https://chatgpt.com/c/0ebce5dc-a44d-47a8-bf0d-4764b9e8c05f

0

ChatGPT ~





```
from PIL import Image, ImageDraw, ImageFont
from random import randint
import imageio
import os
def create_image(porcentage):
  image = Image.new('RGB', (500, 280), 'white')
  draw = ImageDraw.Draw(image)
  try:
     font = ImageFont.truetype("font/ARIAL.TTF", 20)
  except IOError:
     font = ImageFont.load_default()
  draw.rounded_rectangle([(40, 240), (460, 270)], radius=20,
outline="black", width=3, fill="Gray")
  fill_percentage = max(0, min(porcentage, 100))
  fill_length = 44 + (455 * (fill_percentage / 100))
  draw.rounded_rectangle([(44, 242), (int(fill_length), 268)], radius=20,
outline="red", width=3, fill="red")
  draw.text((250, 243), f"{fill_percentage}%", font=font, fill="black")
  return image
def main():
  lista = []
  random_value = randint(1, 101)
  for frame in range(1, random_value):
     lista.append(create_image(frame))
  output_dir = 'gif'
  os.makedirs(output_dir, exist_ok=True)
  output_path = os.path.join(output_dir, 'de_0_a_100.gif')
  imageio.mimsave(output_path, lista, duration=0.1)
  print(random_value)
if __name__ == "__main__":
```

Este script gera uma animação GIF que ilustra o preenchimento de uma barra de progresso de 0% a um valor aleatório entre 1 e 100%. Aqui está uma explicação detalhada do código:

1. Importações:

- `Image, ImageDraw, ImageFont` do PIL (Python Imaging Library) para criar e manipular imagens.
- `randint` da biblioteca `random` para gerar números aleatórios.

main()

- `imageio` para criar o GIF.
- `os` para manipulação de diretórios e caminhos de arquivos.

2. Função `create_image`:

- Cria uma nova imagem de 500x280 pixels com fundo branco.
- Usa `ImageDraw.Draw` para desenhar na imagem.
- Tenta carregar a fonte Arial. Se não encontrar, usa a fonte padrão.
- Desenha um retângulo arredondado que representa o fundo da barra de progresso.
- Calcula o comprimento do preenchimento com base na porcentagem fornecida.
- Desenha o preenchimento da barra e a porcentagem de preenchimento como texto.

3. Função `main`:

- Cria uma lista para armazenar as imagens.
- Gera um valor aleatório entre 1 e 100.
- Cria imagens para cada valor de 1 até o valor aleatório gerado e as adiciona à lista.
- Cria o diretório de saída, se não existir.
- Usa `imageio.mimsave` para salvar a lista de imagens como um GIF.
 Imprime o valor aleatório gerado.
- Imprime o valor aleatorio gerado.

4. Execução do Script:

 A função `main` é executada se o script for executado como o principal (não importado como módulo).

Aqui está o código com alguns comentários adicionais para ajudar na compreensão:

```
Copiar código
python
from PIL import Image, ImageDraw, ImageFont
from random import randint
import imageio
import os
def create_image(porcentage):
   # Cria uma imagem branca de 500x280 pixels
    image = Image.new('RGB', (500, 280), 'white')
    draw = ImageDraw.Draw(image)
    # Tenta carregar a fonte Arial, señão usa a fonte padrão
    try:
        font = ImageFont.truetype("font/ARIAL.TTF", 20)
    except IOError:
        font = ImageFont.load_default()
    # Desenha o fundo da barra de progresso
    draw.rounded_rectangle([(40, 240), (460, 270)], radius=20, outline="black", width=3, f
    # Calcula o comprimento do preenchimento da barra baseado na porcentagem
    fill_percentage = max(0, min(porcentage, 100))
    fill_length = 44 + (455 * (fill_percentage / 100))
    draw.rounded_rectangle([(44, 242), (int(fill_length), 268)], radius=20, outline="red
    # Adiciona o texto da porcentagem
    draw.text((250, 243), f"{fill_percentage}%", font=font, fill="black")
    return image
def main():
    lista = []
   random_value = randint(1, 101) # Gera um valor aleatório entre 1 e 100
    # Cria imagens para cada valor de 1 até o valor aleatório gerado
    for frame in range(1, random_value):
        lista.append(create_image(frame))
    # Cria o diretório de saída se não existir
   output_dir = 'gif'
   os.makedirs(output_dir, exist_ok=True)
    output_path = os.path.join(output_dir, 'de_0_a_100.gif')
    # Salva a lista de imagens como um GIF
    imageio.mimsave(output_path, lista, duration=0.1)
    print(random_value)
if __name__ == "__main__":
    main()
```

.....

um valor aleatório, que é impresso no final da execução.

```
다 C C A 수^
```

Ao executar este script, ele cria um GIF que mostra uma barra de progresso preenchendo de 0% até