In [1]:
```
import kraken
help(kraken)
```

```
Help on package kraken:

NAME
    kraken - entry point for kraken functionality

PACKAGE CONTENTS
    binarization
    ketos
    kraken
    lib (package)
    linegen
    pageseg
    repo
    rpred
    serialization
    transcribe

DATA
    absolute_import = _Feature((2, 5, 0, 'alpha', 1), (3, 0, 0, 'alpha', 0...
    division = _Feature((2, 2, 0, 'alpha', 2), (3, 0, 0, 'alpha', 0), 8192...
    print_function = _Feature((2, 6, 0, 'alpha', 2), (3, 0, 0, 'alpha', 0)...

FILE
    /opt/conda/lib/python3.7/site-packages/kraken/__init__.py
```

In [2]:
```
from kraken import pageseg
help(pageseg)
```

```
Help on module kraken.pageseg in kraken:

NAME
    kraken.pageseg

DESCRIPTION
    kraken.pageseg
    ~~~~~~~~~~~~~~

    Layout analysis and script detection methods.

FUNCTIONS
    detect_scripts(im, bounds, model='/opt/conda/lib/python3.7/site-packages/kr
aken/script.mlmodel', valid_scripts=None)
        Detects scripts in a segmented page.

        Classifies lines returned by the page segmenter into runs of scripts/wr
iting systems.

        Args:
            im (PIL.Image): A bi-level page of mode '1' or 'L'
            bounds (dict): A dictionary containing a 'boxes' entry with a list
of
                           coordinates (x0, y0, x1, y1) of a text line in the i
mage
                           and an entry 'text_direction' containing
                           'horizontal-lr/rl/vertical-lr/rl'.
            model (str): Location of the script classification model or None fo
r default.
            valid_scripts (list): List of valid scripts.

        Returns:
            {'script_detection': True, 'text_direction': '$dir', 'boxes':
            [[(script, (x1, y1, x2, y2)),...]]}: A dictionary containing the te
xt
            direction and a list of lists of reading order sorted bounding boxe
s
            under the key 'boxes' with each list containing the script segmenta
tion
            of a single line. Script is a ISO15924 4 character identifier.

        Raises:
            KrakenInvalidModelException if no clstm module is available.

    segment(im, text_direction='horizontal-lr', scale=None, maxcolseps=2, black
_colseps=False, no_hlines=True, pad=0, mask=None)
        Segments a page into text lines.

        Segments a page into text lines and returns the absolute coordinates of
        each line in reading order.

        Args:
            im (PIL.Image): A bi-level page of mode '1' or 'L'
```

```
            text_direction (str): Principal direction of the text
                                  (horizontal-lr/rl/vertical-lr/rl)
            scale (float): Scale of the image
            maxcolseps (int): Maximum number of whitespace column separators
            black_colseps (bool): Whether column separators are assumed to be
                                  vertical black lines or not
            no_hlines (bool): Switch for horizontal line removal
            pad (int or tuple): Padding to add to line bounding boxes. If int t
he
                                  same padding is used both left and right. If a
                                  2-tuple, uses (padding_left, padding_right).
            mask (PIL.Image): A bi-level mask image of the same size as `im` wh
ere
                                0-valued regions are ignored for segmentation
                                purposes. Disables column detection.

        Returns:
            {'text_direction': '$dir', 'boxes': [(x1, y1, x2, y2),...]}: A
            dictionary containing the text direction and a list of reading orde
r
            sorted bounding boxes under the key 'boxes'.

        Raises:
            KrakenInputException if the input image is not binarized or the tex
t
            direction is invalid.

DATA
    __all__ = ['segment', 'detect_scripts']

FILE
    /opt/conda/lib/python3.7/site-packages/kraken/pageseg.py
```

In [4]:
```
from PIL import Image
im=Image.open('readonly/two_col.png')
display(im)
bounding_boxes=pageseg.segment(im.convert('1'))['boxes']
print(bounding_boxes)
```

CALEB CHADWELL
Daily Staff Reporter

In a statement Tuesday, the Department of Public Safety and Security wrote that DPSS was not aware until Saturday afternoon of an assault on a University lecturer last week, referred in testimony before Ann Arbor City Council Monday.

Khita Whyatt, lecturer of dance in the University of Michigan's School of Music, Theatre & Dance, said in an interview after her testimony on the incident that she did not immediately call the police because she was so shocked, but her department chair contacted the DPSS. Two days after the incident, Whyatt said she was interviewed by two DPSS officers. During her testimony to Council, she called on the University to release a crime report about how she was knocked down and intimidated by unknown assailants. The event follows similar incidents where crime alerts had not been released. The University has released two crime alerts of hate crimes on campus over the past two weeks.

