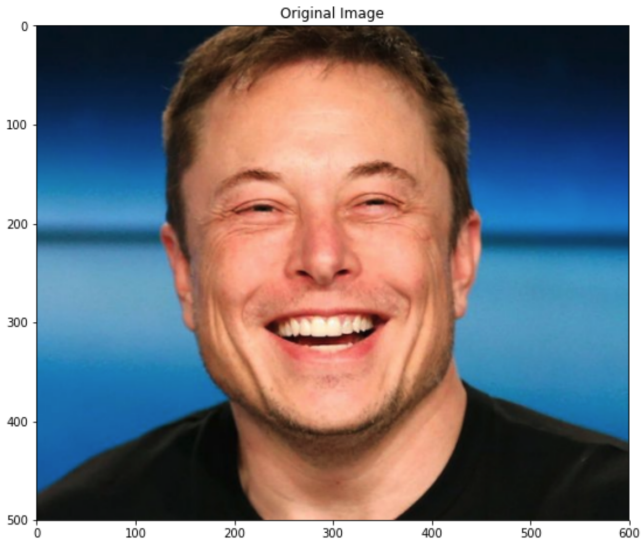
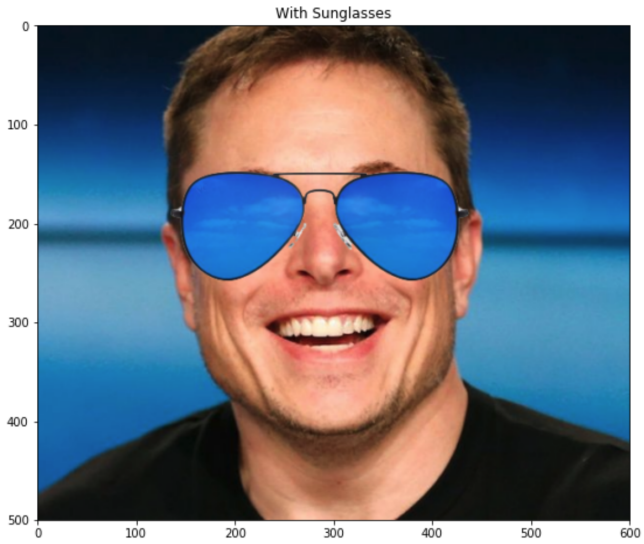
Application: Sunglass Filter

In this section, we will create a simple and fun application using some of the topics we have learned so far.

We will put a sunglass on the face of elon musk and make him look cool (as if he doesn't look so already!).





In [1]:
```
# Import libraries
import cv2
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
import matplotlib
matplotlib.rcParams['figure.figsize'] = (6.0, 6.0)
matplotlib.rcParams['image.cmap'] = 'gray'
```

Load the Face and sunglass Images

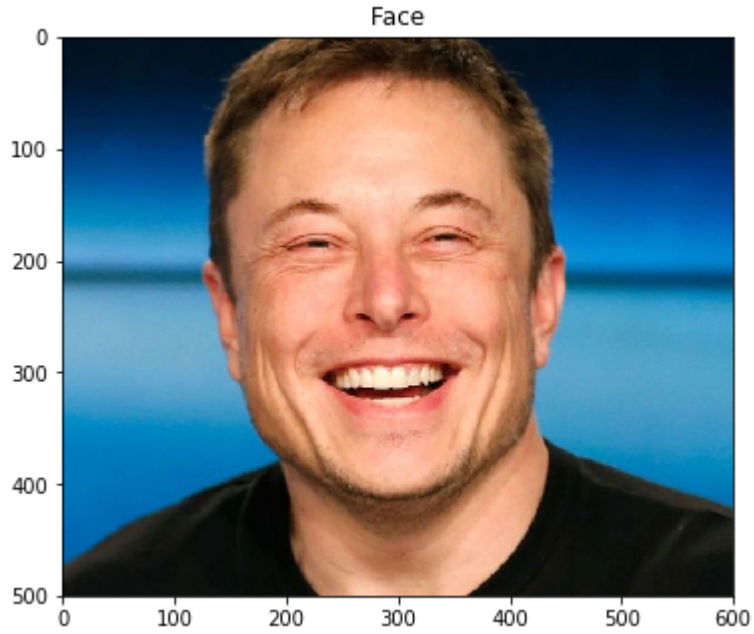
First of all, we will need the two images:

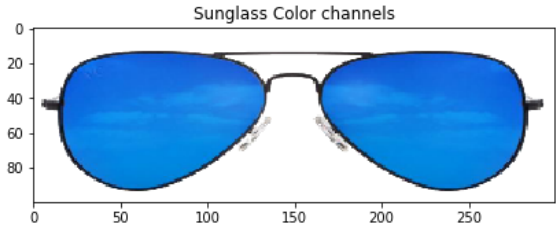
1. Imag of Elon Musk we have been using throughout
2. A png image of a sunglass with an alpha channel

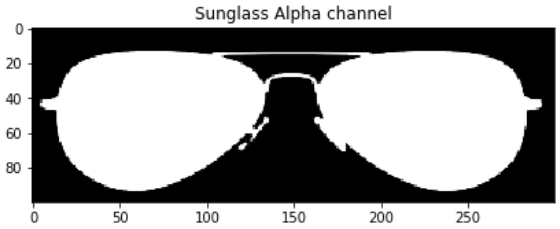
Let us load the images and have a look at them.

