✅ Install Required Libraries

```
pip install transformers
pip install torch torchvision torchaudio
pip install pillow
```

- `transformers`: From Hugging Face – gives access to pre-trained models like BLIP.
- `torch`, `torchvision`, `torchaudio`: PyTorch framework, required to run the BLIP model.
- `pillow`: Python Imaging Library to open and handle images.

🎁 Importing Libraries

```
from transformers import BlipProcessor,
BlipForConditionalGeneration
from PIL import Image
import requests
import torch
```

- `BlipProcessor`: handles image preprocessing and tokenization.
- `BlipForConditionalGeneration`: the BLIP model for generating image captions.
- `Image`: to open and manipulate image files.
- `requests`: not used here (can be removed), usually for downloading images from URLs.
- `torch`: not directly used here but needed under the hood by the model.

🖼️ Open and Convert the Image

```
image = Image.open("content.jpg").convert('RGB')
```

- Loads the image file named `"content.jpg"`.
- `.convert('RGB')` ensures the image is in RGB format (required by BLIP).

📥 Load the Pretrained BLIP Model

```
processor = BlipProcessor.from_pretrained("Salesforce/blip-
image-captioning-base")
```

```
model =
BlipForConditionalGeneration.from_pretrained("Salesforce/blip-
image-captioning-base")
```

- Downloads and loads the **processor** and **model** from Hugging Face's `Salesforce/blip-image-captioning-base`.
- `processor`: handles converting the image into tensors.
- `model`: the neural network that generates a caption.

□ Process Image Input

```
inputs = processor(images=image, return_tensors="pt")
```

- Prepares the image as a tensor that the model can understand.
- `return_tensors="pt"` returns PyTorch tensors (`pt` stands for PyTorch).

📝 Generate the Caption

```
out = model.generate(**inputs, max_length=50)
```

- The model generates a caption for the image.
- `max_length=50` limits the caption to 50 tokens (words or sub-words).

🗣□ Decode the Output

```
caption = processor.decode(out[0], skip_special_tokens=True)
```

- Converts the generated tokens (numbers) into readable text.
- `skip_special_tokens=True` removes tokens like `<s>` or `<pad>` from the result.

🔊 Print the Caption

```
print("Generated Caption:", caption)
```

- Displays the generated caption in the terminal.

🖼️ Display the Image with Caption

```
import matplotlib.pyplot as plt

plt.imshow(image)
plt.title(caption)
plt.axis('off')
plt.show()
```

- `matplotlib.pyplot` is used to display the image in a window.
- `plt.imshow(image)`: shows the image.
- `plt.title(caption)`: sets the generated caption as the title.
- `plt.axis('off')`: hides axis ticks.
- `plt.show()`: renders the final display window.

✅ Summary:

This script takes an image (`content.jpg`), processes it using the **BLIP model**, generates a **natural language caption**, prints it, and shows the image with the caption as a title.

Would you like to modify this to accept image URLs or generate multiple captions?