

Import libraries

In [9]:

```
%matplotlib inline
from tensorflow import keras
from keras.preprocessing import image
import numpy as np
```

Load the saved module

In [10]:

```
model = keras.models.load_model('Downloads/model_data')
```

Load image -> Change to single channel -> Resize image to 28x28

In [11]:

```
from PIL import Image
images=[0]*3
for i in range(3):
    images[i] = Image.open('Downloads/image/'+str(i+1)+'.png').convert('L')
    images[i] = images[i].resize((28,28))
    print(images[i])
```

```
<PIL.Image.Image image mode=L size=28x28 at 0x7FC69CF93BE0>
```

```
<PIL.Image.Image image mode=L size=28x28 at 0x7FC747CF5490>
```

```
<PIL.Image.Image image mode=L size=28x28 at 0x7FC6C00A1550>
```

Convert to numpy and normalize values

In [12]:

```
img_array= np.empty([3, 28, 28])
for i in range(3):
    img_array[i] = np.array(images[i])
    img_array[i] = img_array[i] /255.0
    img_array[i] = img_array[i].reshape(28,28)
```

In [13]:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(2,2))
plt.axis('off')
plt.imshow(img_array[0], cmap=plt.cm.gray)
```

Out[13]:

<matplotlib.image.AxesImage at 0x7fc6c0137e20>



Reshaping

In [14]:

```
img_array = img_array.reshape(3,28,28,1)
img_array.shape
```

Out[14]:

(3, 28, 28, 1)

Predicting values

In [15]:

```
result = model.predict(img_array)
for res in result:
    res = int(np.argmax(res,axis = 0))
    print(res)
```

1
2
4