the map
Shareable / securable on its own	Yes (sharing levels, view layers)	No — rides on the map's sharing
Cleanup burden	One item to manage	Hidden data in every map that used the shortcut

Pick hosted when any of these is true:

- ☐ More than one map, app, or person will use the data
- ☐ Anyone will ever edit it
- ☐ The dataset is more than trivially small
- ☐ You need views, sharing control, or field-level tweaks (Chapter 10)

Pick in-map only when all of these are true:

- ☐ Throwaway or one-time visual check
- ☐ Tiny file
- ☐ You will never need it in another map

Watch out: Dropping the same file into several maps creates independent embedded copies. A fix or restyle in one map changes nothing anywhere else, and months later nobody can say which copy is current. If the data matters at all, publish it once as a hosted feature layer and add that layer everywhere.

Ten-second answers

QUESTION	ANSWER
Why can't I edit this layer?	It's a tile, imagery, map image, or KML layer — or editing is disabled on the item (Chapter 10)
Why no pop-ups?	Tile, vector tile, and media layers carry no queryable attributes
Why did my shapefile land in the ocean?	No .prj in the zip — Chapter 3
Cheapest way to map a spreadsheet?	Add lat/lon columns before upload; geocoding addresses spends credits
Best upload format overall?	Zipped file geodatabase

QUESTION

ANSWER

CSV came in with wrong field types?

Fix types during the upload's field-review step, or re-publish — Chapter 11

Sharing and Permissions Checklist

Sharing in ArcGIS Online is per-item, not per-app. An app can be public while its map or layers are private — the app loads, the content doesn't. Work the chain from the bottom up. For hosted layer mechanics see Compendium Chapter 10 (Hosted Feature Layers); for admin roles and privileges see Compendium Chapter 34 (Administration).

The sharing chain

Every viewer must be able to reach every link. One private link breaks the whole chain.

LEVEL	WHAT IT IS	WHERE TO CHECK
App	Instant App, Dashboard, StoryMap, Experience	App's item page > Share
Web map / scene	The map the app displays	Map's item page > Share
Layers	Each hosted or referenced layer in the map	Each layer's item page > Share
Referenced services	Layers served from Enterprise or external servers	The source server's own security settings
Locator / utility services	Custom geocoders, routing, print services used by the app	Each utility item's Share setting

Rule: the sharing level of every item must be equal to or broader than the app's audience. Public app means public map means public layers.

Chain pre-flight:

- ☐ Open the app's item page and its configuration — identify every map it uses (most app items list their web map in the item details)
- ☐ Open each web map in Map Viewer — walk the full layer list, including group layers, hidden layers, and any basemap overrides; each shares independently
- ☐ Check **Share** on the app, each map, each layer: all match the intended audience
- ☐ Don't work from memory — open the actual map and enumerate; forgotten reference layers are the classic miss
- ☐ Test in a private/incognito browser window while signed out (public) or signed in as a target-audience user (group)

Group sharing steps

For sharing with a specific team instead of the whole org or public.

- ☐ Create the group: **Groups > Create group** — set who can view, who can join, who can contribute
- ☐ Decide membership type: invite-only for controlled access; request-to-join only if you will actively review requests
- ☐ Invite members, or add them directly if your role allows
- ☐ Share the app, map, and all layers to the group (all three levels — the chain rule still applies)
- ☐ If members will edit each other's items, this needs a shared update group — it must be created with that capability enabled; it cannot be switched on later
- ☐ Verify: sign in as a group member (or ask one) and open the app
- ☐ Document the group's purpose in its description so the next admin knows why it exists

Master + view pattern for public data

Never share your editable master layer publicly. Publish a read-only view instead. Details in Compendium Chapter 10 (Hosted Feature Layers).