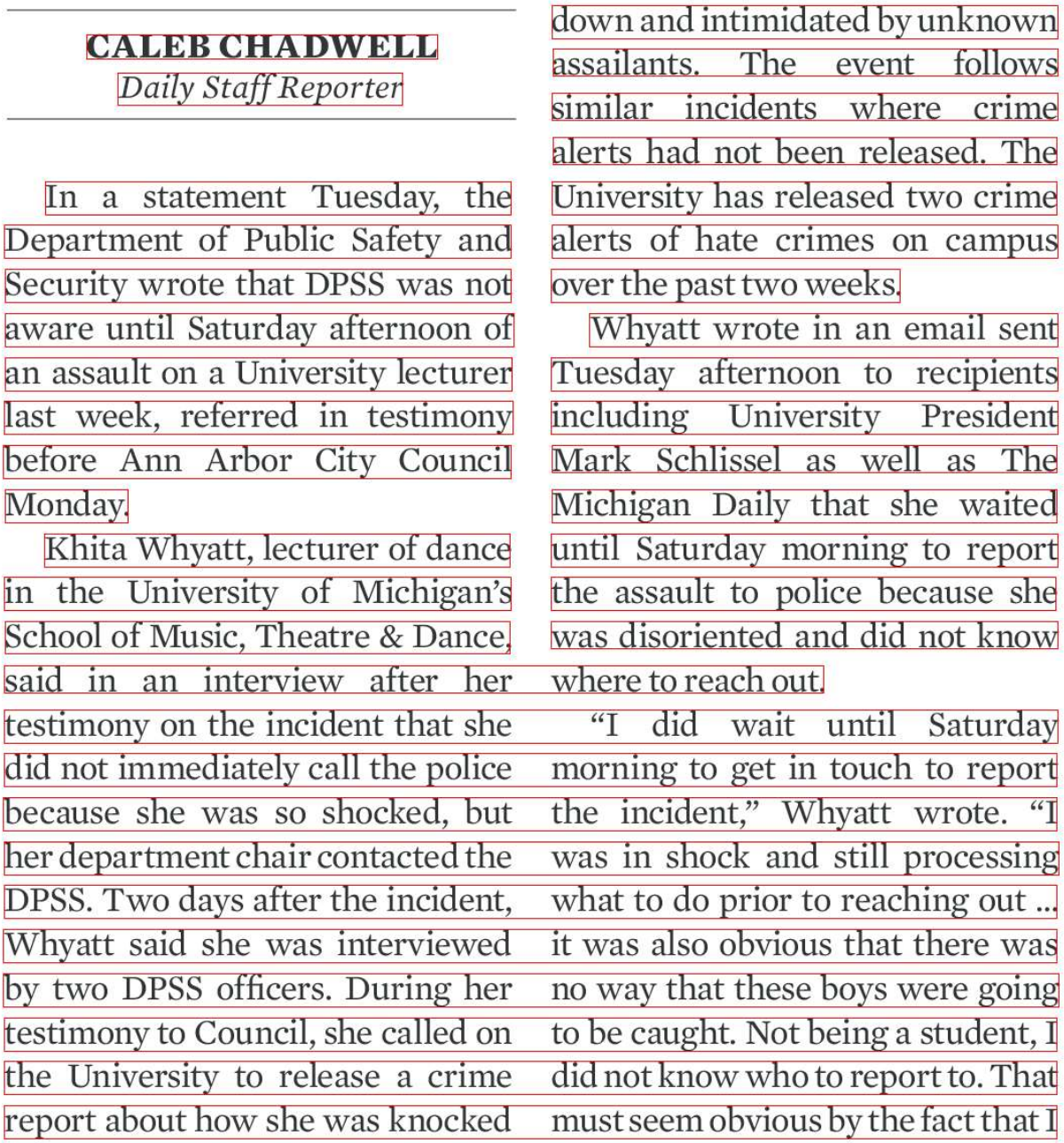
Whyatt wrote in an email sent Tuesday afternoon to recipients including University President Mark Schlissel as well as The Michigan Daily that she waited until Saturday morning to report the assault to police because she was disoriented and did not know where to reach out.

"I did wait until Saturday morning to get in touch to report the incident," Whyatt wrote. "I was in shock and still processing what to do prior to reaching out ... it was also obvious that there was no way that these boys were going to be caught. Not being a student, I did not know who to report to. That must seem obvious by the fact that I

```
[[100, 50, 449, 74], [131, 88, 414, 120], [59, 196, 522, 229], [18, 239, 522, 2
72], [19, 283, 522, 316], [19, 327, 525, 360], [19, 371, 523, 404], [18, 414, 5
24, 447], [17, 458, 522, 491], [19, 502, 141, 535], [58, 546, 521, 579], [18, 5
89, 522, 622], [19, 633, 521, 665], [563, 21, 1066, 54], [564, 64, 1066, 91],
[563, 108, 1066, 135], [564, 152, 1065, 179], [563, 196, 1065, 229], [563, 239,
1066, 272], [562, 283, 909, 316], [600, 327, 1066, 360], [562, 371, 1066, 404],
[562, 414, 1066, 447], [563, 458, 1065, 485], [563, 502, 1065, 535], [562, 546,
1066, 579], [562, 589, 1064, 622], [562, 633, 1066, 660], [18, 677, 833, 704],
[18, 721, 1066, 754], [18, 764, 1065, 797], [17, 808, 1065, 841], [18, 852, 106
7, 885], [18, 895, 1065, 928], [17, 939, 1065, 972], [17, 983, 1067, 1016], [1
8, 1027, 1065, 1060], [18, 1070, 1065, 1103], [18, 1114, 1065, 1147]]
```

In [6]:
```
def show_boxes(img):
    '''Modifies the passed image to show a series of bounding boxes on an image a
    :param img: A PIL.Image object
    :return img:The modifies PIL.Image object
    '''
    from PIL import ImageDraw
    drawing_object=ImageDraw.Draw(img)
    bounding_boxes=pageseg.segment(img.convert('1'))['boxes']

    for box in bounding_boxes:
        drawing_object.rectangle(box, fill=None, outline='red')
    return img
display(show_boxes(Image.open('readonly/two_col.png')))
```



CALEB CHADWELL

Daily Staff Reporter

In a statement Tuesday, the Department of Public Safety and Security wrote that DPSS was not aware until Saturday afternoon of an assault on a University lecturer last week, referred in testimony before Ann Arbor City Council Monday.

