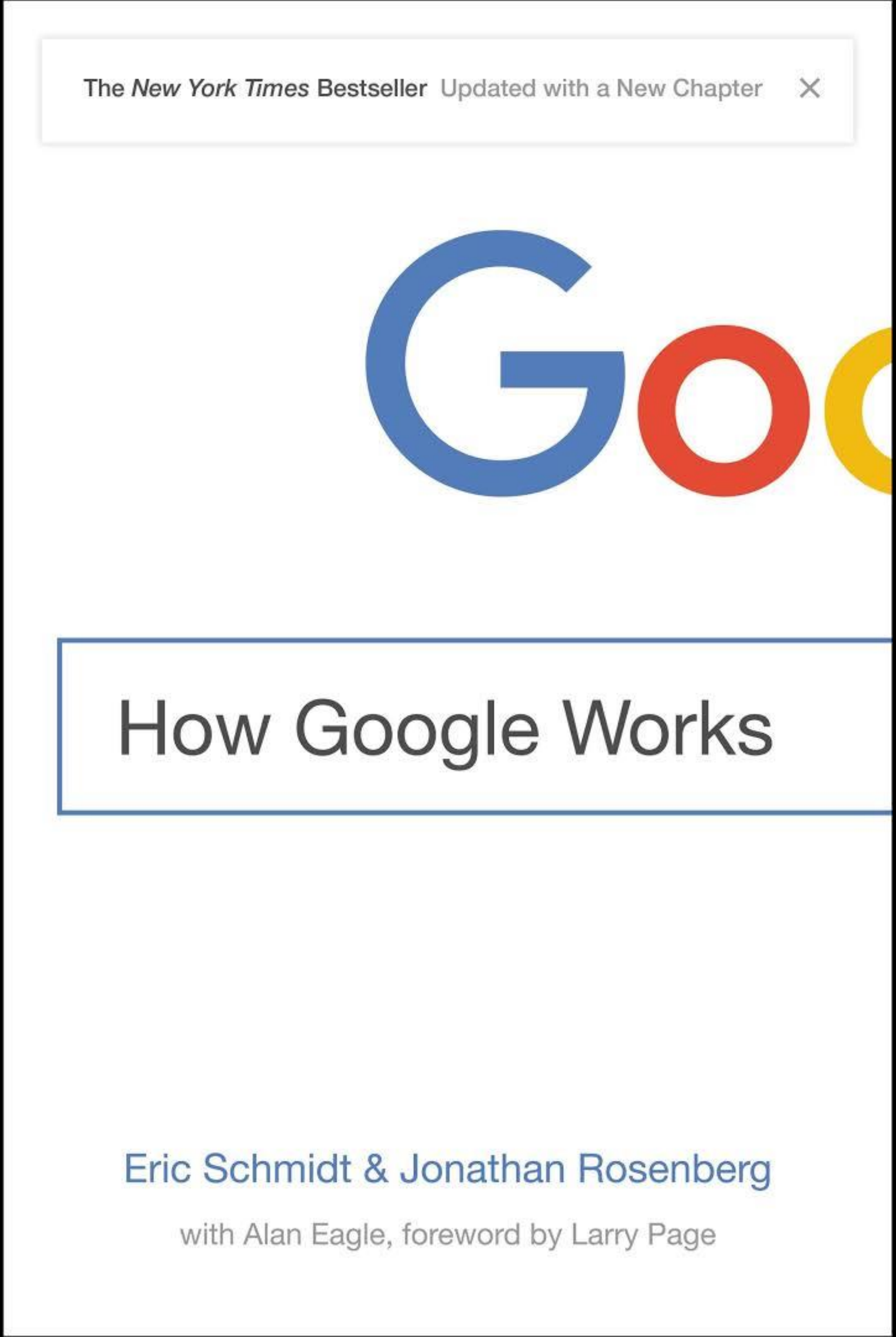
QCon 分享

关于我

- 在google plus frontend infra干过三年
- 主要负责写各种前端工具，维护google css compiler
- 其他的google时间花在了MapReduce team cloud team Security team

