

The ArcGIS Learning Paths

ON THIS PAGE

- [The ArcGIS Learning Paths](#)
 - [The Analyst Path: 30 Days](#)
 - [The Field Operations Path: 21 Days](#)
 - [The Automation and Web Builder Path: 30 Days](#)

The ArcGIS Learning Paths

Three guided routes through the Compendium and Workbook: the Analyst Path (30 days), the Field Operations Path (21 days), and the Automation and Web Builder Path (30 days).

The Analyst Path: 30 Days

A 30-day plan for becoming the person your team hands data to. Not a GIS developer, not a server administrator — the analyst: the one who can find data, judge it, structure it, map it honestly, run the analysis, and ship something people use to make a decision. It leans hard on Volumes A through D, with the workbook doing the hands-on half. Plan on an hour a day for three weeks, closer to two during the capstone.

Who this is for

You have been informally nominated. Maybe your manager said "you're good with spreadsheets, can you make a map of this?" Maybe you inherited a pile of layers from someone who left. Either way, your team now expects you to answer spatial questions, and you want that expectation to be safe.

This path is for someone doing real work in ArcGIS Online: evaluating other people's data, publishing and maintaining your team's data, and running the core analysis workflows. It is not the path for field data collection (Compendium Chapter 30), app building (Volume F), or automation (Volume G). The center of gravity is data and analysis in the browser, with ArcGIS Pro appearing only where the browser runs out.

Prerequisites (honest)

- **The beginner field guide, finished.** This plan assumes you can already sign in, open Map Viewer, add a layer, and save a map without instructions. If any of that is shaky, do the field guide first; this plan does not re-teach it.
- **An ArcGIS account — ideally organizational.** The first two weeks mostly run on a free public account. Week two's publishing work and week four's analysis tools require an organizational account, and some operations consume credits. Read Compendium Chapter 2 (The ArcGIS Ecosystem) on day 5 and find out who your administrator is before you need them.
- **A real question from your team.** The capstone is built around a genuine problem in your domain. Start collecting candidates from day one.
- **An hour a day, honestly given.** The plan has slack days built in, but the exercises build on each other's outputs, so it does not survive cramming.
- **No ArcGIS Pro required.** If you have it, some workbook stretch goals open up. If not, nothing below blocks.

The plan

Reading references are Compendium chapters. Exercises reference the companion workbook by volume. Keep every output — the capstone and milestone checks reuse them.

Week 1 — Learn to distrust maps (Volume A)

DAY	READING	EXERCISE	WHAT YOU PRODUCE
1	Chapter 1 (How GIS Thinks)	Workbook Vol A: the representation exercise (five real-world things, vector or raster)	A graded prediction sheet with one sentence per miss
2	Chapter 3 (Coordinate Systems and Projections)	Workbook Vol A: the Web Mercator measurement exercise	Measured area and distance distortion, plus a two-sentence explanation
3	Chapter 5 (Finding Data)	Workbook Vol A: interrogate a stranger's layer	A written five-line verdict on a layer you did not make
4	Chapter 4 (Cartographic Design)	Workbook Vol A: four stories from one unchanged dataset	Four screenshots, four headlines, and a defense of the one you would publish
5	Chapter 2 (The ArcGIS Ecosystem)	No workbook — instead, inventory your own organization: account type, licenses, who administers it, whether credits are budgeted	A one-page "our ArcGIS setup" note
6	Chapter 6 (Map Viewer Complete Reference) — skim as reference, read the sections you have not touched	Workbook Vol A: spot and repair a bad map	A before/after map pair and the list of what was wrong
7	Reread your week's notes; skim the opening of Chapter 39	Redo whichever exercise felt weakest	A short week-one summary; run the

DAY	READING	EXERCISE	WHAT YOU PRODUCE
	(Troubleshooting Encyclopedia) so you know it exists		Milestone 1 self-check

Week 2 — Build maps worth sharing (Volume B)