- ☐ Keep the master hosted feature layer private or org-only

- ☐ From the master's item page: **Create View Layer**
- ☐ In the view's settings, disable editing entirely
- ☐ Restrict the view's fields — hide collector names, phone numbers, internal notes, anything you don't need exposed. This is the real protection: hiding a field only in the pop-up is cosmetic — it stays queryable through the service
- ☐ Apply a definition filter if only some features are public (e.g., status = approved)
- ☐ Share the view (not the master) to Everyone
- ☐ Build the public map from the view, the internal editing map from the master
- ☐ Re-check the view after any schema change on the master — new fields may appear in the view

Editable-layer exposure check

Run this on every layer shared beyond your team. Anonymous editing on a public layer means anyone on the internet can add, change, or delete your features.

- ☐ Item page > **Settings** > editing section: is editing enabled?
- ☐ If enabled and the layer is public: is that genuinely intended (e.g., a crowdsourcing layer)?
- ☐ If crowdsourcing: restrict operations to add-only; disable update and delete
- ☐ Turn off "editors can see all features" / "editors can edit all features" unless required
- ☐ Disable export/download on the item unless you mean to offer raw data — but remember: on a public feature layer, anyone can still read the data through queries; export settings control convenience, not access
- ☐ Check delete protection is on for anything you cannot afford to lose
- ☐ Confirm attachments don't contain sensitive photos or documents before exposing them
- ☐ Check sync: offline-enabled layers can be pulled in full by anyone with access

Watch out: Sharing an editable master layer to Everyone is the single most damaging mistake in this topic. It hands the public full write access to your

production data — vandalism, mass deletes, and junk features, often unnoticed for weeks. Always publish a read-only view for public consumption and keep the master private.

Public-launch checklist

Final gate before you send the link out.

- ☐
 1. App, map, and every layer shared to Everyone (walk the full chain)
- ☐
 2. Public layers are read-only views, not editable masters
- ☐
 3. Sensitive fields removed at the view level, not just hidden in pop-ups (Compendium Chapter 9) — pop-up hiding doesn't stop service queries
- ☐
 4. No subscriber or premium content in the map — for anonymous users it either demands a login, or, if you've authorized public access to it, consumes your org's credits on their behalf
- ☐
 5. Custom utility services (geocoder, routing, print) are shared and won't leak credits or credentials
- ☐
 6. Tested the exact public URL in a signed-out incognito window — no login prompt, everything draws
- ☐
 7. Tested on a phone — public users are mostly mobile
- ☐
 8. Pop-ups, labels, and symbology reviewed for internal jargon or draft notes

- ☐
 9. Item pages have real titles, summaries, and thumbnails — they're publicly visible metadata
- ☐
 10. Delete protection enabled on the app, map, and layers; ownership assigned to a durable account, not a departing intern

Fast diagnostics

SYMPTOM	LIKELY CAUSE
App loads, layers missing or erroring	A layer shared narrower than the app
Login prompt on a "public" app	Map or a layer still private, or subscriber/premium content in the map
Group member can't see the app	Item not shared to the group, user isn't actually a member, or member is from another org and the item is shared only org-wide
Public user sees edit tools	Editing enabled on the shared layer or view
Layer visible but fields missing	View field restrictions — check whether that's intended

More failure patterns in Compendium Chapter 39 (Troubleshooting Encyclopedia).

Field Apps Chooser and Offline Checklist

Three mobile apps, three metaphors: Field Maps is map-first, Survey123 is form-first, QuickCapture is button-first. Pick by workflow, not by habit. Full treatment of all three:

Compendium Chapter 30 (Field Maps, Survey123, QuickCapture). End-to-end walkthrough: Compendium Chapter 37 (Worked Project - Field Collection).