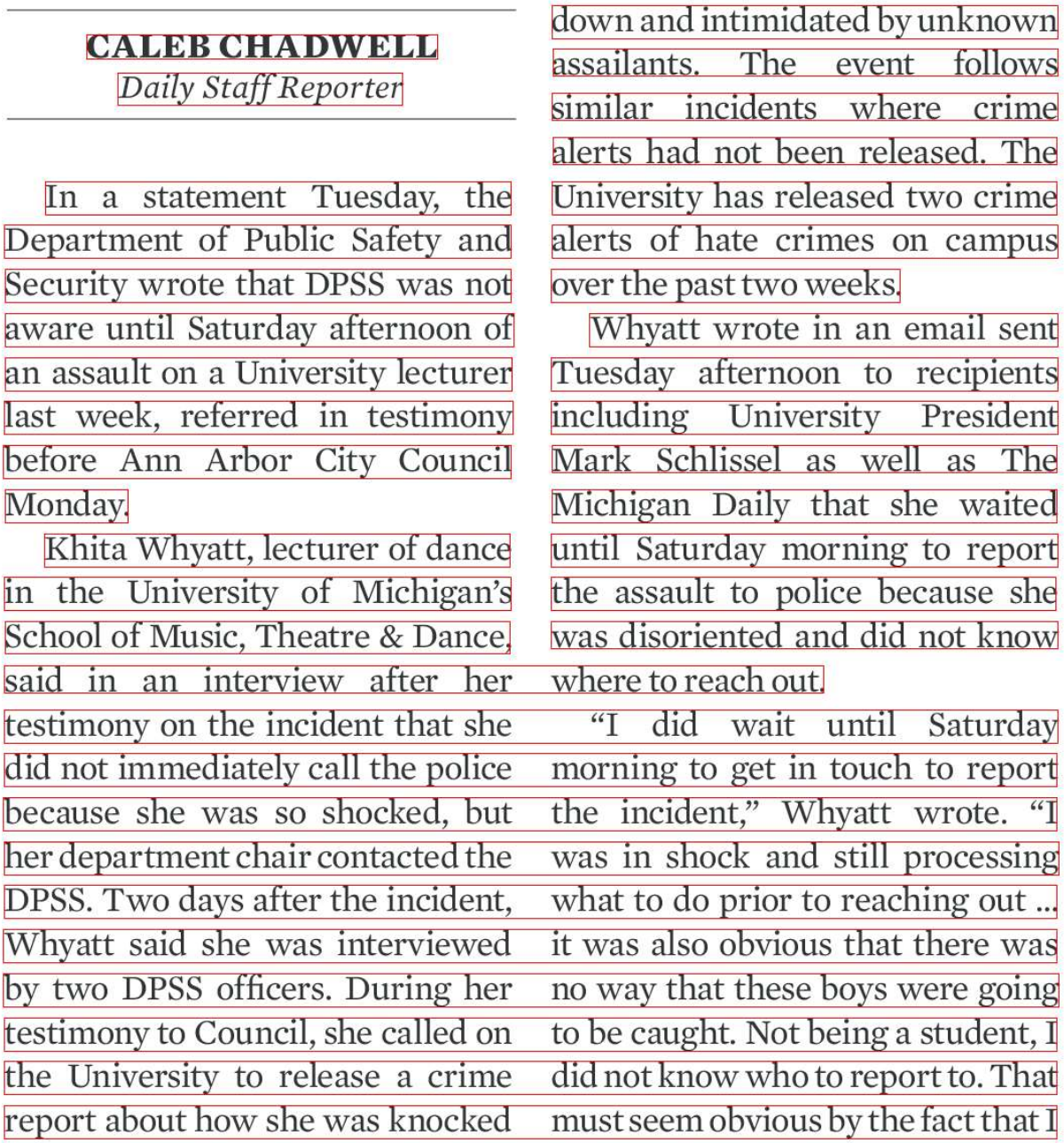
Khita Whyatt, lecturer of dance in the University of Michigan's School of Music, Theatre & Dance, said in an interview after her testimony on the incident that she did not immediately call the police because she was so shocked, but her department chair contacted the DPSS. Two days after the incident, Whyatt said she was interviewed by two DPSS officers. During her testimony to Council, she called on the University to release a crime report about how she was knocked down and intimidated by unknown assailants. The event follows similar incidents where crime alerts had not been released. The University has released two crime alerts of hate crimes on campus over the past two weeks.

Whyatt wrote in an email sent Tuesday afternoon to recipients including University President Mark Schlissel as well as The Michigan Daily that she waited until Saturday morning to report the assault to police because she was disoriented and did not know where to reach out.

"I did wait until Saturday morning to get in touch to report the incident," Whyatt wrote. "I was in shock and still processing what to do prior to reaching out ... it was also obvious that there was no way that these boys were going to be caught. Not being a student, I did not know who to report to. That must seem obvious by the fact that I

In [7]:
```
def show_boxes(img):
    '''Modifies the passed image to show a series of bounding boxes on an image a
    :param img: A PIL.Image object
    :return img:The modifies PIL.Image object
    '''
    from PIL import ImageDraw
    drawing_object=ImageDraw.Draw(img)
    bounding_boxes=pageseg.segment(img.convert('1'),black_colseps=True)['boxes']

    for box in bounding_boxes:
        drawing_object.rectangle(box, fill=None, outline='red')
    return img
display(show_boxes(Image.open('readonly/two_col.png')))
```



CALEB CHADWELL

Daily Staff Reporter

In a statement Tuesday, the Department of Public Safety and Security wrote that DPSS was not aware until Saturday afternoon of an assault on a University lecturer last week, referred in testimony before Ann Arbor City Council Monday.