DAY	READING	EXERCISE	WHAT YOU PRODUCE
8	Chapter 7 (Styling and Smart Mapping)	Workbook Vol B: the styling exercise — restyle a layer with deliberate classification and color choices	A styled map where you can defend every choice
9	Chapter 8 (Arcade from Zero to Fluent), first half	Workbook Vol B: the Arcade exercise, first part	Two or three working expressions driving symbology or labels
10	Chapter 8, second half	Extend yesterday's work: a calculated value in a pop-up	A pop-up that shows something not stored in any field
11	Chapter 9 (Pop-ups, Fields, and Labels)	Workbook Vol B: the pop-up and labeling exercise	A layer with clean aliases, a designed pop-up, and scale-aware labels
12	Chapter 10 (Hosted Feature Layers)	Workbook Vol B: the publishing exercise (organizational account needed)	A hosted feature layer you published, plus a view with different sharing than its parent

DAY	READING	EXERCISE	WHAT YOU PRODUCE
13	Revisit Chapters 4 and 5 with two weeks of hindsight	Assemble everything: one complete web map on a topic you care about — found data, honest styling, working pop-ups, shared to a group	Portfolio map number one
14	Catch-up day: finish anything dangling	Rework the portfolio map after a break from it	Run the Milestone 2 self-check

Week 3 — Own a dataset end to end (Volume C)

DAY	READING	EXERCISE	WHAT YOU PRODUCE
15	Chapter 11 (Creating Data)	Workbook Vol C, Exercise 1: build a CSV from nothing, import it by coordinates and by address	Two imported layers and a note on where each path fails
16	Chapter 12 (Schema Design)	Workbook Vol C, Exercise 2: design the schema with domains	A data dictionary another person could work from
17	Chapter 13 (Editing Workflows in Web and Pro)	Workbook Vol C, Exercise 3: digitize ten features with snapping	A hand-digitized layer with genuinely coincident edges
18	Chapter 14 (Data Quality), first pass	Workbook Vol C, Exercise 4: deliberately damage your data	A seeded-defect dataset and the private list of what you broke
19	Chapter 14 again, focusing on the QA loop	Workbook Vol C, Exercise 5: run the review loop and catch the damage	A QA report and a repaired dataset — score yourself on how many seeded defects you caught

DAY	READING	EXERCISE	WHAT YOU PRODUCE
20	Revisit Chapter 10; skim Chapter 15 (Imagery and Rasters) so raster data stops being a mystery	Publish the tree inventory (or your equivalent) as a hosted feature layer with the schema, domains, and an editing view applied	A live, editable, documented dataset
21	Rest of Chapter 39 as browsing material	Hand your data dictionary to a colleague and have them enter one record while you stay silent	Their questions, written down — each one is a schema defect

Week 4 — Analysis and the capstone (Volume D, then Volume H)

DAY	READING	EXERCISE	WHAT YOU PRODUCE
22	Chapter 16 (Proximity and Overlay)	Workbook Vol D: the buffer and overlay exercise	An analysis result layer answering a written "what is near what" question
23	Chapter 17 (Statistical and Pattern Analysis)	Workbook Vol D: the pattern exercise	A cluster or hot spot map plus a paragraph on what it does not prove
24	Chapter 18 (Network Analysis) or Chapter 19 (Terrain and Raster Analysis) — pick the one your domain actually uses	The matching Workbook Vol D exercise	A drive-time/service-area result or a terrain-derived result
25	Skim Chapter 20 (Space-Time Analysis); browse Chapter 38 (Cookbook) for recipes near your domain	Rerun one full analysis from scratch, unaided, on fresh data	A one-page analysis memo; run the Milestone 3 self-check

DAY	READING	EXERCISE	WHAT YOU PRODUCE
26	Chapter 36 (Worked Project — Dashboard), read end to end	Scope your capstone using the spec below	A one-page capstone spec, reviewed against the checklist
27	—	Capstone: acquire and clean data, design the schema, publish	Documented hosted layers
28	—	Capstone: run the analysis	Result layers plus running methods notes
29	Chapter 28 (Dashboards) or Chapter 26 (Instant Apps) as needed for your product choice	Capstone: assemble the deliverable	A draft product someone else can open
30	Chapter 39 for whatever broke	Capstone: review against your own spec, fix, share	The finished capstone and methods note

A cost warning for week four: several analysis tools and the batch geocoder consume credits on organizational accounts. Before running anything unfamiliar, check what it costs — Compendium Chapter 2 explains the credit model, and your day-5 inventory note should say who to ask.

