1.1.2 — How to Use This Course Page-by-Page (Deliverables per Page) □□

This course is designed like a **production checklist**: every page (lesson) ends with **clear deliverables** you can verify in your Bricks/WordPress build. The goal is that you *never* "kind of understand"—you either **shipped the artifact** (code + Bricks setup + notes), or you didn't.

The Page-by-Page Workflow (Repeatable Every Time) □

1) Before You Start: Prepare the "Lesson Sandbox" (5–10 min)

1. **Pick one page to implement** (e.g., `1.1.2`) and commit to finishing it end-to-end.
2. **Choose a controlled location in Bricks**:
   1. A dedicated *Course Sandbox* page (recommended), or
   2. A specific template/section you won't reuse yet.
3. **Disable variables that hide problems** (temporarily):
   1. Cache/minify plugins (or bypass via dev mode)
   2. CDN optimizations
   3. Aggressive script deferring (until your baseline works)

Deliverable: A dedicated Bricks page/template you can safely break and reset.

---

2) Implementation: Build What the Lesson Asks—Nothing More □□

Each lesson page will include:

1. **Goal** (what you're building)
2. **Why it matters** (the mental model)
3. **Bricks implementation** (exact placement + selectors)
4. **Code** (copy/paste + explanation)
5. **Micro-exercises** (small variations that prove understanding)
6. **Troubleshooting** (common failure modes)

Rule: implement *exactly* the described effect first.
Polish (styling, extra features) comes **after** the effect is reliable.

Deliverable: The animation/behavior works exactly as described in the lesson.

---

3) Verification: Prove It Works (Not Just "Looks OK") □□

For every page, you'll verify four categories:

1. **Functional verification**
   1. Does it run on first load?
   2. Does it run after refresh?
   3. Does it run on hard reload (cache bypass)?
2. **Selector verification**
   1. Does it still work if there are *two* instances of the section?
   2. Does it break if the DOM structure changes slightly?
3. **Responsive verification**
   1. Test at least: mobile, tablet, desktop breakpoints
   2. Confirm no overflow, clipping, or unexpected shifts
4. **Accessibility/motion sanity check**
   1. If motion is significant: consider `prefers-reduced-motion`
   2. Ensure no focus traps or invisible-but-clickable overlays

Deliverable: A short checklist (written by you) marking what you tested.

What “Deliverables per Page” Actually Means □□

Each page produces *artifacts* you keep. Over time, these become your personal GSAP toolkit.

Core Deliverables (Every Page)

1. **Working demo inside Bricks**
   1. A section/page where the effect is visible
   2. Uses robust selectors (usually classes/data attributes)
2. **Saved code snippet**
   1. Stored in your project in a predictable place (see “Folder pattern” below)
   2. Includes a short header comment: what it does + how to apply
3. **One-liner notes**
   1. What surprised you?
   2. What broke first?
   3. What you’d do differently next time

Optional Deliverables (When the page includes them)

1. **Reusable function** (e.g., `initHeroReveal(rootEl)`)
2. **Mini utility** (e.g., a scoped selector helper)
3. **A variation** (e.g., “same reveal but staggered”)

Standard Folder + Naming Pattern (So You Don’t Lose Anything) □□

Even if you’re not using a bundler yet, use a consistent structure.

1. Create a “course lab” area in your theme or child theme (or a snippets plugin):
   1. `gsap-lab/`
   2. `gsap-lab/pages/`
   3. `gsap-lab/utils/`

2. Name files by lesson number:
    1. `pages/01-01-02-workflow.js`
    2. Later: `pages/04-01-first-tween.js`, etc.
3. Add a short header comment at the top of each file:
    1. What it does
    2. Where it’s used (Bricks page/template)
    3. Dependencies (GSAP core, ScrollTrigger, etc.)

Deliverable: A place where your “lesson code” lives permanently, not in random Bricks textareas.

Bricks Placement Rules (So Scripts Don’t Turn Into Chaos) □□

You’ll repeatedly choose *where* code goes. Use these rules while learning:

1. **During learning (recommended):**
    1. Put lesson JS in **one** predictable place (global or the sandbox page)
    2. Keep selectors scoped to the sandbox section to avoid side effects
2. **When you turn a lesson into a reusable component:**
    1. Move code into a “component init” function
    2. Initialize per section instance (important for query loops/repeated sections)

Deliverable: You can explain *why* your code is placed where it is (not “because it worked once”).

The “Two-Instance Test” (Your Fastest Pro Filter) □□

A huge amount of WordPress/Bricks animation pain comes from accidental global targeting.

Do this after most lessons:

1. Duplicate the animated section in Bricks (same page).
2. Reload.
3. Confirm:
    1. Both instances animate correctly
    2. One instance doesn’t hijack the other’s elements

3. Timing doesn’t “double fire”

If it fails, you learn a key professional habit: **scope your selectors**.

Deliverable: A screenshot or note: “Two-instance test: pass/fail + fix”.

What You Should Send Me After Completing Any Page □□

When you finish a lesson page, reply with:

1. **Where you put the code** (Bricks global/page/element OR theme file)
2. **Your selectors** (classes/data attributes you used)
3. **What worked immediately**
4. **What broke**
5. **One screenshot** (optional but helpful) or a short description

Then I can:

1. Suggest improvements (scoping, maintainability)
2. Flag performance/accessibility risks early
3. Provide a cleaner “component version” of your solution

Your Deliverables for *This* Page (1.1.2) □

1. A written **personal workflow checklist** you will follow for each lesson page (copy the sections above and simplify to your taste).
2. A **sandbox page/template in Bricks** dedicated to course experiments.
3. A **project structure decision**:
   1. Where lesson JS will live *for now*
   2. How you will name and store lesson code snippets
4. A quick **two-instance test habit** added to your checklist.

Quick Micro-Exercise (5 minutes) ✍

Create a checklist you can paste at the top of each future lesson file:

1. Goal:
2. Where installed (Bricks/page/global/file):
3. Selectors used:
4. Tested:
    1. Refresh
    2. Hard reload
    3. Two-instance test
    4. Mobile/tablet/desktop
    5. Reduced motion (if relevant)
5. Notes:

Revision #1
Created 2026-04-19 18:55:40 UTC by art10m
Updated 2026-04-19 18:58:09 UTC by art10m