

Reference > saveFrames()

saveFrames()

Captures a sequence of frames from the canvas that can be saved as images.

`saveFrames()` creates an array of frame objects. Each frame is stored as an object with its file type, file name, and image data as a string. For example, the first saved frame might have the following properties:

```
{ ext: 'png', filename: 'frame0', imageData: 'data:image/octet-stream;base64, abc123' }.
```

The first parameter, `filename`, sets the prefix for the file names. For example, setting the prefix to `'frame'` would generate the image files `frame0.png`, `frame1.png`, and so on.

The second parameter, `extension`, sets the file type to either `'png'` or `'jpg'`.

The third parameter, `duration`, sets the duration to record in seconds. The maximum duration is 15 seconds.

The fourth parameter, `framerate`, sets the number of frames to record per second. The maximum frame rate value is 22. Limits are placed on `duration` and `framerate` to avoid using too much memory. Recording large canvases can easily crash sketches or even web browsers.

The fifth parameter, `callback`, is optional. If a function is passed, image files won't be saved by default. The callback function can be used to process an array containing the data for each captured frame. The array of image data contains a sequence of objects with three properties for each frame: `imageData`, `filename`, and `extension`.

Note: Frames are downloaded as individual image files by default.

Examples

▶

■

```
function setup() {
  createCanvas(100, 100);

  describe('A square repeatedly changes color from blue to pink.');
```

```
function draw() {
  let r = frameCount % 255;
  let g = 50;
  let b = 100;
  background(r, g, b);
}
```

```
// Save the frames when the user presses the 's' key.
function keyPressed() {
  if (key === 's') {
    saveFrames('frame', 'png', 1, 5);
  }
}
```

▶

■

```
function setup() {
  createCanvas(100, 100);

  describe('A square repeatedly changes color from blue to pink.');
```

```
function draw() {
  let r = frameCount % 255;
  let g = 50;
  let b = 100;
  background(r, g, b);
}
```

```
// Print 5 frames when the user presses the mouse.
function mousePressed() {
  saveFrames('frame', 'png', 1, 5, printFrames);
}
```

```
// Prints an array of objects containing raw image data,
// filenames, and extensions.
function printFrames(frames) {
  for (let frame of frames) {
    print(frame);
  }
}
```

Syntax

```
saveFrames(filename, extension, duration, framerate, [callback])
```

Parameters

filename	String: prefix of file name.
extension	String: file extension, either 'jpg' or 'png'.
duration	Number: duration in seconds to record. This parameter will be constrained to be less or equal to 15.
framerate	Number: number of frames to save per second. This parameter will be constrained to be less or equal to 22.
callback	Function(Array): callback function that will be executed to handle the image data. This function should accept an array as argument. The array will contain the specified number of frames of objects. Each object has three properties: <code>imageData</code> , <code>filename</code> , and <code>extension</code> .

This page is generated from the comments in [src/image/image.js](#). Please feel free to edit it and submit a pull request!

Related References

blend Copies a region of pixels from another image into this one.	copy Copies pixels from a source image to this image.	delay Changes the delay between frames in an animated GIF.	filter Applies an image filter to the image.
---	---	--	--