

Нейросеть Midjourney

/settings - настройки

/imagine - чтобы вводить свои промты

" :: " — разделяет текст от технической информации

Соотношение объектов " car::80, road::20 "

--hd — высокое разрешение (делает в высоком разрешении все 4 фотки)

--ar 3:2 — соотношение сторон

--no *слово* — без чего-либо

Для качественных промтов:

photorealistic, cinematic composition, cinematic high detail, ultra realistic, cinematic lighting, Shot on 50mm lense, Ultra- Wide Angle, Depth of Field, hyper-detailed, beautifully color-coded, beautifully color graded, Unreal Engine, Cinematic, Color Grading, Editorial Photography, Photography, Photoshoot, Shot on 70mm lense, Depth of Field, DOF, Tilt Blur, Shutter Speed 1/1000, F/22, White Balance, 32k, Super-Resolution, Megapixel, ProPhoto RGB, VR, Lonely, Good, Massive, Halfrear Lighting, Backlight, Natural Lighting, Incandescent, Optical Fiber, Moody Lighting, Cinematic Lighting, Studio Lighting, Soft Lighting, Volumetric, Contre-Jour, Beautiful Lighting, Accent Lighting, Global Illumination, Screen Space Global Illumination, Ray Tracing Global Illumination, Optics, Scattering, Glowing, Shadows, Rough, Shimmering, Ray Tracing Reflections, Lumen Reflections, Screen Space Reflections, Diffraction Grading, Chromatic Aberration, GB, Displacement, Scan Lines, Ray Traced, Ray Tracing Ambient Occlusion, Anti-Aliasing, Fxaa, TAA, RTX, SSAO, Shaders, OpenGL-Shaders, GLSL-Shaders, Post Processing, Post-Production, Cel Shading, Tone Mapping, CGI, VFX, SFX, elegant dynamic pose, photography

Нейросеть ChatGPT скрипт для автоматизации

/**

* Generates text using OpenAI's GPT-3 model

* @param {string} prompt The prompt to feed to the GPT-3 model

* @param {string} cell The cell to append to the prompt
* @param {number} [maxWords=10] The maximum number of words to generate
* @return {string} The generated text
* @customfunction
*/

```
function runOpenAI(prompt, cell, maxWords) {  
  const API_KEY = "ВСТАВИТЬ_СВОЙ_API";  
  maxTokens = 150  
  if (maxWords){maxTokens = maxWords * 0.75}  
  model = "text-davinci-003"  
  prompt = prompt+cell+":",  
  temperature= 0  
  // Set up the request body with the given parameters  
  const requestBody = {  
    "model": model,  
    "prompt": prompt,  
    "temperature": temperature,  
    "max_tokens": maxTokens  
  };  
  console.log(requestBody)  
  // Set up the request options with the required headers  
  const requestOptions = {  
    "method": "POST",  
    "headers": {  
      "Content-Type": "application/json",
```

```
"Authorization": "Bearer "+API_KEY
},
"payload": JSON.stringify(requestBody)
};

// Send the request to the GPT-3 API endpoint for completions
const response =
UrlFetchApp.fetch("https://api.openai.com/v1/completions",
requestOptions);

console.log(response.getContentText())

// Get the response body as a JSON object
const responseBody = JSON.parse(response.getContentText());

let answer= responseBody.choices[0].text

// Return the generated text from the response
return answer
```