

Basic Concepts and Execution

Node.js is a cross-platform JavaScript runtime environment for executing code outside the browser.

- Execution: `node app.js`: Run a script.
- REPL: Type `node` to enter interactive mode.
- `__dirname`, `__filename`: Current directory and file path.
- `process`: Global object providing information about the current process.

Modules and Packages

CommonJS (Traditional)

- `const fs = require('fs');`
- `module.exports = { ... };`

ES Modules (Modern)

- `import fs from 'fs';`
- `export const value = 10;`
- Enabled by "type": "module" in `package.json` or `.mjs` extension.

NPM / PNPM / Bun

- `npm init -y`: Initialize a project.
- `npm install [package]`: Install a dependency.
- `npm run [script]`: Run scripts defined in `package.json`.

Built-in Modules

- `fs`: File system (Sync/Async/Promise).
 - `fs.readFile()`, `fs.writeFile()`.
- `path`: Manage file and directory paths.
 - `path.join()`, `path.resolve()`.
- `http`, `https`: Low-level server creation.
- `os`: System information (CPU, memory, etc.).
- `events`: `EventEmitter` class for event-driven programming.
- `crypto`: Encryption and hashing.

File System (fs/promises)

```
import { readFile, writeFile } from 'fs/promises';
```

```
async function processFile() {
```

```
  const data = await
  readFile('input.txt', 'utf8');
  await writeFile('output.txt',
  data.toUpperCase());
}
```

Express.js (High-level Server)

```
import express from 'express';
const app = express();

app.get('/', (req, res) => {
  res.send('Hello World!');
});
```

```
app.listen(3000, () =>
  console.log('Server running on 3000'));
```

Events (EventEmitter)

```
import EventEmitter from 'events';
const emitter = new EventEmitter();

emitter.on('event_name', (arg) => {
  console.log('Event received:', arg);
});

emitter.emit('event_name', { id: 1 });
```

Streams

Handling large amounts of data efficiently.

- `fs.createReadStream()`.
- `pipe()`: Connect readable and writable streams.

Process and Environment

- `process.env`: Access environment variables.
- `process.exit(0)`: Terminate the process.
- `process.argv`: Access command line arguments.
- `dotenv`: Popular package for loading `.env` files.

Google Style Guide (Node.js/JS)

General Principles

- Use `ES Modules` (`import/export`) for new projects.
- Use `async/await` instead of callbacks for asynchronous operations.
- Prefer `const` over `let`, and avoid `var` entirely.

Error Handling

- Always handle Promise rejections. Use `try-catch` with `async/await`.
- Create custom Error classes for domain-specific errors.
- Never crash the process without logging or cleaning up.

Formatting and Structure

- Indentation: 2 spaces.
- Semicolons: Required.
- Quotes: Use single quotes `'` for strings.
- Filenames: Use `kebab-case.js` or `lower_with_underscores.js` consistently.

Security

- Never include secrets in your source code. Use environment variables.
- Keep dependencies updated using `npm audit`.
- Sanitize all user inputs before processing or DB usage.