

Final Master Algorithm (v2.1)

Complete Command Cheat Sheet

Use this cheat sheet as a **one-stop reference** for all slash commands, their aliases, brief purposes, and example usage. Commands are grouped into four main categories: **Core, Reflective & Analytical, Management & System**, and **Advanced**.

Each table entry provides:

- **Command & Alias**
- **Brief Purpose**
- **Typical Usage (Example)**

You can copy and adapt these tables directly into your PDF layout tool (e.g., Microsoft Word, Google Docs, Adobe InDesign) or use this HTML to generate your PDF.

A. Core Commands

Command & Alias	Brief Purpose	Typical Usage (Example)
/Continue Alias: /cnt	Resumes from the exact stopping point in the previous output, ensuring a seamless flow.	<code>"/Continue"</code> <i>Resumes text mid-sentence.</i>
/GenerateSuperWords Alias: /gsw	Produces advanced or multi-theme keywords to enrich brainstorming.	<code>"/GenerateSuperWords [Theme: Cognitive Expansion]"</code> <i>Returns higher-level, nuanced keywords.</i>
/GeneratePromptPatterns Alias: /gpp	Suggests structural “prompt patterns” (e.g., Explain–Analyze–Reflect) to guide or shape a discussion or analysis.	<code>"/GeneratePromptPatterns [Purpose: Critical Thinking]"</code> <i>Yields patterns like “Compare–Contrast–Conclude.”</i>
/CreateVariableList Alias: /cvl	Compiles placeholder variables (e.g., <code>[MAIN_TOPIC]</code> , <code>[PERSPECTIVE]</code>) for more flexible prompt-building.	<code>"/CreateVariableList"</code> <i>Returns a template-like list of placeholders.</i>
/GenerateBlindspots Alias: /gbs	Identifies easily overlooked factors or risks in a plan, solution, or argument.	<code>"/GenerateBlindspots [Topic: Marketing Strategy]"</code> <i>Outputs potential missed considerations.</i>
/InsightPivotList	Suggests multiple angles or perspectives (“pivots”) for deeper exploration of a single topic.	<code>"/InsightPivotList [Topic: Workplace Automation]"</code> <i>Generates viewpoint pivots (cost–benefit, ethical, etc.).</i>
/GenerateConnectionsList	Generates interrelated ideas or themes to uncover new or cross-domain insights.	<code>"/GenerateConnectionsList [Topic: Neuroscience + Architecture]"</code> <i>Identifies synergy points, e.g. “impact of built environments.”</i>
/ExpandConcept	Broadens a given idea by introducing related themes, tangential topics, or big-picture perspectives.	<code>"/ExpandConcept [Concept: Data Privacy]"</code> <i>Suggests expansions like “Regulatory frameworks, ethical considerations, user trust,” etc.</i>

B. Reflective & Analytical Commands

Below, commands are subdivided into two subcategories for clarity.

B.1. Critical Reflection & Gap Detection

Command & Alias	Brief Purpose	Typical Usage (Example)
/RecursiveCheck Alias: /rc	Initiates repeated questioning to probe deeper into a concept or conclusion.	<code>"/RecursiveCheck [Topic: Employee Turnover]"</code> <i>Generates iterative “why?” queries.</i>
/AssumptionAudit Alias: /aa	Highlights hidden or unexamined assumptions in a proposal or line of reasoning.	<code>"/AssumptionAudit [Topic: Forecasting Sales]"</code> <i>Outputs “Assumption of stable economy,” etc.</i>
/FeedbackLoop	Begins a cycle for refining a response based on user-provided feedback or new data.	<code>"/FeedbackLoop"</code> <i>Prompts for user input, then revises prior outputs.</i>
/ReflectiveOptimization	Revisits and refines existing reflections or partial answers for greater clarity and depth.	<code>"/ReflectiveOptimization [Draft: Proposed Solution to Budget Cuts]"</code>
/ReflectiveGapDetection Alias: /rgd	Identifies omissions or holes in an argument, outline, or plan.	<code>"/ReflectiveGapDetection [Outline: Marketing Plan]"</code> <i>Finds missing competitor analysis or brand positioning.</i>
/RecursiveQuestionCycle Alias: /rqc	Produces multi-layered, iterative “why” or “how” questions to discover core issues.	<code>"/RecursiveQuestionCycle [Topic: AI Regulation]"</code> <i>Digs deeper with each question.</i>
/InsightPivot	Shifts a question or discussion to a fresh angle (e.g., from cost focus to user experience).	<code>"/InsightPivot [Topic: Supply Chain Bottlenecks]"</code>
/GenerateExamples	Offers concrete examples or use cases to illustrate a concept or principle.	<code>"/GenerateExamples [Concept: Behavioral Economics]"</code> <i>Gives real-world scenarios.</i>
/MetaReflection Alias: /mrf	Examines the reasoning process itself, checking for biases or leaps in logic.	<code>"/MetaReflection"</code> <i>Evaluates thoroughness of prior steps.</i>
/ContextualShifts	Reframes a concept across different contexts (individual, societal, global) for broader insights.	<code>"/ContextualShifts [Topic: Remote Work Policies]"</code>
/NegativeImplications Alias: /ni	Identifies potential drawbacks or risks in a proposed solution or strategy.	<code>"/NegativeImplications [Proposal: Aggressive Expansion Strategy]"</code>

