```
!pip install diffusers
!pip install Pillow
```

```
Requirement already satisfied: diffusers in /usr/local/lib/python3.10/dist-packages (0.31.0)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.10/dist-packages (from diffusers) (8.5.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from diffusers) (3.16.1)
Requirement already satisfied: huggingface-hub>=0.23.2 in /usr/local/lib/python3.10/dist-packages (from diffusers) (0.27.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from diffusers) (1.26.4)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from diffusers) (2024.11.6)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from diffusers) (2.32.3)
Requirement already satisfied: safetensors>=0.3.1 in /usr/local/lib/python3.10/dist-packages (from diffusers) (0.4.5)
Requirement already satisfied: Pillow in /usr/local/lib/python3.10/dist-packages (from diffusers) (11.0.0)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.23.2->diffusers
Requirement already satisfied: packaging>=20.9 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.23.2->diffusers)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.23.2->diffusers) (6.
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.23.2->diffusers) (4
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.23.2-
Requirement already satisfied: zipp>=3.20 in /usr/local/lib/python3.10/dist-packages (from importlib-metadata->diffusers) (3.21.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->diffusers) (3.4.
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->diffusers) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->diffusers) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->diffusers) (2024.12.14
Requirement already satisfied: Pillow in /usr/local/lib/python3.10/dist-packages (11.0.0)
```



```
import torch
from diffusers import StableDiffusionPipeline, StableDiffusionImg2ImgPipeline
import requests
from PIL import Image
from io import BytesIO

device = 'cuda' if torch.cuda.is_available() else 'cpu'
print(f"Using device: {device}")

sd_model = StableDiffusionPipeline.from_pretrained(
    "runwayml/stable-diffusion-v1-5",
    torch_dtype=torch.float16
).to(device)

sd_img2img = StableDiffusionImg2ImgPipeline.from_pretrained(
    "runwayml/stable-diffusion-v1-5",
    torch_dtype=torch.float16
).to(device)
```

```
Using device: cuda
model_index.json: 100% 541/541 [00:00<00:00, 36.7kB/s]
Fetching 15 files: 100% 15/15 [00:33<00:00, 2.25s/it]
safety_checker/config.json: 100% 4.72k/4.72k [00:00<00:00, 287kB/s]
(…)ature_extractor/preprocessor_config.json: 100% 342/342 [00:00<00:00, 4.81kB/s]
text_encoder/config.json: 100% 617/617 [00:00<00:00, 8.99kB/s]
tokenizer/merges.txt: 100% 525k/525k [00:00<00:00, 853kB/s]
scheduler/scheduler_config.json: 100% 308/308 [00:00<00:00, 2.81kB/s]
tokenizer/special_tokens_map.json: 100% 472/472 [00:00<00:00, 4.20kB/s]
model.safetensors: 100% 1.22G/1.22G [00:22<00:00, 50.5MB/s]
model.safetensors: 100% 492M/492M [00:12<00:00, 49.4MB/s]
diffusion_pytorch_model.safetensors: 100% 3.44G/3.44G [00:31<00:00, 218MB/s]
unet/config.json: 100% 743/743 [00:00<00:00, 7.82kB/s]
tokenizer/tokenizer_config.json: 100% 806/806 [00:00<00:00, 7.27kB/s]
vae/config.json: 100% 547/547 [00:00<00:00, 8.50kB/s]
tokenizer/vocab.json: 100% 1.06M/1.06M [00:01<00:00, 1.60MB/s]
diffusion_pytorch_model.safetensors: 100% 335M/335M [00:10<00:00, 24.2MB/s]
Loading pipeline components...: 100% 7/7 [00:15<00:00, 2.67s/it]
Loading pipeline components...: 100% 7/7 [00:21<00:00, 2.99s/it]
```

Preposições obtidas para exemplos a serem usados nas imagens:

1. Emitir papel-moeda e autorizar uso de terras indígenas
2. Garantir segurança nuclear e regulamentar censos
3. Destinar receita tributária para programas sociais

Estilo Visual

```
prompt_1 = "Create a highly detailed and realistic image of paper currency being issued in a modern printing facility. Show paper bills
image = sd_model(prompt_1, num_inference_steps=10).images[0]

generator = torch.Generator("cuda").manual_seed(1021)

image.save("papel_moeda.png")
print("Image saved as papel_moeda.png")
image
```

