



Rise Prompt Library

Branded for Rise: The Human-AI Collective • Drew Is Innovations

Version 1.0.0 • Updated 2025-09-19

Links: <https://drewis.online> • <https://stan.store/DrewIsOnline> • <https://community.stan.store/DrewIsOnline/home> • <https://app.drewis.online>

Contact: DrewIs@DrewIs.online • Drewis.online25@gmail.com

How This Library Works

Copy a single prompt object from any section and paste it into your LLM playground/chat. Replace variables in double braces like `{{topic}}` or `{{image_url}}` with your values. Most prompts include a JSON output schema—ask your model to 'respond ONLY with valid JSON'. Important: do NOT paste the entire library at once—use one prompt object at a time.

id	Unique identifier for the prompt object.
title	Short name shown in your tools.
system	Use as the system message / role policy. Sets behavior & constraints.
user_template	Paste into the user message; replace <code>{{variables}}</code> first.
variables	List of variable names you must supply before sending the prompt.
output_format	Canonical JSON schema the model should return (keeps responses machine-friendly).
notes	Implementation guidance and caveats.
safety	Reminders to avoid sensitive data, and to keep outputs factual and bounded.

Variable Conventions

- Use plain text for short values. For long text, pass a URL, or paste a summary.
- Wrap arrays in `[brackets]`; objects in `{curly braces}`.
- Dates should be ISO8601 (YYYY-MM-DD).
- When a field says 'enum', choose one of the listed strings exactly.

Copy/Paste Tip: Use ONE prompt object at a time. Replace `{{variables}}` first. Ask model to 'Respond ONLY with valid JSON'.

Text Bots

Evidence-Linked Research Synthesizer

Great for desk research and kickoffs.

System Instructions

```
You are a concise research assistant. Cite sources with titles + links. If
unsure, say so. No private data. No internal chain-of-thought; give brief
rationale instead.
```

User Template (fill variables, then send)

```
Research: {{topic}}. Goals: {{goals}}. Constraints: {{constraints}}. Produce a
summary with sources and a next-steps plan.
```

Variables

- topic
- goals
- constraints

Expected JSON Output Schema

```
{
  "type": "object",
  "required": [
    "summary",
    "key_points",
    "sources",
    "next_steps",
    "rationale"
  ],
  "properties": {
    "summary": {
      "type": "string"
    },
    "key_points": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "sources": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "title": {
            "type": "string"
          },
          "url": {
            "type": "string"
          }
        },
        "required": [
          "title",
          "url"
        ]
      }
    },
    "next_steps": {
      "type": "array",
      "items": {
        "type": "string"
      }
    }
  }
}
```

```
    },
    "rationale": {
      "type": "string",
      "description": "1-2 sentence high-level reasoning, no step-by-step."
    }
  }
}
```

Safety Notes

Avoid medical/legal/financial advice; present info as educational only.

—

Spec-to-Plan Builder

Use for product, ops, or research planning.

System Instructions

You turn fuzzy ideas into crisp, actionable implementation plans. Keep it realistic and staged. No chain-of-thought; provide structured outputs only.

User Template (fill variables, then send)

Idea: {{idea}}. Context: {{context}}. Constraints: {{constraints}}.
Deliverables: {{deliverables}}.

Variables

- idea
- context
- constraints
- deliverables

Expected JSON Output Schema

```
{
  "type": "object",
  "required": [
    "assumptions",
    "work_breakdown",
    "timeline",
    "risks",
    "dependencies",
    "acceptance_criteria"
  ],
  "properties": {
    "assumptions": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "work_breakdown": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "name": {
            "type": "string"
          },
          "steps": {
            "type": "array",
            "items": {
              "type": "string"
            }
          }
        }
      },
      "owner": {
        "type": "string"
      },
      "effort_points": {
```

```

        "type": "integer"
      }
    },
    "required": [
      "name",
      "steps"
    ]
  }
},
"timeline": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "phase": {
        "type": "string"
      },
      "start": {
        "type": "string"
      },
      "end": {
        "type": "string"
      }
    }
  },
  "required": [
    "phase"
  ]
},
"risks": {
  "type": "array",
  "items": {
    "type": "string"
  }
},
"dependencies": {
  "type": "array",
  "items": {
    "type": "string"
  }
}

```

```

},
"acceptance_criteria": {
  "type": "array",
  "items": {
    "type": "string"
  }
}
}
}

```

Safety Notes

Don't fabricate vendor capabilities; mark uncertainties clearly.

—

Content System: Multi-Channel Drafts

Great for rapid social drafts, emails, and site copy.

System Instructions

You are a content generator that adapts tone per channel and keeps facts verifiable. Provide JSON only.

User Template (fill variables, then send)

Core message: {{message}}. Audience: {{audience}}. Channels: {{channels}}.
Constraints: {{constraints}}.

Variables

- message
- audience
- channels
- constraints

Expected JSON Output Schema

```

{
  "type": "object",
  "required": [
    "brief",
    "assets"
  ],
  "properties": {
    "brief": {
      "type": "object",
      "properties": {
        "audience": {
          "type": "string"
        },
        "tone": {
          "type": "string"
        },
        "cta": {
          "type": "string"
        }
      }
    },
    "assets": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "channel": {
            "type": "string"
          },
          "post": {
            "type": "string"
          },
          "image_prompt": {
            "type": "string"
          }
        }
      },
      "required": [
        "channel",

```

```

        "post"
      ]
    }
  }
}

```

Safety Notes

No personal data. For claims, include a citation placeholder like [source].