B.2. Problem-Solving & Decision Tools

Command & Alias	Brief Purpose	Typical Usage (Example)
/ConductRootCauseAnalysis Alias: /rca	Decomposes a problem to find the fundamental source(s).	<code>"/ConductRootCauseAnalysis [Problem: Software Bugs]"</code>
/DevelopDecisionMatrices Alias: /ddm	Creates a matrix/table to compare multiple options or solutions.	<code>"/DevelopDecisionMatrices [Options: Vendor A vs. Vendor B vs. Vendor C]"</code>
/PerformSWOTAnalysis Alias: /psa	Analyzes Strengths, Weaknesses, Opportunities, and Threats for a given topic.	<code>"/PerformSWOTAnalysis [Topic: Launching a New Product Line]"</code>
/ConductScenarioPlanning Alias: /csp	Explores possible future scenarios and strategies (e.g., best-case, worst-case).	<code>"/ConductScenarioPlanning [Context: Economic Downturn]"</code>
/DevelopContingencyPlans Alias: /dcp	Lays out backup options for each scenario in case conditions change unexpectedly.	<code>"/DevelopContingencyPlans [Challenge: Supply Chain Disruption]"</code>

/AnalyzeRiskFactors Alias: /arf	Identifies and evaluates possible risks, their severity, and likelihood.	"/AnalyzeRiskFactors [Project: Global Expansion]"
/OptimizeWorkflows Alias: /owf	Suggests ways to improve efficiency or eliminate bottlenecks in processes.	"/OptimizeWorkflows [Department: Customer Support]"
/IdentifyCognitiveBiases	Recognizes common biases (confirmation bias, anchoring, etc.) that may skew analysis.	"/IdentifyCognitiveBiases [Topic: Hiring Process]"
/GenerateCounterarguments	Produces opposing viewpoints to ensure balanced consideration.	"/GenerateCounterarguments [Position: Working Remotely Increases Productivity]"
/GenerateAnalogiesBetweenFields	Draws parallels across different domains (e.g., "medicine vs. cybersecurity").	"/GenerateAnalogiesBetweenFields [Fields: Quantum Computing & Supply Chain]"
/ExploreEthicalImplications	Investigates moral or ethical concerns around a plan or technology.	"/ExploreEthicalImplications [Topic: AI-Powered Hiring]"
/ConductCulturalComparisons	Compares how different cultures approach a concept or issue.	"/ConductCulturalComparisons [Business Strategy: US vs. Japan]"
/AnalyzeHistoricalPrecedents	Looks at historical examples to inform current decisions.	"/AnalyzeHistoricalPrecedents [Topic: Economic Crisis Management]"
/IntegrateArtAndScience	Merges artistic creativity with scientific data or methods for innovative solutions.	"/IntegrateArtAndScience [Concept: Urban Planning]"
/ExplorePhilosophicalUnderpinnings	Probes fundamental philosophical angles or theories behind a topic.	"/ExplorePhilosophicalUnderpinnings [Theme: Technology & Free Will]"
/SynthesizeScientificFindings	Combines multiple data sets or studies into cohesive conclusions.	"/SynthesizeScientificFindings [Research: Climate Change Reports]"

C. Management & System Commands

Commands designed to manage the system, generate structured outputs, or provide administrative insights.

Command & Alias	Brief Purpose	Typical Usage (Example)
/SystemPromptReview	Displays or summarizes ChatGPT's current "system prompt" or underlying instructions.	"/SystemPromptReview"
/ProfileAssessment	Requests the system's understanding of the user's context or goals based on recent interactions.	"/ProfileAssessment"
/CommandFeedbackSummary	Summarizes how effectively commands have been used and suggests optimization.	"/CommandFeedbackSummary"
/ShowStructuresOfOutput	Reveals text structures (headings, bulleting) the system uses to format responses.	"/ShowStructuresOfOutput"
/ShowVisualStructuresOfOutput	Outlines available visual output formats (tables, mind maps) and their best use cases.	"/ShowVisualStructuresOfOutput"
/GenerateMindMap	Creates a conceptual mind map for a topic, showing main nodes and branches.	"/GenerateMindMap [Topic: Personal Finance]"
/CreateListReflectiveQuestions	Compiles questions that target deeper reflection on a specific concept.	"/CreateListReflectiveQuestions [Theme: Leadership Styles]"
/GenerateDivergentQuestions	Produces broad, creative questions to explore multiple angles or solutions.	"/GenerateDivergentQuestions [Problem: Reducing Plastic Waste]"
/ConvergentThinkingMode	Refines a range of ideas into a smaller set of the	"/ConvergentThinkingMode [Brainstorm

