



[Get ALL AI Automation + Agent Templates >>](#)

Prompts & Links - I Built the Ultimate Video AI Automation for Reels and Shorts (n8n No-Code)

(Follow the instructions in this video: https://youtu.be/5iLxMk5UA_I)

Templates:

AI Automation Template for Instant Setup

<https://fabimarkl.com/automation-templates/#newsletter-video>

Google Sheet Template (must follow YT tutorial for correct setup):

<https://docs.google.com/spreadsheets/d/1tfr2xoUEmzBkLSyToDgFuHZYVHOcTKlhGMBOZlw9EJM/copy>

Links:

Free Trial for N8N Account:

<https://n8n.io/>

Create New Gmail Account:

<https://accounts.google.com/signin>

Create PIAPI Account:

<https://piapi.ai/>

Get API Key for OpenAI:

<https://platform.openai.com/>

Setup JSON2Video Account:

<https://json2video.com/>

Setup Make.com Account:

<https://www.make.com/>

Setup Upload Posts Account (TikTok)

<https://www.upload-post.com/>

[Click Here to Get AI Agent Template for Instant Setup >>](#)

Clean Up Email Content:

```
const rawHtml = $('Gmail Trigger').first().json.textAsHtml|| '';

function decodeHtmlEntities(str) {
  return str
    .replace(/&nbsp;/g, ' ')
    .replace(/&zwnj;/g, '')
    .replace(/&mdash;/g, '-')
    .replace(/&ndash;/g, '-')
    .replace(/&rsquo;/g, '')
    .replace(/&lsquo;/g, '')
    .replace(/&ldquo;/g, '"')
    .replace(/&rdquo;/g, '"')
    .replace(/&quot;/g, '"')
    .replace(/&#39;/g, '"')
    .replace(/&apos;/g, '"')
    .replace(/&lt;/g, '<')
    .replace(/&gt;/g, '>')
    .replace(/&amp;/g, '&')
    .replace(/&#x[\da-f]+;/gi, ''); // removes weird unicode like
&#x1F916;
}

// Step-by-step cleaning
let cleaned = rawHtml

// Strip everything before the first real paragraph
.replace(/.*?<p>\.*?Good morning\./is, '*Good morning.*')

// Remove footer: from "Update your email preferences" onward
.replace(<p>Update your email preferences[\s\S]*$/i, '')

// Strip all remaining <a href=""> tags, keeping just the
visible text
.replace(<a [^>]+>(.*?)</a>/gi, '$1')

// Replace <br> and </p> with line breaks
.replace(<br\s*\/?>/gi, '\n')
.replace(<\/p>/gi, '\n\n')

// Strip remaining HTML tags
.replace(<[>]*>/g, '')

// Decode HTML entities
.trim();
```

```
cleaned = decodeHtmlEntities(cleaned);

// Final cleanup: multiple newlines -> max 2
cleaned = cleaned.replace(/\n{3,}/g, '\n\n').trim();

return [{ json: { cleanedText: cleaned } }];
```

PIAPI Music Generator:

Name: X-API-KEY

<https://api.piapi.ai/api/v1/task>

```
{
  "model": "Qubico/diffrhythm",
  "task_type": "txt2audio-base",
  "input": {
    "style_prompt": "YOUR_MUSIC_PROMPT"
  },
  "config": {}
}
```

OpenAI Generate Video Content:

You are a video strategist. Your task is to break down the newsletter below into 3 short-form vertical video scripts, each designed for platforms like Instagram Reels, YouTube Shorts, or TikTok.

#RULE:

- this is the absolute rule you must comply with for any text you generate in the following: NEVER EVER use any quotation marks "" as this will give me error later in my json code.

Each video must include:

- A compelling title (acts as the hook, in plain English). Give it a clear, curiosity-driven title that sounds like a hook for TikTok or Instagram Reels (simple, not techy); it must be directly related to the core newsletter topic and unique angle of the specific video; must read like a short world-class curiosity-building copywriting headline that builds a highly intriguing open loop in the reader's mind

- Exactly 4 scenes per video

Each scene must have:

- A photorealistic image prompt written in clear and simple English
- The prompt must be realistic and safe (no violence, nudity, or NSFW content)
- Avoid abstract or conceptual elements like dashboards, charts, graphs, or digital UIs
- The image must be something that Flux.ai can generate easily and accurately
- Prompts should describe realistic physical scenes or objects, such as people, environments, or situations
- Each image must be specific to the voiceover script in that scene – they should visually support the message
- Use concrete subjects: e.g., "a businesswoman typing on a laptop," not "an AI system optimizing workflows"
- NEVER use quotation marks in any part of the prompt.

