ImageChops

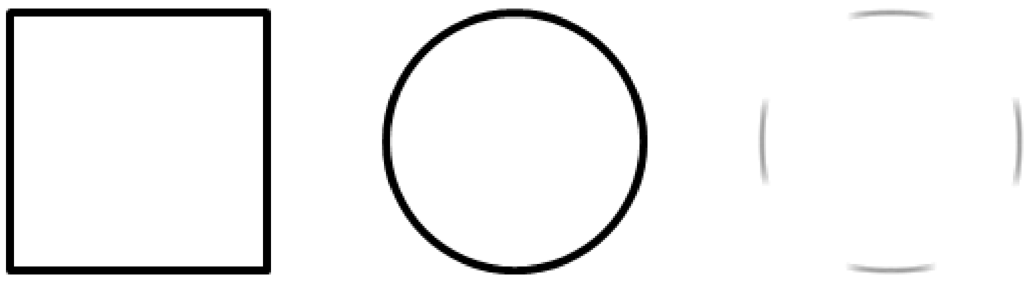
Nagaraj. Kulkarni

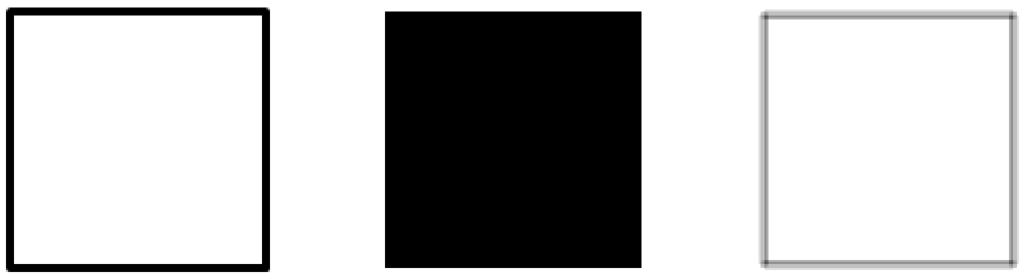
nkulkarni64@gatech.edu

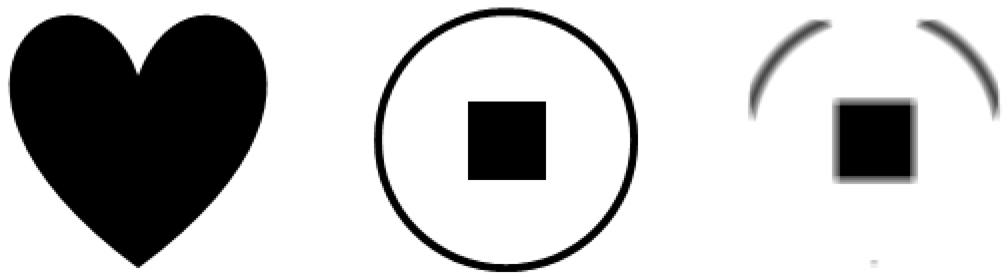
`PIL.ImageChops.add`(*image1, image2, scale=1.0, offset=0*)

Adds two images, dividing the result by scale and adding the offset. If omitted, scale defaults to 1.0, and offset to 0.0.

```
out = ((image1 + image2) / scale + offset)
```

