```
#diffusers is a hugging face page for using diffusion models from huggingface hub
!pip install diffusers transformers
```

```
Requirement already satisfied: diffusers in /usr/local/lib/python3.10/dist-packages (0.27.2)
Requirement already satisfied: transformers in /usr/local/lib/python3.10/dist-packages (4.38.2)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.10/dist-packages (from diffusers) (7.1.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from diffusers) (3.13.3)
Requirement already satisfied: huggingface-hub>=0.20.2 in /usr/local/lib/python3.10/dist-packages (from diffusers) (0.20.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from diffusers) (1.25.2)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from diffusers) (2023.12.25)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from diffusers) (2.31.0)
Requirement already satisfied: safetensors>=0.3.1 in /usr/local/lib/python3.10/dist-packages (from diffusers) (0.4.2)
Requirement already satisfied: Pillow in /usr/local/lib/python3.10/dist-packages (from diffusers) (9.4.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from transformers) (24.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (6.0.1)
Requirement already satisfied: tokenizers<0.19,>=0.14 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.15.2)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages (from transformers) (4.66.2)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.20.2->diffusers
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.20.2-
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.10/dist-packages (from importlib-metadata->diffusers) (3.18.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->diffusers) (3.3.
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->diffusers) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->diffusers) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->diffusers) (2024.2.2)
```

```
from diffusers import StableDiffusionPipeline
import matplotlib.pyplot as plt
import torch
```

```
The cache for model files in Transformers v4.22.0 has been updated. Migrating your old cache. This is a one-time only operation. You
    0/0 [00:00<?, ?it/s]
```

```
# !apt-get update
# !apt-get install -y nvidia-driver-470
```

```
model_id1 = "dreamlike-art/dreamlike-diffusion-1.0"

pipe = StableDiffusionPipeline.from_pretrained(model_id1, torch_dtype=torch.float16)
pipe = pipe.to("cuda")
```

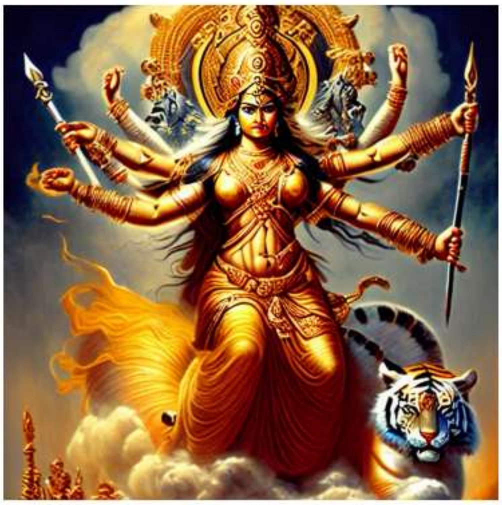
```
Cannot initialize model with low cpu memory usage because `accelerate` was not found
```
pip install accelerate
```
.
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:88: UserWarni
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public mo
  warnings.warn(
model_index.json: 100%                                  511/511 [00:00<00:00, 35.2kB/s]
Fetching 13 files: 100%                                  13/13 [00:16<00:00,  1.69s/it]
model.safetensors: 100%                              246M/246M [00:02<00:00, 109MB/s]
text_encoder/config.json: 100%                          612/612 [00:00<00:00, 12.9kB/s]
tokenizer/merges.txt: 100%                            525k/525k [00:00<00:00, 2.12MB/s]
tokenizer/special_tokens_map.json: 100%                 472/472 [00:00<00:00, 5.99kB/s]
(…)ature_extractor/preprocessor_config.json: 100%       518/518 [00:00<00:00, 6.40kB/s]
tokenizer/tokenizer_config.json: 100%                   807/807 [00:00<00:00, 9.44kB/s]
tokenizer/vocab.json: 100%                          1.06M/1.06M [00:00<00:00, 7.11MB/s]
scheduler/scheduler_config.json: 100%                   341/341 [00:00<00:00, 7.33kB/s]
unet/config.json: 100%                                  901/901 [00:00<00:00, 14.3kB/s]
vae/config.json: 100%                                   577/577 [00:00<00:00, 8.96kB/s]
diffusion_pytorch_model.safetensors: 100%           1.72G/1.72G [00:15<00:00, 64.9MB/s]
diffusion_pytorch_model.safetensors: 100%              167M/167M [00:01<00:00, 129MB/s]
Loading pipeline components...: 100%                     5/5 [00:14<00:00,  1.59s/it]
```

```
prompt2 = """dreamlike, Goddess Durga coming down from the heaven with a weapon in one hand and other hand in the pose of blessing. Ange
reflecting from her eyes.
She is in the form of a soldier and savior coming to protect the world from misery. She is accompanied by her tiger. Make sure to keep i
"""

image = pipe(prompt2).images[0]
print('[PROMPT]: ',prompt2)
plt.imshow(image);
plt.axis('off');
```

```
100%                                              50/50 [00:13<00:00,   3.91it/s]
```

```
[PROMPT]:  dreamlike, Goddess Durga coming down from the heaven with a weapon in one
reflecting from her eyes.
She is in the form of a soldier and savior coming to protect the world from misery. S
```