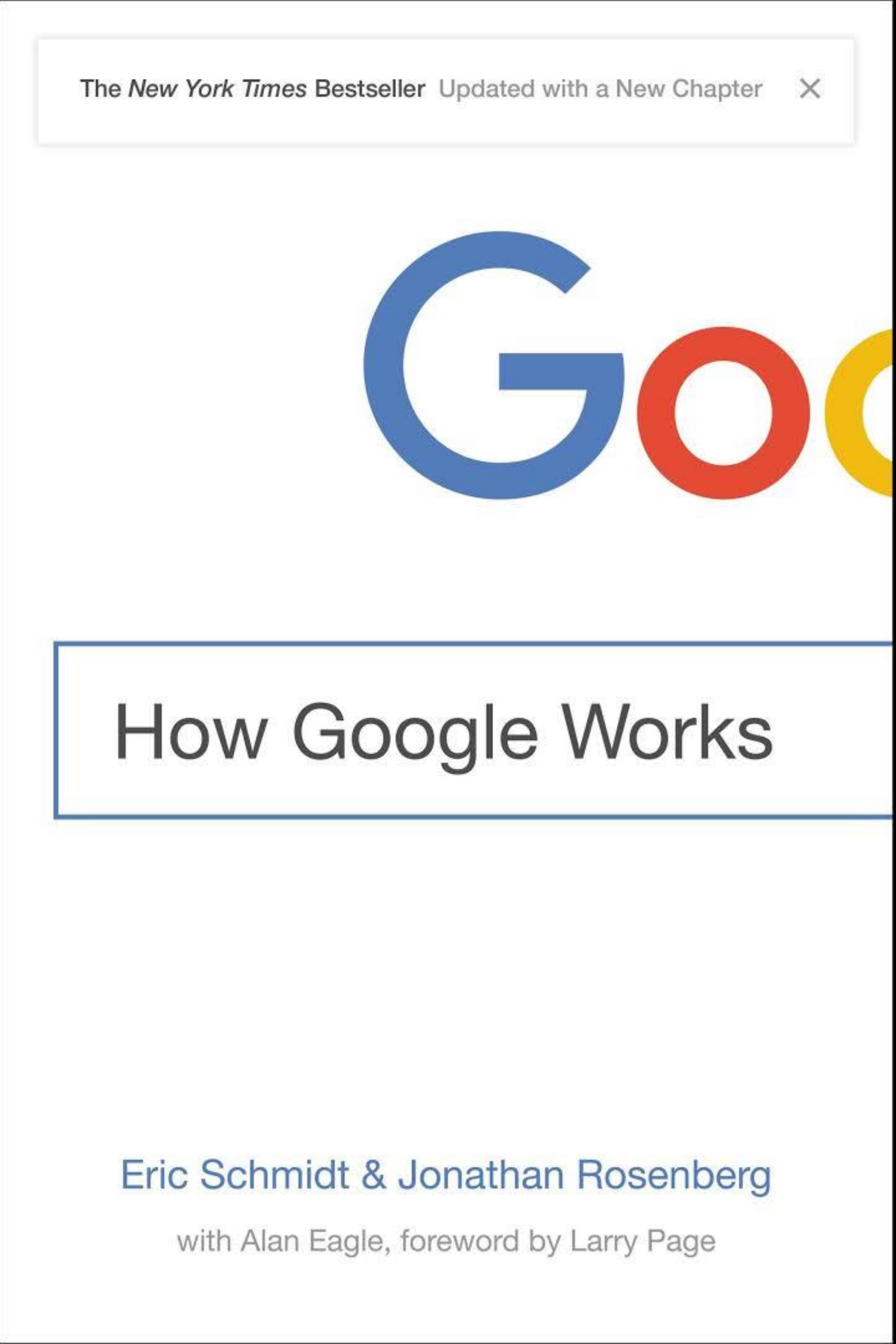


The *New York Times* Bestseller Updated with a New Chapter

How Google Works

Eric Schmidt & Jonathan Rosenberg

with Alan Eagle, foreword by Larry Page



The *New York Times* Bestseller Updated with a New Chapter

How Google Works

Eric Schmidt & Jonathan Rosenberg

with Alan Eagle, foreword by Larry Page

Almost everything happens in Google doesn’t apply to the outside

该演讲希望传达的观点

- 讲讲故事，大家听听就算了

激进 vs 保守

前端技术最激进的公司，没有之一

- maps: world’s first truly Ajax application
- gmail: hundred millions lines of javascript code
- spreadsheet: complex formula calculation in browser

前端技术最激进的公司，没有之一

- Offline docs: combine appcache and browser storage to offer native application experience
- adword frontend: GWT
- wave: GWT