Decision matrix

FACTOR	FIELD MAPS	SURVEY123	QUICKCAPTURE
Core metaphor	Map with editable layers	Standalone smart form	Big buttons, one tap per record
Form complexity	Moderate: conditional visibility, calculated expressions, required fields	Highest: full branching logic, repeats, constraints, multi-page forms (XLSForm or web designer)	Minimal: each button writes preset values, with at most a quick single-entry prompt
Map-centricity	High. You navigate to and select features on a map	Low. Map appears only as a location question	Low. Location captured automatically in the background
Capture speed	Moderate: tap map, fill form	Slowest: work through the form	Fastest: single tap, usable from a moving vehicle
Edit existing features	Yes: inspect, update, move geometry	Limited: existing records via the Inbox only	No. Capture-only
Related records	Yes: view and edit related tables from the parent feature	Repeats publish as related records; editing existing ones is limited	Not practical
Attachments	Photos, video, and files from the device, per feature	Photos, audio, signatures, sketches, image annotation	Photos tied to the button press

FACTOR	FIELD MAPS	SURVEY123	QUICKCAPTURE
Offline	Yes: pre-planned or on-device map areas	Yes: forms work offline; add an offline basemap for context	Yes: projects run offline by default
Typical jobs	Asset inspection, maintenance, inventory against known features	Structured surveys, permits, inspections with heavy logic and signatures	Windshield surveys, damage tallies, wildlife counts, incident pins

Ten-second chooser

- You must **see or edit existing features** on a map → **Field Maps**.
- You need **branching logic, repeats, or signatures** → **Survey123**.
- You need **one-tap speed** and no form at all → **QuickCapture**.
- You need map context *and* a complex form → Field Maps for the map, launching a **Survey123 form from a pop-up link** on the feature. Chapter 30 covers the app-link pattern.
- Data lands in hosted feature layers either way: publishing, views, and sync settings are Compendium Chapter 10 (Hosted Feature Layers); field design, domains, and relationships are Chapter 12 (Schema Design).

Offline pre-flight checklist

Run this before any crew leaves the building. Web vs. Pro editing behavior: Compendium Chapter 13 (Editing Workflows).

Layers and map

- ☐ Sync enabled on every editable hosted feature layer (item page settings).
- ☐ Offline mode enabled on the web map itself (map item settings, offline section).
- ☐ Basemap is export-enabled, or you have sideloaded a tile package to the device. A standard basemap that blocks export is the most common "can't take offline" cause.

- ☐ Offline map areas pre-defined where crews work in the same zones (Field Maps Designer); otherwise confirm on-device download works.
- ☐ Editing settings match the job: add-only vs. full edit, and whether editors see only their own features.

Form and schema

- ☐ Domains and default values set so pick lists work without connectivity (Chapter 12).
- ☐ Attachments enabled on the layer if photos are required.
- ☐ Photo capture size set in the app settings, not left at device maximum.
- ☐ Related tables published in the same service as the parent layer if the workflow needs them.

People and devices

- ☐ Every field account can sign in and holds the right user type and app license (Compendium Chapter 34, Administration).
- ☐ Full dry run on the actual device in airplane mode: open map, collect a feature, attach a photo, reconnect, sync, verify the record in the office map.
- ☐ Sync cadence agreed: end of shift minimum, Wi-Fi for attachment-heavy loads.
- ☐ Someone in the office owns watching for sync errors during the campaign.

Watch out: Never change a layer's schema, disable sync, or overwrite the service while crews hold offline areas. Doing so invalidates their downloaded copies, and unsynced field edits can become unrecoverable. Freeze the schema for the life of the field campaign; if a change is unavoidable, have every device sync in first, then re-download after the change.