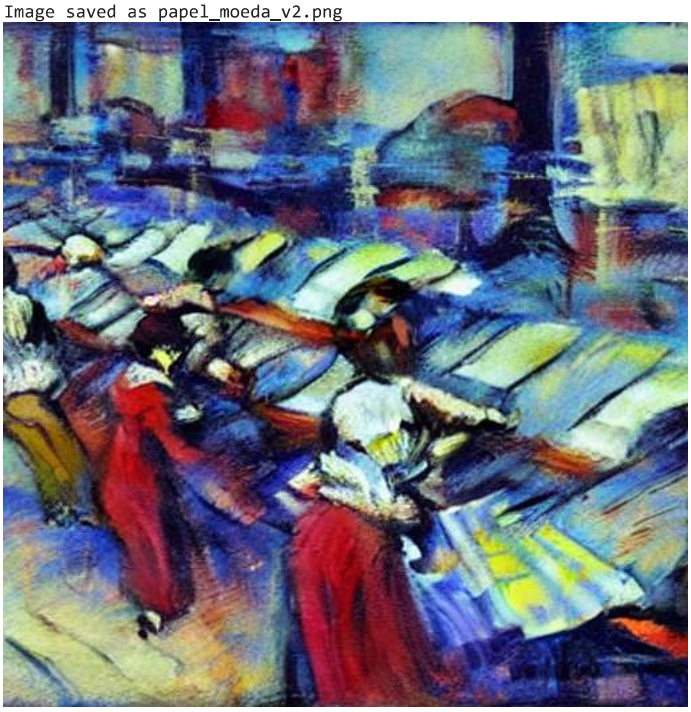
```
Token indices sequence length is longer than the specified maximum sequence length for this model (90 > 77). Running this sequence t
The following part of your input was truncated because CLIP can only handle sequences up to 77 tokens: ['texture of the paper and th
100%                                                  10/10 [00:01<00:00,  6.09it/s]
Image saved as papel_moeda.png
```

```
prompt_1 = "Create an impressionist painting of paper currency being issued in a vibrant and dynamic printing facility. Use broad, textu
image = sd_model(prompt_1, num_inference_steps=10).images[0]

generator = torch.Generator("cuda").manual_seed(1021)

image.save("papel_moeda_v2.png")
print("Image saved as papel_moeda_v2.png")
image
```

```
100%                                                 10/10 [00:01<00:00,  5.96it/s]
Image saved as papel_moeda_v2.png
```



```
prompt_1 = "Create a colorful and exaggerated cartoon-style image of paper currency being issued in a bustling printing facility. Show ca
image = sd_model(prompt_1, num_inference_steps=10).images[0]

generator = torch.Generator("cuda").manual_seed(1021)

image.save("papel_moeda_v2.png")
print("Image saved as papel_moeda_v3.png")
image
```

```
The following part of your input was truncated because CLIP can only handle sequences up to 77 tokens: ['create a lively and fun atm
100%                                                 10/10 [00:01<00:00,  6.00it/s]
Image saved as papel_moeda_v3.png
```

Composição

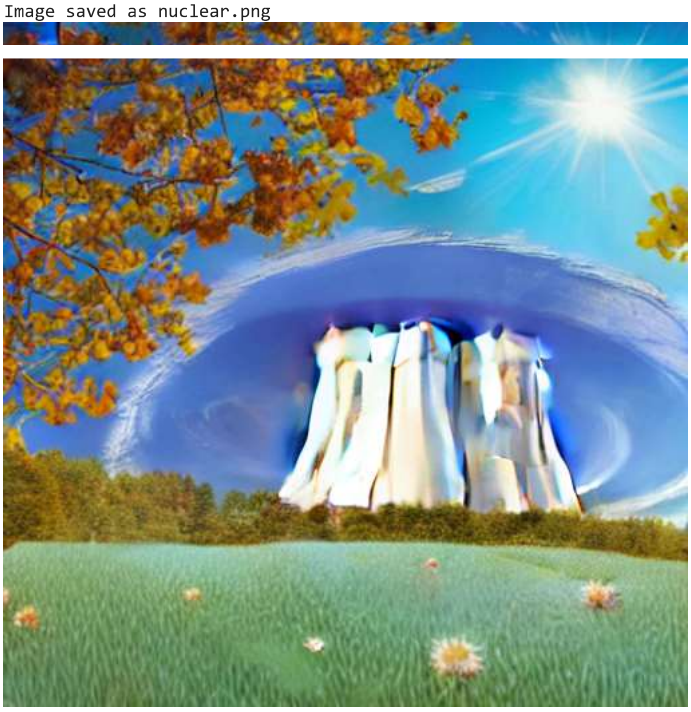
```
prompt_3 = "Create an image of a nuclear power plant placed in the center of the scene, surrounded by a serene and balanced background t

image = sd_model(prompt_3, num_inference_steps=10).images[0]
```

```
generator = torch.Generator("cuda").manual_seed(1021)

image.save("nuclear.png")
print("Image saved as nuclear.png")
image
```

```
100%                                                 10/10 [00:01<00:00,  5.99it/s]
Image saved as nuclear.png
```