Milestones

Milestone 1 — end of day 7: you read maps critically. Self-check: open any public web map you have never seen. Within ten minutes, state what data model each layer uses, what the projection is doing to areas or distances, whether the classification flatters or distorts the data, and whether you would trust the underlying layers — citing the item pages, not your gut. If you cannot get through this in ten minutes, repeat the Vol A exercises on new layers before moving on.

Milestone 2 — end of day 14: you build defensible web maps. Self-check: starting from a Living Atlas layer you have never used, produce a styled, labeled, pop-up-

configured web map with at least one working Arcade expression, shared to a group, in under an hour, without opening the Compendium. Speed matters here because this is the task your team will interrupt you with.

Milestone 3 — end of day 25: you answer questions with analysis. Self-check: take a "where should we / which ones are / how many within" question from your domain. Choose the tool, state before running it what it will cost in credits, run it, and write one paragraph of findings that includes at least one honest limitation. If you cannot name a limitation, you do not understand the tool yet — reread the relevant Volume D chapter's discussion of what the method assumes.

Capstone spec

Days 26 through 30. Build a Chapter 36-style project — raw data to published product — in your own domain, on a question someone on your team actually has. Not a tutorial dataset. If no question exists, ask three coworkers what they wish they could see on a map.

Required deliverables:

1. **A published, documented dataset.** At least one hosted feature layer you created or substantially cleaned, with designed fields, at least one coded value domain, a filled-in item page, and correct sharing. A view layer with different permissions if anyone besides you should see it.
2. **At least two analysis operations** from Volume D whose outputs feed the final product — for example a proximity result and a summary or pattern result. Each must answer a stated question, not decorate the map.
3. **A shareable product:** a web map at minimum; a dashboard or Instant App if the audience needs numbers or filtering. It must be usable by someone who was not in your head — test it on that colleague from day 21.
4. **A one-page methods note:** data sources with their vintage and known gaps, every processing step, the analysis settings you chose and why, and what the result does not show. This note is what separates an analyst from someone who makes maps.

Acceptance criteria: a stranger can open the product and answer the original question in under two minutes; every layer traces back to a documented source; you can rerun

the whole workflow from the methods note alone; and nothing in the styling overstates what the analysis found.

What you can credibly claim afterward

You can honestly say: you evaluate spatial data before trusting it; you publish and maintain hosted feature layers with real schemas, domains, and views; you build clean, honest web maps with Arcade-driven pop-ups and labels; you run and interpret core proximity, overlay, and pattern analysis; and you document your work so it survives your vacation. That is a working definition of "the GIS person on the team," and you can defend it with the portfolio map, the published inventory, and the capstone.

You cannot yet claim: ArcGIS Pro fluency beyond the basics (Volume E is its own climb), Python automation (Chapter 31), administration of other people's accounts (Chapter 34), or statistical authority — Chapter 17 makes you a careful user of pattern tools, not a statistician. Say "I can find out" for those, then go find out. The Compendium's other paths pick up where this one stops.

The Field Operations Path: 21 Days

You have a crew, a job that currently runs on paper or a shared spreadsheet, and a mandate to move it to mobile data collection. This path takes you from that starting point to a piloted, go-live-ready field operation in 21 days: one week of data foundations, one week building the layers and forms, and one week standing up the operation, running a pilot, and deciding — with evidence — whether to flip the switch.

Who this is for

You are the person standing up mobile collection for other people. Maybe you are the office GIS person who got volunteered, the operations lead who knows the work but not the software, or the consultant setting this up for a client. The defining trait: your success is measured by whether a crew that is not you can collect clean data on day one, not by whether you personally understand every setting.

This path is not for someone building a one-off survey for themselves (skim Compendium Chapter 30 (Field Maps, Survey123, QuickCapture) and go), and it is not a general GIS education — it deliberately skips analysis, cartography, and most of ArcGIS Pro.

Prerequisites (honest)

- **An ArcGIS Online organizational account with publishing privileges.** Not a free public account. You need to publish hosted feature layers and share them to groups. If you do not know what account type you have, day 1 sorts that out — but if your organization has no ArcGIS subscription at all, stop and resolve that first; nothing in this plan works without it.
- **Accounts for your crew, or a plan to get them.** Field workers need their own sign-ins with an appropriate user type. Budget conversations about licensing take longer than any technical step in this plan. Start that conversation on day 1, not day 18.
- **At least one field device you can test on** — a phone or tablet running Field Maps, ideally the same model your crew will carry. Two is better.
- **Access to whoever does the work now.** You will need roughly an hour of a crew member's time in week 1 and a half day for the pilot in week 3. If you cannot get that, this plan produces a system nobody will use.
- **Time:** roughly one to two hours a day. The tables assume that. Days are sequential but weeks have slack built in — a missed day is recoverable, a missed week is not.