Khita Whyatt, lecturer of dance in the University of Michigan's School of Music, Theatre & Dance, said in an interview after her testimony on the incident that she did not immediately call the police because she was so shocked, but her department chair contacted the DPSS. Two days after the incident, Whyatt said she was interviewed by two DPSS officers. During her testimony to Council, she called on the University to release a crime report about how she was knocked down and intimidated by unknown assailants. The event follows similar incidents where crime alerts had not been released. The University has released two crime alerts of hate crimes on campus over the past two weeks.

Whyatt wrote in an email sent Tuesday afternoon to recipients including University President Mark Schlissel as well as The Michigan Daily that she waited until Saturday morning to report the assault to police because she was disoriented and did not know where to reach out.

"I did wait until Saturday morning to get in touch to report the incident," Whyatt wrote. "I was in shock and still processing what to do prior to reaching out ... it was also obvious that there was no way that these boys were going to be caught. Not being a student, I did not know who to report to. That must seem obvious by the fact that I

In [9]:
```
char_width=25
def calculate_line_height(img):
    '''calculate the average height of a line from a given image
    :param img:A PIL.Image object
    :return:The average height of line in pixels
    '''
    bounding_boxes=pageseg.segment(img.convert('1'))['boxes']
    height_accumulator=0
    for box in bounding_boxes:
        height_accumulator=height_accumulator+box[3]-box[1]
    return int((height_accumulator)/len(bounding_boxes))
line_height=calculate_line_height(Image.open('readonly/two_col.png'))
print(line_height)
```

```
31
```

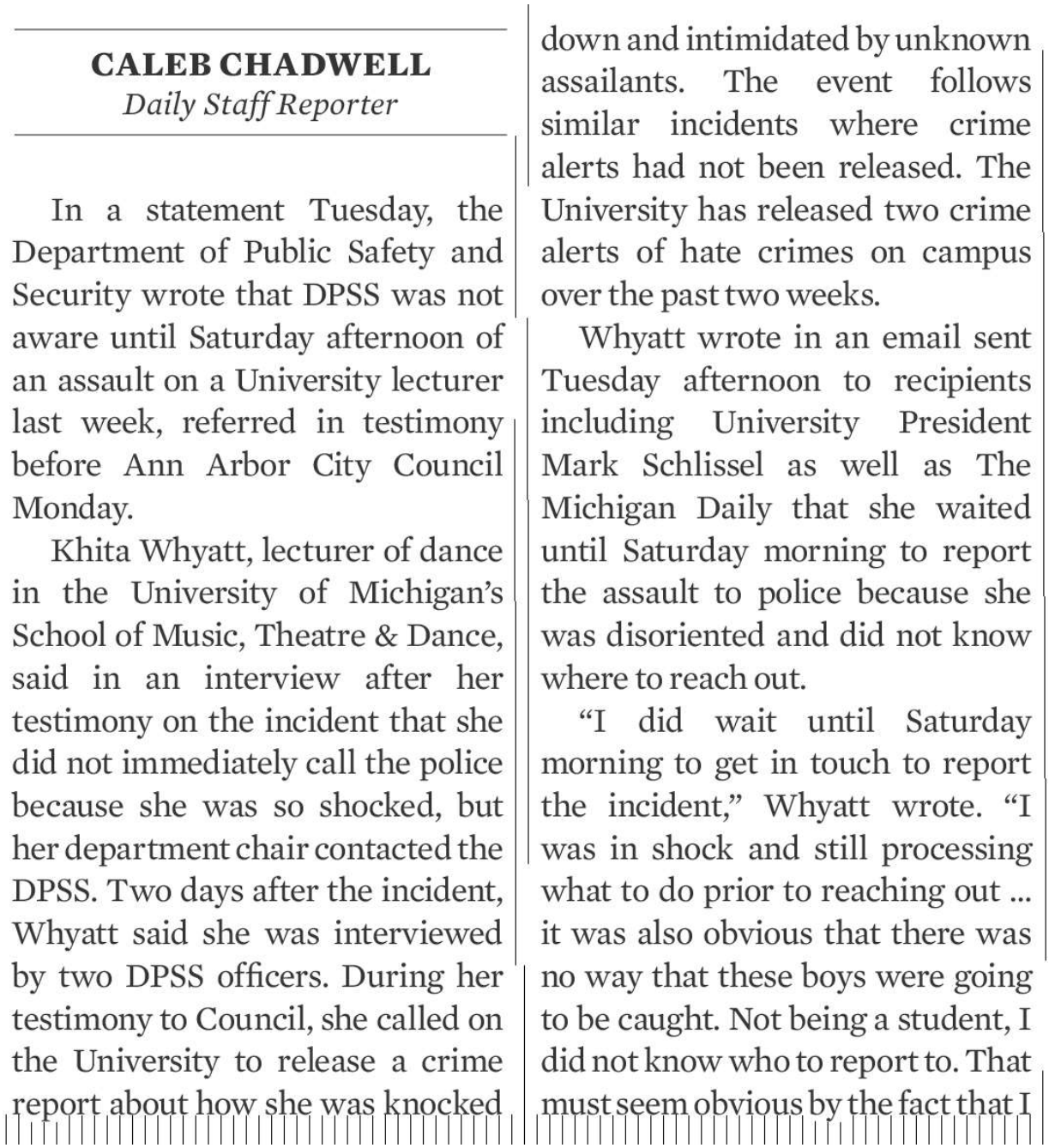
In [25]:
```
gap_box=(0,0,char_width,line_height*6)
gap_box
```

