

The ArcGIS FAQ

ON THIS PAGE

- [The ArcGIS FAQ: Fifty Real Questions](#)
 - [The ArcGIS FAQ: Fifty Real Questions](#)

The ArcGIS FAQ: Fifty Real Questions

Fifty genuinely-asked questions with direct answers, grouped by theme, each pointing to the Compendium chapter that goes deeper.

The ArcGIS FAQ: Fifty Real Questions

Fifty questions people actually ask, answered directly, each with a pointer to the Compendium chapter that treats the topic in full. Read the answer here to decide; read the chapter to master.

Getting Started

Q: I'm brand new. Should I start with ArcGIS Online or ArcGIS Pro?

Start with ArcGIS Online. It runs in a browser, needs no installation, and teaches the concepts that transfer everywhere: layers, attributes, styling, pop-ups, sharing. Move to Pro when you hit a wall Online genuinely cannot handle — heavy data preparation, advanced geoprocessing, print cartography, or file formats you cannot upload. People

who begin in Pro usually spend their first weeks fighting the interface instead of learning GIS. *Depth: Chapter 2.*

Q: How is this different from Google Maps? I can already look up a map of anything.

Google Maps shows you a map someone else made. GIS lets you build maps from data — your data — and then interrogate it: what falls inside this boundary, what sits within a mile of that pipe, which parcels changed since last year. The map is an output; the database behind it is the point. Once you start thinking of layers as tables that happen to have geometry, the rest of ArcGIS falls into place. *Depth: Chapter 1.*

Q: What's the difference between vector and raster, and why does everyone keep asking?

Vector is discrete things: points, lines, and polygons, each with a row in an attribute table. Raster is a grid of cells, each holding a value — elevation, reflectance, temperature. The distinction matters because the tools, storage, styling, and analysis for each are almost entirely separate. Mapping assets, boundaries, or incidents puts you in vector territory. Working with elevation, imagery, or any continuous surface puts you in raster territory. *Depth: Chapter 1.*

Q: I added my data and it landed in the middle of the ocean. What happened?

Your data's coordinate system is missing or wrongly declared. Points stacked at a single spot in the Gulf of Guinea mean coordinates near (0,0) — usually numbers being read as latitude and longitude when they are actually something else, like a projected grid. The fix is to find out what coordinate system the data was really created in and define it correctly — define, not reproject. Reprojecting data that is mislabeled just moves the error. *Depth: Chapter 3.*

Q: Where do I get decent data to practice with?

The Living Atlas first — it is Esri's curated collection of authoritative, ready-to-use layers, available directly inside Map Viewer and Pro. After that, the open data portals run by your city, county, state, and national agencies. Judge every dataset the same way: who published it, when it was last updated, and whether the attribute table actually contains what the title promises. *Depth: Chapter 5.*

Accounts and Money

Q: Do I have to pay to use ArcGIS Online?

Not to learn it. A free public account — licensed for personal, noncommercial use — lets you make and share maps and learn most of the core concepts, but it cannot publish hosted feature layers, run the credit-consuming analysis tools, or work inside an organization's private sharing. If you are learning, use a public account or a trial. If you are doing real work for an organization, you need a seat in a paid organizational subscription — there is no sustainable free path for production work. *Depth: Chapter 2.*

Q: What are credits, and how do people accidentally burn through them?

Credits are ArcGIS Online's usage currency, consumed by things like data storage, geocoding, some analysis tools, and demographic enrichment. The classic accidents: batch-geocoding a large address file, running an analysis tool without reading the credit estimate it shows you, and storing large volumes of attachments in hosted layers. Always read the estimate dialog before you confirm an operation, and ask your administrator whether budgets are set on your account. *Depth: Chapter 2.*

Q: My admin asked what user type I need and I have no idea.

Answer with what you do, not with a product name. If you only view maps and dashboards, you need a viewer-level type. If you edit data in the field, you need a mobile or contributor-level type. If you create maps, layers, and apps, you need a creator-level type. If you need ArcGIS Pro, the professional-level types bundle it. Buying the highest type for everyone is the most common way organizations waste money. *Depth: Chapter 2.*

Q: Why can't I publish a hosted layer? The option is grayed out.