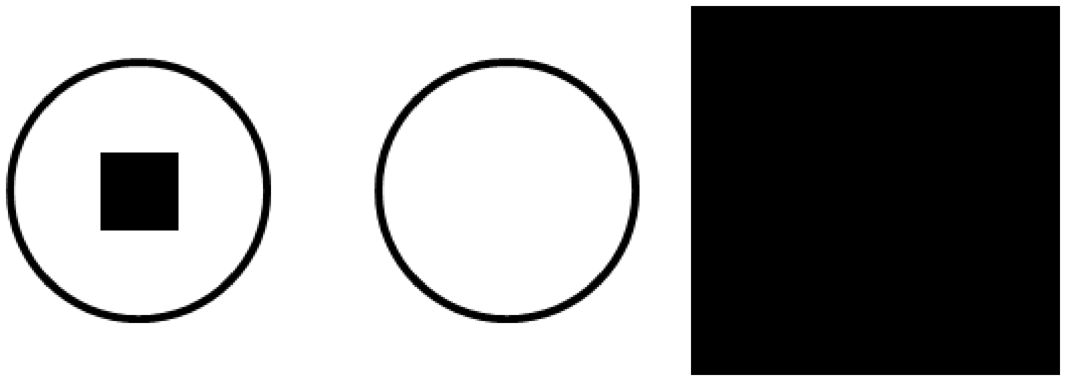


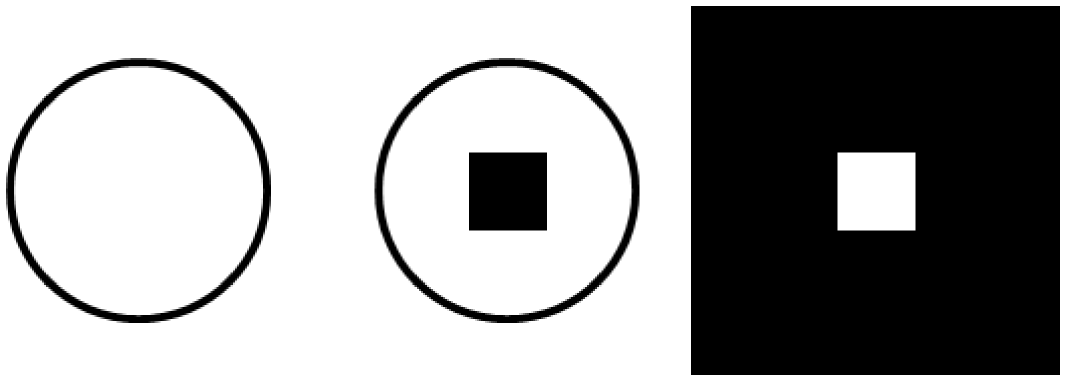




`PIL.ImageChops.subtract`(*image1*, *image2*, *scale=1.0*, *offset=0*)

Subtracts two images, dividing the result by scale and adding the offset. If omitted, scale defaults to 1.0, and offset to 0.0.