	most viable ones.	Output: 10 Product Ideas]"
/GenerateTaskList	Compiles actionable steps (a "to-do" list) to implement a plan or solve a problem.	"/GenerateTaskList [Project: Launching a Personal Blog]"
/OutputTables	Formats key points or data into a table for clarity.	"/OutputTables [Data: Sales by Quarter in 2024]"
/RecursiveAdjustmentCycle	Continuously refines answers or solutions based on prior outputs and new feedback.	"/RecursiveAdjustmentCycle [Topic: Writing a White Paper]"
/ExpandMetaInquiry	Broadens inquiry-based prompts to include questions about the process of inquiry itself.	"/ExpandMetaInquiry [Topic: Why pursue brand loyalty?]"
/GenerateCommandMap	Displays how commands relate or feed into one another (dependencies, suggested flows).	"/GenerateCommandMap"
/InsightfulAngleList	Lists potential new angles or perspectives for a given subject.	"/InsightfulAngleList [Topic: Corporate Social Responsibility]"
/ListNewCommands	Suggests new command structures that might address recent user inquiries or trends.	"/ListNewCommands"
/GenerateProcessMap	Creates a step-by-step workflow or flowchart for complex tasks or projects.	"/GenerateProcessMap [Goal: Implementing Agile in a Startup]"
/GenerateInterconnections	Identifies potential links between separate concepts, encouraging integrated exploration.	"/GenerateInterconnections [Concepts: Blockchain, Healthcare, Security]"
/GenerateAlternativeSolutions	Suggests additional solutions or approaches to a problem for broader innovation.	"/GenerateAlternativeSolutions [Issue: Declining Employee Morale]"
/MetaSynthesis	Combines key insights from different responses or commands into a single, cohesive synthesis.	"/MetaSynthesis"

D. Advanced Commands

High-level commands for **systemic analyses**, **multi-perspective feedback**, and **future scenario modeling**. Subdivided for easier reading.

D.1. Complex Systemic Analysis

Command & Alias	Brief Purpose	Typical Usage (Example)
/GenerateSystemicHypotheses Alias: /gsh	Produces interconnected hypotheses about a complex system, considering multiple variables or feedback loops.	"/GenerateSystemicHypotheses [Topic: Education Reform]"
/BlindspotAssessmentCycle	Iteratively uncovers hidden blind spots from multiple perspectives.	"/BlindspotAssessmentCycle [Strategy: Expanding into Emerging Markets]"
/CausalLayeredAnalysis	Examines a topic at multiple layers (litany, systemic causes, worldviews, myths/metaphors).	"/CausalLayeredAnalysis [Problem: Decline in Voter Turnout]"
/GenerateRecursiveExploration	Revisits a topic at increasing depths, refining each pass.	"/GenerateRecursiveExploration [Topic: Corporate Governance Ethics]"
/InterconnectedThemesAnalysis Alias: /ita	Detects overarching patterns or themes across different arguments or domains.	"/InterconnectedThemesAnalysis [Ideas: AI in Finance, AI in Healthcare, AI in Education]"
/GenerateDynamicDecisionTree	Builds a branching decision tree that adapts based on earlier outcomes.	"/GenerateDynamicDecisionTree [Situation: Choosing a Marketing

		Strategy]"
/ScenarioResilienceAssessment	Evaluates how robust or “resilient” scenarios or strategies are under changing conditions.	"/ScenarioResilienceAssessment [Plan: Launching a Mobile App]"
/GameTheoryScenarioGeneration Alias: /gtsg	Simulates strategic scenarios using game theory logic (multiple players, possible moves, payoffs).	"/GameTheoryScenarioGeneration [Context: Negotiating a Trade Deal]"
/GenerateSequentialSimulations	Produces a series of potential outcomes for a plan or strategy, iterating from one result to the next.	"/GenerateSequentialSimulations [Topic: Rolling Out Policy Changes]"
/TransdisciplinarySynthesis Alias: /tds	Integrates insights from multiple disciplines into a cohesive solution.	"/TransdisciplinarySynthesis [Problem: Income Inequality]"
/ConceptualEvolutionMap	Maps how a concept has changed over time or across fields, noting key shifts or trends.	"/ConceptualEvolutionMap [Concept: AI Ethics]"
/SystemSynergyMapping	Identifies synergy points where elements combine to create enhanced outcomes.	"/SystemSynergyMapping [Ecosystem: University + Tech Startups + Local Government]"
/AnalogicalSynthesis	Draws complex analogies across domains to spark innovative connections.	"/AnalogicalSynthesis [Fields: Climate Change & Healthcare Resource Allocation]"
/GenerateComplexPatternRecognition	Detects intricate patterns or trends in multi-faceted data or arguments.	"/GenerateComplexPatternRecognition [Dataset: Customer Churn + Social Media Engagement]"
/ContinuousLearningPath	Outlines a progressive, evolving learning roadmap for a specified domain.	"/ContinuousLearningPath [Topic: Quantum Computing Basics to Advanced]"
/GenerateAdaptiveHeuristic	Creates heuristics that shift based on new data or changing environments.	"/GenerateAdaptiveHeuristic [Context: Allocating Project Resources]"