Top five sync gotchas

#	GOTCHA	SYMPTOM	PREVENTION
1	Schema changed or sync disabled mid-	Sync fails outright; edits stranded on	Freeze schema during fieldwork. Sync everyone in before any layer

#	GOTCHA	SYMPTOM	PREVENTION
	campaign	devices	change (see Watch out above)
2	Basemap won't download	Offline area stalls or errors before crews leave	Use an export-enabled basemap or sideload a tile package; limit the download to the zoom levels the job actually needs
3	Attachment bloat	Sync hangs or fails on cellular; huge upload queues	Set photo size in the app, require Wi-Fi for photo-heavy syncs, split large offline areas per crew
4	Crews never sync	Days of edits on one device; office map stale; device loss = data loss	Enforce end-of-shift sync as policy; enable auto-sync where connectivity allows; monitor last-sync per user
5	Two crews edit the same feature	Later sync silently wins; earlier edit disappears	Assign non-overlapping work areas; use assignment fields or editor tracking; review edits daily while memories are fresh

Sync error messages and their fixes are catalogued in Compendium Chapter 39 (Troubleshooting Encyclopedia).

Quick reference: what each app cannot do

APP	HARD LIMITS TO REMEMBER
Field Maps	Form logic is simpler than Survey123; no repeats inside the form itself (use related tables)
Survey123	Weak at browsing or editing the existing feature map; Inbox is the only path to existing records
QuickCapture	Essentially form-free: preset values plus at most a quick prompt; no editing, no related records; wrong tool the moment users must answer real

Schema Design Checklist

Use this before you publish a layer or hand a form to field crews. Full treatment: Compendium Chapter 12 (Schema Design). Schema decisions are cheap before data exists and expensive after.

Field-type selection

YOU ARE STORING	USE	WHY / NOTES
Category from a known list	Text + coded value domain	Enforces valid values; renders as a dropdown in forms
Free-form notes	Text, generous length	Growing length later is usually possible; shrinking is not safe
Small whole numbers (counts, ratings)	Short integer	Smallest adequate type
Large whole numbers	Long integer	Population, IDs without leading zeros
Measurements, percentages, money	Double	Skip Float; the savings are not worth precision surprises
Dates and times	Date	Never store dates as text — you lose sorting, filtering, and time analysis (Compendium Chapter 20)

YOU ARE STORING	USE	WHY / NOTES
Yes/no	Text or short integer + two-value coded domain	Most geodatabase formats have no true boolean field
IDs with leading zeros (parcel, phone, ZIP)	Text	Numeric types strip leading zeros permanently
Unique record identifier	GlobalID	System-maintained; required for offline sync and used as the key for attachments and relationships
Photo, document	Attachments (not a field)	See attachments decision below

Watch out: A field's type is permanent once data exists. You cannot convert a populated text field to a date or number field in place — the only fix is add a new field, calculate values across, repair every map, form, pop-up, and app that referenced the old field, then delete it. Storing dates or numbers as text is the single most expensive schema mistake. Decide types before the first row exists.

Naming rules

RULE	DETAIL
Treat field names as permanent	Hosted feature layers offer no rename; a geodatabase rename tool exists but still breaks every map, form, and query that used the old name. Get the name right first — the alias is the safe thing to change
Lowercase, underscores, no spaces	<code>install_date</code> , not <code>Install Date!</code>
Start with a letter	Never a number or underscore

RULE	DETAIL
Keep names short	Some formats truncate long names on export (shapefile is brutal — Compendium Chapter 11)
Avoid SQL reserved words	<code>date</code> , <code>order</code> , <code>user</code> , <code>desc</code> , <code>add</code> cause query errors later
Set a human-readable alias on every field	Aliases drive pop-ups, forms, and legends; they are cheap to change anytime
No PII in field names	Names leak into URLs and REST endpoints (Compendium Chapter 32)

Coded domain vs free text

CHOOSE CODED DOMAIN WHEN	CHOOSE FREE TEXT WHEN
The value list is known and finite (status, material, species)	Genuinely unpredictable content (comments, descriptions)
You will filter, symbolize, or count by the value	The value is never used for filtering or symbology
Multiple people will enter data	One careful editor, throwaway dataset
Field crews use forms (dropdowns beat typing — Compendium Chapter 30)	The list would exceed practical dropdown length

Rule of thumb: if you will ever chart or filter by it, use a domain. "Oak / oak / OAK / Oaktree" in free text costs a cleanup project later; adding a code to a domain costs a minute.

