🟩 Introduction to REPL (Read-Eval-Print Loop) 🚀

REPL stands for **Read-Eval-Print Loop** (or **Read, Evaluate, Print, and Loop**).
It is an **interactive programming environment** that allows you to execute JavaScript code **one statement at a time**.

🔹 Key Features of Node.js REPL

- Can be opened in your terminal using the command:

```
node
```

- Useful for **testing small JavaScript snippets** quickly, similar to browser DevTools.
- Exiting REPL:

```
Ctrl + C
```

Press twice to exit.

- Supports **multi-line expressions**, variable declaration, function testing, and quick debugging

---

🧩 How REPL Works (Step by Step)

1. Read

- The REPL **reads the user’s input** (single or multiple lines of code).
- Parses it into a **data structure** that the JavaScript engine can understand.

2. Eval (Evaluate)

- The parsed input is **evaluated/executed** by the JavaScript engine.
- If the expression is valid, REPL computes the **result**.

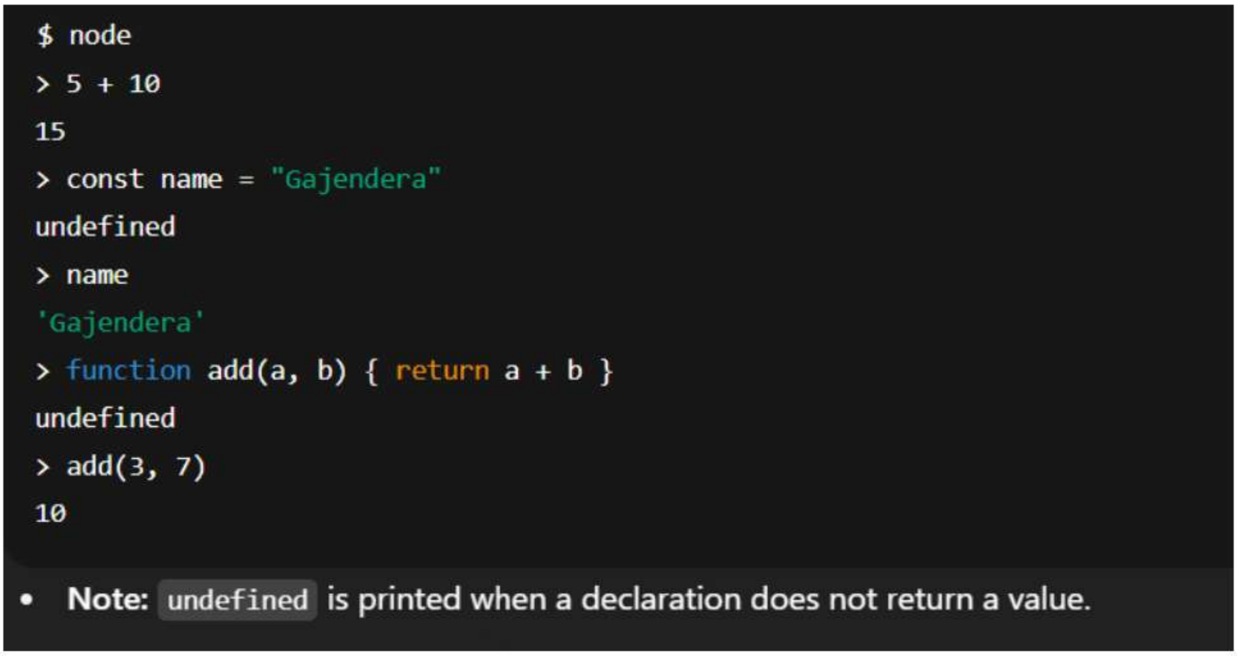
3. Print

- The **result** of the evaluated expression is printed back to the console.
- Allows the user to **see the output immediately**.

4. Loop

- The process **repeats continuously**, waiting for the next input.
- Continues until the user exits REPL.

⚡ Example Usage



```
$ node
> 5 + 10
15
> const name = "Gajendera"
undefined
> name
'Gajendera'
> function add(a, b) { return a + b }
undefined
> add(3, 7)
10
```

- **Note:** `undefined` is printed when a declaration does not return a value.

- **Note:** `undefined` is printed when a declaration does not return a value.

✅ Why REPL is Useful

- Quick **experimentation** without writing a full script.
- Ideal for **learning JavaScript** or testing **Node.js APIs**.
- Helps with **debugging and prototyping** small pieces of code.

💡 Tip

- You can use **special REPL commands**:

Command	Description
`.help`	Show all REPL commands
`.break`	Exit multi-line mode
`.clear`	Reset context
`.exit`	Exit REPL