**Chapter 1**

**Presentation Format:**

JavaScript Grammar is not a complete JavaScript reference or manual. The subjects were reduced to only what’s important in modern-day JavaScript environment.

Namely: imports, classes, constructors, key principles behind functional programming, including many features ranging from ES5 - ES10 are covered in this book.

**Creative Communication:**

Some of JavaScript is easy, some of it is difficult. Not everything can be explained by source code alone. Some things are based on intangible ideas or principles.

Throughout this tutorial book you will come across many creative communication devices, designed to make the learning process a bit easier and perhaps more fun. One example of that is color-coded diagrams.

**Theory:**

Not all subjects require extensive theory. On the other hand, some things won’t make any sense without it. Additional discussion will be included, where it becomes absolutely necessary, in order to fully understand a particular concept.

**Color-Coded Diagrams:**

A significant amount of effort went into creating diagrams describing fundamental ideas behind JavaScript.

They were designed for communicative value, hopefully they will speed up the learning process in places where hard to grasp abstract ideas need to be explained visually.

There are two types of diagrams in this book: abstract ideas and source code close ins.

**Source code:**

Source code listings will be provided to cement the foundational principles from preceding text.

Example:

// Create (instantiate) a sparrow from class Bird

let sparrow = new Bird(“sparrow”,”gray”);

sparrow.fly();

sparrow.walk();

sparrow.lay\_egg();

sparrow.talk(); //error,only parrot can talk

This is an example of instantiating sparrow object from Bird class and using some of its methods.

**Code close ins:**

Most of the source code is accompanied by source code listings.

But when we need to close in on a particular important subject, a slightly larger diagram with source code and additional color-coded highlighting will be shown.

For example, here is exploration of an anonymous function when used in the context of a event callback function:

setTimeout(function() {

console.log(“print something in 1 second”);

}, 1000);