You do not need: prior GIS experience, ArcGIS Pro (everything here runs in the browser and on the device), or any coding.

The plan

Reading references are Compendium chapters. Exercises reference the companion workbook by volume — do the exercises attached to the chapters you read that day, then apply the same technique to your own operation. "What you produce" is the real artifact for your deployment, not the workbook output.

Week 1 — Foundations sprint

The goal this week is to understand the platform's moving parts and design your data before you build anything permanent. Everything you make this week is disposable except the schema document.

DAY	READING	EXERCISE	WHAT YOU PRODUCE
1	Ch 2 (The ArcGIS Ecosystem)	Workbook Vol A: accounts and sharing exercises	A one-page inventory: your account type, your crew's account situation, who administers the org, and what still needs purchasing or assigning
2	Ch 11 (Creating Data)	Workbook Vol C: create a layer three different ways	Two or three throwaway test layers; a note on which creation path fits your operation
3	Ch 12 (Schema Design), first half — fields and types	Workbook Vol C: field-type exercises	Schema draft v1: every field your crew will fill in, with type, on paper or in a doc
4	Ch 12, second half — domains, defaults, relationships	Workbook Vol C: domain exercises	Schema draft v2: every categorical field converted to a coded-value domain; sensible defaults chosen
5	Ch 13 (Editing Workflows in Web and Pro)	Workbook Vol C: web editing exercises	Notes from editing a test layer in Map Viewer yourself — you cannot support editors if you have never edited
6	Ch 14 (Data Quality)	Workbook Vol C: validation exercises	A one-page QA plan: which fields are required, what values are impossible, who reviews incoming records and when
7	Catch-up; re-skim Ch 12	None — talk to a crew member instead	A written narrative of the current field workflow, start to finish, in the crew's own words

Day 7 matters more than it looks. The most common failure in field deployments is a form that models what the office imagines, not what the crew does. Get the narrative before you build the form.

Week 2 — Forms and layers

Now you build the real thing: the production hosted layer, its views, and the Field Maps form.

DAY	READING	EXERCISE	WHAT YOU PRODUCE
8	Ch 10 (Hosted Feature Layers)	Workbook Vol B: publishing exercises	Your production feature layer, published with the week-1 schema, editing enabled, sync-capable
9	Ch 10, views and performance sections	Workbook Vol B: view exercises	Two hosted feature layer views: an editable view shared to the crew's group, and a read-only view for everyone else. Field apps and dashboards point at views, never the parent layer
10	Ch 30 (Field Maps, Survey123, QuickCapture), Field Maps sections	Workbook Vol F: form-builder exercises	Form v1 built in Field Maps Designer (<code>app launcher > Field Maps Designer > your map > Forms</code>): field order matching the physical workflow, domains as choice lists, required fields marked
11	Ch 30, offline sections	Workbook Vol F: offline exercises	Offline enabled on the map, with a map area covering your work zone; a note on basemap size and download time on your actual device
12	Ch 9 (Pop-ups, Fields, and Labels), skim for field display	Workbook Vol F: conditional visibility exercises	Form v2: conditional logic so crews only see fields relevant to the record type; clean display names replacing raw field names

DAY	READING	EXERCISE	WHAT YOU PRODUCE
13	None — device day	None	A device test log: install Field Maps on two devices, collect a handful of real-shaped records, go offline (airplane mode), collect more, sync, verify everything arrived intact
14	Re-skim Ch 14	None	A revision list from a 30-minute dry run with one friendly crew member watching over your shoulder — every hesitation they show is a form defect

Week 3 — Operation week

The system exists; now build the office-side view of it, rehearse against a known-good reference, pilot with the real crew, and decide about go-live.