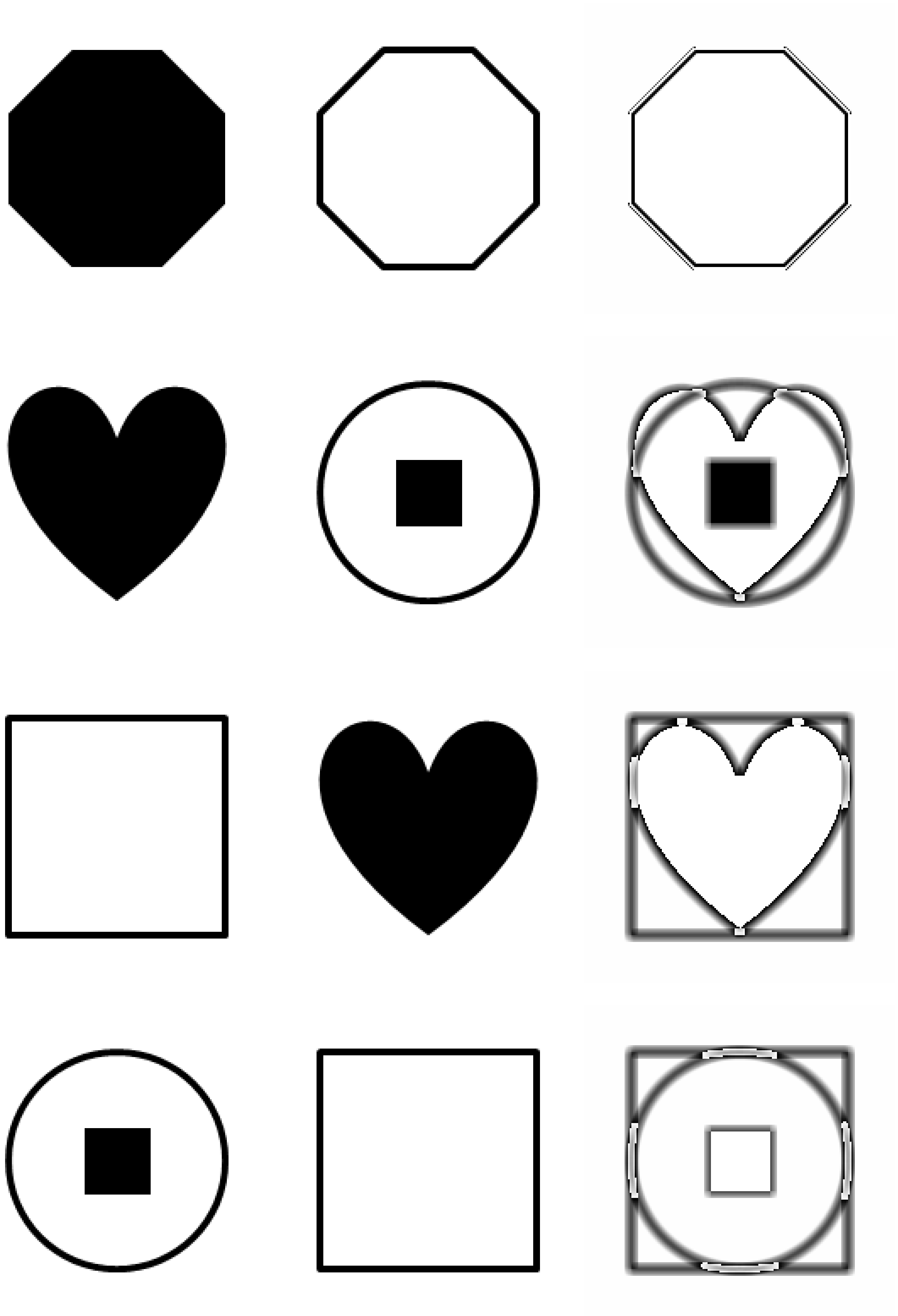




`PIL.ImageChops.add_modulo`(*image1*, *image2*)

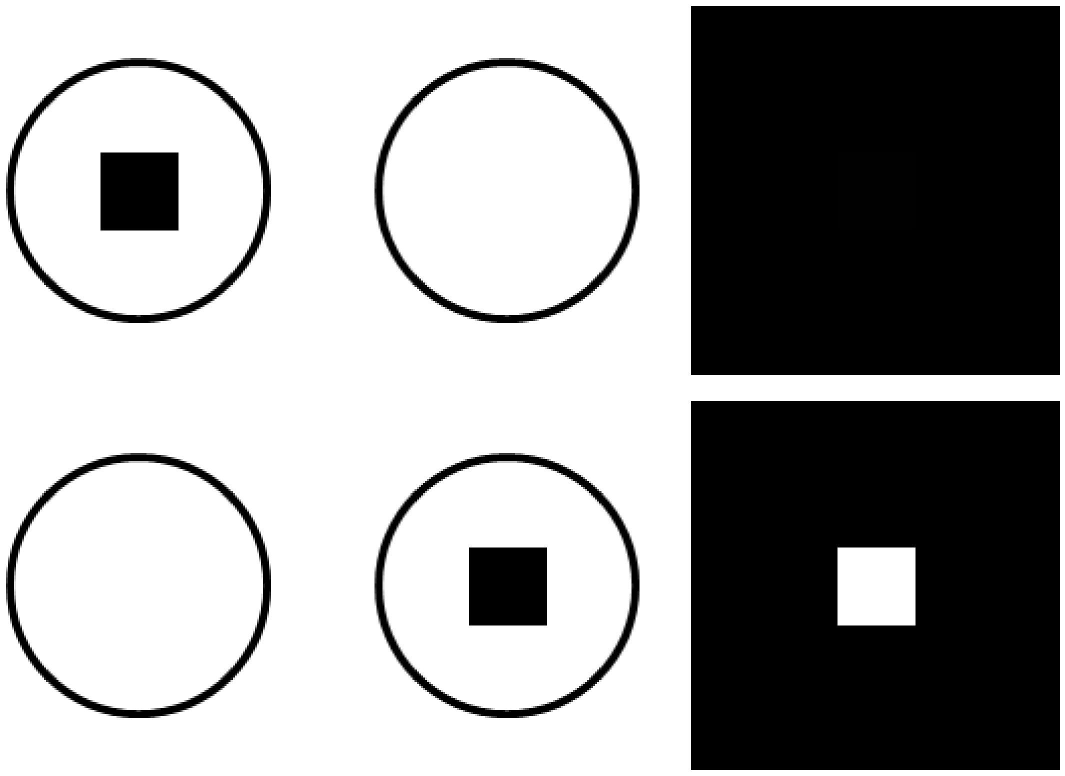
Add two images, without clipping the result.

```
out = ((image1 + image2) % MAX)
```



`PIL.ImageChops.subtract_modulo`*(image1, image2)*

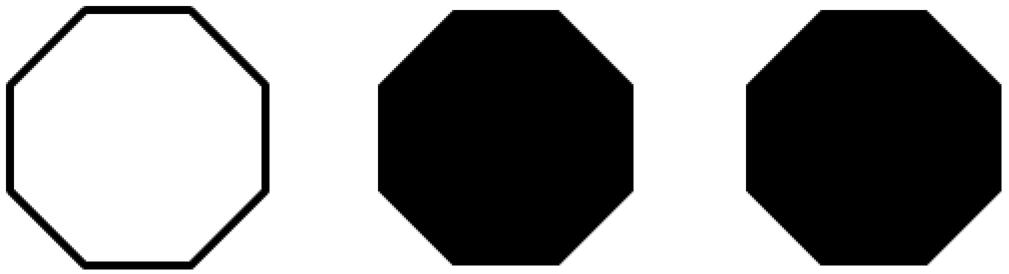
Subtract two images, without clipping the result.



