In [24]:

```
import numpy as np
import matplotlib.pyplot as plt
import time
```

In [25]:

```
black_image = np.zeros((500,1280))
```

In [26]:

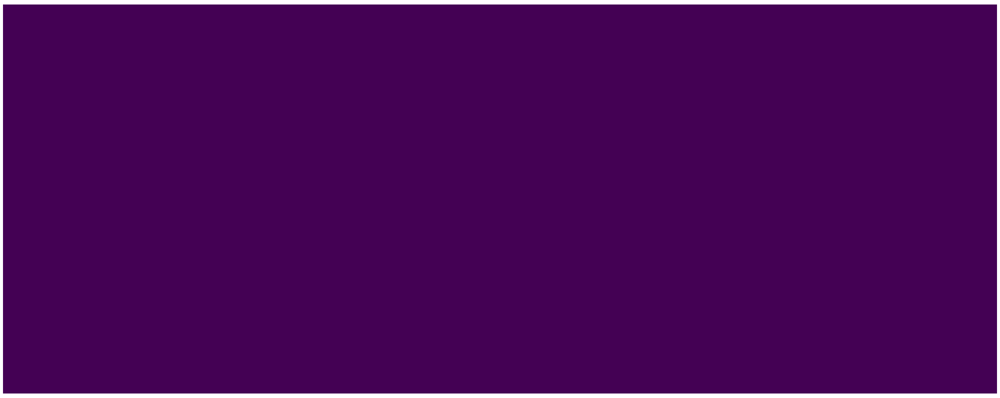
```
black_image.shape
```

Out[26]:

```
(500, 1280)
```

In [27]:

```
plt.imshow(black_image)
plt.axis("off")
plt.show()
```



In [28]:

```
for i in range(500):
    array_created[0:i] = 255
    plt.imshow(black_image)
    time.sleep(0.5)
```

```
---------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
Cell In[28], line 2
      1 for i in range(500):
----> 2     array_created[0:i] = 255
      3     plt.imshow(black_image)
      4     time.sleep(0.5)

NameError: name 'array_created' is not defined
```

In [29]:

```
array_created[0:250,0:250] = 255
```

```
--------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
Cell In[29], line 1
----> 1 array_created[0:250,0:250] = 255

NameError: name 'array_created' is not defined
```

In [ ]:

```
array_created
```

In [ ]:

```
plt.imshow(array_created)
```

In [30]:

```
import numpy as np

# Define the dimensions of the black image
width = 640
height = 480

# Create a black image using a NumPy array
black_image = np.zeros((height, width, 3), dtype=np.uint8)

# Display or use the black image as needed
print(black_image)
```

```
[[[0 0 0]
  [0 0 0]
  [0 0 0]
  ...
  [0 0 0]
  [0 0 0]
  [0 0 0]]

 [[0 0 0]
  [0 0 0]
  [0 0 0]
  ...
  [0 0 0]
  [0 0 0]
  [0 0 0]]

 [[0 0 0]
  [0 0 0]
  [0 0 0]
  ...
  [0 0 0]
  [0 0 0]
  [0 0 0]]

 ...

 [[0 0 0]
  [0 0 0]
  [0 0 0]
  ...
  [0 0 0]
  [0 0 0]
  [0 0 0]]

 [[0 0 0]
  [0 0 0]
  [0 0 0]
  ...
  [0 0 0]
  [0 0 0]
  [0 0 0]]

 [[0 0 0]
  [0 0 0]
  [0 0 0]
  ...
  [0 0 0]
  [0 0 0]
  [0 0 0]]]
```

In [21]:

```
import cv2
import numpy as np
import time

# Load the given image
image_path = 'my.png'  # Replace with the actual path to the image
image = cv2.imread(image_path)

# Get the dimensions of the given image
height, width, _ = image.shape

# Create a black screen with the same dimensions as the given image
black_screen = np.zeros((height, width, 3), dtype=np.uint8)

# Define the time delay between each pixel column pass (in seconds)
delay = 7.9 / 1000  # Convert milliseconds to seconds

width = list(range(width))
width = width[::-1]

# Iterate over each pixel column of the given image
for col in width:
    # Copy the pixel column to the corresponding column of the black screen
    black_screen[:, col] = image[:, col]

    # Display the current frame or perform further processing as needed
    cv2.imshow('Moving Image', black_screen)
    cv2.waitKey(1)  # Display each frame for a very short duration

    # Delay for the specified duration
    time.sleep(delay)

# Wait for a key press to close the window
cv2.waitKey(0)
cv2.destroyAllWindows()
```

