# Documentation: qairt\_converter\_parser.py

This script is a procedural-style Python utility that automates the generation of qairt-converter command-line commands using parameters provided through a YAML configuration file. The output is a JSON file containing the generated commands.  
  
Key Components:  
- argparse is used to parse CLI inputs for YAML config, output path, and execution flag.  
- YAML config is read using PyYAML to get all input parameters.  
- Each specific optional argument (like input shape, output tensor, data type, layout) is connected to the required '--input\_network' argument.  
- Commands are appended in a list and stored in JSON format.  
- If --execute is passed, subprocess.Popen is used to run those commands.  
- It's a script-based approach and not object-oriented.  
  
Pros:  
- Simple structure.  
- Easy to read and modify for small-scale automation.  
  
Limitations:  
- Not modular or reusable.  
- Harder to extend or maintain.

# Documentation: alternate\_qairt\_converter\_automation.py

This is an object-oriented version of the qairt-converter automation tool. It follows clean software design practices by encapsulating logic in a class and modularizing functionalities.  
  
Key Components:  
- Class: `QairtConverterAutomation` encapsulates all operations.  
- Method `load\_yaml\_config`: Loads parameters from a YAML file.  
- Method `build\_commands`: Constructs command lists based on input network and associated arguments.  
- Method `save\_commands\_to\_json`: Saves the generated commands to a JSON file.  
- Method `execute\_commands`: Executes the constructed commands if `--execute` flag is passed.  
- Logging is handled with `loguru` for better diagnostics.  
  
Advantages:  
- Modular and extensible.  
- Aligns with best practices for larger automation systems.  
- Easier to integrate, maintain, and test.  
  
This version mirrors a production-style structure as seen in industry-grade CLI tools and is based on the `ToolArguments` architecture the user referenced.