这些流行的事情和G基本无关

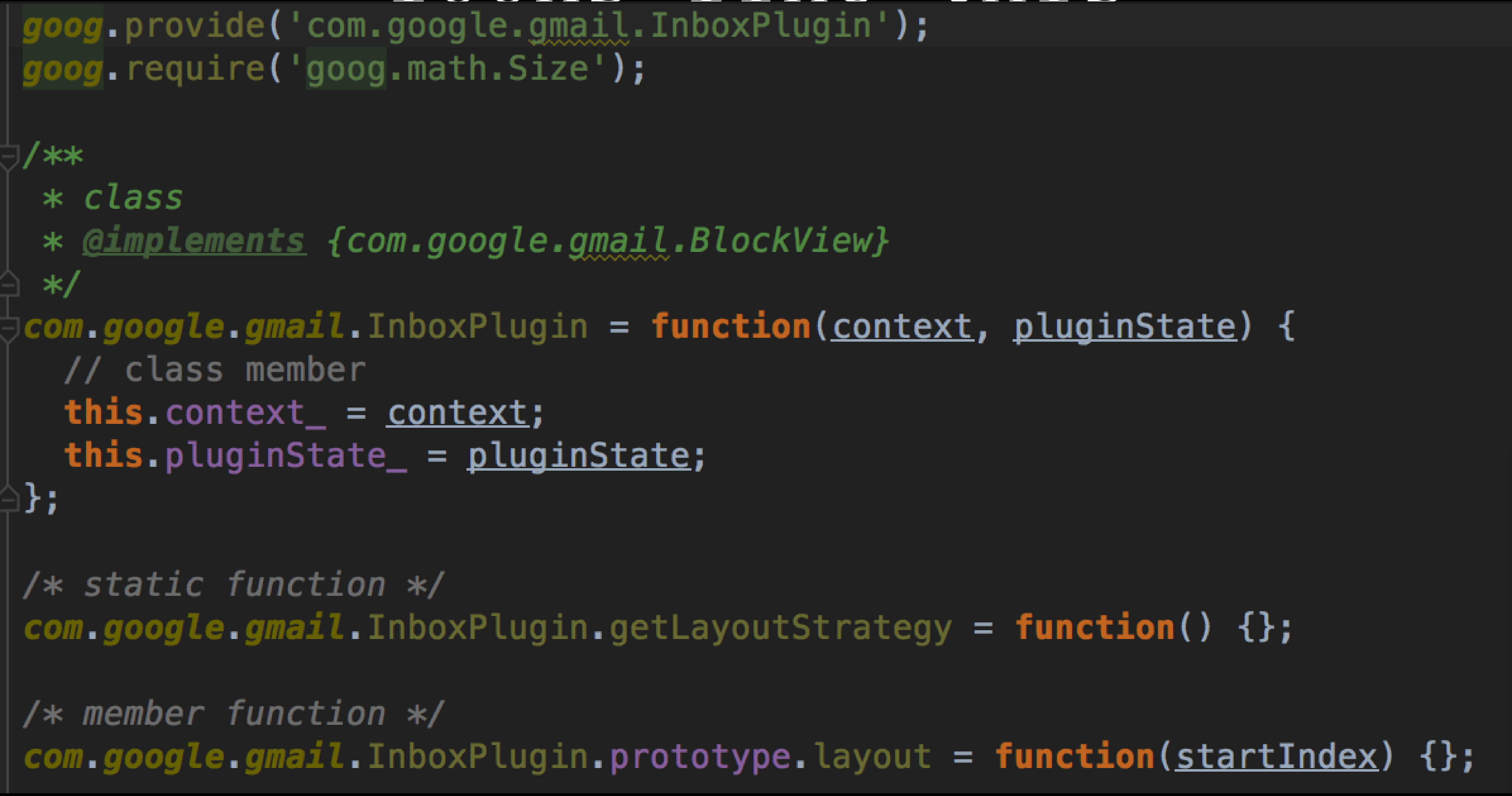
- node
- npm
- web pack
- es6
- less
- babel
- AMD
- …

G’ s world

- plain javascript
- closure
  - closure js
  - closure template
  - closure style*
- gwt
  - compile java to js
- angular