You are missing a privilege, not a skill. Publishing hosted feature layers is a privilege granted by your role, and roles are assigned separately from user types. Ask your administrator to check your role — either move you to one that includes publishing or create a custom role that adds it. Nothing you do inside Map Viewer will make the option appear. *Depth: Chapter 34.*

Q: We're paying for licenses nobody uses. How do we find out?

Your administrator can run member and item reports from the organization settings: last login dates, credit consumption per member, and item usage tell you almost everything. Downgrade dormant creator seats to viewer types, reclaim add-on licenses that were assigned once and forgotten, and make this a recurring review rather than a crisis audit. *Depth: Chapter 34.*

Data

Q: Can I just upload my Excel spreadsheet?

Yes, if it has location in it. Latitude and longitude columns publish directly with no geocoding step; street addresses have to be geocoded, which consumes credits in proportion to the row count. Either way the resulting hosted layer occupies storage. Clean the sheet first: a single header row, no merged cells, no formulas, no stray notes below the data. Most upload failures are spreadsheet-hygiene failures, not platform failures. *Depth: Chapter 11.*

Q: What is a hosted feature layer, actually?

It is your data living in a cloud database and served over the web. Unlike a file you attached to an item, a hosted feature layer can be queried, filtered, edited by many people at once, styled differently in every map that uses it, and fed to apps — one copy of the data, many products built on it. It is the fundamental unit of ArcGIS Online; nearly everything else in the platform builds on top of it. *Depth: Chapter 10.*

Q: Are shapefiles bad? People online keep telling me to stop using them.

Not bad — old. Field names are capped at ten characters, dates and nulls are handled poorly, there is a hard file-size ceiling, and a "shapefile" is really a loose bundle of files that break when separated. Use file geodatabases in Pro and hosted feature layers online as your working formats, and treat the shapefile as what it now is: an exchange format for moving data between systems. *Depth: Chapter 11.*

Q: What are domains, and should I bother setting them up?

Bother. A domain is a controlled pick-list or valid range attached to a field, so editors choose "Oak Street" from a list instead of typing "Oak St", "oak street", and "Oka Street" into three different rows. Every hour spent on domains before data collection

saves many hours of cleanup after. If a field's values could be enumerated, they should be. *Depth: Chapter 12.*

Q: Someone edited my layer and wrecked it. How do I protect my data?

Stop handing out the editable layer. Create a hosted feature layer view, give the audience the view, and grant editing only where it is genuinely needed — views can restrict editing, hide fields, and filter features without duplicating data. Turn on editor tracking so every change records who and when. The pattern is one authoritative layer, several purpose-built views. *Depth: Chapter 10.*

Q: How do I keep two copies of the same dataset in sync?

You stop having two copies. Maintain one authoritative hosted layer and serve every audience through views or maps that reference it — that is what the platform is designed for. If a second copy is unavoidable because an internal system of record exists, make the sync a scheduled script that overwrites or appends on a known cadence, never a human remembering to re-upload. Compendium Chapter 31 (Python) covers the scripting side. *Depth: Chapter 10.*

Maps and Styling

Q: My map looks amateurish and I can't articulate why.

It is almost always one of three things: too many competing colors, default symbology left everywhere, or a busy basemap fighting your data for attention. Switch to a muted basemap, decide on the single thing the map must communicate, and let one visual variable — color, size, or transparency — carry that message. Remove every layer that does not serve the point. Restraint reads as competence. *Depth: Chapter 4.*

Q: What is smart mapping actually doing when I pick a style?

Reading your field's data type and statistical distribution, then proposing sensible defaults: class breaks positioned around the data, a color ramp matched to the basemap, size ranges that stay legible. Treat the suggestion as a competent first draft, not a verdict — drag the handles to breakpoints that mean something in your domain, because the defaults know your numbers but not your subject. *Depth: Chapter 7.*

Q: How do I show only some of my features without making a whole new layer?