Defaults and required flags

SETTING	USE IT FOR	CAUTION
Default value	The most common value (status = "Active", condition = "Unknown")	A wrong default gets silently accepted at scale — prefer "Unknown" over a plausible-looking guess
Required (non-nullable)	Fields the record is meaningless without	Every required field slows field entry; require only what you would reject a record over
Domain + default together	High-volume field collection	Fastest possible form entry

Editor tracking and GlobalIDs

- Turn both on before anyone edits. Editor tracking is a layer setting — in ArcGIS Online, on the hosted feature layer's item Settings tab (Compendium Chapter 10). GlobalIDs are present on most hosted layers automatically and are added when sync is enabled; if you publish from Pro, add them before publishing.
- Editor tracking records who created and last edited each feature, and when. For a hosted layer it is usually your only audit trail, and it powers ownership-based editing restrictions.
- GlobalIDs are required for offline sync in Field Maps and key the attachments plumbing. Enabling them mid-project on an offline workforce invites sync trouble — do it on day one.
- Retroactive enabling backfills nothing: edits made before tracking was on have no history.

Attachments: yes or no

Enable attachments when photos or documents belong to a specific feature (pole photo, permit PDF). Skip them when:

- Images are large and numerous — storage counts against your organization (Compendium Chapter 2).

- The file relates to the whole dataset, not one feature — store it as a separate item.
- You need to query by content — attachments are opaque blobs; store descriptive fields alongside.

Enabling attachments later is possible, but decide before designing field forms so crews capture photos from day one.

Related table or more fields

SIGNAL	VERDICT
One value per feature, ever (install date, owner)	Field on the feature
Many records per feature over time (inspections, readings, maintenance visits)	Related table
You keep adding <code>inspection1_date</code> , <code>inspection2_date</code> ...	Stop — that is a related table
Data edited by a different team on a different schedule	Related table

Relationships need a key: GlobalID on the parent, GUID field on the child. Design them together — retrofitting keys onto loose tables is manual work. Details in Compendium Chapter 12.

Cost of change after data exists

CHANGE	COST
Field alias, pop-up, symbology	Trivial — change anytime
Add a field, add a domain code	Easy

CHANGE	COST
Add a domain to an existing free-text field	Moderate — must clean existing values first
Enable attachments, editor tracking	Moderate — no backfill of history
Increase text length	Usually possible
Rename a field, change its type, shrink text length	Effectively a rebuild — not supported in place on hosted layers, and even where a geodatabase tool can rename, every dependent map, form, and app breaks. Plan on add/calculate/delete plus repairs
Change geometry type or the layer's spatial reference	Republish from scratch (Compendium Chapter 10)

12-point pre-collection checklist

1. ☐ Every field's type checked against the table above — no dates or numbers stored as text
2. ☐ Field names lowercase, no spaces, no reserved words; final (not placeholders)
3. ☐ Every field has a plain-language alias
4. ☐ Every categorical field has a coded value domain, including an "Unknown" or "Other" code
5. ☐ Defaults set on high-volume fields; no misleading defaults
6. ☐ Required flags limited to genuinely mandatory fields
7. ☐ GlobalIDs enabled
8. ☐ Editor tracking enabled
9. ☐ Attachments decision made and configured
10. ☐ Repeating data moved to related tables with keys defined

11. [] Test features created, edited in the real form (Compendium Chapter 13), then deleted
12. [] Schema reviewed by one person who did not design it
- If all twelve pass, publish. If any fail, fix them now — every row collected multiplies the cost of the fix.

Styling Decision Guide

Match your data's shape to a style family first, then pick classification and color. Full click paths and every style option: Compendium Chapter 7 (Styling and Smart Mapping). Color theory and visual hierarchy: Compendium Chapter 4 (Cartographic Design).

Step 1: What shape is your data?

