[1]:

```
import numpy as np
import matplotlib.pyplot as plt
import skimage.data as data
import skimage.segmentation as seg
import skimage.filters as filters
import skimage.draw as draw
import skimage.color as color
```

[2]:

```
def image_show(image, nrows=1, ncols=1, cmap='gray'):
    fig, ax = plt.subplots(nrows=nrows, ncols=ncols, figsize=(2, 2))
    ax.imshow(image, cmap='gray')
    ax.axis('off')
    return fig, ax
```

[3]:

```
# import the image
from skimage import io
image = io.imread('ballon.jpg')
image_show(image);
```

[4]:

```
image_gray = color.rgb2gray(image)
image_show(image_gray);
```

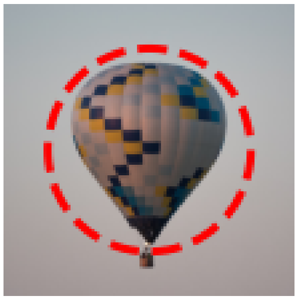
[5]:

```
def circle_points(resolution, center, radius):
    """ Generate points which define a circle on an image.Centre refers to the centre of the circle"""
    radians = np.linspace(0, 2*np.pi, resolution)
    c = center[1] + radius*np.cos(radians)#polar co-ordinates
    r = center[0] + radius*np.sin(radians)

    return np.array([c, r]).T
# Exclude last point because a closed path should not have duplicate points
points = circle_points(200, [400, 400], 280)[:-3]
```

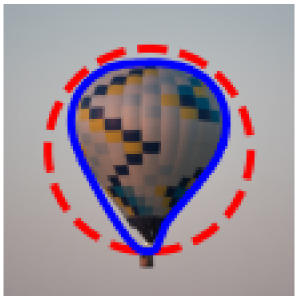
[6]:

```
fig, ax = image_show(image)
ax.plot(points[:, 0], points[:, 1], '--r', lw=3)
```



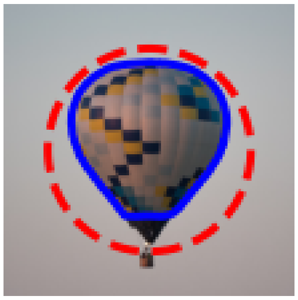
[7]:

```
snake = seg.active_contour(image_gray, points)
fig, ax = image_show(image)
ax.plot(points[:, 0], points[:, 1], '--r', lw=3)
ax.plot(snake[:, 0], snake[:, 1], '-b', lw=3);
```



[8]:

```
snake = seg.active_contour(image_gray, points,alpha=0.06,beta=0.3)
fig, ax = image_show(image)
ax.plot(points[:, 0], points[:, 1], '--r', lw=3)
ax.plot(snake[:, 0], snake[:, 1], '-b', lw=3);
```

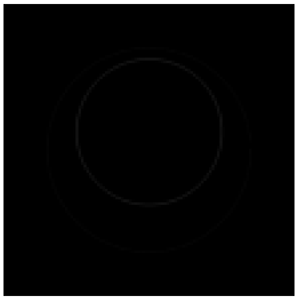


[9]:

```
image_labels = np.zeros(image_gray.shape, dtype=np.uint8)
```

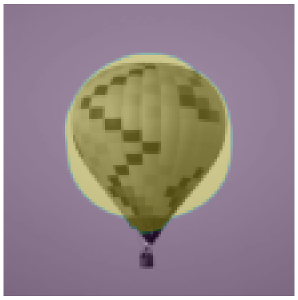
[10]:

```
indices = draw.circle_perimeter(350, 400, 200)#from here
image_labels[indices] = 1
image_labels[points[:, 1].astype(np.int), points[:, 0].astype(np.int)] = 2
image_show(image_labels);
```



[11]:

```
image_segmented = seg.random_walker(image_gray, image_labels)
# Check our results
fig, ax = image_show(image_gray)
ax.imshow(image_segmented == 1, alpha=0.3);
```



[12]:

```
image_segmented = seg.random_walker(image_gray, image_labels, beta = 3000)
# Check our results
fig, ax = image_show(image_gray)
ax.imshow(image_segmented == 1, alpha=0.3);
```