- A voice script of at least 20 words, written in clear, casual language, like you're explaining it to a friend. NEVER use quotation marks in any part of the script.

- Each video should cover a distinct aspect of the newsletter (e.g., problem, concept, use cases)
- Avoid any complex terms unless you explain them
- Do not come up with any content that isn't directly part of the newsletter content

Output format:

- Never use quotation marks "" for any of the text you generate as this will give me errors in JSON.

- Respond using only the following format and your generated text for each variable, with NO extra text, no quotation marks "", no explanations, and no markdown:

Videol_Title:
Videol_Image1:
Videol_Voice1:
Videol_Image2:
Videol_Voice2:
Videol_Image3:
Videol_Voice3:
Videol_Image4:
Videol_Voice4:

```
Video2_Title:
Video2_Image1:
Video2_Voice1:
Video2_Image2:
Video2_Voice2:
Video2_Image3:
Video2_Voice3:
Video2_Image4:
Video2_Voice4:
```

```
Video3_Title:
Video3_Image1:
Video3_Voice1:
Video3_Image2:
Video3_Voice2:
Video3_Image3:
Video3_Voice3:
Video3_Image4:
Video3_Voice4:
```

Here is the full newsletter text to turn into 3 short videos:
{ { \$('Cleanup2').item.json.cleanedText } }

Remove Quotation Marks from Video Script:

```
const text = $input.first().json.choices[0].message.content;

// Remove all types of quotation marks from the full string
const cleaned = text.replace(/[""'']/g, '');

return [
  {
    json: {
      cleaned_content: cleaned
    }
  }
];
```

Split Variables Function:

```
const input = $input.first().json.cleaned_content;

// Helper function to extract values between markers
function extract(label, text) {
```

```

const regex = new RegExp(`${label}:\s*([^\n]*)`, "i");
const match = text.match(regex);
return match ? match[1].trim() : "";
}

const output = {
  Video1_Title: extract("Video1_Title", input),
  Video1_Image1: extract("Video1_Image1", input),
  Video1_Voice1: extract("Video1_Voice1", input),
  Video1_Image2: extract("Video1_Image2", input),
  Video1_Voice2: extract("Video1_Voice2", input),
  Video1_Image3: extract("Video1_Image3", input),
  Video1_Voice3: extract("Video1_Voice3", input),
  Video1_Image4: extract("Video1_Image4", input),
  Video1_Voice4: extract("Video1_Voice4", input),

  Video2_Title: extract("Video2_Title", input),
  Video2_Image1: extract("Video2_Image1", input),
  Video2_Voice1: extract("Video2_Voice1", input),
  Video2_Image2: extract("Video2_Image2", input),
  Video2_Voice2: extract("Video2_Voice2", input),
  Video2_Image3: extract("Video2_Image3", input),
  Video2_Voice3: extract("Video2_Voice3", input),
  Video2_Image4: extract("Video2_Image4", input),
  Video2_Voice4: extract("Video2_Voice4", input),

  Video3_Title: extract("Video3_Title", input),
  Video3_Image1: extract("Video3_Image1", input),
  Video3_Voice1: extract("Video3_Voice1", input),
  Video3_Image2: extract("Video3_Image2", input),
  Video3_Voice2: extract("Video3_Voice2", input),
  Video3_Image3: extract("Video3_Image3", input),
  Video3_Voice3: extract("Video3_Voice3", input),
  Video3_Image4: extract("Video3_Image4", input),
  Video3_Voice4: extract("Video3_Voice4", input)
};

return [{ json: output }];

```

JSON2Video Credentials:

```

{
  "headers": {
    "x-api-key": "INSERT_YOUR_API"
  }
}

```

Add Text to Video (<https://api.json2video.com/v2/movies>):

(Voices: <https://json2video.com/ai-voices/azure/>)