YOUR DATA	STYLE FAMILY	NOTES
Categories (land use, status, type)	Types (unique symbols)	Qualitative colors. Merge rare categories into "Other" if you have many.
Counts / totals (population, crash count)	Counts and Amounts (size)	Counts get size , not color. Colored polygons of raw counts mislead.
Rates / ratios (percent, per capita, density)	Counts and Amounts (color) — choropleth	Only for normalized values. Set the divisor in the style pane or precompute the field.
Value with a meaningful midpoint (change, +/- from average)	Above and Below	Diverging ramp centered on zero, average, or a target.

YOUR DATA	STYLE FAMILY	NOTES
Dense point data (thousands of overlapping points)	Clustering or binning (aggregation), or heat map	See Step 4 table.
Totals inside polygons, shown as texture	Dot density	One dot = N units. Zoomed-out (small-scale) views only.
Two variables	Color and Size, or Relationship (bivariate)	Strongest combo: rate as color + count as size. Bivariate grids need a 10-second legend read — test on someone.
"Which category wins here"	Predominance	Compares competing count fields per polygon; opacity = strength of the win.
Location only, no attribute	Single symbol (Location)	Don't invent meaning; one symbol is often the right answer.

If a field can be styled with a computed value instead of raw data (a ratio, a category from thresholds), write an Arcade expression rather than adding a field — Compendium Chapter 8 (Arcade from Zero to Fluent).

Step 2: Classification method — each one has a bias

METHOD	HOW IT SPLITS	ITS HONEST BIAS	REACH FOR IT WHEN
Natural breaks (Jenks)	Minimizes variance within classes	Finds "groups" even in smooth data; breaks are dataset-specific, so two maps aren't comparable	One standalone map, clumpy data, no better reason

METHOD	HOW IT SPLITS	ITS HONEST BIAS	REACH FOR IT WHEN
Quantile	Equal feature count per class	Manufactures visual contrast; near-identical values land in different classes	Rankings; evenly spread data; you want every class populated
Equal interval	Equal-width value ranges	Skewed data dumps most features into one class; map looks empty	Familiar ranges (percentages, temperature); legend must be simple
Standard deviation	Distance from the mean	Assumes roughly normal data; legend baffles general readers	Analyst audiences; "how unusual is this place"
Manual	You choose breaks	Your bias — which is fine when breaks come from policy or domain thresholds	Regulatory cutoffs, round numbers, comparing maps across dates or regions
Unclassed (continuous)	No breaks; smooth ramp	Hides exact values; hard to cite a number	Showing overall pattern, not lookup

Checklist before committing:

- ☐ Looked at the histogram in the style pane (outliers? skew?)
- ☐ Kept classes to roughly three to seven
- ☐ Rounded break values in the legend to numbers a human can repeat
- ☐ If maps will be compared side by side, used identical manual breaks on all of them

Step 3: Color ramp rules

RAMP	USE FOR	RULE
Sequential (light to dark, one hue)	Ordered values, low to high	Dark = more on light basemaps; flip when the basemap is dark. Never break this.
Diverging (two hues meeting at neutral)	Values around a meaningful midpoint	The midpoint must mean something (zero, average, target) — otherwise use sequential.
Qualitative (distinct hues)	Unordered categories	Keep it to about seven distinct hues; beyond that, group into "Other." No hue should look "higher" than another.

Colorblind rule: never let red vs. green carry the only meaning. Prefer ramps flagged colorblind-friendly in the ramp picker, and vary lightness as well as hue so the ramp survives grayscale. When in doubt, run the map through a color-vision simulator before publishing.

Watch out: The single most damaging styling mistake is a choropleth of raw counts — coloring polygons by totals when the polygons differ in size or population. Big empty counties look "high" simply because they're big. Normalize (per capita, per square mile, percent) or switch to size-based symbols. No classification method or ramp can rescue an unnormalized count map.

