

COLOR MAPPING

```
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.colors import ListedColormap

x = np.random.rand(50)
y = np.random.rand(50)
colors = np.random.rand(50)

cmap = ListedColormap(['red', 'green', 'blue'])

plt.scatter(x, y, c=colors, cmap=cmap)
plt.colorbar()
plt.show()
```

2. # Customizing Colormaps

#You can modify existing colormaps by changing properties like color limits.

```
import numpy as np
from matplotlib import pyplot as plt
import matplotlib as mpl

data = np.random.rand(4, 4)

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(7, 4))
```

```
ax1.imshow(data)
ax1.set_title("Default colormap")

mpl.rc('image', cmap='plasma')

ax2.imshow(data)
ax2.set_title("Modified colormap")

plt.show()
```

COLOR CONTURING

3.# Implementation of matplotlib function

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

feature_x = np.arange(0, 50, 2)
feature_y = np.arange(0, 50, 3)

# Creating 2-D grid of features
[X, Y] = np.meshgrid(feature_x, feature_y)

fig, ax = plt.subplots(1, 1)

Z = np.cos(X / 2) + np.sin(Y / 4)

# plots contour lines
ax.contour(X, Y, Z)

ax.set_title('Contour Plot')
```

```
ax.set_xlabel('feature_x')
```

```
ax.set_ylabel('feature_y')
```

```
plt.show()
```

Example #2: Plotting of contour using `contourf()` which plots filled contours.

```
# Implementation of matplotlib function
```

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
feature_x = np.linspace(-5.0, 3.0, 70)
```

```
feature_y = np.linspace(-5.0, 3.0, 70)
```

```
# Creating 2-D grid of features
```

```
[X, Y] = np.meshgrid(feature_x, feature_y)
```

```
fig, ax = plt.subplots(1, 1)
```

```
Z = X ** 2 + Y ** 2
```

```
# plots filled contour plot
```

```
ax.contourf(X, Y, Z)
```

```
ax.set_title('Filled Contour Plot')
```

```
ax.set_xlabel('feature_x')
```

```
ax.set_ylabel('feature_y')
```

```
plt.show()
```