# JavaScript Compiler Optimization

[V8](https://v8.dev/) is Google’s engine for compiling our JavaScript. Firefox has its own engine called Spider Monkey; it is quite like V8 but there are differences.The purpose of V8 is to optimize our code and our purpose should be to write readable and maintainable code

* Speculative optimization

This post is all about speculative compilation, or just *speculation* for short, in the context of the JavaScriptCore virtual machine. Speculative compilation is ideal for making dynamic languages, or any language with enough dynamic features, run faster

* Hidden classes for dynamic lookups

Hidden Class is not a javascript feature, it’s a V8’s feature. Every object/primitives that you define are mapped to a certain hidden class. We as programmer should not change the object structure because it creates a new hidden class every time a property is added or deleted

* Function inlining

When executing a given piece of code, whenever you call a standard function the execution time is slightly higher than dumping there the code contained into that function. Dumping every time the whole code contained in a function is on the other end unmainteinable because it obviously leads to a whole mess of duplication of code.

*Inlining* solves the performance and maintainability issue by letting you declare the function as *inline* (at least in C++), so that when you call that function - instead of having your app jumping around at runtime - the code in the inline function is injected at compile time every time that given function is called.

**Garbage collection**

Garbage collection is a way of managing application memory automatically. The job of the garbage collector is to reclaim memory occupied by unused objects .The way how the garbage collector knows that objects are no longer in use is that no other object has references to them. It frees up memory that no longer is used.