$ node zero_to_builder.js

English edition



JAVASCRIPT + TYPESCRIPT

ZERO TO BUILDER

The JavaScript + TypeScript book that teaches you to **build**, not just memorize.

WHAT'S INSIDE

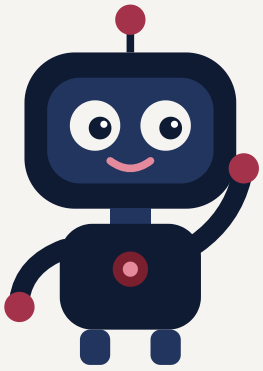
- Step-by-step explanations
- 8 real projects
- Quizzes in every chapter
- Exercises at 3 levels
- Final exam
- Answer key + solution code
- Oops corners: common errors
- AI study tips
- Source code included

Elmasryx

Beginner friendly · No experience needed · English + Arabic editions

CHAPTER 00 · START HERE

Hey! Let's turn you into a builder.



BIT · YOUR GUIDE

Hi, I'm **Bit**. I'll be on every page with you. When something looks scary, I'll jump in and make it simple. Deal?

JavaScript runs inside **every web browser on Earth**. It's what makes websites come alive: buttons that react, menus that open, live chats, games. **TypeScript** is the same JavaScript with a safety belt that catches mistakes before your users see them. Learn both, and they pay you back:

Build websites
Every interactive website on the internet uses it.

Get hired
Front-end, back-end, and full-stack jobs ask for it.

Freelance
Clients pay for forms, dashboards, and web apps.

One language, everywhere
Websites, servers, mobile and desktop apps.

66% of developers use JavaScript, the #1 language in the world's biggest developer survey. And TypeScript became the #1 language on GitHub in August 2025.

Sources: Stack Overflow Developer Survey 2025 · GitHub Octoverse 2025

Elmasryx

CHAPTER 00 · SETUP

Ready in 10 minutes.

Good news: you already own the most important tool. Your **web browser** can run JavaScript. You only need one free notebook to write it in.

1 Get VS Code

Download it free from **code.visualstudio.com** and install it. That's where you'll write your code.

2 Make a folder with two files

Create a folder called `my-js`. Inside it, create `index.html` and `app.js`. Put this in `index.html`:

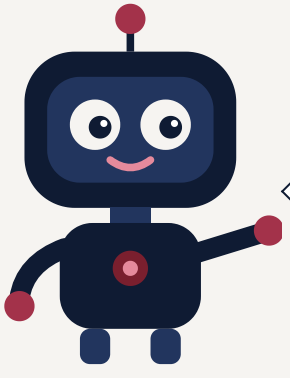
INDEX.HTML

```html
<!DOCTYPE html>
<html>
  <head><meta charset="UTF-8"></head>
  <body>
    <h1>My first JS page</h1>
    <script src="app.js"></script> 1
  </body>
</html>
```

3 Open the console

Double-click `index.html` to open it in Chrome or Edge. Press **F12** (Mac: ⌥ ⌘ I) and click the **Console** tab. That's your JavaScript screen.

1 This line tells the page: "run the JavaScript in `app.js`." You'll write all your code there.



BIT'S TIP

Changed `app.js`? Save, then press **F5** to refresh. No laptop? Search "online JavaScript playground" and run the examples in your browser.

One line. Your first program.

Open `app.js` and type this one line. Save, then refresh the page.

APP.JS

```
console.log(1)("Hello, world!"2);3
```

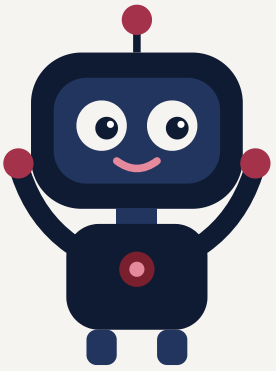
1. `console.log` means **"show this in the console."**
2. Text goes inside **quotes**. That's how JavaScript knows it's text, not a command.
3. The **semicolon** ends the line, like a full stop ends a sentence.

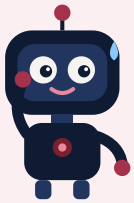
OUTPUT

```
Hello, world!
```

BIT

You just told a computer what to do, **and it listened.** Want it on the page itself? Try `alert("Hi!");` and refresh. A pop-up says hi.





OOPS CORNER

Forgot the quotes? JavaScript thinks `Hello` is a name it should know:

```
Uncaught ReferenceError: Hello is not defined
```

Fix: put your text back inside quotes. The red text in the console tells you the line number, too.

CHAPTER 01

01 JavaScript Basics

Every app you use does three things: it **remembers** things, it **calculates**, and it **talks** to you. After this chapter, your code will do all three.

`let` `const` Boxes with labels

WHAT IT DOES

Stores a value under a name, so you can use it again later.

WHY YOU NEED IT

A shopping cart remembers items. A game remembers your score. That memory is variables.

APP.JS

```
const① name = "Sara";
let② age = 24;
age = age + 1;③   // happy birthday!

console.log(name, "is now", age);
```

1. `const` = a box that **never changes**. Use it by default.
2. `let` = a box you **can change** later.
3. Take the old value, add 1, put it back in the same box.

OUTPUT

```
Sara is now 25
```



OOPS CORNER

Tried to change a `const`? JavaScript protects it:

```
TypeError: Assignment to constant variable.
```

Fix: if the value must change, use `let`.

CHAPTER 01 · TYPES AND MATH

Text, numbers, true or false.

"Sara"	24	true	null
`string` Text. Always in quotes.	`number` Whole or decimal: 9.99	`boolean` Yes or no. true / false.	`empty` "Nothing here on purpose."

`${ }` Put values inside text

WHAT IT DOES
Builds a sentence that mixes text and values, cleanly.

WHY YOU NEED IT
Every "Hi, Sara!" and "Total: $42" on a website is built this way.

MATH.JS

```
const item = "Coffee";
const price = 3.5;
const qty = 4;

const total = price * qty;
console.log(`${item} x ${qty} = $${total}`);
console.log(10 % 3);
```

1. Math works like a calculator: `+ - * /`.
2. Backticks `` ` `` (not quotes!) let you drop values in with `${ }`.
3. `%` gives the **remainder**: 10 ÷ 3 = 3, remainder **1**.

OUTPUT

```
Coffee x 4 = $14
1
```

CHAPTER 01 · TALK TO THE USER

Let your page have a conversation.

`prompt()` Ask the user

WHAT IT DOES
Shows a box where the user types an answer. Your code gets it back.

WHY YOU NEED IT
Programs that listen are useful: calculators, quizzes, forms.

APP.JS

```
const name = prompt(1)("What's your name?");
const age = Number(2)(prompt("How old are you?"));

console.log(`Hi ${name}! In 10 years: ${age + 10}`);
```

1. `prompt` shows a question and returns what the user typed, always as **text**.
2. `Number()` turns text like `"21"` into the number `21`, so math works.

OUTPUT

```
Hi Omar! In 10 years: 31
```



OOPS CORNER

Skip `Number()` and JavaScript glues text together instead of adding:

`"21" + 10 → "2110"`

Fix: wrap anything you'll do math with in `Number()`.

BIT

Real websites use input boxes, not `prompt`. You'll build those in Chapter 6. For now, `prompt` keeps things simple.

