Unlock your potential and master web development

PROGRAMMING STEP BY STEP AND MORE AN INSTRUCTIVE GUIDE

UP-TO-DATE, VERSATILE, ADVANCED

BOOK 3 - Tercera Parte

JavaScript. POO, Prototype

Rene F. Ruano Domínguez

DEDICATION

TO THE GENERATIONS OF THE FUTURE, MILLENNIALS, BABY BOOMERS AND BEYOND.

This How-To is dedicated to everyone: To you, the brilliant young minds shaping the world with your ingenuity and creativity. To the millennials, who navigate the digital age with ease and constantly seek new ways to innovate. To the baby boomers, who pioneered the technological revolution and now contribute their experience and wisdom. To the retirees who dream of staying up-to-date, filling their free time with useful hours, and exercising their minds.

Prologue

Book 3 – Third Part: JavaScript. OOP, Prototype continues the "**Programming Step by Step and More. Instructive Series**", diving into intermediate and advanced JavaScript. This volume expands the guide to the language, covering fundamentals and complex concepts through hundreds of hands-on exercises that strengthen the reader's skills with updated tools for today's professional environment.

We explore Object-Oriented Programming (OOP) and its connection to JavaScript's prototype-based inheritance, offering clear explanations of prototypes, the prototype chain, and properties **like __proto__, Object.getPrototypeOf(), and prototype.** The book also examines **ES6** classes and compares them with traditional constructor functions.

This volume stands as a valuable contribution to the developer community, offering a solid foundation for modern application design and reflecting the author's commitment to delivering advanced knowledge in an accessible and effective way. Every page showcases his ability to simplify complex ideas and his dedication to excellence, making this work an essential resource for anyone aiming to grow as a developer.

With admiration and gratitude,

Idalmy Baluja Conde

GRATITUDE

To my family, my colleagues, my friends.

Thank you, thank you so much for the unconditional support you've given me since I started this project. And to those who didn't believe in me, I'm grateful for motivating me to prove that it was worth it. Without the encouragement, suggestions, trust they placed in me, and the personal challenge of creating a useful product, this project would not have been possible.

BOOK 3 - Tercera Parte

JavaScript. POO, Prototype.

René F. Ruano Domínguez

1 OBJECTIVES AND SCOPE OF BOOK 3

The *Programming Step by Step and More* series has become a definitive guide for anyone aiming to master web development through a clear, concise, and structured methodology built on a Learning Management System.

This third volume takes JavaScript mastery to the next level. Readers will not only deepen their understanding of the language but also dive into advanced application development, exploring everything from **Object-Oriented Programming (OOP)** to the internal architecture of **Prototypes**—core pillars for building dynamic, efficient, and scalable software in today's environment.

1.1 Thematic Content: JavaScript, OOP, and Prototypes

In this volume, we explore the concepts that enable developers to write modular, high-performance code. The course is designed for an intermediate-to-advanced level, preparing developers to tackle real-world challenges with a solid understanding of how JavaScript works "under the hood."

1.1.1 Module I: Advanced Object-Oriented Programming (OOP)

We approach OOP not just as theory, but as a practical tool for building maintainable architecture.

- **Core Principles:** Encapsulation (including internal data control), Abstraction, and the use of abstract classes.
- **Modern Inheritance:** ES6 class syntax and the creation of complex hierarchies.
- **Polymorphism in JavaScript**:
    - Subtype Polymorphism (Inheritance)
    - Parametric Polymorphism (Higher-Order Functions)
    - Ad Hoc Polymorphism (Duck Typing)
- **Technical Comparison:** How JavaScript implements OOP compared to traditional languages like Java or C++.

1.1.2 Module II: The World of Prototypes

JavaScript is unique for its prototype-based inheritance model. We break down this system to optimize memory usage and performance.

- **Inheritance Mechanics:** Using Object.create(), getPrototypeOf(), and setPrototypeOf().
- **Prototype Chain:** How the JS engine searches for and assigns properties.
- **Critical Differences**: __proto__ vs. prototype, and how to restore the constructor property.
- **Extending Native Objects**: Techniques and risks of modifying prototypes of Array, String, and Date.
- **Under the Hood:** How ES6 classes are internally transformed into prototypes.

Traditional Constructor Functions and **Modern ES6 Classes**

Characteristic	Constructor Functions (Traditional)	ES6 Classes (Modern)
Syntax	Based on function and this.	Based on the class keyword.
Methods	Manually added to .prototype.	Defined inside the class body.
Inheritance	Requires Object.create() and manual constructor adjustment.	Uses extends and super keywords.
Hoisting	Functions are hoisted (usable before declaration).	Classes are not hoisted (must be declared before use).
Strict Mode	Not enforced by default.	Class bodies always run in Strict Mode.
Readability	Can become confusing in deep hierarchies.	Cleaner, more organized, and easier to maintain.
Nature	The foundational model of JavaScript.	Syntactic sugar over the prototype-based model.

1.1.3 Module III: Design Patterns, Composition, and Practice

- **Creational Patterns**: Implementation of Prototype, Factory, Singleton, and Builder patterns.
- **Composition Over Inheritance:** When and why to use Mixins and "has-a" relationships instead of "is-a."
- **Capstone Project:** Development of a modular web page applying prototypal inheritance and professional file organization (CSS, HTML, JS).
- **Intensive Practice:** Over 60 hands-on exercises strategically distributed to reinforce learning.

