

# Most commonly used JavaScript functions for backend development.

## 1. Core JavaScript Functions

These are frequently used in both frontend and backend development:

### Data Manipulation

- `JSON.parse(string)` – Converts a JSON string to an object.
- `JSON.stringify(object)` – Converts an object to a JSON string.
- `Object.keys(obj)` – Returns an array of keys in an object.
- `Object.values(obj)` – Returns an array of values in an object.
- `Object.entries(obj)` – Returns an array of `[key, value]` pairs.
- `Array.map(callback)` – Transforms an array and returns a new one.
- `Array.filter(callback)` – Filters an array based on a condition.
- `Array.reduce(callback, initialValue)` – Reduces an array to a single value.

### String Manipulation

- `str.split(separator)` – Splits a string into an array.
- `str.trim()` – Removes whitespace from both ends of a string.
- `str.toLowerCase()` / `str.toUpperCase()` – Changes case.
- `str.includes(substring)` – Checks if a string contains a substring.

### Error Handling

- `try { ... } catch (err) { ... }` – Handles errors in asynchronous and synchronous operations.
- `throw new Error("message")` – Throws a custom error.

### Asynchronous Handling

- `setTimeout(fn, delay)` – Executes a function after a delay.
- `setInterval(fn, interval)` – Executes a function repeatedly at intervals.
- `Promise.resolve(value)` – Creates a resolved promise.
- `Promise.reject(error)` – Creates a rejected promise.
- `Promise.all([...])` – Waits for multiple promises to resolve.
- `Promise.race([...])` – Resolves with the first settled promise.

---

## 2. Node.js-Specific Functions

Node.js provides several built-in modules for backend operations.

### File System (fs)

- `fs.readFile(path, callback)` – Reads a file asynchronously.
- `fs.writeFile(path, data, callback)` – Writes data to a file.
- `fs.appendFile(path, data, callback)` – Appends data to a file.
- `fs.unlink(path, callback)` – Deletes a file.

### Path Handling (path)

- `path.join(...segments)` – Joins multiple path segments.
- `path.resolve(...segments)` – Resolves an absolute path.

### HTTP & Express.js

- `app.get('/route', (req, res) => { ... })` – Handles GET requests.
- `app.post('/route', (req, res) => { ... })` – Handles POST requests.
- `app.put('/route', (req, res) => { ... })` – Handles PUT requests.
- `app.delete('/route', (req, res) => { ... })` – Handles DELETE requests.
- `res.json(data)` – Sends JSON response.
- `res.status(code).send(data)` – Sends response with status code.

### Database Handling (MongoDB, SQL, etc.)

- `db.collection('users').find(query).toArray()` – Fetches data in MongoDB.
- `db.collection('users').insertOne(document)` – Inserts a document.
- `db.collection('users').updateOne(query, update)` – Updates a document.
- `db.collection('users').deleteOne(query)` – Deletes a document.
- `db.query('SELECT * FROM users', callback)` – Executes an SQL query.

---

## 3. Utility Functions for Backend

- `require(moduleName)` – Imports a Node.js module (CommonJS).
- `import moduleName from 'module'` – Imports an ES6 module.
- `console.log(value)` – Logs data to the console.
- `crypto.randomBytes(size)` – Generates secure random bytes (for tokens).

- `process.env.VARIABLE` – Accesses environment variables