Out[25]:
```
(0, 0, 25, 186)
```

In [26]:
```
def gap_check(img, location):
    '''check a given image at a location(x,y) if it fits the description of a giv
    :param img: A PIL.Image file
    :param location:a tuple(x,y) which is a pixel location of that image
    :return:True if it fits the definition of a gap_box else false
    '''
    for x in range(location[0],location[0]+gap_box[2]):
        for y in range(location[1], location[1]+gap_box[3]):
            if x<img.width and y<img.height:
                if img.getpixel((x,y))!=255:
                    return False
    return True
```

In [27]:
```
def draw_sep(img, location):
    '''Draw a line at the middle of the gap discovered at the location
    :param img: A PIL.Image file
    :param location: A tuple (x,y) in pixel location
    '''
    from PIL import ImageDraw
    drawing_object=ImageDraw.Draw(img)
    x1=location[0]+int(gap_box[2]/2)
    x2=x1
    y1=location[1]
    y2=y1+gap_box[3]
    drawing_object.rectangle((x1,y1,x2,y2), fill='black', outline='black')
```

In [28]:
```
def process_img(img):
    '''Takes in an image of text and adds black vertical bars to break up columns
    :param img: A PIL.Image file
    :return: A modified PIL.Image file
    '''
    for x in range(img.width):
        for y in range(img.height):
            if (gap_check(img, (x,y))):
                draw_sep(img, (x,y))
    return img
i=Image.open('readonly/two_col.png').convert('L')
i=process_img(i)
display(i)
```



CALEB CHADWELL
Daily Staff Reporter

In a statement Tuesday, the Department of Public Safety and Security wrote that DPSS was not aware until Saturday afternoon of an assault on a University lecturer last week, referred in testimony before Ann Arbor City Council Monday.

Khita Whyatt, lecturer of dance in the University of Michigan's School of Music, Theatre & Dance, said in an interview after her testimony on the incident that she did not immediately call the police because she was so shocked, but her department chair contacted the DPSS. Two days after the incident, Whyatt said she was interviewed by two DPSS officers. During her testimony to Council, she called on the University to release a crime report about how she was knocked down and intimidated by unknown assailants. The event follows similar incidents where crime alerts had not been released. The University has released two crime alerts of hate crimes on campus over the past two weeks.

Whyatt wrote in an email sent Tuesday afternoon to recipients including University President Mark Schlissel as well as The Michigan Daily that she waited until Saturday morning to report the assault to police because she was disoriented and did not know where to reach out.

"I did wait until Saturday morning to get in touch to report the incident," Whyatt wrote. "I was in shock and still processing what to do prior to reaching out ... it was also obvious that there was no way that these boys were going to be caught. Not being a student, I did not know who to report to. That must seem obvious by the fact that I

In [30]:
```
display(show_boxes(i))
```

CALEB CHADWELL

Daily Staff Reporter

In a statement Tuesday, the Department of Public Safety and Security wrote that DPSS was not aware until Saturday afternoon of an assault on a University lecturer last week, referred in testimony before Ann Arbor City Council Monday.

Khita Whyatt, lecturer of dance in the University of Michigan's School of Music, Theatre & Dance, said in an interview after her testimony on the incident that she did not immediately call the police because she was so shocked, but her department chair contacted the DPSS. Two days after the incident, Whyatt said she was interviewed by two DPSS officers. During her testimony to Council, she called on the University to release a crime report about how she was knocked down and intimidated by unknown assailants. The event follows similar incidents where crime alerts had not been released. The University has released two crime alerts of hate crimes on campus over the past two weeks.

Whyatt wrote in an email sent Tuesday afternoon to recipients including University President Mark Schlissel as well as The Michigan Daily that she waited until Saturday morning to report the assault to police because she was disoriented and did not know where to reach out.

"I did wait until Saturday morning to get in touch to report the incident," Whyatt wrote. "I was in shock and still processing what to do prior to reaching out ... it was also obvious that there was no way that these boys were going to be caught. Not being a student, I did not know who to report to. That must seem obvious by the fact that I