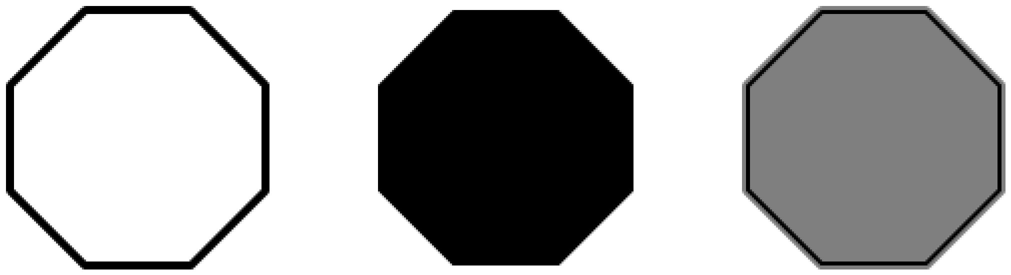
`PIL.ImageChops.blend`*(image1, image2, alpha)*

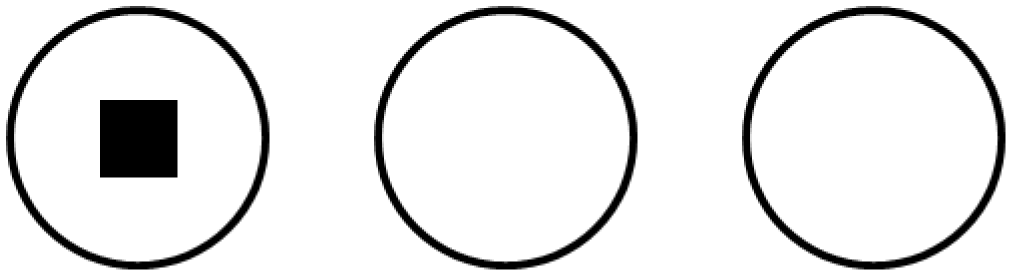
Blend images using constant transparency weight. Alias for `PIL.Image.Image.blend()`.

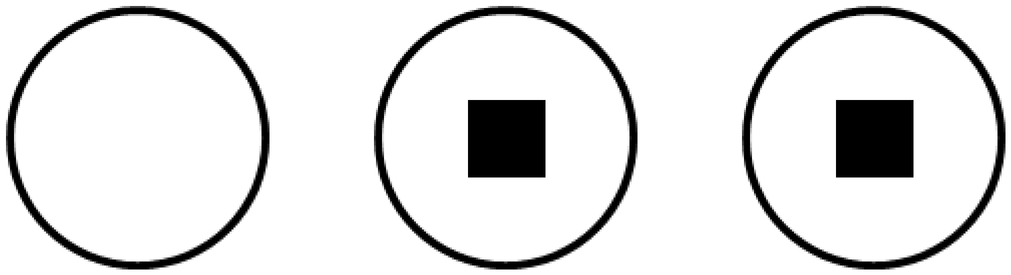
Alpha=1 = weight given to the second figure



