1. What's Repeatable in AI Agents or Modeling (Strategy → Tactics)

When you zoom out, every AI agent or modeling initiative has **repeatable layers**:

- **Strategy Layer (Repeatable Concepts)**
  - **Problem Definition**: Define scope (diagnostic → predictive → prescriptive).
  - **Value Lens**: Tie agent/model to measurable KPIs (cost, cycle time, accuracy, adoption).
  - **System Boundary**: Clarify what the agent controls vs. just observes (closed-loop vs. advisory).
  - **Architecture Choices**: Choose between symbolic (rules), statistical (ML), or hybrid (neuro-symbolic).
  - **Scaling Strategy**: Pilot → rollout → orchestration → governance.
- **Tactical Layer (Repeatable Steps)**
  - **Data Acquisition**: Web scraping, APIs, sensors, logs.
  - **Data Structuring**: Cleaning, feature extraction, embeddings.
  - **Modeling Cycle**: Baseline → train → validate → deploy → monitor.
  - **Integration**: Expose via API, dashboard, or workflow automation.
  - **Feedback Loops**: User input, continuous learning, A/B testing.
  - **Ops Practices**: CI/CD for models, monitoring drift, retraining triggers.

👉 The repeatable piece is the **pipeline pattern** — define → collect → structure → model → deploy → refine.

---

2. Fundamentals of Scripting (Applies to AI + Scraping)

For both AI agents and scraping tools, the **fundamentals** are the same:

1. **Environment Setup**: Install packages, import libraries.
2. **Inputs**: Accept variables (URLs, queries, params).
3. **Process Logic**: Loops, conditionals, function definitions.
4. **Data Handling**: Parse, clean, transform.
5. **Persistence**: Store results (files, DB, Google Sheets).
6. **Error Handling**: Try/except, logging, retries.
7. **Automation**: Scheduling, scaling, modularizing.

👉 Think in **functions** that each do one clear thing — scrape_page(), parse_job(), write_to_sheets().