

loadStrings()

Loads a text file to create an `Array`.

The first parameter, `path`, is always a string with the path to the file. Paths to local files should be relative, as in `loadStrings('/assets/data.txt')`. URLs such as '<https://example.com/data.txt>' may be blocked due to browser security.

The second parameter, `successCallback`, is optional. If a function is passed, as in `loadStrings('/assets/data.txt', handleData)`, then the `handleData()` function will be called once the data loads. The array created from the text data will be passed to `handleData()` as its only argument.

The third parameter, `failureCallback`, is also optional. If a function is passed, as in `loadStrings('/assets/data.txt', handleData, handleFailure)`, then the `handleFailure()` function will be called if an error occurs while loading. The `Error` object will be passed to `handleFailure()` as its only argument.

Note: Data can take time to load. Calling `loadStrings()` within `preload()` ensures data loads before it's used in `setup()` or `draw()`.

Examples



```
let myData;

// Load the text and create an array.
function preload() {
  myData = loadStrings('/assets/test.txt');
}

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

  background(200);

  // Select a random line from the text.
  let phrase = random(myData);

  // Style the text.
  textAlign(LEFT, CENTER);
  textStyle('Courier New');
  textSize(12);

  // Display the text.
  text(phrase, 10, 50, 90);

  describe(`The text "${phrase}" written in black on a gray
background.`);
}
```

```
let lastLine;

// Load the text and preprocess it.
function preload() {
  loadStrings('/assets/test.txt', handleData);
}

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

  background(200);

  // Style the text.
  textAlign(LEFT, CENTER);
  textStyle('Courier New');
  textSize(12);

  // Display the text.
  text(lastLine, 10, 50, 90);

  describe('The text "I talk like an orange" written in black
on a gray background.');
}

// Select the last line from the text.
function handleData(data) {
  lastLine = data[data.length - 1];
}
```

```
let lastLine;

// Load the text and preprocess it.
function preload() {
  loadStrings('/assets/test.txt', handleData, handleError);
}

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

  background(200);

  // Style the text.
  textAlign(LEFT, CENTER);
  textStyle('Courier New');
  textSize(12);

  // Display the text.
  text(lastLine, 10, 50, 90);

  describe('The text "I talk like an orange" written in black
on a gray background.');
}

// Select the last line from the text.
function handleData(data) {
  lastLine = data[data.length - 1];
}

// Log any errors to the console.
function handleError(error) {
```

Syntax

```
loadStrings(path, [successCallback], [errorCallback])
```

Parameters

<code>path</code>	String: path of the text file to be loaded.
<code>successCallback</code>	Function: function to call once the data is loaded. Will be passed the array.
<code>errorCallback</code>	Function: function to call if the data fails to load. Will be passed an <code>Error</code> event object.

Returns

`String[]`: new array containing the loaded text.

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

Related References

[addChild](#)
Adds a new child element and returns a reference to it.

[getAttributeCount](#)
Returns the number of attributes the element has.

[getChild](#)
Returns the first matching child element as a new p5.XML object.

[getChildren](#)
Returns an array with the element's child elements as new p5.XML objects.