Alpha=0.5

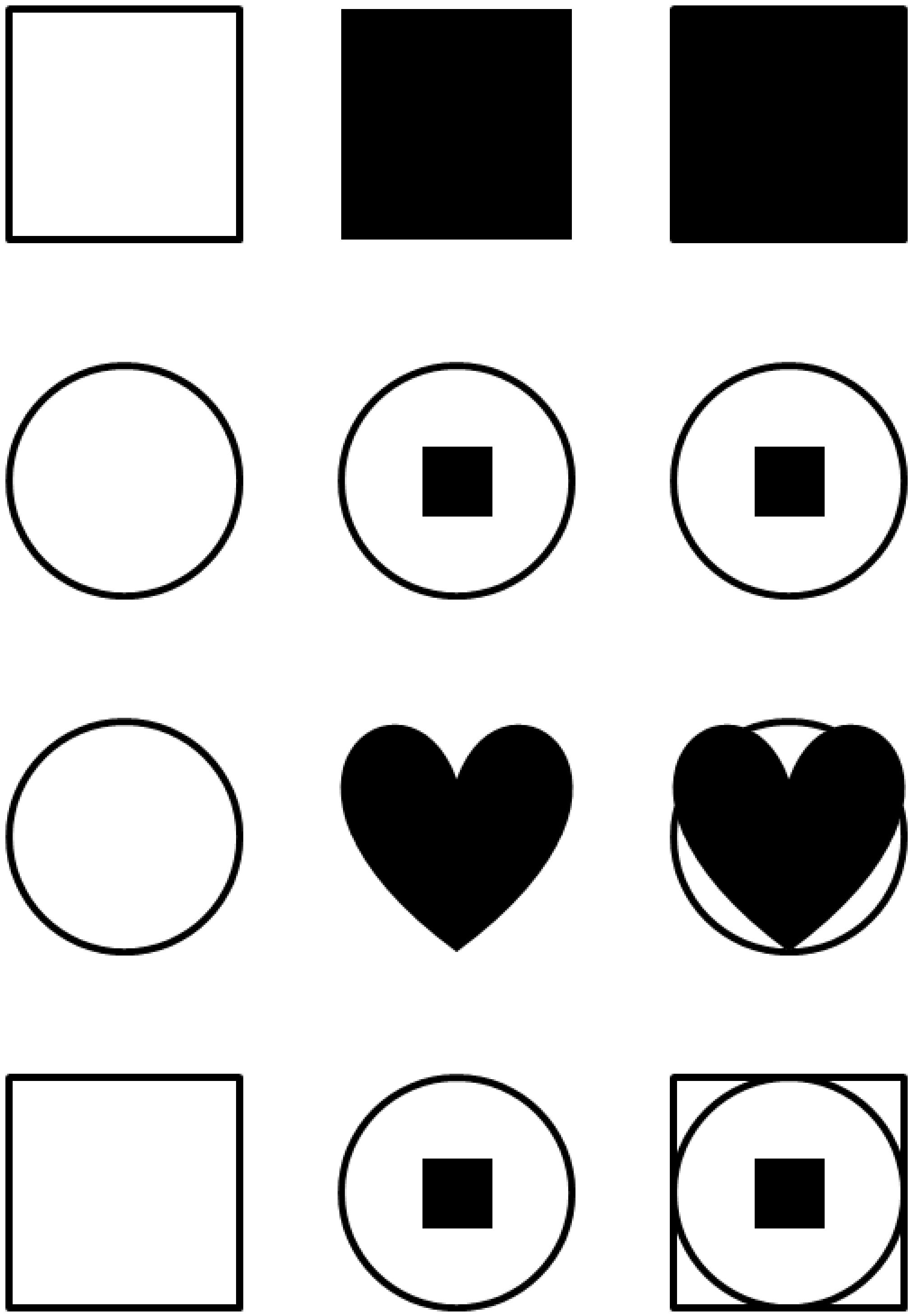






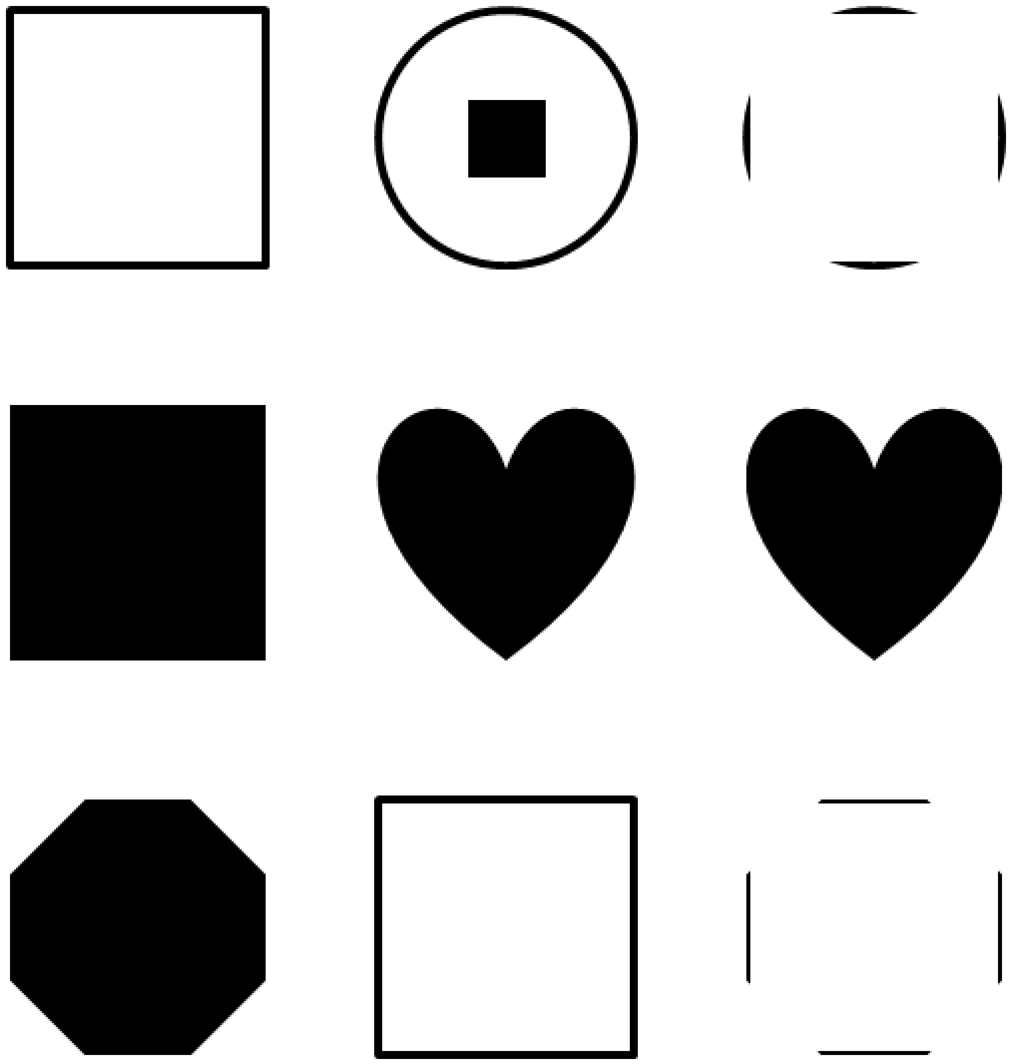
`PIL.ImageChops.darker`***(image1, image2)***