Step 4: Dense points — clustering vs. binning vs. heat map vs. dot density

TECHNIQUE	WHAT THE READER SEES	BEST WHEN	AVOID WHEN
Clustering	Countable circles with numbers; expands on zoom	Readers need to drill into individual features; interactive web maps	Exact positions matter at every scale; static exports

TECHNIQUE	WHAT THE READER SEES	BEST WHEN	AVOID WHEN
Binning	Fixed geometric cells colored/labeled by count	Consistent area-to-area comparison; very large layers (server-side aggregation helps performance — Compendium Chapter 10)	Bin edges would imply false boundaries in a small dataset
Heat map	Smooth density surface	First-glance "where is activity concentrated" at small scales	Readers need values or counts; few points (a dozen points make a fake hotspot); print output. Weight by a field if magnitude matters, or it shows point density only
Dot density	Scatter texture inside polygons	Totals across regions, showing composition with multiple fields	Zoomed in — dot placement is random within polygons and implies false precision

A heat map is a visual impression, not analysis. For statistically defensible hot spots, use the pattern-analysis tools in Compendium Chapter 17 (Statistical and Pattern Analysis).

Ten-second pre-publish check

- ☐ Counts → size; rates → color
- ☐ Choropleth normalized
- ☐ Break values defensible and rounded
- ☐ Ramp direction matches intuition on this basemap
- ☐ Survives a colorblind simulator and grayscale
- ☐ Checked at the actual zoom levels the audience will use

- ☐ Legend reads in under ten seconds without you explaining it

Working in Pro instead of Map Viewer? Same decisions apply; the mechanics live in Compendium Chapter 23 (Symbology and Labeling).

Troubleshooting Triage Card

Run the three checks in order. Most failures die at check 1 or 2. If all three pass, go to the referenced chapter for the full diagnosis tree.

Before anything else: the 60-second sweep

- ☐ Hard-refresh the browser (or try an incognito window) — cached apps lie.
- ☐ Confirm which account you are signed into, and which organization.
- ☐ Open the item page of the thing that is broken. Errors often show there first.
- ☐ Ask: did anything change since it last worked? (Sharing, schema, rename, delete.)

The triage table

SYMPTOM	FIRST THREE CHECKS, IN ORDER	FULL DIAGNOSIS
Layer invisible (in the list but nothing draws)	1. Layer visibility toggle on, including any parent group layer. 2. Visible range — zoom in and out past the scale thresholds. 3. A filter or definition query is excluding every feature.	Compendium Chapter 39 (Troubleshooting Encyclopedia); styling causes in Chapter 7 (Styling and Smart Mapping)
CSV points in the wrong place (ocean off Africa, wrong country)	1. A pile of points at 0,0 — "Null Island," in the ocean off West Africa — means blank or unparseable coordinate values, not a projection problem. 2. Latitude and longitude columns	Compendium Chapter 11 (Creating Data); coordinate theory in Chapter 3 (Coordinate Systems and Projections)

SYMPTOM	FIRST THREE CHECKS, IN ORDER	FULL DIAGNOSIS
	mapped correctly at import — not swapped — and values in decimal degrees, not DMS text or projected coordinates; watch commas used as decimal separators. 3. Correct coordinate system chosen at import time.	
Cannot edit (tools greyed out, edits rejected)	1. Editing is enabled in the layer's item page settings. 2. Your account has edit privileges and access to the layer. 3. You are on the source layer or an editable view — not a read-only view or a copy.	Compendium Chapter 13 (Editing Workflows); views in Chapter 10 (Hosted Feature Layers)
Sharing broken (others get errors or a blank map)	1. Every layer in the map is shared at the same level as the map itself — update sharing from the map item, which flags mismatched layers. 2. The recipient is in the group or organization you shared to. 3. The map includes subscriber or premium content that demands sign-in from public viewers.	Compendium Chapter 10 (Hosted Feature Layers); admin side in Chapter 34 (Administration)
Slow map	1. Feature count and geometry complexity of the heaviest layer — simplify or generalize. 2. Visible ranges — detailed layers should only draw when zoomed in. 3. Optimization settings on the layer's item page (drawing optimization, caching/tiles for static data).	Compendium Chapter 10 (Hosted Feature Layers, performance section)

