Grayness

This package provides basic image editing functions. All of them work with Raster Data (e.g. “normal” Images like PNG or JPEG). The `grayscale-image` function also works with Vector Data (SVG).

The Following functions are available:

- blur-image()
- crop-image()
- flip-image-horizontal()
- flip-image-vertical()
- grayscale-image()
- show-image()
- transparent-image()

blur-image

performs a Gaussian blur on the imagedata.

Warning: This operation is very slow, especially for large sigmas.

Parameters

```
blur-image(
  imagebytes: bytes,
  sigma: int,
  ..args: any
) -> content
```

imagebytes `bytes`

Raw imagedata provided by the read function

Example:

```
#let data = read("file.webp", encoding:none)
#blur-image(data)
```

sigma `int`

a measure of how much to blur by (standard deviation)

Example:

```
#let data = read("file.webp", encoding:none)
#blur-image(data, 5)
```

Default: 5

..args `any`

Arguments to pass to typst image function i.e. width, height, alt and fit

Example:

```
#let data = read("file.webp", encoding:none)
#blur-image(data, 7, width: 50%, height: 80%)
```

crop-image

Crop the given imagedata to the specified width and height

Parameters

```
crop-image(
  imagebytes: bytes,
  crop-width: int,
  crop-height: int,
  start-x: int,
  start-y: int,
  ..args: any
) -> content
```

imagebytes `bytes`

Raw imagedata provided by the read function

Example:

```
#let data = read("file.webp", encoding:none)
#crop(data, 0, 0, 150, 200)
```

crop-width `int`

horizontal size (in pixels) of the crop window

crop-height `int`

vertical size (in pixels) of the crop window

start-x `int`

left starting coordinate (in pixels) of the crop window

Default: `0`

start-y `int`

top starting coordinate (in pixels) of the crop window

Default: `0`

..args `any`

Arguments to pass to typst image function i.e. width, height, alt and fit

Example:

```
#let data = read("file.webp", encoding:none)
#crop-image(data, 0, 70, 120, 250, width: 50%, height: 80%)
```

flip-image-horizontal

Flip the provided imagedata horizontally

Parameters

```
flip-image-horizontal(
  imagebytes: bytes,
  ..args: any
) -> content
```

imagebytes `bytes`

Raw imagedata provided by the read function

Example:

```
#let data = read("file.webp", encoding:none)
#flip-image-horizontal(data)
```

..args `any`

Arguments to pass to typst image function i.e. width, height, alt and fit

Example:

```
#let data = read("file.webp", encoding:none)
#flip-image-horizontal(data, width: 50%, height: 80%)
```

flip-image-vertical

Flip the provided imagedata vertically

Parameters

```
flip-image-vertical(
  imagebytes: bytes,
  ..args: any
) -> content
```

imagebytes `bytes`

Raw imagedata provided by the read function

Example:

```
#let data = read("file.webp", encoding:none)
#flip-image-vertical(data)
```

..args `any`

Arguments to pass to typst image function i.e. width, height, alt and fit

Example:

```
#let data = read("file.webp", encoding:none)
#flip-image-vertical(data, width: 50%, height: 80%)
```

grayscale-image

Create a grayscale-image representation of the provided imagedata (Raster or SVG)

Parameters

```
grayscale-image(
  imagebytes: bytes,
  ..args: any
) -> content
```

imagebytes `bytes`

Raw imagedata provided by the read function

Example:

```
#let data = read("file.webp", encoding:none)
#grayscale-image(data)
```

..args `any`

Arguments to pass to typst image function i.e. width, height, alt and fit

You must pass `format: "svg"` as argument if you use a SVG Image as your input.

Example:

```
#let data = read("file.svg", encoding:none)
#grayscale-image(data, width: 50%, height: 80%, format: "svg")
```

show-image

Displays an image from bytes in formats not natively supported by typst

Supported formats are:

- Bmp
- Dds
- Farbfeld
- Gif
- Hdr
- Ico

- Jpeg
- OpenExr
- Png
- Pnm
- Qoi
- Tga
- Tiff
- WebP

Parameters

```
show-image(
  imagebytes: bytes,
  ..args: any
) -> content
```

imagebytes `bytes`

Raw imagedata provided by the read function

Example:

```
#let data = read("file.webp", encoding:none)
#show-image(data)
```

..args `any`

Arguments to pass to typst image function i.e. width, height, alt and fit

Example:

```
 #let data = read("file.webp", encoding:none)
 #show-image(data, width: 50%, height: 80%)
```

transparent-image

Adds transparency to the provided image data

Parameters

```
transparent-image(
  imagebytes: bytes,
  alpha: ratio,
  ..args: any
) -> content
```

imagebytes `bytes`

Raw imagedata provided by the read function

Example:

```
#let data = read("file.webp", encoding:none)
#transparent-image(data)
```

alpha `ratio`

remaining amount of visibility

0% = fully transparent, 100% = fully opaque

Example:

```
#let data = read("file.webp", encoding:none)
#transparent-image(data, 70%)
```

Default: `50%`

..args `any`

Arguments to pass to typst image function i.e. width, height, alt and fit

Example:

```
#let data = read("file.webp", encoding:none)
#transparent-image(data, width: 50%, height: 80%)
```