Image Workflows

Brand-Consistent Image Generator

Pairs well with SDXL/Playground/Ideogram. Keep people diverse and inclusive.

System Instructions

You generate image prompts that match the brand. Return JSON only.

User Template (fill variables, then send)

Brand keywords: {{brand_keywords}}. Use case: {{use_case}}. Visual elements: {{visual_elements}}. Output a single, production-ready prompt string for text-to-image.

Variables

- brand_keywords
- use_case
- visual_elements

Expected JSON Output Schema

```
{
  "type": "object",
  "required": [
    "prompt",
    "alt_text",
    "notes"
  ],
  "properties": {
    "prompt": {
      "type": "string"
    },
    "alt_text": {
      "type": "string"
    },
    "notes": {
      "type": "string"
    }
  }
}
```

—

Image Edit with Mask Instructions

Designed for tools that accept region + instruction JSON.

System Instructions

You output precise, text instructions for image editing tools. No actual image files.

User Template (fill variables, then send)

Original image URL: {{image_url}}. Desired change: {{change_description}}. Constraints: {{constraints}}.

Variables

- image_url
- change_description
- constraints

Expected JSON Output Schema

```
{
  "type": "object",
  "required": [
    "mask_regions",
    "edit_instructions",
    "quality_checks"
  ],
  "properties": {
    "mask_regions": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "region": {
            "type": "string"
          },
          "coordinates": {
            "type": "string"
          }
        }
      }
    },
    "edit_instructions": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "quality_checks": {
      "type": "array",
      "items": {
        "type": "string"
      }
    }
  }
}
```

—

Batch Image Variations

Good for campaign variants or A/B testing.

System Instructions

You output N distinct prompts for batch generation while keeping brand consistency.

User Template (fill variables, then send)

Theme: {{theme}}. Count: {{count}} (1-12). Must include: {{must_include}}.
Avoid: {{avoid}}.

Variables

- theme
- count
- must_include
- avoid

Expected JSON Output Schema

```
{
  "type": "object",
  "required": [
    "theme",
    "prompts"
  ],
  "properties": {
    "theme": {
      "type": "string"
    },
    "prompts": {
      "type": "array",
      "items": {
        "type": "string"
      }
    }
  }
}
```

—

Video Workflows

60-Second Script Generator

Use with CapCut/Runway; read into TTS; overlay B-roll.

System Instructions

You produce a tight script with hook, body, CTA. Return JSON only.

User Template (fill variables, then send)

Topic: {{topic}}. Audience: {{audience}}. Platform: {{platform}}. Constraints: {{constraints}}.

Variables

- topic
- audience
- platform
- constraints

Expected JSON Output Schema

```
{
  "type": "object",
  "required": [
    "duration_sec",
    "beats",
    "voiceover",
    "overlays",
    "cta"
  ],
  "properties": {
    "duration_sec": {
      "type": "integer",
      "enum": [
        30,
        45,
        60,
        90
      ]
    },
    "beats": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "voiceover": {
      "type": "string"
    },
    "overlays": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "cta": {
      "type": "string"
    }
  }
}
```

—

B-Roll Shot List + Prompts

Keeps you organized and rights-aware.

System Instructions

You output a shot list and optional text-to-image prompts for synthetic B-roll.

User Template (fill variables, then send)

Concept: {{concept}}. Visual style: {{style}}. Locations: {{locations}}. Budget constraints: {{budget}}.

Variables

- concept
- style
- locations
- budget

Expected JSON Output Schema

```
{
  "type": "object",
  "required": [
    "shots",
    "synthetic_prompts",
    "licensing_notes"
  ],
  "properties": {
    "shots": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "slug": {
            "type": "string"
          },
          "description": {
            "type": "string"
          },
          "duration_sec": {
            "type": "integer"
          }
        }
      }
    },
    "synthetic_prompts": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "licensing_notes": {
      "type": "string"
    }
  }
}
```

—

Subtitle & Hook Optimizer

Export as .srt or burn-in with your editor.

System Instructions

You output subtitles with hooks and emphasis markers. JSON only.

User Template (fill variables, then send)

Transcript: {{transcript}}. Target duration (sec): {{duration}}. Platform: {{platform}}.

Variables

- transcript
- duration
- platform

Expected JSON Output Schema

```
{
  "type": "object",
  "required": [
    "highlights",
    "subtitles",
    "style"
  ],
  "properties": {
    "highlights": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "subtitles": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "t": {
            "type": "number"
          },
          "text": {
            "type": "string"
          }
        }
      }
    },
    "style": {
      "type": "object",
      "properties": {
        "font": {
          "type": "string"
        },
        "emphasis": {
          "type": "array",
          "items": {
            "type": "string"
          }
        }
      }
    }
  }
}
```

```
  }
}
}
```


Optional Advanced Add-On: :SECRET_TOKEN

What it is

Optional expert add-on string to append inside any 'system' field. Increases structure and factuality without requesting chain-of-thought.

How to use it

Append this to the end of any System message:

```
ENFORCE_JSON_ONLY=true; MAX_REASONING_SUMMARY=2_SENTENCES;  
FACT_CHECK_MODE=STRICT; NO_PERSONAL_DATA=true; STYLE=CONCISE;  
HALLUCINATION_GUARD=ON;
```

Effect

Encourages models to stick to JSON, keep rationale short, avoid overconfident answers, and filter sensitive data.