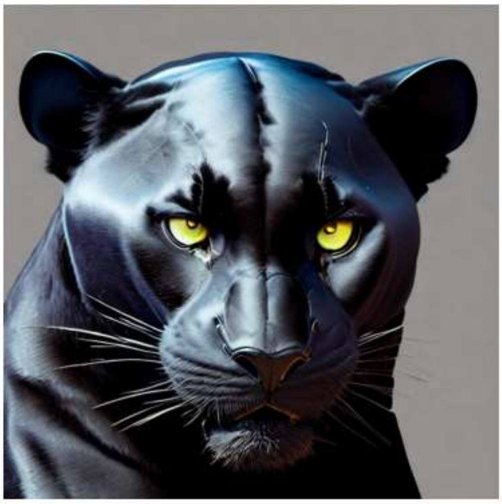


```
prompt2 = """generate a image of blackpanter
"""

image = pipe(prompt2).images[0]
print('[PROMPT]: ',prompt2)
plt.imshow(image);
plt.axis('off');
```

```
100%                                              50/50 [00:12<00:00,   4.10it/s]
```

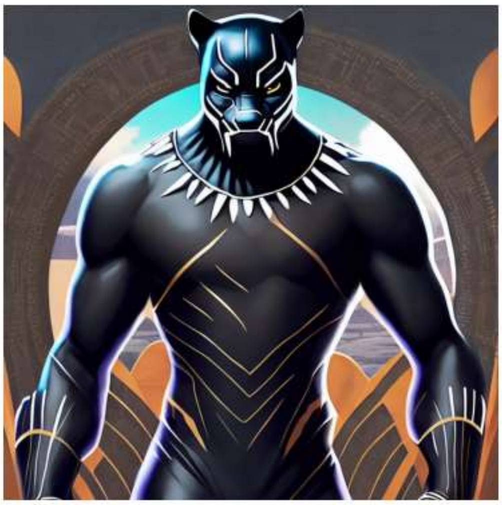
```
[PROMPT]:  generate a image of blackpanter
```



```
prompt2 = """generate a image of blackpanter and wakanda forever
"""


image = pipe(prompt2).images[0]
print('[PROMPT]: ',prompt2)
plt.imshow(image);
plt.axis('off');
```

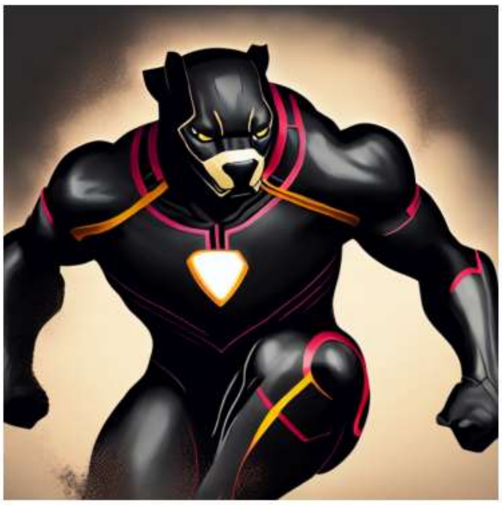
```
100%                                              50/50  [00:12<00:00,    4.03it/s]
[PROMPT]:  generate a image of blackpanter and wakanda forever
```



```
prompt2 = """generate a image of blackpanter and ironman
"""

image = pipe(prompt2).images[0]
print('[PROMPT]: ',prompt2)
plt.imshow(image);
plt.axis('off');
```