Filters. In Map Viewer, select the layer and open its filter pane from the settings toolbar, then build the expression. A filter belongs to that map only and never touches the underlying data, so different maps can filter the same layer differently. If you need a restriction the audience cannot remove, use a hosted feature layer view instead.

Depth: Chapter 6.

Q: Do I really need to learn Arcade?

If you make web maps other people use, yes — but far less of it than you fear. Arcade formats pop-up values, computes new values on the fly, and drives labels and symbology without touching your schema. An afternoon covers most of what real maps need. It is the highest-leverage small skill in web mapping. *Depth: Chapter 8.*

Q: My pop-ups show thirty fields with names like ADDR_ST_1. How do I fix them?

Configure the pop-up instead of accepting the default field dump. Set readable aliases on the fields, then build the pop-up from only the handful of values a reader actually needs — text with embedded values, or a short curated list. A pop-up is an answer to "what am I looking at," not a database export. *Depth: Chapter 9.*

Q: Why do my labels disappear when I zoom out?

Two mechanisms. First, labels and layers have visible ranges, and yours may be set to switch off beyond a scale. Second, the labeling engine silently drops labels that would overlap, and zooming out packs features together until most labels lose that fight. Check the visible range settings first, then accept that dense data cannot be fully labeled at small scales — label where labels are readable. *Depth: Chapter 9.*

Sharing and Collaboration

Q: I shared my map but my colleague sees an empty map or gets errors.

The web map and each layer inside it are separate items with separate sharing levels, and you shared the map without sharing the layers. Open each layer's item page and set its sharing at least as broad as the map's — ArcGIS Online usually offers to fix this for you when you share the map, and you should let it. This is the single most common sharing failure on the platform. *Depth: Chapter 10.*

Q: What's the difference between sharing to everyone, my organization, and a group?

Everyone means the public internet, no sign-in required. Organization means anyone with a login in your org. A group means exactly its members, which can include invited people from other orgs. Default to groups — they map to real teams and audiences — and reserve "everyone" for products you deliberately publish to the world, because public items get found. *Depth: Chapter 6.*

Q: Can I put a map on my own website?

Yes. The easiest path is building a simple app — an Instant App works well — and embedding it in an iframe; app templates are designed to live inside other pages. If you need full control over behavior and appearance, that is the Maps SDK for JavaScript. Either way, everything the page uses must be shared with everyone, or visitors will hit a login wall. *Depth: Chapter 26.*

Q: People opening my public link get asked to sign in. Why?

Something in the map is not actually public. Usually it is a layer still shared to the organization; occasionally it is subscriber or premium content, which always requires an organizational login no matter how the map is shared. Walk every layer's sharing level, and replace subscriber content with a public equivalent if the map must work for anonymous visitors. *Depth: Chapter 39.*

Q: Can people edit only the records they created and not each other's?

Yes — that is ownership-based access control. In the hosted feature layer's item page settings, enable editing and choose the option restricting editors to features they added. Pair it with editor tracking so ownership is recorded, and serve any read-everything audience through a separate view. This is the standard pattern for crowdsourced or multi-crew collection. *Depth: Chapter 10.*

Apps Without Code

Q: Dashboard, StoryMap, Instant Apps, Experience Builder — how do I pick?

Pick by what the audience does. Monitoring numbers and live status at a glance: Dashboard. A narrative someone scrolls through: StoryMap. One focused map task,

built in minutes: Instant Apps. A multi-page product combining maps, forms, and custom layout: Experience Builder. Start with the simplest tool that does the job and escalate only when you hit its edges — Experience Builder is the last resort, not the first. *Depth: Chapters 26–29.*

Q: Web AppBuilder is going away. What happens to my old apps?

You rebuild them in Experience Builder — plan for that, because there is no converter that carries everything across. Inventory the widgets each app actually uses, find their Experience Builder equivalents, and rebuild the highest-traffic apps first. Treat it as a chance to cut features nobody used rather than a faithful port. *Depth: Chapter 29.*

Q: Can my dashboard update in real time?