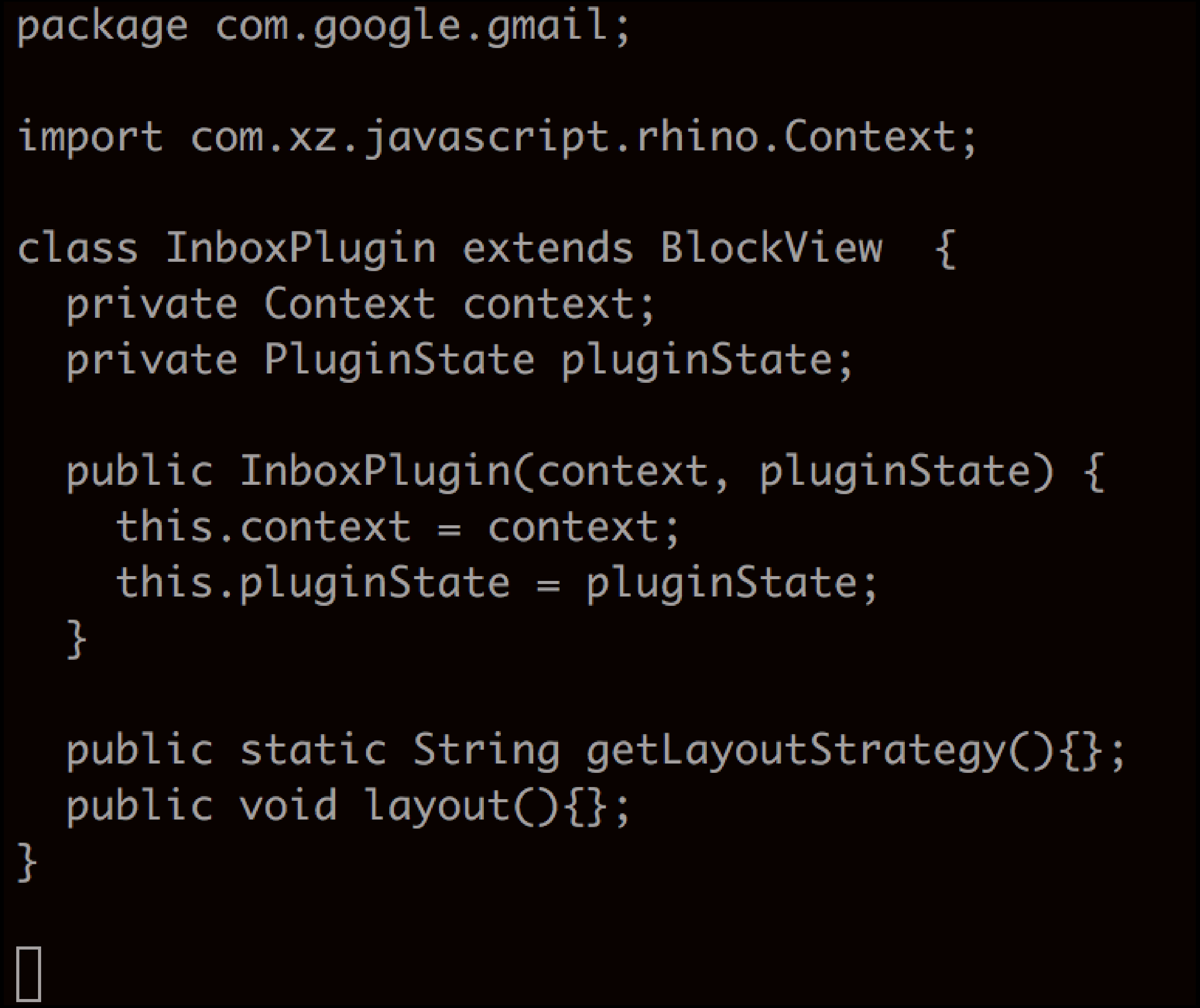
- polymer
- …

90% of js code in google looks like this



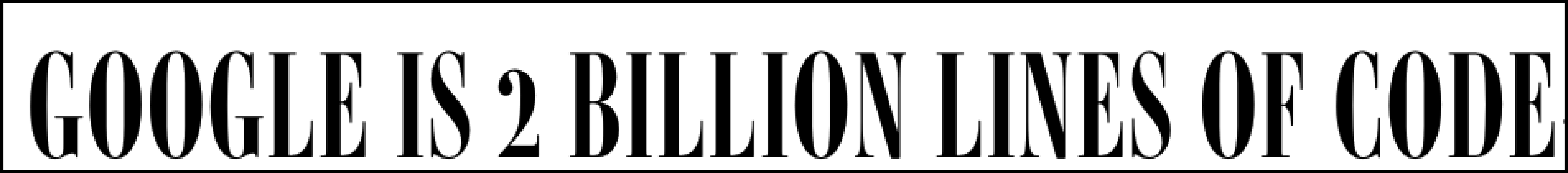
```
goog.provide('com.google.gmail.InboxPlugin');
goog.require('goog.math.Size');

/**
 * class
 * @implements {com.google.gmail.BlockView}
 */
com.google.gmail.InboxPlugin = function(context, pluginState) {
  // class member
  this.context_ = context;
  this.pluginState_ = pluginState;
};

/* static function */
com.google.gmail.InboxPlugin.getLayoutStrategy = function() {};

/* member function */
com.google.gmail.InboxPlugin.prototype.layout = function(startIndex) {};
```

Google loves Java, not JS



```
package com.google.gmail;

import com.xz.javascript.rhino.Context;

class InboxPlugin extends BlockView  {
  private Context context;
  private PluginState pluginState;

  public InboxPlugin(context, pluginState) {
    this.context = context;
    this.pluginState = pluginState;
  }

  public static String getLayoutStrategy(){};
  public void layout(){};
}
```