DAY	READING	EXERCISE	WHAT YOU PRODUCE
15	Ch 28 (Dashboards)	Workbook Vol F: dashboard exercises	An operations dashboard on the read-only view: records collected today, a map of latest records, a count by status, a list sorted by newest
16	Ch 37 (Worked Project — Field Collection), end to end	None — this chapter is the exercise	A gap list: everything the worked project does that your setup does not, marked keep/skip with a reason
17	Ch 34 (Administration), sharing and groups sections only	Workbook Vol G: group exercises	Hardened sharing: crew group has exactly the editable view and map; nothing shared publicly by accident; a second admin who is not you
18	None — pilot prep	None	The pilot kit: a one-page crew cheat sheet (sign in, open map, collect, sync), devices charged and

DAY	READING	EXERCISE	WHAT YOU PRODUCE
			provisioned, test records deleted from the production layer
19	None — pilot day	None	Pilot data plus a timestamped issue log (protocol below)
20	Re-skim Ch 12 and Ch 10 before touching schema	None	Triaged fixes applied; a change log of exactly what you altered and why
21	None	None	The completed go-live checklist and a written go/no-go decision with a date

Pilot-day protocol (day 19)

Run the pilot as a real half-day of work, not a demo. One rule: **the old process runs in parallel**. The crew collects on paper or spreadsheet as usual *and* in Field Maps. That gives you a ground-truth comparison and means a total pilot failure costs nothing.

1. Brief the crew for ten minutes with the cheat sheet. Do not train for an hour; the form should not need it.
2. Send them out. You stay reachable but do not hover.
3. Watch the dashboard as records arrive. Note sync gaps, missing required fields, wrong domain choices.
4. Debrief the same day, while memory is fresh: what was slower than paper, what was confusing, what they invented workarounds for.
5. That evening, compare digital records to the parallel paper records field by field. Every mismatch is a finding.

Classify findings into three buckets: form fixes (reorder, rename, add a choice — cheap, do them day 20), schema fixes (new field, changed type — do them day 20 with care, since data now exists; see Compendium Chapter 12 on what changes safely after publish), and process fixes (training, cheat sheet, who syncs when — document, do not build).

Go-live checklist (day 21)

- Every crew member has signed in on their own device and completed one sync.
- Offline map area downloads on the crew's actual data plan and device in acceptable time.
- Required fields and domains stop the bad records you predicted in the week-1 QA plan — test by trying to submit garbage.
- Dashboard reflects a new record within a minute or two of sync.
- Someone other than you can answer "my sync failed, now what" (write the answer down; Compendium Chapter 39 (Troubleshooting Encyclopedia) covers the common failures).
- A rollback trigger is written and agreed: the specific condition under which you revert to the old process (for example, a defined share of records needing office correction, or repeated sync loss).
- The old process stays available, unannounced-deprecated, for the first weeks. Rollback is only cheap if the fallback still exists.
- You export a snapshot of the layer (item page > Export Data) before go-live and on a schedule after — your undo button for bad edits at scale.

Milestones

Milestone 1 — end of day 7: the design is done before the build starts. You have a schema with domains, a QA plan, and a workflow narrative from the crew. *Self-check:* explain every field in your schema out loud in one sentence each, including who fills it in and what happens if it is wrong. Any field you cannot justify gets cut. Any free-text field that could be a domain gets converted — free text in the field is data cleaning in the office, forever.

Milestone 2 — end of day 13: a record survives the full round trip. A record collected offline on a real device syncs into the production layer with every attribute intact. *Self-check:* hand the device to someone else, in airplane mode, with no instructions beyond the cheat sheet. If they cannot create and later sync a valid record, the failure is the form's, not theirs — fix it before day 14.

Milestone 3 — end of day 20: the pilot happened and the findings are closed. Real crew, real work, parallel paper trail, triaged issue log, fixes applied and logged. *Self-check:* count the pilot records that needed office correction. If it is more than a small fraction, do not go live — extend the pilot a week rather than launch a system the crew will learn to distrust. Distrust is the true rollback risk: crews quietly return to paper and you find out a month later.

Capstone spec

The capstone is the **go-live package** — everything a stranger would need to run or audit your deployment:

1. **Production hosted feature layer** with domains on all categorical fields, plus separate editable (crew) and read-only (stakeholder) views.
2. **A Field Maps map** with the form (logical order, conditional visibility, required fields) and offline enabled with a sized map area.
3. **An operations dashboard** on the read-only view showing at minimum: today's count, newest records on a map, and a status breakdown.
4. **The paper trail:** crew cheat sheet, QA plan, pilot issue log with resolutions, change log, rollback trigger, and the go/no-go decision memo.
5. **Evidence:** the pilot dataset itself, plus the paper-versus-digital comparison.

