# Buffer From

A [ponyfill](#) for `Buffer.from`, uses native implementation if available.

## Installation

```
npm install --save buffer-from
```

## Usage

```
const bufferFrom = require('buffer-
        from')

console.log(bufferFrom([1, 2, 3, 4]))
//=> <Buffer 01 02 03 04>

const arr = new Uint8Array([1, 2, 3, 4])
console.log(bufferFrom(arr.buffer, 1,
        2))
//=> <Buffer 02 03>

console.log(bufferFrom('test', 'utf8'))
//=> <Buffer 74 65 73 74>
```

# API

```
const buf = bufferFrom('test')
console.log(bufferFrom(buf))
//=> <Buffer 74 65 73 74>
```

## bufferFrom(array)

- array `<Array>`
  Allocates a new Buffer using an array of octets.

## bufferFrom(arrayBuffer[, byteOffset[, length]])

- arrayBuffer `<ArrayBuffer>` The .buffer property of a
  TypedArray or ArrayBuffer
- byteOffset `<Integer>` Where to start copying from
  arrayBuffer. **Default:** 0
- length `<Integer>` How many bytes to copy from
  arrayBuffer. **Default:** arrayBuffer.length -
  byteOffset

When passed a reference to the .buffer property of a
TypedArray instance, the newly created Buffer will share the
same allocated memory as the TypedArray.

The optional byteOffset and length arguments specify a
memory range within the arrayBuffer that will be shared by
the Buffer.

## bufferFrom(buffer)

- buffer `<Buffer>` An existing Buffer to copy data from
  Copies the passed buffer data onto a new Buffer instance.

## bufferFrom(string[, encoding])

- string `<String>` A string to encode.
- encoding `<String>` The encoding of string. **Default:**
  'utf8'

Creates a new Buffer containing the given JavaScript string
string. If provided, the encoding parameter identifies the
character encoding of string.

# See also

- buffer-alloc A ponyfill for Buffer.alloc
- buffer-alloc-unsafe A ponyfill for Buffer.allocUnsafe