All code MUST look the same



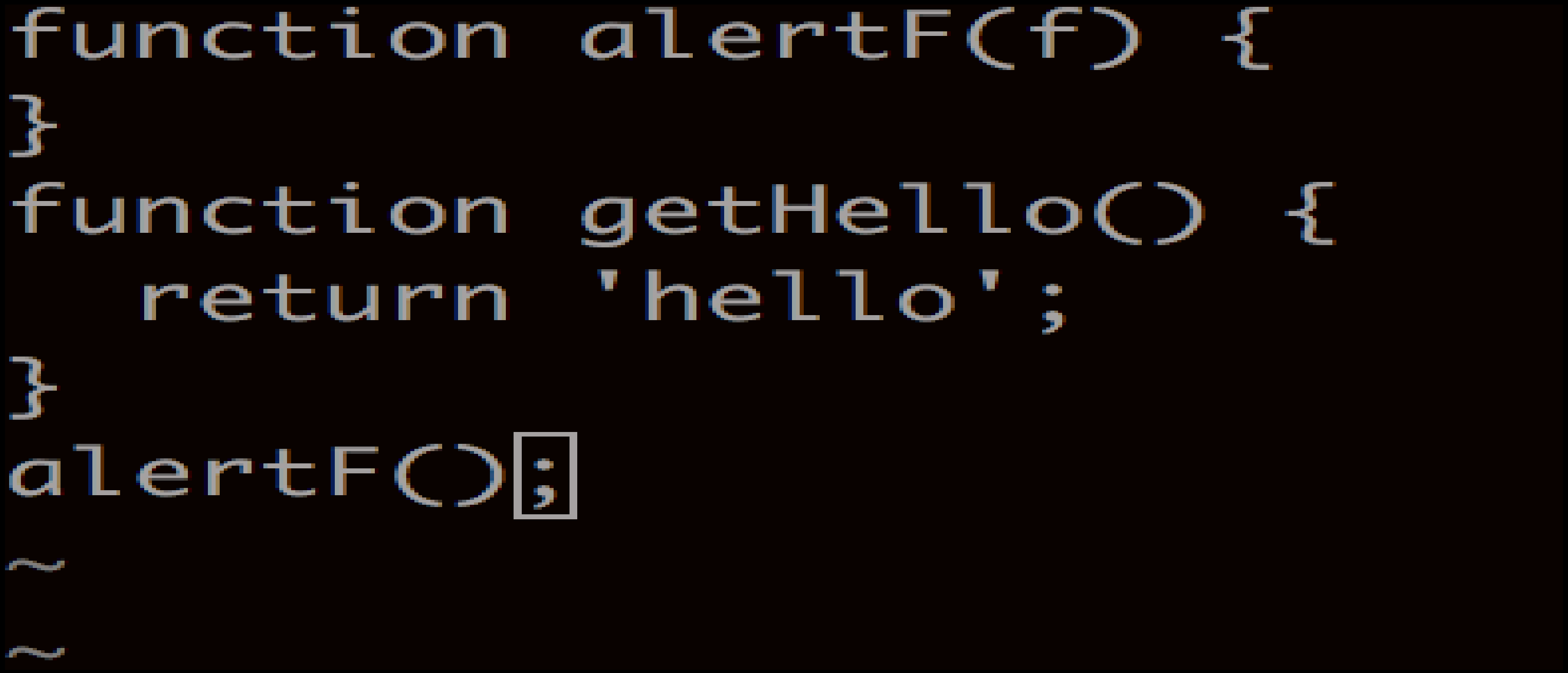
GOOGLE IS 2 BILLION LINES OF CODE

- 技术
- 工具
- 组织架构

技术层面

一些可能是比较特殊的技术