Done means: a crew member can collect offline and sync unaided; garbage input is rejected at the form; the dashboard shows work as it lands; and you can state the rollback condition from memory.

What you can credibly claim afterward

You can claim you have designed, built, piloted, and launched a mobile data collection operation on ArcGIS: schema and domain design, hosted feature layers with editable and read-only views, Field Maps forms with offline sync, an operations dashboard, and a piloted rollout with a rollback plan. That is a legitimate line on a resume and a defensible answer in an interview, with the pilot report as proof.

You cannot yet claim: Survey123 or QuickCapture experience beyond awareness (Compendium Chapter 30 covers when each beats Field Maps), spatial analysis of the

data you now collect (Volume D is the follow-on), or ArcGIS Enterprise field deployments (Compendium Chapter 35). The natural next step: after a month of live data, take the collected layer into Compendium Chapter 16 (Proximity and Overlay) and Chapter 17 (Statistical and Pattern Analysis) and make the data earn its keep.

The Automation and Web Builder Path: 30 Days

This path takes you from "comfortable with computers, new to ArcGIS" to someone who can script the platform, call its services directly, and stand up a small web page that consumes them. It is not a cartography or analysis path — you will make maps only far enough to understand what your code is manipulating. The destination is automation: scheduled data refreshes, REST calls you wrote yourself, and a working grasp of where Python, ModelBuilder, and JavaScript each belong.

Who this is for

Take this path if at least two of these describe you:

- You already write code in some language, or you have built spreadsheets complex enough that you wish they were code.
- Your job (or the job you want) involves keeping GIS data current, wiring ArcGIS into other systems, or building lightweight web tools on top of it.
- You get impatient clicking through the same dialog five times and immediately wonder how to loop it.
- You have inherited someone's scripts or models and need to maintain them.

Skip this path if you have never touched ArcGIS and your real goal is analysis or map production. This path deliberately skips most of Volumes C, D, and E outside the automation chapters, and it will not teach you cartography or spatial statistics.

Prerequisites (honest)

Be honest with yourself here, because week 3 onward assumes all of this:

- **Basic programming literacy.** You can read a `for` loop, a function definition, and a dictionary/object literal in Python or JavaScript without panic. If you have never programmed at all, spend two weeks on a general Python introduction first — this path teaches ArcGIS-flavored Python, not Python itself.
- **Comfort with a text editor and files.** You know what a file path is, can unzip an archive, and can tell a CSV from a JSON file.
- **An ArcGIS Online account with publishing privileges.** You will publish hosted feature layers repeatedly. A free public account is not enough; you need an organizational account (a trial works) with a role that allows publishing. Compendium Chapter 2 (The ArcGIS Ecosystem) explains account types — read it before day 1 if unsure.
- **ArcGIS Pro installed for weeks 3 and 4.** Weeks 1 and 2 are browser-only. Without Pro you can still do most of the Python week via the ArcGIS API for Python in a notebook, but the ModelBuilder and geoprocessing days become reading-only and your capstone options narrow.
- **About an hour a day, five days a week.** Roughly 60–90 minutes per day for 4 weeks. Days 6 and 7 are catch-up buffer, not extra content.

You do *not* need prior GIS coursework, a math background, or JavaScript framework experience. The web-building week uses plain HTML and script tags.

The plan

Exercises reference the companion workbook by volume; readings are Compendium chapters. Skim what you already know, but do every exercise — this path is judged by what you produce.

Week 1 — Platform fluency (Volume B)

You cannot automate a platform you do not understand by hand. This week you do everything through the browser, deliberately, so later scripts have meaning.