(Image Models: <https://json2video.com/ai-voices/azure/>)

```
{
  "resolution": "instagram-story",
  "scenes": [
    {
      "elements": [
        {
          "type": "image",
          "model": "flux-schnell",
          "prompt": "{{ $json.Video1_Image1 }}",
          "duration": -2,
          "resize": "cover",
          "zoom": 3,
          "pan": "right"
        },
        {
          "type": "voice",
          "model": "azure",
          "voice": "{{ $('Get Voice').all().find(item =>
item.json.Setting === 'Voice').json.Value }}",
          "text": "{{ $json.Video1_Voice1 }}"
        }
      ]
    },
    {
      "elements": [
        {
          "type": "image",
          "model": "flux-schnell",
          "prompt": "{{ $json.Video1_Image2 }}",
          "duration": -2,
          "resize": "cover",
          "zoom": 3,
          "pan": "left"
        },
        {
          "type": "voice",
          "model": "azure",
          "voice": "{{ $('Get Voice').all().find(item =>
item.json.Setting === 'Voice').json.Value }}",
          "text": "{{ $json.Video1_Voice2 }}"
        }
      ]
    }
  ]
}
```

```

    ]
  },
  {
    "elements": [
      {
        "type": "image",
        "model": "flux-schnell",
        "prompt": "{{ $json.Video1_Image3 }}",
        "duration": -2,
        "resize": "cover",
        "zoom": 3,
        "pan": "top"
      },
      {
        "type": "voice",
        "model": "azure",
        "voice": "{{ $('Get Voice').all().find(item =>
item.json.Setting === 'Voice').json.Value }}",
        "text": "{{ $json.Video1_Voice3 }}"
      }
    ]
  },
  {
    "elements": [
      {
        "type": "image",
        "model": "flux-schnell",
        "prompt": "{{ $json.Video1_Image4 }}",
        "duration": -2,
        "resize": "cover",
        "zoom": 3,
        "pan": "bottom"
      },
      {
        "type": "voice",
        "model": "azure",
        "voice": "{{ $('Get Voice').all().find(item =>
item.json.Setting === 'Voice').json.Value }}",
        "text": "{{ $json.Video1_Voice4 }}"
      }
    ]
  }
],
"elements": [
  {
    "type": "text",
    "text": "{{ $json.Video1_Title }}"
  }
]

```



```

    "settings": {
      "font-family": "Arial",
      "font-size": "60px",
      "color": "#000000",
      "background-color": "#FFFFFF",
      "horizontal-position": "center",
      "vertical-position": "top",
      "margin-top": "100px",
      "word-break": "break-word",
      "overflow-wrap": "break-word",
      "font-weight": "bold",
      "text-align": "center",
      "width": "80%",
      "padding": "0px",
      "line-height": "1.2",
      "margin": "50px 0 0 0",
"padding": "0 10px 0 10px",
"border-radius": "15px"
    }
  },
  {
    "type": "subtitles",
    "settings": {
      "style": "classic",
      "position": "bottom",
      "font-family": "Arial",
      "font-size": "100px",
      "word-color": "#ffd700",
      "shadow-color": "#260B1B",
      "line-color": "#F1E7F4",
      "shadow-offset": 0,
      "box-color": "#260B1B",
      "outline-color": "#000000",
      "outline-width": 8
    }
  },
  {
    "id": "music-track",
    "type": "audio",
    "src": "{{ $('Get Music 1').first().json.data.output.audio_url }}",
    "volume": 0.3,
    "duration": -2
  }
]
}

```

Get video:

`https://api.json2video.com/v2/movies?id={{ $json.project }}`

Video ID:

```
function generateRandomCode(length = 20) {
  const chars =
'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';
  let result = '';
  for (let i = 0; i < length; i++) {
    result += chars[Math.floor(Math.random() * chars.length)];
  }
  return result;
}

return [
  {
    json: {
      code: generateRandomCode()
    }
  }
];
```

Caption Text Generator:

You are writing a TikTok caption. You must follow this format and obey these rules strictly.

Write three caption text using the exact structure of the caption examples below and these three topics:

Topic 1:

```
{{ $('Split1').item.json.Video1_Voice1 }}
{{ $('Split1').item.json.Video1_Voice2 }}
{{ $('Split1').item.json.Video1_Voice3 }}
{{ $('Split1').item.json.Video1_Voice4 }}
```

Topic 2:

```
{{ $('Split1').item.json.Video2_Voice1 }}
{{ $('Split1').item.json.Video2_Voice2 }}
{{ $('Split1').item.json.Video2_Voice3 }}
{{ $('Split1').item.json.Video2_Voice4 }}
```

Topic 3:

```
{{ $('Split1').item.json.Video3_Voice1 }}  
{{ $('Split1').item.json.Video3_Voice2 }}  
{{ $('Split1').item.json.Video3_Voice3 }}  
{{ $('Split1').item.json.Video3_Voice4 }}
```

Keep:

- Same structure and tone
- Similar sentence count and layout

Caption Examples:

Example 1:

10x better ChatGPT outputs with these simple tricks. You can use these in DeepSeek, Claude, Gemini, etc too I just mention ChatGPT because it's well known.

What's your 1 tip for beginners??

#ai #chatgpt #chatgptprompts #artificialintelligence

Example 2:

One of the EASIEST platforms to get started building Ai agents if you have zero prior experience with coding or workflow automations. It's called MindStudio.ai and I like how they have lots of prebuilt Ai agents ready-to-go.