Compares the two images, pixel by pixel, and returns a new image containing the darker values.



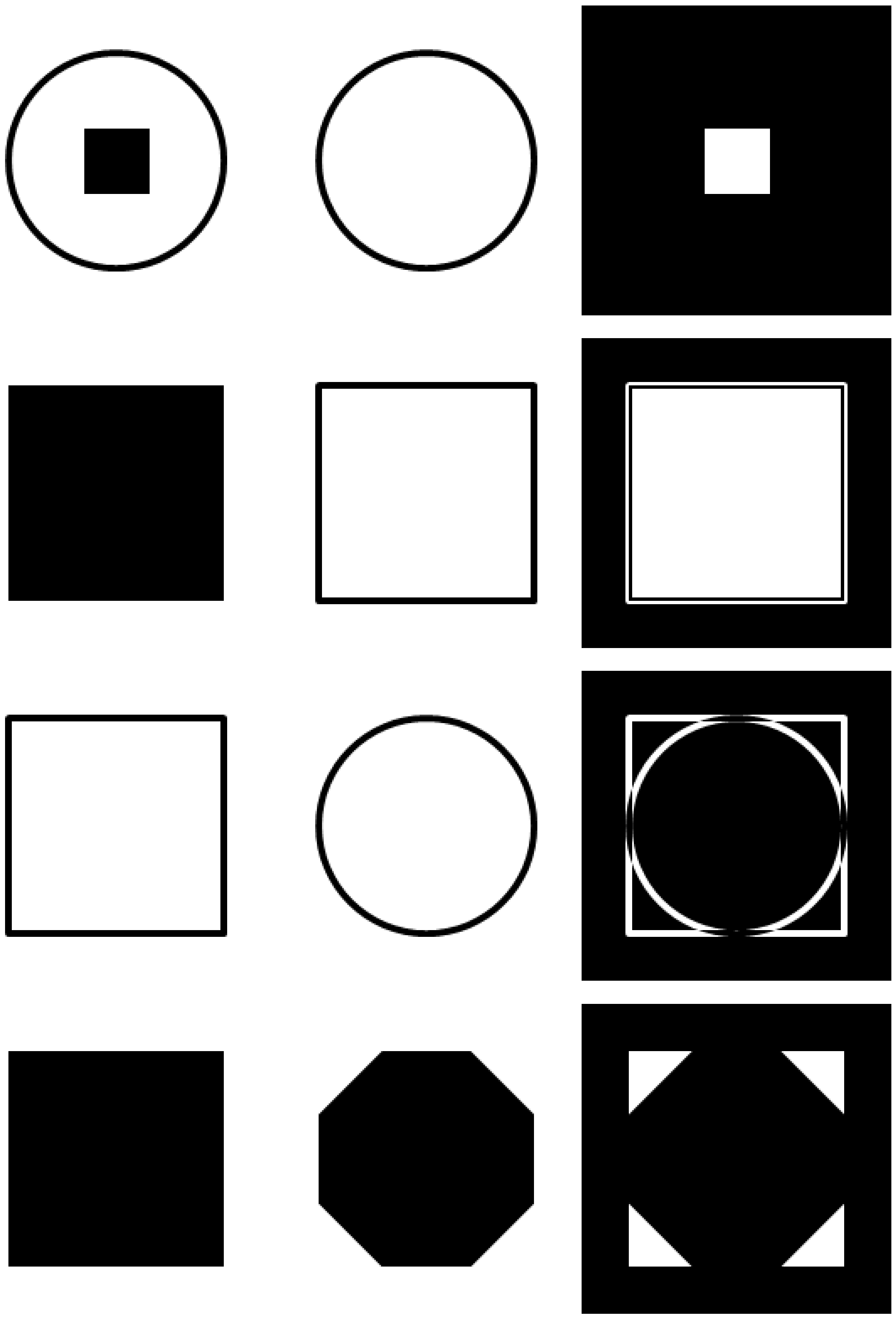
`PIL.ImageChops.lighter`*(image1, image2)*

Compares the two images, pixel by pixel, and returns a new image containing the lighter values.