Project structure and files:

- 10-03proto_style.css
- 10-03proto_header.html
- 10-03proto_article.html
- 10-03proto_footer.html
- 10-03proto_app.js
- 10-03proto_index.html
- 10-03proto_main.js

1.2 Professional Benefits and Advantages

a. Abstraction Skills: Ability to simplify complex problems into modular, reusable solutions.
b. Resource Optimization: Efficient memory usage through shared methods in the prototype chain.
c. Framework Readiness: A solid foundation for understanding how libraries like React, Vue, or Angular work internally.
d. AI Integration: In a market reshaped by AI, mastering JavaScript is essential. Leading tools and APIs (such as TensorFlow.js or OpenAI) rely on advanced object manipulation to integrate AI models directly in the browser.

Learning web development at this level is a high-return investment. It not only provides competitive technical skills but also fosters creativity and professional growth in an industry that rewards those who understand the essence of technology.

CONCLUSIONS BOOK 3, PART THREE

In this Book 3, Part Three: JavaScript, OOP, Prototype, we have delved deeper into the fundamentals of Object-Oriented Programming (OOP), exploring both its practical application and its integration with the prototype-based inheritance model. Throughout the modules, a detailed understanding of key concepts has been provided, such as creating and modifying prototypes, the prototype chain, and the differences between properties like __proto__, Object.getPrototypeOf(), and prototype.

Furthermore, the class syntax introduced in ES6 has been analyzed, comparing it to traditional constructor functions and demonstrating how both are prototype-based.

Design patterns such as Prototype, Factory, Singleton, and Builder have been implemented, offering practical examples that illustrate their usefulness in application development. The importance of following best practices has also been highlighted, such as the proper use of methods like hasOwnProperty(), in, and for...in, and memory optimization through prototypical inheritance. The principle of "Composition over Inheritance" has also been addressed, providing clear guidelines for deciding when composition is preferable to inheritance.

The chapter culminated in a practical project that integrated all the concepts learned, enabling the creation of a modular and responsive website through the application of prototypical inheritance and composition. This approach not only reinforced theoretical understanding but also prepared participants to tackle real-world challenges in developing efficient and maintainable JavaScript applications.

In summary, this chapter has provided a solid foundation for mastering OOP in JavaScript, combining theory and practice to foster the development of essential skills in the field of modern web development.

5. REFERENCES

1. MDN Web Docs - Encapsulation in JavaScript
   https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Details_of_the_Object_Model#encapsulation
2. Abstraction in JavaScript.
   https://medium.com/javascript-scene/javascript-factory-functions-with-es6-4d224591a8b1
3. Class Inheritance in JavaScript
   https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Classes#subclassing_with_extends
4. Polymorphism in JavaScript
   https://www.freecodecamp.org/news/object-oriented-programming-concepts-in-javascript/
5. Prototypal Inheritance in JavaScript
   https://developer.mozilla.org/en-US/docs/Web/JavaScript/Inheritance_and_the_prototype_chain
6. Higher-Order Functions in JavaScript
   https://eloquentjavascript.net/05_higher_order.html?form=MG0AV3
7. Polymorphism and Hierarchies in JavaScript
   Polymorphism and Hierarchies in JavaScript
8. Prototypal vs Classical Inheritance in JavaScript
   https://www.codementor.io/@dariogarciamoya/understanding-prototypal-vs-classical-inheritance-in-javascript-du1086uy0
9. Class Syntax in JavaScript
   https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Classes
10 Object Literals in JavaScript
11. Constructor Functions in JavaScript
12. Using the new operator in JavaScript

ABOUT THE AUTHOR.

After several decades dedicated to improving the energy efficiency of heating and cooling equipment and systems, I faced a major life change: retirement. Upon retiring, I felt disoriented and without purpose, which affected my emotional well-being. To overcome this feeling, I sought a new activity that would keep me active and motivated.

I decided to share my experiences and the knowledge I had accumulated over more than four decades working in the industry. This is how this project was born—with the goal of publishing articles, calculation tools, and guides on the various disciplines I had practiced throughout my engineering career, including web programming, which I used to optimize project management, investments, finance, and marketing. Based on my own experience, web programming is a powerful tool for solving complex problems in both industrial environments and technical services, as well as in everyday life.

My commitment to learning and technological renewal has driven me to stay updated and deepen my knowledge in new areas.

This **Third Part of PROGRAMMING STEP BY STEP AND MORE** was created from the conviction that JavaScript and the understanding of POO, Prototype are essential in today's digital world.

This project is designed to help people of all ages enhance their professional skills and achieve their career goals. Whether starting a new career or seeking new ways to apply one's experience, this resource provides the tools and opportunities needed to grow.

Learning a new programming language, such as JavaScript, is an excellent way to make good use of free time, strengthen one's résumé, and stay up to date in an increasingly digital job market.

The Author

PROGRAMMING STEP BY STEP AND MORE ®
An Instructive Guide - Book 3 · Part Three
JavaScript. POO, Prototype
Published in Paperback and Digital
Miami, Florida, USA · 2025
English-language edition © 2025 René F. Ruano Domínguez
Translated from Spanish by René F. Ruano Domínguez
Originally published in Spanish as *"PROGRAMANDO PASO A PASO Y MÁS* – INSTRUCTIVO – LIBRO 3: JavaScript. POO, Prototype

Publisher: Corporate Luxury Group, LLC