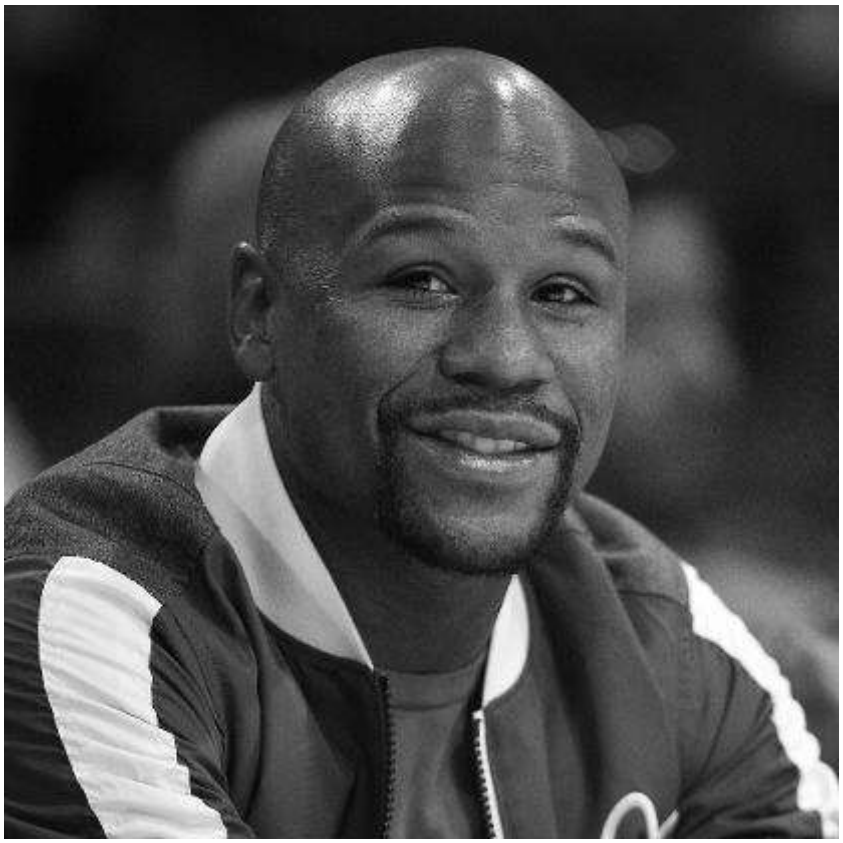
In [58]:
```
#Comparing data structures
import cv2 as cv
img=cv.imread('readonly/floyd.jpg')
gray=cv.cvtColor(img, cv.COLOR_BGR2GRAY)
import inspect
inspect.getmro(type(gray))
```

Out[58]:
```
(numpy.ndarray, object)
```

In [59]:
```
gray
```

Out[59]:
```
array([[ 40,  39,  39, ...,  77,  76,  75],
       [ 43,  42,  42, ...,  76,  75,  75],
       [ 39,  39,  39, ...,  76,  75,  74],
       ...,
       [ 21,  22,  24, ..., 219, 223, 209],
       [ 18,  20,  22, ..., 196, 206, 196],
       [ 16,  18,  20, ..., 168, 182, 176]], dtype=uint8)
```

In [60]:
```
from PIL import Image
image=Image.fromarray(gray,'L')
display(image)
```



In [61]:
```
import numpy as np
single_dim=np.array([25,50,25,10,10])
double_dim=np.array([single_dim])
double_dim
```

Out[61]:
```
array([[25, 50, 25, 10, 10]])
```

In [62]:
```
display(Image.fromarray(double_dim, 'L'))
```

-

In [63]:
```
double_dim.shape
```

Out[63]:
```
(1, 5)
```

In [64]:
```
img.shape
```

Out[64]:
```
(416, 416, 3)
```

In [65]:
```
first_pixel=img[0][0]
first_pixel
```

Out[65]:
```
array([33, 35, 53], dtype=uint8)
```

In [68]:
```
print('Original Image')
print(gray)
print('New Image')
image1d=np.reshape(gray,(1,gray.shape[0]*gray.shape[1]))
print(image1d)
```

```
Original Image
[[ 40  39  39 ...  77  76  75]
 [ 43  42  42 ...  76  75  75]
 [ 39  39  39 ...  76  75  74]
 ...
 [ 21  22  24 ... 219 223 209]
 [ 18  20  22 ... 196 206 196]
 [ 16  18  20 ... 168 182 176]]
New Image
[[ 40  39  39 ... 168 182 176]]
```

In [70]:
```
import cv2 as cv
img=cv.imread('readonly/two_col.png')
gray=cv.cvtColor(img, cv.COLOR_BGR2GRAY)
```

In [71]:
```
gray[2:4,1:3]
```

Out[71]:
```
array([[255, 255],
       [255, 255]], dtype=uint8)
```

In [72]:
```
np.count_nonzero(gray[2:4,1:3])
```

Out[72]:
```
4
```

In [73]:
```
white_matrix=np.full((12,12),255,dtype=np.uint8)
display(Image.fromarray(white_matrix,'L'))
white_matrix
```

Out[73]:
```
array([[255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255]],
      dtype=uint8)
```

In [74]:
```
white_matrix[:,6]=np.full((1,12),0,dtype=np.uint8)
display(Image.fromarray(white_matrix,'L'))
white_matrix
```


Out[74]:
```
array([[255, 255, 255, 255, 255, 255,   0, 255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255, 255,   0, 255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255, 255,   0, 255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255, 255,   0, 255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255, 255,   0, 255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255, 255,   0, 255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255, 255,   0, 255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255, 255,   0, 255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255, 255,   0, 255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255, 255,   0, 255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255, 255,   0, 255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255, 255,   0, 255, 255, 255, 255, 255]],
      dtype=uint8)
```

In [81]:
```
#OpenCV
import cv2 as cv
face_cascade=cv.CascadeClassifier('readonly/haarcascade_frontalface_default.xml'
eye_cascade=cv.CascadeClassifier('readonly/haarcascade_eye.xml')
```

