

# pjs.fiber.wrap()

This API wraps an asynchronous function so that it can be called in a top-down manner within a [fiber](#).

The API assumes that the function accepts a callback as its last parameter in the following format:

```
function(error, response)
```

This is typical for virtually all asynchronous Node.js requests.

The API can also accept an object which contains a set of asynchronous functions, in which case it will act the same as the [pjs.fiber.wrapAll\(\)](#) API.

## Parameters

- asynchronous function
- *this* argument (*optional*)
  - Typically, the parent object of the function is passed; this is required if internally the function uses the *this* keyword.

## Alternative Parameters

- object containing asynchronous functions

## Return Value

A top-down function is returned.

## Exception Handling

If the callback receives an error object, an error is thrown.

## Shortcut

`pjs.wrap()` is a shortcut for `pjs.fiber.wrap()`.

## Examples

### Wrap fs.readFile in a fiber

```
const fs = require('fs');
var readFile = pjs.wrap(fs.readFile);
var text = readFile("somefile.txt", 'utf8');
```

### Wrap all fs API in a fiber

```
const fs = pjs.wrap(require('fs'));
var text = fs.fiber.readFile("somefile.txt", 'utf8');
```

## Video Tutorial