Effectively, yes. Dashboards read live hosted layers, and you can set a refresh interval on the layers so new records appear without anyone reloading the page. For "a crew member submitted a record and the count ticked up within a minute," standard refresh intervals are all you need; true second-by-second streaming requires heavier infrastructure most projects never need. *Depth: Chapter 28.*

Q: My app looks great on my monitor and broken on a phone.

Because you authored it on a monitor. Instant Apps handle responsive layout for you, which is a real argument for them. Experience Builder makes you responsible: it has separate layout modes per screen size, and an app only handles phones if you configured the phone layout. Test on a real phone before launch, every time, no exceptions. *Depth: Chapter 29.*

Fieldwork

Q: Field Maps or Survey123 — which one do I actually need?

Map-first versus form-first. If field workers navigate a map, find assets, and create or edit features in place, use Field Maps. If the job is answering a structured questionnaire — inspections, surveys, incident reports — and the map is incidental, use Survey123. The two can be linked for combined workflows, but start from where the worker's attention lives: the map or the form. *Depth: Chapter 30.*

Q: Can crews collect data where there's no cell signal?

Yes, and it works well when prepared properly. Enable offline use on the layers, define offline map areas, and have crews download them before leaving coverage; edits queue on the device and sync when connectivity returns. The failure mode is skipping the dry run — test the full download, collect, and sync round trip before the real field day.

Depth: Chapter 30.

Q: The GPS points from our phones are meters off. Is something broken?

No — that is what consumer phone GPS delivers. Expect accuracy of several meters, worse under tree canopy or between buildings. If your work genuinely needs better, pair the device with an external GNSS receiver and configure it in the app. If a few meters is acceptable — and for most inspection and observation work it is — spend the money elsewhere. *Depth: Chapter 30.*

Q: Where do the photos my crew takes actually end up?

As attachments stored on the individual feature, inside the hosted feature layer. Enable attachments on the layer before collection starts, and photos captured in Field Maps or Survey123 ride along with each record and appear in pop-ups. Be aware they count against your organization's storage, and a photo-heavy collection campaign adds up faster than the point geometry does. *Depth: Chapter 30.*

ArcGIS Pro

Q: Pro is painfully slow. Is it me or the machine?

Usually both, in a fixable way. Check where the data lives first: layers read over a network drive or VPN are the top cause of a sluggish Pro, and copying data locally often transforms performance. Then look at the display — dozens of layers, complex symbology, and dense labeling cost real time. Finally, the machine: Pro genuinely wants a dedicated GPU and generous RAM, and laptops without them will always feel heavy.

Depth: Chapter 39.

Q: What's the relationship between projects and geodatabases? Where are my actual files?

The project file stores your maps, layouts, and connections — not your data. Data lives in geodatabases and folders that the project points at, and every layer in a map is a

pointer to a source on disk. That is why moving or renaming source folders breaks maps with red exclamation marks. Keep a project and its default geodatabase together, and repair broken sources through the layer's properties. *Depth: Chapter 21.*

Q: There are hundreds of geoprocessing tools. How do I find the right one?

Search by the verb of what you want, in the geoprocessing tool search — "clip," "merge," "join" — and read the tool's short description before running anything. Then internalize the core dozen: buffer, clip, intersect, union, dissolve, merge, spatial join, select by attributes and location. Those cover the vast majority of real work; the long tail exists for specialists. *Depth: Chapter 22.*

Q: How do I make a proper PDF map for printing?

Layouts. Create a new layout from the Insert ribbon, choose the page size, add a map frame, then the furniture: scale bar, legend, north arrow, title, credits. Export it to PDF from the Share ribbon. If you need many maps from one template — one per district, one per parcel — that is a map series, which generates the whole set from a single layout. *Depth: Chapter 24.*

Q: Should I learn ModelBuilder or go straight to Python?

