

createCapture()

Creates a `<video></video>` element that "captures" the audio/video stream from the webcam and microphone.

`createCapture()` returns a new [p5.MediaElement](#) object. Videos are shown by default. They can be hidden by calling `capture.hide()` and drawn to the canvas using `image()`.

The first parameter, `type`, is optional. It sets the type of capture to use. By default, `createCapture()` captures both audio and video. If `VIDEO` is passed, as in `createCapture(VIDEO)`, only video will be captured. If `AUDIO` is passed, as in `createCapture(AUDIO)`, only audio will be captured. A constraints object can also be passed to customize the stream. See the [W3C documentation](#) for possible properties. Different browsers support different properties.

The 'flipped' property is an optional property which can be set to `{flipped:true}` to mirror the video output. If it is true then it means that video will be mirrored or flipped and if nothing is mentioned then by default it will be `false`.

The second parameter, `callback`, is optional. It's a function to call once the capture is ready for use. The callback function should have one parameter, `stream`, that's a [MediaStream](#) object.

Note: `createCapture()` only works when running a sketch locally or using HTTPS. Learn more [here](#) and [here](#).

Examples

```
function setup() {
  noCanvas();

  // Create the video capture.
  createCapture(VIDEO);

  describe('A video stream from the webcam.')
}
```

```
let capture;

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

  // Create the video capture and hide the element.
  capture = createCapture(VIDEO);
  capture.hide();

  describe('A video stream from the webcam with inverted colors.');
}

function draw() {
  // Draw the video capture within the canvas.
  image(capture, 0, 0, width, width * capture.height / capture.width);

  // Invert the colors in the stream.
  filter(INVERT);
}
```

```
let capture;

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

  // Create the video capture with mirrored output.
  capture = createCapture(VIDEO, { flipped:true });
  capture.size(100,100);

  describe('A video stream from the webcam with flipped or mirrored output.');
}
```

```
function setup() {
  createCanvas(480, 120);

  // Create a constraints object.
  let constraints = {
    video: {
      mandatory: {
        minWidth: 1280,
        minHeight: 720
      },
      optional: [{ maxFrameRate: 10 }]
    },
    audio: false
  };

  // Create the video capture.
  createCapture(constraints);

  describe('A video stream from the webcam.');
}
```

Syntax

```
createCapture([type], [flipped], [callback])
```

Parameters

<code>type</code>	<code>String Constant Object</code> : type of capture, either <code>AUDIO</code> or <code>VIDEO</code> , or a <code>constraints</code> object. Both video and audio audio streams are captured by default.
<code>flipped</code>	<code>Object</code> : flip the capturing video and mirror the output with <code>{flipped:true}</code> . By default it is <code>false</code> .
<code>callback</code>	<code>Function</code> : function to call once the stream has loaded.

Returns

`p5.MediaElement`: new [p5.MediaElement](#) object.

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

Related References

[addClass](#)
Adds a class to the element.

[attribute](#)
Adds an attribute to the element.

[center](#)
Centers the element either vertically,

[child](#)
Attaches the element as a child of



[Resources](#)
[Reference](#)
[Tutorials](#)
[Examples](#)
[Contribute](#)
[Community](#)
[About](#)
[Start Coding](#)
[Donate](#)

[Information](#)
[Download](#)
[Contact](#)
[Copyright](#)
[Privacy Policy](#)
[Terms of Use](#)

Socials
[GitHub](#) ↗
[Instagram](#) ↗
[X](#) ↗
[YouTube](#) ↗
[Discord](#) ↗
[Forum](#) ↗

Donate Today! Support p5.js and the Processing Foundation.