Note: I am Using Scipy v1.1.0. In case of any errors you can install scipy 1.1.0 by the following command: pip install scipy==1.1.0

In [1]:
```
import scipy
scipy.__version__
```

Out[1]:
```
'1.1.0'
```

To download images Click on this url: https://drive.google.com/drive/folders/1pcaTwofZGfoCxZ3Hv2X6vW6xf_1i88eb?usp=sharing (https://drive.google.com/drive/folders/1pcaTwofZGfoCxZ3Hv2X6vW6xf_1i88eb?usp=sharing)

Import Libraries

In [2]:
```
from skimage import data
from scipy.misc import imread, imresize
import numpy as np
from scipy import ndimage
import matplotlib.pyplot as plt
```

Image Negatives Transformation

Formula of Image Negative Transformation is: s =(L-1)-r

Original Image

In [5]:
```
original_image =imread('image_negative.png',True, 'L')    #read image as grey scale image
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: DeprecationWarning: `imread` is deprecate
d!
`imread` is deprecated in SciPy 1.0.0, and will be removed in 1.2.0.
Use ``imageio.imread`` instead.
  """Entry point for launching an IPython kernel.
```

Processed Image

In [8]:
```
processed_img =original_image.copy()
```

In [9]:
```
processed_img =processed_img.astype(np.uint8)

print(processed_img.dtype)
processed_img=(256-1)-original_image
print(np.amin(processed_img),np.amax(processed_img))
```

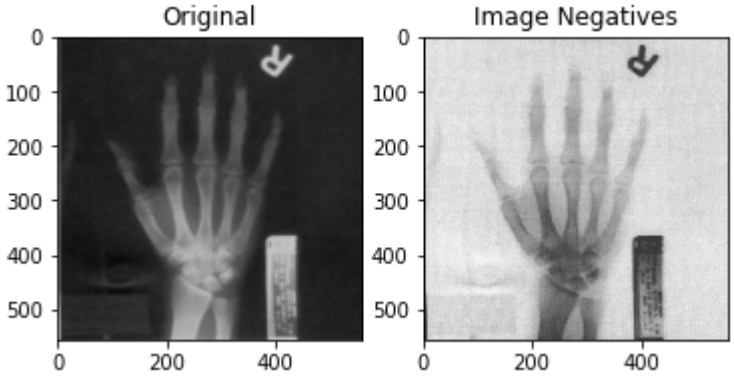
```
uint8
0.0 236.0
```

In [10]:
```
processed_img[processed_img<0]=0
processed_img[processed_img>=255]=255
```

In [11]:
```
fig, axes = plt.subplots(1, 2)
ax = axes.ravel()

ax[0].imshow(original_image,cmap=plt.cm.gray,interpolation='bilinear')
ax[0].set_title("Original")
ax[1].imshow(processed_img, cmap=plt.cm.gray,interpolation='bilinear')
ax[1].set_title("Image Negatives")
plt.imshow(processed_img,cmap=plt.cm.gray)
```

Out[11]:
```
<matplotlib.image.AxesImage at 0x2c4e43d6908>
```



Demo of Array

In [13]:
```
print(original_image[1:5,1:5])
print(processed_img[1:5,1:5])
```

```
[[254. 237. 175. 125.]
 [252. 235. 167. 103.]
 [251. 232. 159.  85.]
 [254. 231. 148.  72.]]
[[  1.  18.  80. 130.]
 [  3.  20.  88. 152.]
 [  4.  23.  96. 170.]
 [  1.  24. 107. 183.]]
```