GrayNess

This Package provides the following basic image editing functions:

- blur_image()
- convert_image()
- crop_image()
- flip_image_horizontal()
- flip_image_vertical()
- grayscale()
- transparent_image()

blur_image

performs a Gaussian blur on the imagedata.

Warning: This operation is **SLOW**

Parameters

```
blur_image(
  imagebytes: bytes,
  sigma: int str,
  ..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` or `str`

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%)
```

convert_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

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

imagebytes `bytes`

Raw imagedata provided by the read function

Example:

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

crop_image

Crop the given imagedata to the specified width and height

Parameters

```
crop_image(
  imagebytes: bytes,
  startx: int str,
  starty: int str,
  width: int str,
  height: int str,
  ..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)
```

startx `int` or `str`

left starting coordinate (in pixels) of the crop window

Default: `0`

starty `int` or `str`

top starting coordinate (in pixels) of the crop window

Default: `0`

width `int` or `str`

horizontal size (in pixels) of the crop window

height `int` or `str`

vertical size (in pixels) of the crop window

..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

Create a grayscale represenation of the provided imagedata

Parameters

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

imagebytes `bytes`

Raw imagedata provided by the read function

Example:

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