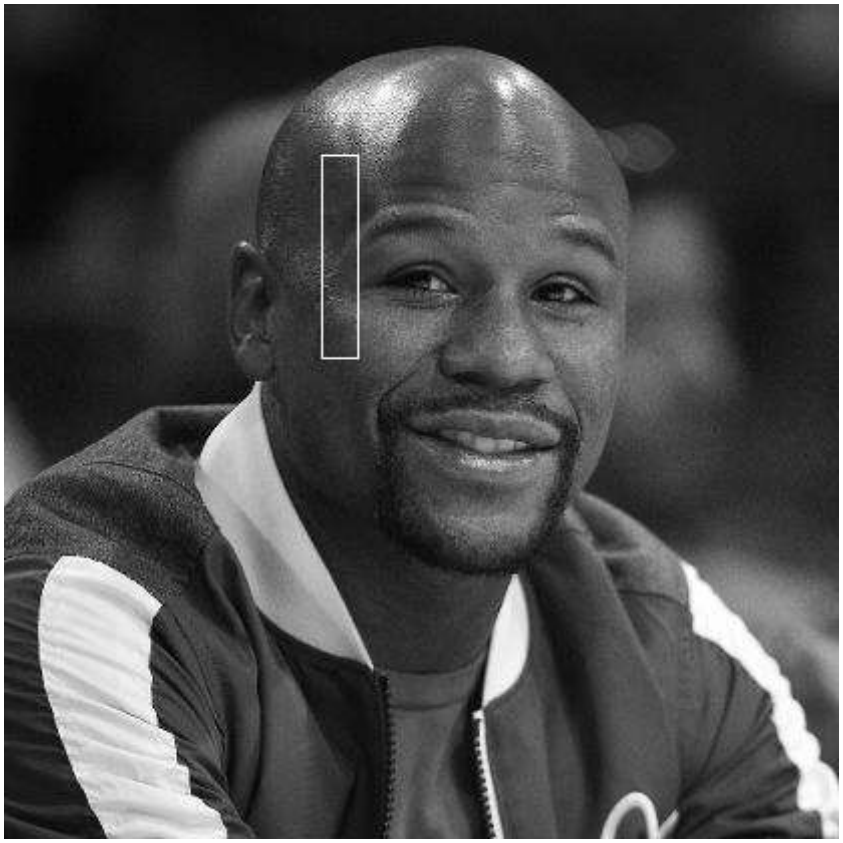
In [82]:
```
img=cv.imread('readonly/floyd.jpg')
gray=cv.cvtColor(img, cv.COLOR_BGR2GRAY)
faces=face_cascade.detectMultiScale(gray)
faces
```

Out[82]:
```
array([[158,  75, 176, 176]], dtype=int32)
```

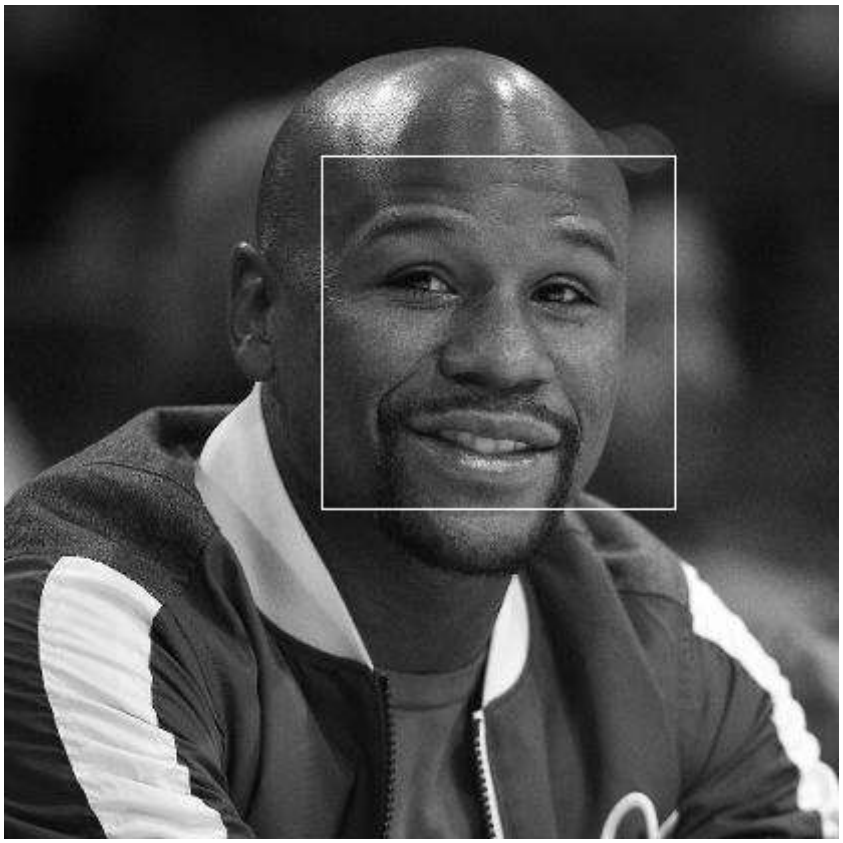
In [84]:
```
faces.tolist()[0]
```

Out[84]:
```
[158, 75, 176, 176]
```

In [85]:
```
from PIL import Image
pil_img=Image.fromarray(gray,mode='L')
from PIL import ImageDraw
drawing=ImageDraw.Draw(pil_img)
rec=faces.tolist()[0]
drawing.rectangle(rec, outline='white')
display(pil_img)
```



In [87]:
```
pil_img=Image.fromarray(gray, mode='L')
drawing=ImageDraw.Draw(pil_img)
drawing.rectangle((rec[0],rec[1],rec[0]+rec[2],rec[1]+rec[3]), outline='white')
display(pil_img)
```



In [89]:
```
img=cv.imread('readonly/msi_recruitment.gif')
display(Image.fromarray(img))
```