DAY	READING	EXERCISE	WHAT YOU PRODUCE
1	Ch 6 (Map Viewer Complete)	Workbook Vol B: build a web map from two Living Atlas layers; set	A saved web map with sensible defaults

DAY	READING	EXERCISE	WHAT YOU PRODUCE
	Reference)	visibility ranges and bookmarks	
2	Ch 7 (Styling and Smart Mapping)	Workbook Vol B: restyle the map three ways (single symbol, class breaks, unique values)	One map, three deliberate styles
3	Ch 8 (Arcade from Zero to Fluent), first half	Workbook Vol B: write three Arcade expressions — display field, conditional symbol, formatted pop-up value	Three expressions you can explain line by line
4	Ch 9 (Pop-ups, Fields, and Labels)	Workbook Vol B: configure a pop-up with field formatting, one Arcade element, and an image	A demo-worthy pop-up
5	Ch 10 (Hosted Feature Layers)	Workbook Vol B: publish a CSV as a hosted feature layer; create a filtered view with reduced fields	A hosted layer, a view, and notes on what a view actually is

Day 5 matters most: hosted feature layers and views are the substrate everything else here sits on. If the difference between a layer, an item, and a service is fuzzy at week's end, reread Chapter 10 before moving on.

Week 2 — Services literacy (Chapters 32 and 10)

The week most GIS training skips, and the one that separates builders from clickers. Every hosted layer is a REST service; this week you bypass the ArcGIS interfaces and talk to the services directly.

DAY	READING	EXERCISE	WHAT YOU PRODUCE
1	Ch 32 (REST Services), first half	Workbook Vol G: find your week-1 layer's REST endpoint; browse the service directory; identify layer id, fields, capabilities	A written map of one service's REST anatomy

DAY	READING	EXERCISE	WHAT YOU PRODUCE
2	Ch 32, second half	Workbook Vol G: run browser <code>query</code> operations – where clause, attribute-only, geometry filter, JSON output	Five saved query URLs, each with a one-line note
3	Ch 32 + Ch 10 (performance sections)	Workbook Vol G: inspect editing capabilities in item settings; enable editing on a view without touching the parent; verify with a query	Notes on where capabilities live, plus proof you changed one safely
4	Ch 10 (publishing and updating workflows)	Workbook Vol G: overwrite or append data via the item page; watch what happens to the service, views, and dependent maps	A cause-and-effect log: "I did X, downstream Y happened"
5	Review Ch 32	Workbook Vol G: generate a token, call a query with it, watch it fail after expiry	Working notes on how authentication actually behaves

By Friday you should be able to look at any ArcGIS URL and predict what is behind it. That instinct is what week 3's code builds on.

Week 3 — Python, both flavors (Chapters 31 and 25)

ArcGIS Python comes in two flavors: ArcPy (desktop geoprocessing, runs where Pro runs) and the ArcGIS API for Python (talks to portals and services over the web, runs anywhere). Confusing them is the most common beginner error. This week you use both, plus ModelBuilder as a deliberate contrast so you know when *not* to write code.

DAY	READING	EXERCISE	WHAT YOU PRODUCE
1	Ch 31 (Python for ArcGIS), the "which	Workbook Vol G: in Pro's Python window, describe a	Your first ArcPy session, saved and commented

DAY	READING	EXERCISE	WHAT YOU PRODUCE
	Python where" sections	layer, list fields, run one geoprocessing tool	
2	Ch 31, ArcPy sections	Workbook Vol G: write a standalone ArcPy script looping over a geodatabase's feature classes, reporting counts and geometry types	A runnable <code>.py</code> file with a loop, not a transcript
3	Ch 31, ArcGIS API for Python sections	Workbook Vol G: from a notebook, connect to your org, search for your week-1 items, query your hosted layer	A notebook that touches nothing on disk and everything in the cloud
4	Ch 25 (ModelBuilder)	Workbook Vol E: rebuild day 2's logic as a model; export the model to Python and read what it generated	The same workflow twice, plus three sentences on which you would maintain and why
5	Ch 22 (Geoprocessing in Depth), environments sections	Workbook Vol G: parameterize day 2's script, then run it from a command line outside Pro	A script a scheduler could call with no human present

The day-4 contrast is the point of the week: ModelBuilder is superb for visual, linear, shareable workflows and poor at branching, error handling, and web interaction. Form your own opinion and write it down.

Week 4 — Build week: web output and hardened geoprocessing (Chapters 33 and 22)

Now you build outward in two directions: a web page that consumes a service, and geoprocessing robust enough to run unattended.

