Executive Summary

In this paper, we explain the history of and motivation behind the JavaScript programming language. We then use this background as a framework for explaining the use of prototypes in JavaScript.

In the JavaScript Overview section, we go over when, where, why, and by who JavaScript was developed. We continue into a discussion of some derivative and related languages as well as its formalization into ECMAScript. After that, we go over the type system used by JavaScript including a formal analysis with inductive definitions. Next, we talk about which programming language paradigms that it falls into. We also discuss JavaScript's use of first-class functions and how it handles operator precedence. We end this section with the major revisions and additions that the language has had since its inception.

In the JavaScript Prototypes section, we begin by defining the concept of a prototype and explaining the implications of their use. Following that is a simple example of how to implement an object using a prototype. We then compare and contrast prototype-based programming with the class-based programming of the Java language. We give examples of the differences in implementation of an object's constructor, attributes, and functions. Here we show a formal analysis of the differences between inheritance and subtyping in JavaScript and Featherweight Java. We then present several concrete examples of the differences implemented in both JavaScript and Featherweight Java.