SYMPTOM	FIRST THREE CHECKS, IN ORDER	FULL DIAGNOSIS
Misaligned data (consistent offset from basemap)	1. Datum mismatch — compare each dataset's geographic coordinate system. 2. Correct datum transformation applied in the map or during projection. 3. Define-vs-project confusion: was the wrong coordinate system <i>assigned</i> to data that actually needed <i>reprojecting</i>?	Compendium Chapter 3 (Coordinate Systems and Projections)
Sync conflict (offline edits missing or overwritten)	1. Read the sync error in Field Maps before assuming loss — retry the sync. 2. Did two editors change the same feature? The last edit synced wins by default. 3. Sync and offline settings on the layer item — were they changed mid-project?	Compendium Chapter 30 (Field Maps, Survey123, QuickCapture)
Geocode wrong (addresses land in the wrong spots)	1. Address fields mapped to the right locator fields — city or ZIP stuffed into the street column is the classic. 2. Country/region setting on the locator matches your data. 3. Match scores — review low-score candidates and rematch instead of accepting everything.	Compendium Chapter 11 (Creating Data, geocoding paths); credit cost in Chapter 2 (The ArcGIS Ecosystem)
Pop-up empty (or shows the wrong fields)	1. Pop-ups are enabled on that specific layer — not just the map. 2. Pop-up field list — fields hidden or removed in the pop-up configuration.	Compendium Chapter 9 (Pop-ups, Fields, and Labels)

SYMPTOM	FIRST THREE CHECKS, IN ORDER	FULL DIAGNOSIS
	3. Arcade expression errors — test each expression in the editor.	
Credits drain (balance dropping faster than expected)	1. Org credit usage report — which tool, which user, which item. 2. Feature storage — bloated attachments, abandoned hosted layers, duplicate copies. 3. Repeated geocoding jobs or scheduled analysis running unattended.	Compendium Chapter 2 (The ArcGIS Ecosystem); budgets and allocation in Chapter 34 (Administration)
Analysis fails (tool errors out or hangs)	1. Inputs: valid geometries, overlapping extents, feature counts within reason for the tool. 2. Your account has the privilege and credits for that tool. 3. The actual error text in the tool results or geoprocessing history — not just the red X.	Compendium Chapter 39 (Troubleshooting Encyclopedia); per-tool detail in Chapters 16–20 (Vol D)
Login/license tangle ("no license," wrong content, can't sign in)	1. Right destination: your organization's URL, not the generic sign-in or a different org. 2. Your user type and licenses cover the app you are opening — check with your admin. 3. In Pro: open the Portals page in the project's backstage settings — confirm the correct portal is set active and you are signed in.	Compendium Chapter 2 (The ArcGIS Ecosystem); assignment fixes in Chapter 34 (Administration)

Watch out: Never "fix" a broken hosted feature layer by deleting it and republishing. The new item gets a new ID, which silently breaks every map, app, dashboard, form, and field deployment pointing at the old one — and can destroy field-collected data that only lived in that layer. Fix layers in place (overwrite,

update definition, edit settings). See Compendium Chapter 10 (Hosted Feature Layers).

If all three checks pass: escalate with evidence

Collect these before filing a ticket or asking a colleague. It halves the round trips.

- ☐ Item ID of the broken layer, map, or app.
- ☐ Exact error text, plus a screenshot.
- ☐ Does it fail in an incognito window? For a different user?
- ☐ Does the layer's REST endpoint respond? (Open the service URL from the item page — Compendium Chapter 32, REST Services.)
- ☐ When did it last work, and what changed since?

Full symptom-by-symptom trees live in Compendium Chapter 39 (Troubleshooting Encyclopedia).