In [3]:
```
# Load the Face Image
faceImagePath = "musk.jpg"
faceImage = cv2.imread(faceImagePath)

faceImage = cv2.resize(faceImage,(600,500))

plt.imshow(faceImage[:,:,::-1]);plt.title("Face")

# Load the Sunglass image with Alpha channel
# (http://pluspng.com/sunglass-png-1104.html)
glassimagePath = "sunglass.png"
glassPNG = cv2.imread(glassimagePath,-1)

# Resize the image to fit over the eye region
glassPNG = cv2.resize(glassPNG,(300,100))
print("image Dimension ={}".format(glassPNG.shape))

# Separate the Color and alpha channels
glassBGR = glassPNG[:,:,0:3]
glassMask1 = glassPNG[:,:,3]

# Display the images for clarity
plt.figure(figsize=[15,15])
plt.subplot(121);plt.imshow(glassBGR[:,:,::-1]);plt.title('Sunglass Color chan
nels');
plt.subplot(122);plt.imshow(glassMask1,cmap='gray');plt.title('Sunglass Alpha
 channel');
```

```
image Dimension =(100, 300, 4)
```







Find the eye region

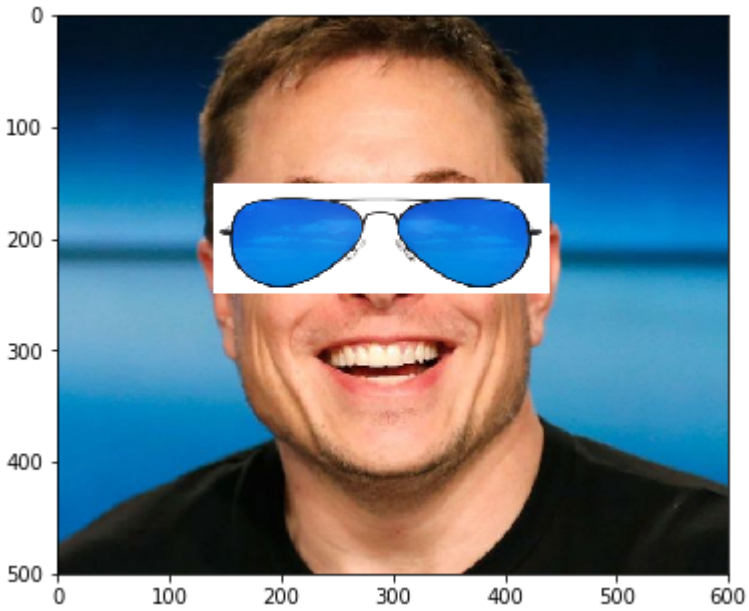
From the face image, we can see that the eye region lies approximately from row #150 to row #250 and col #140 to col #440. Thus, this will be region where the sunglass is to be placed.

Using Naive Replace

The easiest way that might have come to your mind would be to just replace the eye region with that of the sunglass. Let us give it a try.

In [4]:
```
# Make a copy
faceWithGlassesNaive = faceImage.copy()

# Replace the eye region with the sunglass image
faceWithGlassesNaive[150:250,140:440]=glassBGR

plt.imshow(faceWithGlassesNaive[...,::-1])
```

Out[4]: <matplotlib.image.AxesImage at 0x208c8096dd8>



You can see that the output is not even close. This is because the BGR image is opaque and will either have a black or white rectangle around the sunglass.

We need to use a mask along with the sunglass image in order to get the desired output. Let us see how it can be done with very simple steps using the tools we have learned till now.

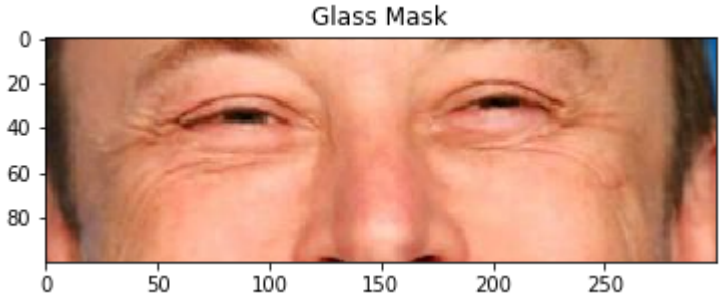
Using Arithmetic Operations and Alpha Mask

In order to put the sunglass on top of the eye region, we need to follow these steps:

1. Create a binary mask with 3-channels using the single channel mask.
2. Extract the eye region from the face image
3. Multiply the Mask with the sunglass to get the masked sunglass
4. Multiply the negative of Mask with the eye region to create a hole in the eye region for the sunglass to be placed.
5. Add the masked sunglass and eye regions to get the combined eye region with the sunglass.
6. Replace the eye region in the original image with that of the output we got in the previous step. This is the final output

