Lab 1 — Plugin registry (decorator ➜ metaclass)

Goal (20 minutes): Build a plugin system where plugins register themselves by name.

Tasks

1. Implement a class decorator `@register(name=...)` that adds the decorated class to a module-level `registry` dict.
   - Key: `name` if provided, else `cls.plugin_name` if defined, else the class name.
   - Reject duplicate names (raise clear `ValueError`).
   - Add `discover()` that returns sorted plugin names.
2. (Stretch) Implement `PluginMeta` so that any subclass of `BasePlugin` auto-registers itself.
   - Use `__abstract__ = True` on `BasePlugin` to skip registration.
   - Keep the same naming rule.
3. Write example plugins (`resize`, `crop`, `emoji`) and show discovery and usage.

Run `pytest -q` to check your work; `solution.py` shows a reference.