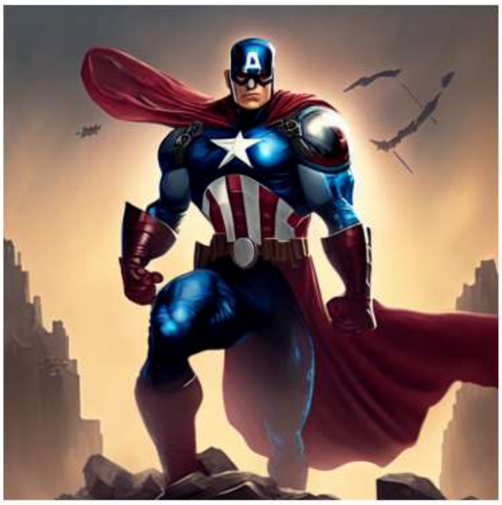
```
100%                                              50/50  [00:12<00:00,    3.96it/s]
[PROMPT]:  generate a image of blackpanter and ironman
```



```
prompt2 = """generate a image of avenger forever
"""

image = pipe(prompt2).images[0]
print('[PROMPT]: ',prompt2)
plt.imshow(image);
plt.axis('off');
```

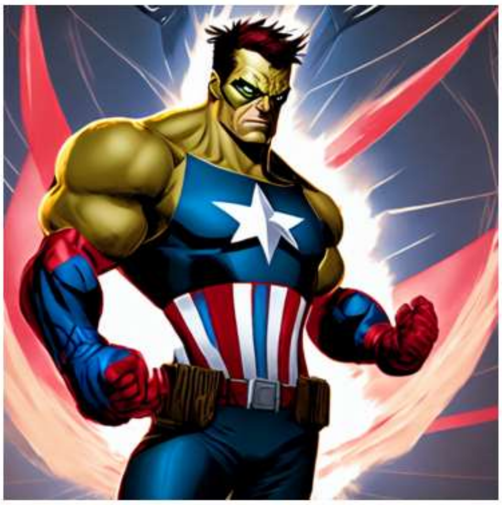
```
100%                                          50/50  [00:12<00:00,    3.82it/s]
[PROMPT]:  generate a image of avenger forever
```



```
prompt2 = """generate a image of marvel avenger
"""

image = pipe(prompt2).images[0]
print('[PROMPT]: ',prompt2)
plt.imshow(image);
plt.axis('off');
```

```
100%                                          50/50  [00:13<00:00,    3.80it/s]
[PROMPT]:  generate a image of marvel avenger
```



```
prompt2 = """generate a image of blackpanter
"""

image = pipe(prompt2).images[0]
print('[PROMPT]: ',prompt2)
plt.imshow(image);
plt.axis('off');
```

```
100%                                        50/50 [00:13<00:00,  3.79it/s]
[PROMPT]:  generate a image of blackpanter
```

```
def generate_image(pipe, prompt, params):
  img = pipe(prompt, **params).images

  num_images = len(img)
  if num_images>1:
    fig, ax = plt.subplots(nrows=1, ncols=num_images)
    for i in range(num_images):
      ax[i].imshow(img[i]);
      ax[i].axis('off');

  else:
    fig = plt.figure()
    plt.imshow(img[0]);
    plt.axis('off');
  plt.tight_layout()
#num_images_per_prompt
params = {'num_inference_steps': 100, 'num_images_per_prompt': 2}

generate_image(pipe, prompt2, params)
```

```
100%                                        100/100 [00:47<00:00,  2.15it/s]
```

```
prompt2 = """generate a image of omar in iron man suit

"""

image = pipe(prompt2).images[0]
print('[PROMPT]: ',prompt2)
plt.imshow(image);
plt.axis('off');
```

```
100%                                          50/50  [00:12<00:00,   3.84it/s]
[PROMPT]:  generate a image of omar in iron man suit
```