Getting Started with JavaScript & Node.js

AiCE Software Engineering Foundations — JavaScript & Node.js

You already know Python from beginner to advanced level. This section adds a second pillar to your skillset: JavaScript. Because you already understand programming fundamentals, this section moves quickly, leans heavily on comparisons to Python, and focuses only on what you'll actually use.

1. What Is JavaScript?

JavaScript is a programming language, just like Python — you write instructions, the computer executes them. But JavaScript has one special quality: it is the only language that runs natively inside a web browser. Every button, dropdown, animation, and live notification on a website is JavaScript.

In 2009, Ryan Dahl created Node.js — a runtime that lets JavaScript run outside the browser, directly in a terminal or on a server. That's why JavaScript can be written and run in a terminal, exactly like Python. Companies like Netflix, Uber, Paystack, and Flutterwave all use JavaScript and Node.js to power their backends.

2. JavaScript vs. Python — A Quick Comparison

Since you already know Python, the fastest way to learn JavaScript is to see it side by side with what you already know:

Concept	Python	JavaScript
Print output	print("Hello")	console.log("Hello")
Declaring a variable	name = "Adesina"	let name = "Adesina";
Declaring a constant	No true constant (convention: ALL_CAPS)	const age = 26;
Single-line comment	# comment	// comment
Multi-line comment	""" comment """	/* comment */

Concept	Python	JavaScript
Boolean values	True / False	true / false
Checking a value's type	type(value)	typeof value

The concepts are identical — only the syntax changes. Keep this comparison in mind throughout this section: every time something in JavaScript feels strange, ask yourself "what is the Python equivalent?"

3. Installing Node.js

Open the Ubuntu terminal and run the following, pressing Enter and typing y when prompted:

```
sudo apt update
sudo apt install nodejs
sudo apt install npm
```

npm stands for Node Package Manager — it does the same job in Node.js that pip3 does in Python. Confirm both installed correctly by checking their versions:

```
node --version
npm --version
```

Both should print a version number. JavaScript can now run in this terminal.

4. Running JavaScript Files

Just as a Python file has the .py extension, a JavaScript file has the .js extension. Just as Python files run with python3 filename.py, JavaScript files run with:

```
node filename.js
```

Set up a working directory exactly as you did for Python: open GitHub, create a new repository named aice-javascript-programming, generate a token, and clone the repository. That cloned folder is your working directory for this entire section.

5. console.log, Comments & Semicolons

In JavaScript, output is printed with console.log() instead of print(). Comments use // for a single line and /* */ for multiple lines, instead of # in Python. Professional JavaScript ends statements with a semicolon ; — though this is technically optional, since JavaScript code without semicolons still runs.

Create a first file named hello.js:

```
console.log("Hello, Africa!");
console.log("Name:", "Adesina", "| Age:", 25);
console.log(10 + 5);
console.log("AiCE".toUpperCase());
```

Save it, then run it:

```
node hello.js
```

Line	What It Shows
console.log("Hello, Africa!");	Prints text to the terminal — the same job print() does in Python.
console.log("Name:", "Adesina", ...)	console.log can print multiple values at once, each separated by a comma.
console.log(10 + 5);	Mathematical calculations can be done directly inside console.log.
console.log("AiCE".toUpperCase());	JavaScript has string methods, just like Python — here, converting text to uppercase.

Key Takeaways

JavaScript runs in the browser and, since Node.js, in the terminal too — installed on Ubuntu with apt, run with node filename.js, exactly the shape of the Python workflow you already know. console.log() replaces print(), // and /* */ replace #, and semicolons are the professional convention even though JavaScript doesn't strictly require them. With Node.js installed and your first file running, the next lesson covers how JavaScript actually stores information — declaring variables with let and const, the core data types, and template literals, JavaScript's answer to Python's f-strings.