In [7]:

```
import cv2
import numpy as np
import time

# Load the given image
image_path = 'my.png'  # Replace with the actual path to the image
image = cv2.imread(image_path)

# Get the dimensions of the given image
height, width, _ = image.shape
```

In [9]:

```
height,width
```

Out[9]:

```
(670, 1030)
```

In [11]:

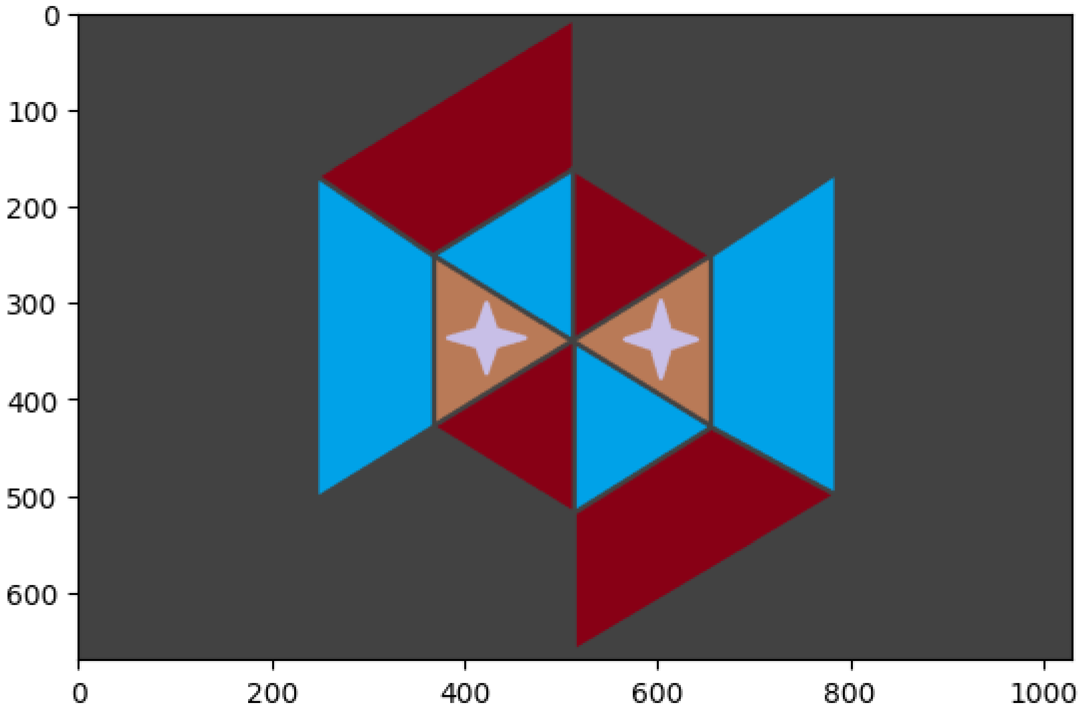
```
# Create a black screen with the same dimensions as the given image
black_screen = np.zeros((height, width, 3), dtype=np.uint8)
```

In [23]:

```
plt.imshow(black_screen)
```

Out[23]:

```
<matplotlib.image.AxesImage at 0x212a9d241f0>
```



In [31]:

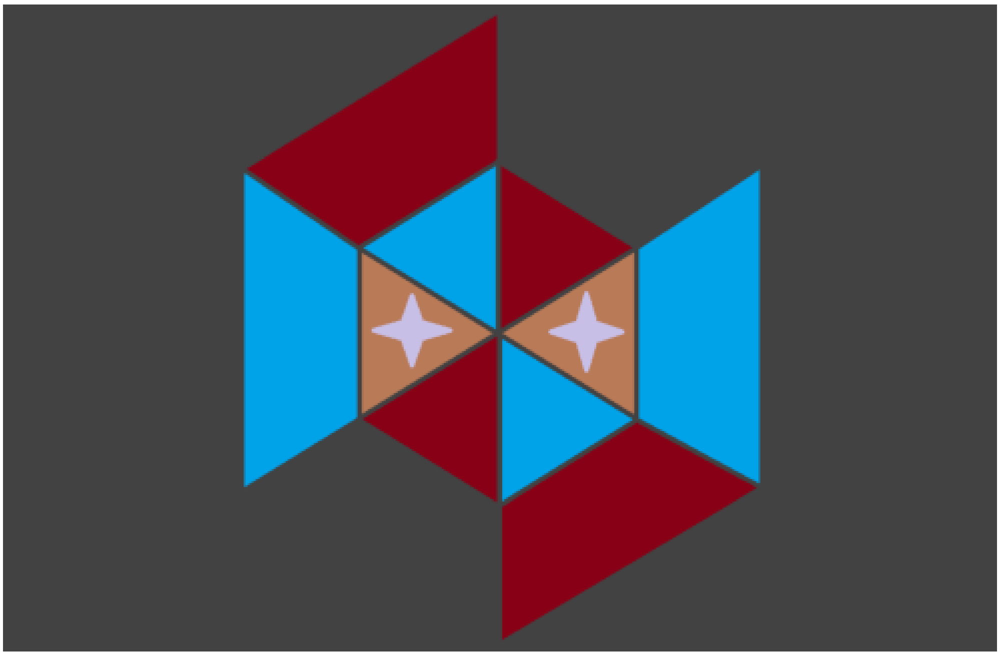
```
image,black_screen
```

```
Out[31]:
```

```
(array([[[66, 66, 66],
         [66, 66, 66],
         [66, 66, 66],
         ...,
         [66, 66, 66],
         [66, 66, 66],
         [66, 66, 66]],

        [[66, 66, 66],
         [66, 66, 66],
         [66, 66, 66],
         ...,
         [66, 66, 66],
         [66, 66, 66],
         [66, 66, 66]],

        [[66, 66, 66],
         [66, 66, 66],
         [66, 66, 66],
         ...,
         [66, 66, 66],
         [66, 66, 66],
         [66, 66, 66]],

        ...,

        [[66, 66, 66],
         [66, 66, 66],
         [66, 66, 66],
         ...,
         [66, 66, 66],
         [66, 66, 66],
         [66, 66, 66]],

        [[66, 66, 66],
         [66, 66, 66],
         [66, 66, 66],
         ...,
         [66, 66, 66],
         [66, 66, 66],
         [66, 66, 66]],

        [[66, 66, 66],
         [66, 66, 66],
         [66, 66, 66],
         ...,
         [66, 66, 66],
         [66, 66, 66],
         [66, 66, 66]]], dtype=uint8),
 array([[[66, 66, 66],
         [66, 66, 66],
         [66, 66, 66],
         ...,
         [66, 66, 66],
         [66, 66, 66],
         [66, 66, 66]],

        [[66, 66, 66],
         [66, 66, 66],
         [66, 66, 66],
         ...,
         [66, 66, 66],
         [66, 66, 66],
         [66, 66, 66]],

        [[66, 66, 66],
         [66, 66, 66],
         [66, 66, 66],
         ...,
         [66, 66, 66],
         [66, 66, 66],
         [66, 66, 66]]
```

In [79]:

```
width1 = 1030
width2 = 1030
for i in range(int(1030/10)):
    width2 = width2 - 10
    plt.imshow(black_screen[:,width2:width1])
plt.axis("off")
plt.show()
```



```
      [[66, 66, 66],
       [66, 66, 66],
       [66, 66, 66],
       ...,
       [66, 66, 66],
       [66, 66, 66],
       [66, 66, 66]]], dtype=uint8))
```

In [78]:

```
width3 = 0
width4 = 0
for i in range(int(1030/10)):
    width4 = width4 + 10
    plt.imshow(black_screen[:,width3:width4])
plt.axis("off")
plt.show()
```