DAY	READING	EXERCISE	WHAT YOU PRODUCE
1	Ch 33 (Maps SDK for JavaScript Primer), first half	Workbook Vol G: one HTML file that loads the SDK and shows a basemap centered on your area	A page of code that opens in a browser
2	Ch 33, second half	Workbook Vol G: add your hosted feature layer by URL; add a legend and a pop-up	The same page, now showing <i>your</i> data
3	Ch 33 + Ch 32	Workbook Vol G: add one query-driven element — a count, a filter, or a dropdown that changes a definition expression	A page that queries the service and reacts to the result
4	Ch 22 (Geoprocessing in Depth)	Workbook Vol G: harden your week-3 script — try/except around fragile steps, log to a file, exit nonzero on failure	A script you would trust at 3 a.m.
5	Skim Ch 34 (Administration), automation sections	Capstone assembly (see spec below)	Capstone draft

Days 6–7 of week 4: finish and self-review the capstone.

Milestones

Milestone 1 — end of week 2: Services literacy. You can take any hosted feature layer and, without opening Map Viewer, answer: what fields does it have, how many features match a given where clause, and is it editable? *Self-check:* pick a public Living Atlas feature layer you have never seen. Using only browser tabs and the REST endpoint, produce its field list and a filtered feature count in under ten minutes. If you reach for Map Viewer, you are not there yet.

Milestone 2 — end of week 3: Two-flavor fluency. For a given task, you can state whether ArcPy, the ArcGIS API for Python, or ModelBuilder is the right tool, and you

have working code in both Python flavors. *Self-check*: explain in writing why your day-3 notebook could run on a machine without ArcGIS Pro installed and why your day-2 script could not. If the explanation is fuzzy, reread the environment sections of Compendium Chapter 31.

Milestone 3 — end of week 4, day 3: A page that consumes a service. You have a self-contained HTML file that renders your own hosted layer and runs at least one live query against it. *Self-check*: open the file on a different machine. If it works with no setup, you built it right. If it only works on your machine, find the hardcoded assumption.

Capstone spec

Two deliverables, designed as a pair. Budget the last three days.

Part 1 — An automated weekly data-refresh design. A short design document plus a working script for its core step. Choose a dataset that genuinely changes — a public CSV feed or open data portal export with a real update cadence. The document must specify:

- Source: where new data comes from, in what format.
- Transform: what cleaning or field mapping happens, and in which flavor of Python.
- Load: overwrite, append, or truncate-and-load — and why (Compendium Chapter 10 covers the trade-offs).
- Schedule: what triggers the run (OS scheduler, hosted notebook scheduling, or similar) — described, not implemented.
- Failure handling: what happens when the source is down, the schema changed, or the write fails; who finds out and how.
- Downstream contract: which maps, views, and your Part 2 page depend on the layer, and what the refresh must never break.

The script must perform the transform-and-load step end to end at least once against your real hosted layer. Scheduling and alerting may stay on paper; the data movement may not.

Part 2 — A small hosted web page consuming the service. The week-4 page, finished: your refreshed hosted layer rendered with the Maps SDK for JavaScript, one interactive

query element, a legend, and a "data as of" indicator read from the service or its data rather than hardcoded. Host it anywhere a browser can reach — a simple static host is fine. The point is that a URL exists.

The pair must connect. Run your refresh script, reload your page, and the page reflects the new data with no edits to the page. That demonstration — write path and read path meeting at one hosted feature layer — is the whole path in one motion.

Self-review: give the design doc to the most skeptical technical person you know and ask them to attack the failure-handling section. If it survives, ship it.

What you can credibly claim afterward

Say these in an interview and be able to back each one:

- "I understand the ArcGIS REST services architecture and can query and inspect hosted feature layers at the endpoint level."
- "I write automation in both ArcPy and the ArcGIS API for Python, and I know which one a given task calls for."
- "I can evaluate whether a workflow belongs in ModelBuilder or a script, and defend the choice."
- "I have designed and partially implemented an automated data-refresh pipeline against a hosted feature layer, including its failure modes."
- "I have built and hosted a web page with the Maps SDK for JavaScript that consumes a live feature service."

Do **not** claim production web development skills (one page with script tags is a primer, not a portfolio), ArcGIS Enterprise administration (Compendium Chapter 35 territory you only skimmed), or spatial analysis competence (Volume D, skipped entirely). If your next goal is any of those, this path gave you the substrate — services literacy and a Python footing — to take them on deliberately rather than from zero.