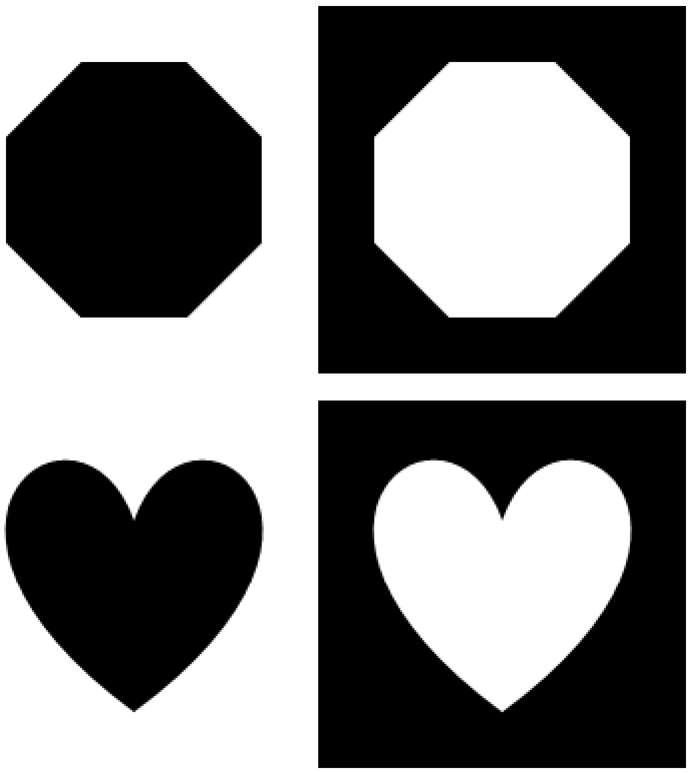
`PIL.ImageChops.difference`*(image1, image2)*

Returns the absolute value of the pixel-by-pixel difference between the two images.



