
Image 1.0

September 9, 2024



Contents

Image	2
Crop	2
GetColorAt	2
GetDimensions	3
Scale	3
Color distance	3
Creating a color object	4
Predefined colors	4

Image

The `Image` module is used to manipulate images.

Crop

The `crop` method is used to crop an image. It takes the following parameters:

- `imageBytes` the image to crop in base64 format
- `x` the x-coordinate of the top-left corner of the crop area
- `y` the y-coordinate of the top-left corner of the crop area
- `width` the width of the crop area
- `height` the height of the crop area

Example

```
1 var imageModule = Module.load('Image', { version: 'vX.Y.Z' });
2 var croppedImage = imageModule.crop(imageBytes, 10, 10, 100, 100);
```

GetColorAt

The `getColorAt` method is used to get the color of a pixel at a given coordinate. It takes the following parameters:

- `imageBytes` the image to crop in base64 format
- `x` the x-coordinate of the pixel
- `y` the y-coordinate of the pixel

Example

```
1 var imageModule = Module.load('Image', { version: 'vX.Y.Z' });
2 var color = imageModule.getColorAt(imageBytes, 10, 10);
3 // color is an object with the following properties: red, green, blue
```

GetDimensions

The `getDimensions` method is used to get the dimensions of an image. It takes the following parameters:

- `imageBytes` the image to crop in base64 format

Example

```
1 var imageModule = Module.load('Image', { version: 'vX.Y.Z' });
2 var dimensions = imageModule.getDimensions(imageBytes);
3 var width = dimensions.width;
4 var height = dimensions.height;
```

Scale

The `scale` method is used to scale an image. It takes the following parameters:

- `imageBytes` the image to crop in base64 format
- `factor` the factor by which to scale the image

Example

```
1 var imageModule = Module.load('Image', { version: 'vX.Y.Z' });
2 // Double the size of the image
3 var scaledImage = imageModule.scale(imageBytes, 2);
```

Color distance

The `colorDistance` method is used to calculate the difference between two colors. It takes the following parameters:

- `color1` the first color object
- `color2` the second color object

The return value is a number in [0;1] where 0 means the colors are identical and 1 means the colors are as different as black and white.

Example

```
1 var imageModule = Module.load('Image', { version: 'vX.Y.Z' });
2 var color1 = imageModule.RED;
3 var color2 = imageModule.toColorObject(145, 10, 17);
4 var distance = imageModule.colorDistance(color1, color2);
```

Creating a color object

The `toColorObject` method is used to create a color object. It takes the following parameters:

- `red` the red component of the color
- `green` the green component of the color
- `blue` the blue component of the color

The return value is an object with the following numerical properties: `red`, `green`, `blue`.

Example

```
1
2 var imageModule = Module.load('Image', { version: 'vX.Y.Z' });
3 var color = imageModule.toColorObject(145, 10, 17);
```

Predefined colors

The `Image` module provides the following predefined colors for the sake of convenience:

- `RED`
- `GREEN`
- `BLUE`
- `WHITE`
- `BLACK`

Example

```
“javascript var imageModule = Module.load('Image', { version: 'vX.Y.Z' }); var color = imageModule.RED;
```