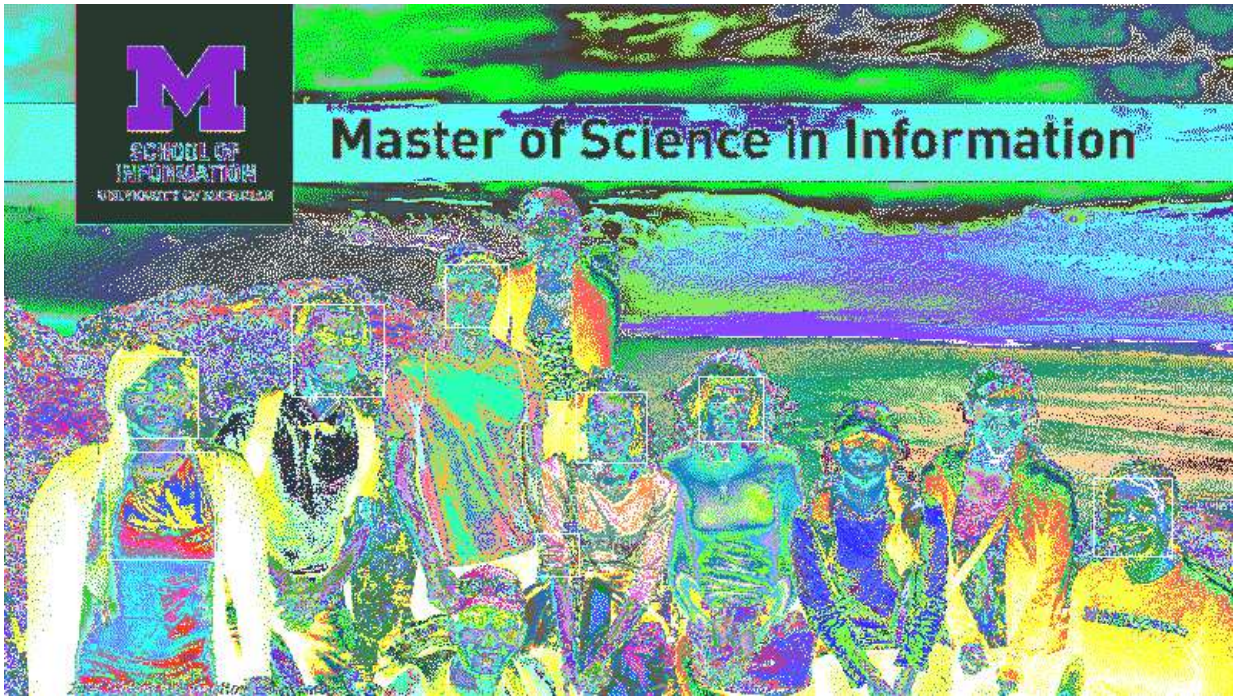
```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-89-92fac8edbb4b> in <module>
      1 img=cv.imread('readonly/msi_recruitment.gif')
----> 2 display(Image.fromarray(img))

/opt/conda/lib/python3.7/site-packages/PIL/Image.py in fromarray(obj, mode)
   2506     .. versionadded:: 1.1.6
   2507     """
-> 2508     arr = obj.__array_interface__
   2509     shape = arr['shape']
   2510     ndim = len(shape)

AttributeError: 'NoneType' object has no attribute '__array_interface__'
```

In [90]:
```
pil_img=Image.open('readonly/msi_recruitment.gif')
opencv_version=pil_img.convert('L')
opencv_version.save('msi_recruitment.png')
```

In [92]:
```
cv_img=cv.imread('msi_recruitment.png')
faces=face_cascade.detectMultiScale(cv_img)
pil_img=Image.open('readonly/msi_recruitment.gif')
drawing=ImageDraw.Draw(pil_img)
for x,y,w,h in faces:
    drawing.rectangle((x,y,x+w,y+h),outline='white')
display(pil_img)
```



In [93]:
```
pil_img.mode
```

Out[93]:
```
'P'
```

In [95]:
```
pil_img=Image.open('readonly/msi_recruitment.gif')
pil_img=pil_img.convert('RGB')
pil_img.mode
```

Out[95]:
```
'RGB'
```

In [96]:
```
drawing=ImageDraw.Draw(pil_img)
for x,y,w,h in faces:
    drawing.rectangle((x,y,x+w,y+h),outline='white')
display(pil_img)
```

In [ ]:
```
cv_img_bin=cv.threshold(img,120,255,cv.THRESH_BINARY)[1] # returns a list, we wa
# Now do the actual face detection
faces = face_cascade.detectMultiScale(cv_img_bin)
# Now lets see the results
show_rects(faces)
```

In [ ]:
```
faces = face_cascade.detectMultiScale(cv_img,1.05)
# Show those results
show_rects(faces)
# Now lets also try 1.15
faces = face_cascade.detectMultiScale(cv_img,1.15)
# Show those results
show_rects(faces)
# Finally lets also try 1.25
faces = face_cascade.detectMultiScale(cv_img,1.25)
# Show those results
show_rects(faces)
```

In [ ]:
```
%timeit face_cascade.detectMultiScale(cv_img,1.05)
```

In [ ]:
```
%timeit face_cascade.detectMultiScale(cv_img,1.15)
```

In [ ]:
```
#More jupyter widgets
from ipywebrtc import CameraStream, ImageRecorder
help(CameraStream)
```

In [ ]:
```
camera = CameraStream.facing_user(audio=False)
# The next object we want to look at is the ImageRecorder
help(ImageRecorder)
```

In [ ]:
```
image_recorder = ImageRecorder(stream=camera)
image_recorder.recording=True
# Now lets download the image
image_recorder.download()
# Then lets inspect the type of the image
type(image_recorder.image)
```

In [ ]:
```
import PIL.Image
# And lets import io
import io
# And now lets create a PIL image from the bytes
img = PIL.Image.open(io.BytesIO(image_recorder.image.value))
# And render it to the screen
display(img)
```