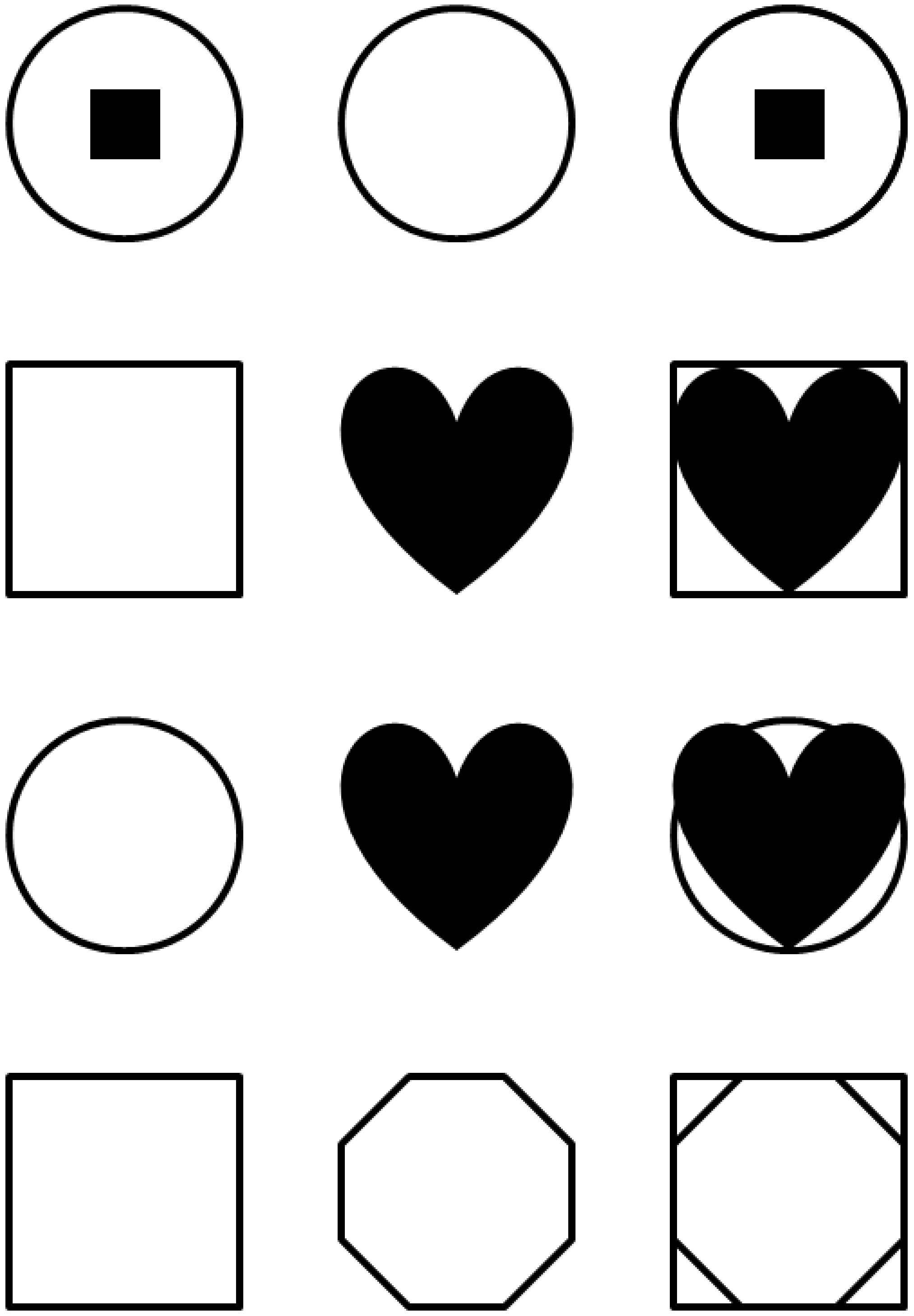
`PIL.ImageChops.invert(`*`image`*`)`

Invert an image (channel).



`PIL.ImageChops.multiply`*(image1, image2)*

Superimposes two images on top of each other. If you multiply an image with a solid black image, the result is black. If you multiply with a solid white image, the image is unaffected.



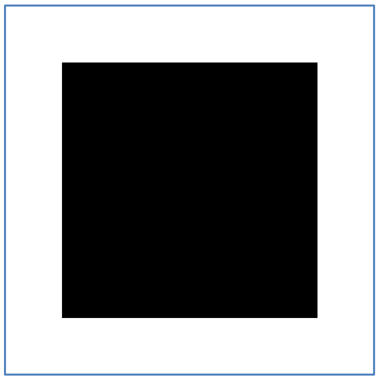
`PIL.ImageChops.offset`*(image, xoffset, yoffset=None)*

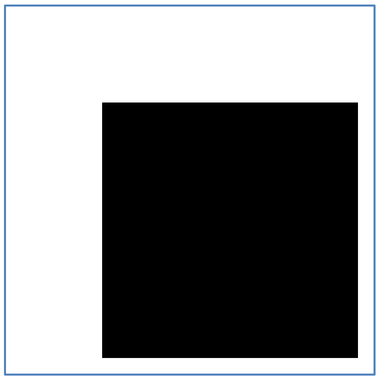
Returns a copy of the image where data has been offset by the given distances. Data wraps around the edges. If **yoffset** is omitted, it is assumed to be equal to **xoffset**.

Parameters:

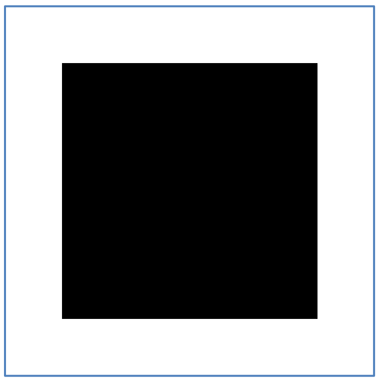
- **xoffset** – The horizontal distance.
- **yoffset** – The vertical distance. If omitted, both distances are set to the same value.

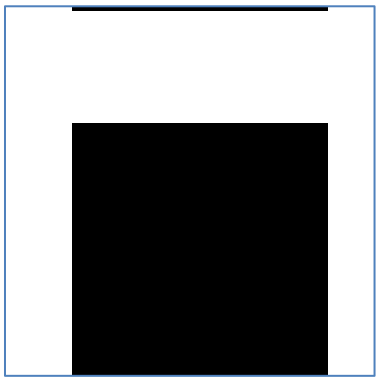
```
out=ImageChops.offset(image1, xoffset=20, yoffset=None)
```





```
out=ImageChops.offset(image1, xoffset=5, yoffset=30)
```





`PIL.ImageChops.screen`(*image1*, *image2*)

Superimposes two inverted images on top of each other.

```
out = MAX - ((MAX - image1) * (MAX - image2) / MAX)
```