ModelBuilder first if code intimidates you: it chains tools visually, pays off the same afternoon, and exports to Python when you outgrow it. Go straight to Python if you already write any code, need loops over many datasets, or plan to schedule anything. The wrong answer is neither — anyone repeating the same multi-tool workflow by hand more than twice should automate it. *Depth: Chapter 25.*

Automation and Development

Q: How do I make my data update itself every night?

Write the update as a script using the ArcGIS API for Python — typically truncate-and-append or overwrite against the hosted layer — then schedule it. ArcGIS Notebooks can run on a schedule in the cloud, or run the script from a task scheduler on a machine you control. Log every run and alert on failure; a silent broken sync is worse than no sync. *Depth: Chapter 31.*

Q: What is this REST API everyone keeps mentioning?

Every service in ArcGIS is a web endpoint that returns JSON, and every client — Map Viewer, Field Maps, your dashboard — is just calling those endpoints. Learn to read a feature service URL, open it in a browser, and run a query against it, and half of the platform stops being magic. It is the single best investment for troubleshooting, because you can see exactly what the server returns. *Depth: Chapter 32.*

Q: Are arcpy and the ArcGIS API for Python the same thing?

No, and confusing them wastes weeks. arcpy is the desktop geoprocessing engine exposed to Python — it installs with Pro (and with ArcGIS Server), requires a license, and is for crunching data. The ArcGIS API for Python manages your web GIS — users, groups, items, hosted layers — is free to install, and runs anywhere Python runs. Desktop data work is arcpy; ArcGIS Online administration and content automation is the API. Mature workflows often use both. *Depth: Chapter 31.*

Q: Do I need to be a developer to build a custom web app?

For most needs, no — the configurable app builders go further than people expect, and you should exhaust them first because custom code is a permanent maintenance commitment. When you genuinely need behavior no builder offers, the Maps SDK for JavaScript is approachable: basic JavaScript plus the documentation samples gets a working map on a page quickly. *Depth: Chapter 33.*

Career and Learning

Q: Is GIS still a viable career, or is it dissolving into everything else?

Both statements are true, and the second is good news. Pure push-button technician roles are shrinking, while spatial skills embedded in planning, utilities, environment, public safety, logistics, and data science keep growing. Position yourself as someone who solves problems with spatial data and can automate the boring parts, not as an operator of one vendor's software. *Depth: Chapter 40.*

Q: Do I need a degree or a certificate to get hired?

A relevant degree helps most for government positions with rigid HR screens; certificates matter far less than people hope. What consistently works is demonstrable output — real projects a hiring manager can click through — plus fundamentals you can

explain out loud. If you must choose between a paid certificate and a month spent building portfolio projects, build the projects. *Depth: Chapter 40.*

Q: What should actually be in a GIS portfolio?

Two or three finished, public products built on real data, each with a short write-up of the decisions you made: a dashboard, a field collection workflow, a story map or analysis with a clear question and answer. Finished and explained beats numerous and half-done. The Compendium's worked projects exist precisely to be portfolio templates — do them with your own local data, not the sample data. *Depth: Chapter 36.*

Q: Do I have to learn to code?

To get started, no. To keep advancing, essentially yes. Python is the clearest line between people who operate the software and people who make systems run, and it shows up in a large share of GIS job postings above entry level. Start with Arcade because it is small and pays off immediately inside maps you already make, then move to Python for automation. *Depth: Chapter 31.*

Q: What do employers actually test in GIS interviews?

Fundamentals over button locations. Expect projections (explain why two layers misalign), table joins and relates, data quality thinking, and — most telling — whether you can turn a vague request into layers, an analysis, and a deliverable. Practice narrating one complete project out loud, from raw data to published product, and most interviews take care of themselves. *Depth: Chapter 1.*

Q: Where do I go when I'm stuck and the documentation doesn't help?

Esri Community forums, a web search containing the exact error text in quotes, and a disciplined isolation habit: does it fail in a fresh map, with a different layer, in a different browser, under a different account? Bisecting the problem identifies the culprit in half the cases before you ever post a question — and makes the question answerable when you do. *Depth: Chapter 39.*