#ai #mindstudio #aiagent #aitools #entrepreneur #greenscreenvideo

Example 3:

Save this ChatGPT tutorial for beginners, walking through the basics and various options! If you're new to AI, stick to ChatGPT and dive in, learning how to use it for your work or personal goals.

- step by step ChatGPT tutorial for beginners?
- new to AI and want to learn ChatGPT?
- basics of AI and ChatGPT guide?

#ai #chatgpt #learning #artificialintelligence
#tiktoklearningcampaign

FINAL OUTPUT FORMAT (no markdown formatting):

Caption_Text1:

Caption_Text2:

Caption_Text3:

DO NOT return any explanations. Only return the rewritten sections.

Split Caption Text:

```
const rawText = $input.first().json.captionText;

const blocks =
$input.first().json.choices[0].message.content.split(/Caption_Text
\d+:/)
  .map(b => b.trim())
  .filter(b => b.length > 0);

const result = blocks.map((block, index) => ({
  json: {
    caption: block,
    captionNumber: index + 1
  }
}));

return result;
```

Cleanup Email Subject:

```
let subject = $('Gmail Trigger').first().json.subject || "";

// Remove multiple "Fwd:" (case-insensitive), optional whitespace,
and emojis
subject = subject
  .replace(/(Fwd:\s*)+/gi, "") // Remove one or more "Fwd:"
  .replace(/[\p{Emoji_Presentation}\p{Extended_Pictographic}]/gu,
  "") // Remove most emojis
  .trim();

return [{ json: { cleanedSubject: subject } }];
```

Google Sheet Save Captions Variable:

```
{{ $('Split').all()[1].json.caption }}
```

TikTok Post Module (upload-post.com):

<https://api.upload-post.com/api/upload>

Authorization

Apikey YOUR_API_KEY

Write Date in Sheet:

```
{{formatDate(now; "  YYYY-MM-DD hh:mm A")}}
```

Make.com Publisher Webhook Response:

```
{
  "platform": "Facebook",
  "postTitle": "{{92.`Post Title`}}",
  "Email": "{{92.`User Email`}}",
  "publishTime": "{{formatDate(now; "YYYY-MM-DD HH:mm:ss")}}"
}
```

Email Publishing Response:

Your video "{{ \$json.body.postTitle }}" was published to {{ \$json.body.platform }}

This is a confirmation message that your video "{{ \$json.body.postTitle }}" was published to {{ \$json.body.platform }} at {{ \$json.body.publishTime }}

Cost Breakdown

Updated Video Workflow Cost Breakdown

Tool	30 videos/mo	60 videos/mo	90 videos/mo	120 videos/mo
n8n	\$24.00	\$24.00	\$24.00	\$24.00
n8n (local)	\$0.00	\$0.00	\$0.00	\$0.00
OpenAI	\$0.65	\$1.29	\$1.94	\$2.58
Cloudinary	\$0.00	\$0.00	\$0.00	\$0.00
JSON2Video	\$5.10	\$10.20	\$15.30	\$20.40
AI Music API	\$0.60	\$1.20	\$1.80	\$2.40
Make.com	\$0.00	\$0.00	\$10.59	\$10.59
Upload-Post.com (TikTok)	\$14.00	\$14.00	\$14.00	\$14.00
Cost per video (n8n webapp)	\$1.48	\$0.85	\$0.75	\$0.62
TOTAL (n8n webapp)	\$44.35	\$50.69	\$67.63	\$73.97
Cost per video (n8n local)	\$0.68	\$0.44	\$0.48	\$0.42
TOTAL (n8n local)	\$20.35	\$26.69	\$43.63	\$49.97

Updated Video Workflow Cost Breakdown (mit flux-pro)

Tool	30 videos/mo	60 videos/mo	90 videos/mo	120 videos/mo
n8n	\$24.00	\$24.00	\$24.00	\$24.00
n8n (local)	\$0.00	\$0.00	\$0.00	\$0.00
OpenAI	\$0.65	\$1.29	\$1.94	\$2.58
Cloudinary	\$0.00	\$0.00	\$0.00	\$0.00
JSON2Video (flux-pro)	\$21.90	\$43.80	\$65.70	\$87.60
AI Music API	\$0.60	\$1.20	\$1.80	\$2.40
Make.com	\$0.00	\$0.00	\$10.59	\$10.59
Upload-Post.com (TikTok)	\$14.00	\$14.00	\$14.00	\$14.00
Cost per video (n8n webapp)	\$2.04	\$1.41	\$1.31	\$1.18
TOTAL (n8n webapp)	\$61.15	\$84.29	\$118.03	\$141.17
Cost per video (n8n local)	\$1.24	\$1.00	\$1.04	\$0.98
TOTAL (n8n local)	\$37.15	\$60.29	\$94.03	\$117.17