

Map		General (cont)		Teorico (cont)		Express	
Map()	A JavaScript object that stores key-value pairs, where keys can be any type of data, and values can be any data type.	let, const, var	Variable declaration keywords: let (block-scoped), const (block-scoped and immutable), var (function-scoped).	JSON (JavaScript Object Notation)	Lightweight data-interchange format that's easy for humans to read and write, and for machines to parse and generate.	express()	The function used to create an instance of an Express application.
set(key, value)	Adds a new key-value pair to the Map. If the key already exists, it updates the value.	Promises	Represents a value that may be available in the future. It is used for handling asynchronous operations.	RESTful APIs	Design principles for building web services that are stateless and use HTTP methods for CRUD operations (GET, POST, PUT, DELETE).	app.listen(port)	Starts the Express server and listens on a specified port. For example: <code>app.listen(3000)</code> to listen on port 3000.
get(key)	Retrieves the value associated with the key from the Map. If the key doesn't exist, returns undefined.	async/await	Used to simplify asynchronous code. <code>await</code> waits for a promise to resolve, and <code>async</code> makes a function asynchronous.	CSS Flexbox	Layout model that allows for responsive and flexible design, aligning elements in rows or columns.	app.get(route, callback)	Defines a route handler for GET requests. For example: <code>app.get('/', (req, res) => res.send('Hello World'))</code>
has(key)	Checks if a key exists in the Map. Returns true or false.			CSS Grid	2D layout system for creating grid-based layouts with rows and columns.	app.post(route, callback)	Defines a route handler for POST requests. For example: <code>app.post('/submit', (req, res) => res.send('Data received'))</code>
delete(key)	Removes a key-value pair from the Map.	Teorico		CORS (Cross-Origin Resource Sharing)	A mechanism that allows restricted resources on a web page to be requested from another domain.	app.put(route, callback)	Defines a route handler for PUT requests. Typically used for updating data.
clear()	Removes all key-value pairs from the Map.	DOM (Document Object Model)	The structure that represents the HTML document as a tree of nodes, allowing dynamic manipulation.			app.delete(route, callback)	Defines a route handler for DELETE requests. Usually used to remove resources.
forEach(callback)	Iterates over each key-value pair in the Map and executes a provided function.	AJAX (Asynchronous JavaScript and XML)	Technique for making asynchronous HTTP requests to fetch data without refreshing the page.				
General							
console.log()	Outputs a message to the console, commonly used for debugging.						



By **HoshiYouta**

cheatography.com/hoshiyouta/

Published 24th February, 2025.
 Last updated 24th February, 2025.
 Page 1 of 2.

Sponsored by **ApolloPad.com**
 Everyone has a novel in them. Finish Yours!
<https://apollopad.com>

Express (cont)

`app.use(middleware)` Adds middleware to the Express app. Middleware functions execute during the lifecycle of a request. For example: `app.use(express.json())` to parse JSON in request bodies.

`res.send()` Sends a response to the client. Can send various types of data such as strings, objects, or arrays.

`res.json()` Sends a JSON response. Converts an object or array to JSON format.

NodeJs

`require()` Imports modules, JSON, or local files into your Node.js application.

`fs module` Provides an API for interacting with the file system, allowing you to read, write, and modify files.

NodeJs (cont)

`module.exports` Defines what a module exports so that other files can use it.

`fs.readFile(path, encoding, callback)` Asynchronously reads the contents of a file. The callback receives two arguments: an error (if any) and the data.

`fs.writeFile(path, data, encoding, callback)` Asynchronously writes data to a file, replacing the file if it already exists. The callback handles error (if any).

`fs.appendFile(path, data, encoding, callback)` Asynchronously appends data to a file. If the file does not exist, it is created. The callback handles error (if any).

`fs.copyFile(src, dest, callback)` Asynchronously copies a file from src to dest. The callback handles error (if any).

Events

`addEventListener()` Attaches an event handler to an element for a specific event.

`removeEventListener()` Removes an event handler from an element.

`event.target` Returns the element that triggered the event.

`event.preventDefault()` Cancels the default behavior of an event (e.g., prevents form submission).

`event.stopPropagation()` Stops the event from propagating up or down the DOM tree.

`DOMContentLoaded` Fires when the HTML has been completely loaded, but before stylesheets, images, and subframes finish loading.



By HoshiYouta

cheatography.com/hoshiyouta/

Published 24th February, 2025.

Last updated 24th February, 2025.

Page 2 of 2.

Sponsored by [ApolloPad.com](https://apollopad.com)

Everyone has a novel in them. Finish Yours!

<https://apollopad.com>