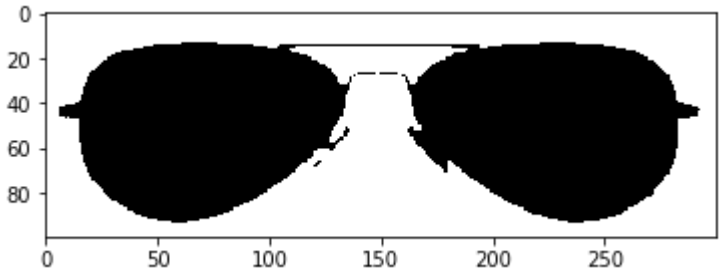
In [5]:
```
glassMask_1 = cv2.merge((glassMask1,glassMask1,glassMask1))
glassMask_1 = np.uint8(glassMask_1/255)
#plt.imshow(1 - glassMask_1[:,:,2]);plt.title("Glass Mask")

# Get the eye region from the face image
eyeROI= faceImage[150:250,140:440]
plt.imshow(eyeROI[:,:,::-1]);plt.title("Glass Mask")
```

Out[5]:
```
Text(0.5, 1.0, 'Glass Mask')
```



In [6]:
```
plt.imshow(np.float32(1- glassMask_1[:,:,1]))
```

Out[6]:
```
<matplotlib.image.AxesImage at 0x208c7c4bc18>
```

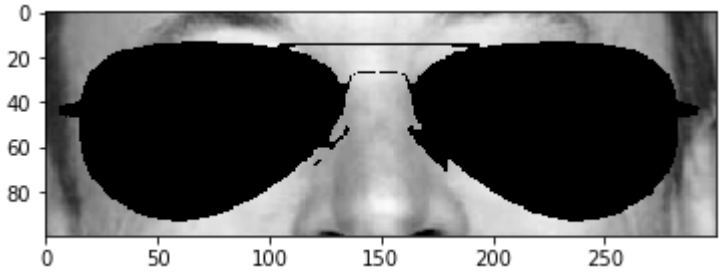


In [7]:
```
1 - glassMask_1[:,:,1][20]
```

Out[7]:
```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 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, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 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, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], dtype=uint8)
```

In [8]:
```
plt.imshow(np.float32(eyeROI[:,:,1])*np.float32(1- glassMask_1[:,:,1]))
```

Out[8]:
```
<matplotlib.image.AxesImage at 0x208c7ca56a0>
```

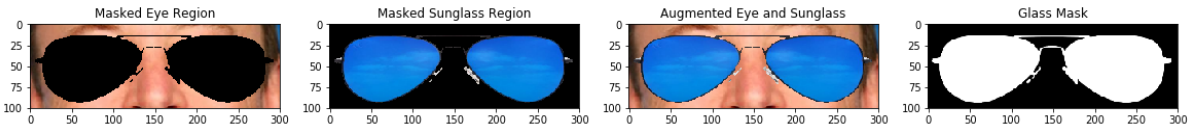


In [9]:
```
a = np.array([[1,2], [3,4]])
b = np.array([[1,2], [3,4]])
a*b
```

Out[9]:
```
array([[ 1,  4],
       [ 9, 16]])
```

In [11]:
```
# Make the dimensions of the mask same as the input image.
# Since Face Image is a 3-channel image, we create a 3 channel image for the m
ask
glassMask = cv2.merge((glassMask1,glassMask1,glassMask1))

# Make the values [0,1] since we are using arithmetic operations
glassMask = np.uint8(glassMask/255)

# Make a copy
faceWithGlassesArithmetic = faceImage.copy()

# Get the eye region from the face image
eyeROI= faceWithGlassesArithmetic[150:250,140:440]

# Use the mask to create the masked eye region
maskedEye = cv2.multiply(eyeROI,(1-  glassMask ))

# Use the mask to create the masked sunglass region
maskedGlass = cv2.multiply(glassBGR,glassMask)

# Combine the Sunglass in the Eye Region to get the augmented image
eyeRoiFinal = cv2.add(maskedEye, maskedGlass)

# Display the intermediate results
plt.figure(figsize=[20,20])
plt.subplot(141);plt.imshow(maskedEye[...,::-1]);plt.title("Masked Eye Region"
)
plt.subplot(142);plt.imshow(maskedGlass[...,::-1]);plt.title("Masked Sunglass
 Region")
plt.subplot(143);plt.imshow(eyeRoiFinal[...,::-1]);plt.title("Augmented Eye an
d Sunglass")
plt.subplot(144);plt.imshow(glassMask1);plt.title("Glass Mask")
```

Out[11]: Text(0.5, 1.0, 'Glass Mask')



In [12]:
```
# Replace the eye ROI with the output from the previous section
faceWithGlassesArithmetic[150:250,140:440]=eyeRoiFinal

# Display the final result
plt.figure(figsize=[20,20]);
plt.subplot(121);plt.imshow(faceImage[:,:,::-1]); plt.title("Original Image");
plt.subplot(122);plt.imshow(faceWithGlassesArithmetic[:,:,::-1]);plt.title("Wi
th Sunglasses");
```