D.2. Meta-Reflection & Self-Improvement

Command & Alias	Brief Purpose	Typical Usage (Example)
/MultiPerspectiveFeedbackLoop	Collects feedback from multiple viewpoints, then refines the approach or answer accordingly.	"/MultiPerspectiveFeedbackLoop [Topic: Proposed Healthcare Policy]"
/RecursiveRefinementCycle	Iteratively revisits a question or solution, integrating feedback or new angles at each pass.	"/RecursiveRefinementCycle [Thesis Draft: Impacts of Social Media on Mental Health]"
/MetaCognitionAudit Alias: /mca	Evaluates the quality of reasoning, identifying potential oversights or cognitive shortcuts.	"/MetaCognitionAudit" <i>Examines logic for biases or leaps.</i>
/GenerateFractalModeling	Uses fractal principles to map repeating, self-similar structures in a complex system.	"/GenerateFractalModeling [Topic: Organizational Hierarchy Patterns]"
/EmergentPropertyAssessment	Looks for properties that only arise when components interact (emergent behaviors).	"/EmergentPropertyAssessment [System: Social Networks & Viral Trends]"
/RecursiveAnalogyLayering	Builds layered analogies step by step, providing deeper understanding of abstract concepts.	"/RecursiveAnalogyLayering [Concept: Blockchain Security]"
/MetaAssumptionsCheck	Examines underlying assumptions across multiple steps or viewpoints.	"/MetaAssumptionsCheck [Discussion: Free Trade Policies]"
/GenerateComplexInterdependencies	Maps how various parts of a system mutually influence each other.	"/GenerateComplexInterdependencies [System: Global Food Supply Chain]"

/ContextualLayerShifting	Shifts analysis across different contextual layers (e.g., individual → group → societal) for deeper insight.	"/ContextualLayerShifting [Issue: Internet Privacy]"
/GenerateSystemDiagnostics	Analyzes the performance or alignment of multiple command sequences, suggesting improvements.	"/GenerateSystemDiagnostics"
/SystemFeedbackAnalysis Alias: /sfa	Reviews user/system feedback from recent interactions to refine subsequent responses or commands.	"/SystemFeedbackAnalysis"
/SelfOptimizationCycle Alias: /soc	Implements a loop of self-improvement for command usage or the system's approach.	"/SelfOptimizationCycle"
/GenerateAdminReport	Summarizes command efficiency, usage patterns, and satisfaction levels for administrative oversight.	"/GenerateAdminReport"
/CommandDependencyMap	Shows interdependencies among commands, illustrating logical flows or synergy.	"/CommandDependencyMap"
/MetaCommandFeedbackCycle	Runs a recursive check on the effectiveness of recent command choices, suggesting ways to refine them.	"/MetaCommandFeedbackCycle"

Notes on Usage & Further Tips

1. Context is Key

Many commands (especially advanced ones) thrive on *detailed* context or examples. The richer your input, the more nuanced the output.

2. Combine Commands

It's often helpful to *chain* commands. For instance, use `/GenerateBlindspots` first, then run `/NegativeImplications` on the newly discovered issues.

3. Watch Out for Overuse

Some commands (e.g., `/RecursiveQuestionCycle`, `/GameTheoryScenarioGeneration`) can produce repetitive or lengthy outputs if not guided with specifics.

4. Aliases & Tags

Shortcut aliases (e.g., `/rqc` for `/RecursiveQuestionCycle`) speed up usage. Tags are optional but help in searching or referencing commands.

5. Periodic Maintenance

Keep a **CHANGELOG** if you modify or add commands. Regularly review which commands are used frequently (or rarely) to refine your custom GPT environment.

End of Complete Cheat Sheet

You now have a **no-truncation** list of **all** the commands in the Final Master Algorithm (v2.1). Feel free to use this HTML in any "HTML to PDF" tool to download a well-formatted version.