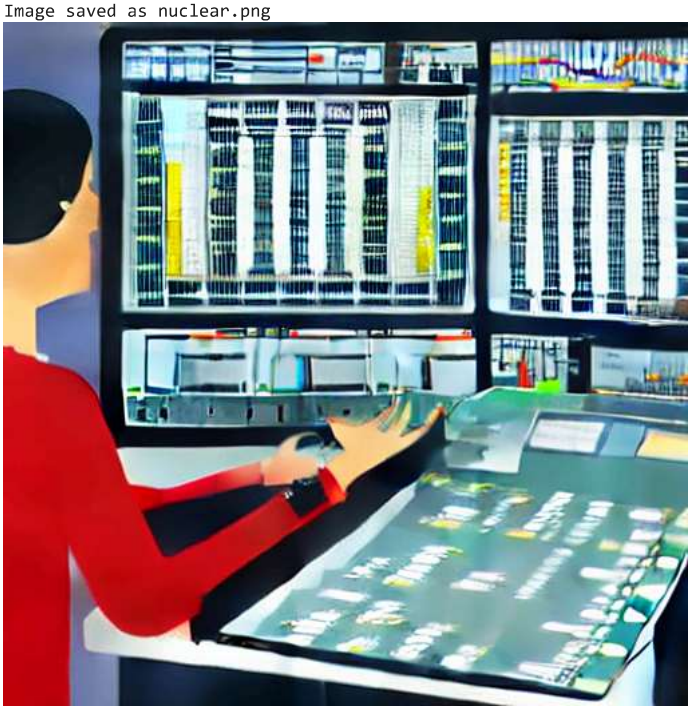
```
prompt_3 = "Create an image illustrating a nuclear power plant using a dynamic perspective: a close-up of a hand operating a secure nucl

image = sd_model(prompt_3, num_inference_steps=10).images[0]

generator = torch.Generator("cuda").manual_seed(1021)

image.save("nuclear_v2.png")
print("Image saved as nuclear.png")
image
```

```
100%                                                 10/10 [00:01<00:00,  5.88it/s]
Image saved as nuclear.png
```



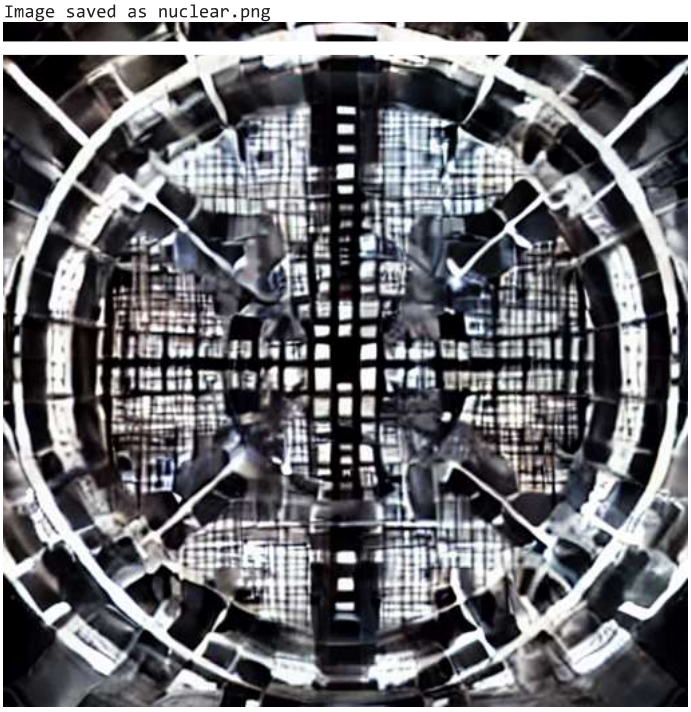
```
prompt_3 = "Create an image of a nuclear vault using a symmetrical layout. The composition should be perfectly balanced, with a clean

image = sd_model(prompt_3, num_inference_steps=10).images[0]
```

```
generator = torch.Generator("cuda").manual_seed(1021)

image.save("nuclear_v3.png")
print("Image saved as nuclear.png")
image
```

```
100%  10/10 [00:01<00:00,  5.97it/s]
Image saved as nuclear.png
```



Negative prompt

```
prompt_3 = "Create an image illustrating social programs, with people interacting in a supportive community setting, vibrant and optimis

image = sd_model(prompt_3, num_inference_steps=10).images[0]

generator = torch.Generator("cuda").manual_seed(1021)

image.save("nuclear_v3.png")
print("Image saved as nuclear.png")
image
```

```
100%  10/10 [00:01<00:00,  5.98it/s]
Image saved as nuclear.png
```