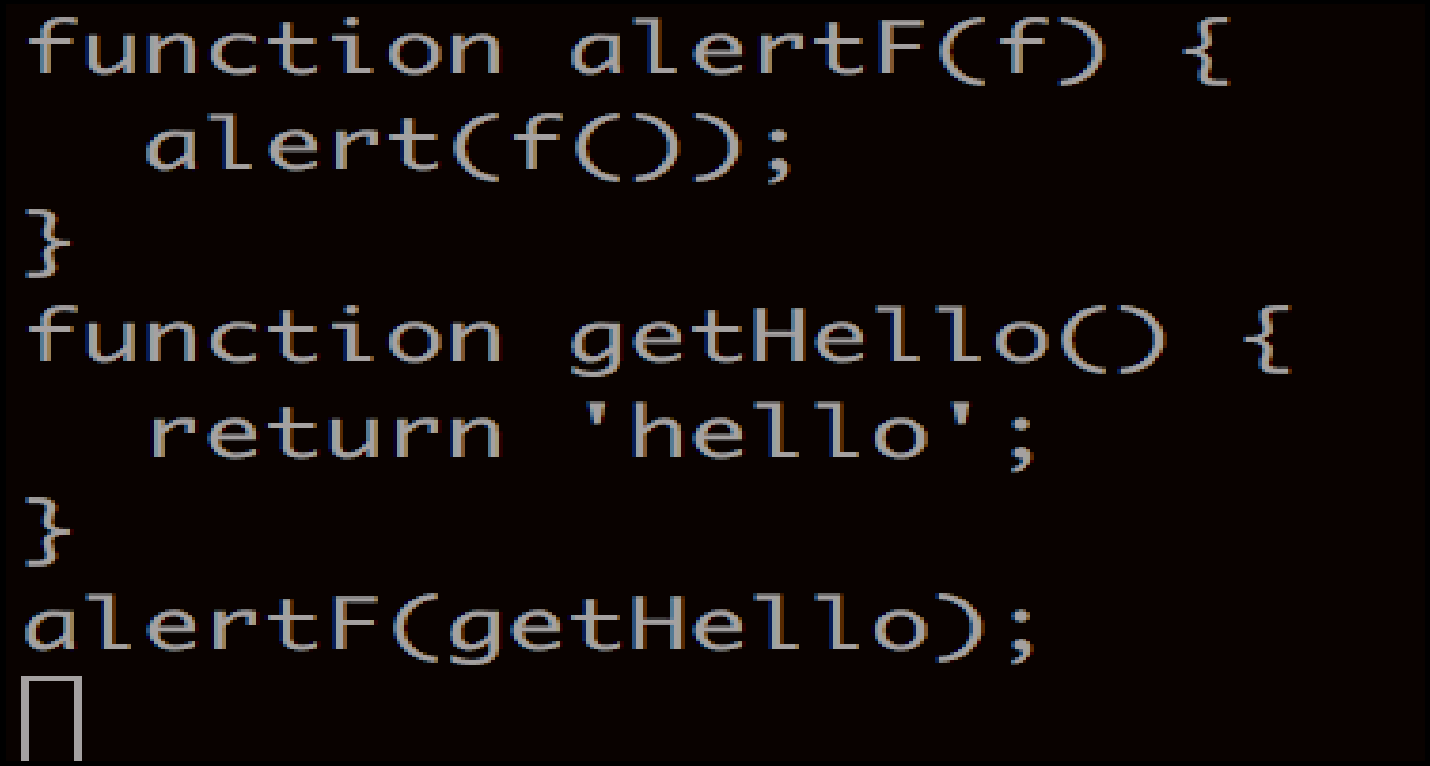
- dead code removal



```
function alertF(f) {
}
function getHello() {
   return 'hello';
}
alertF();
~
~
```

一些可能是比较特殊的技术

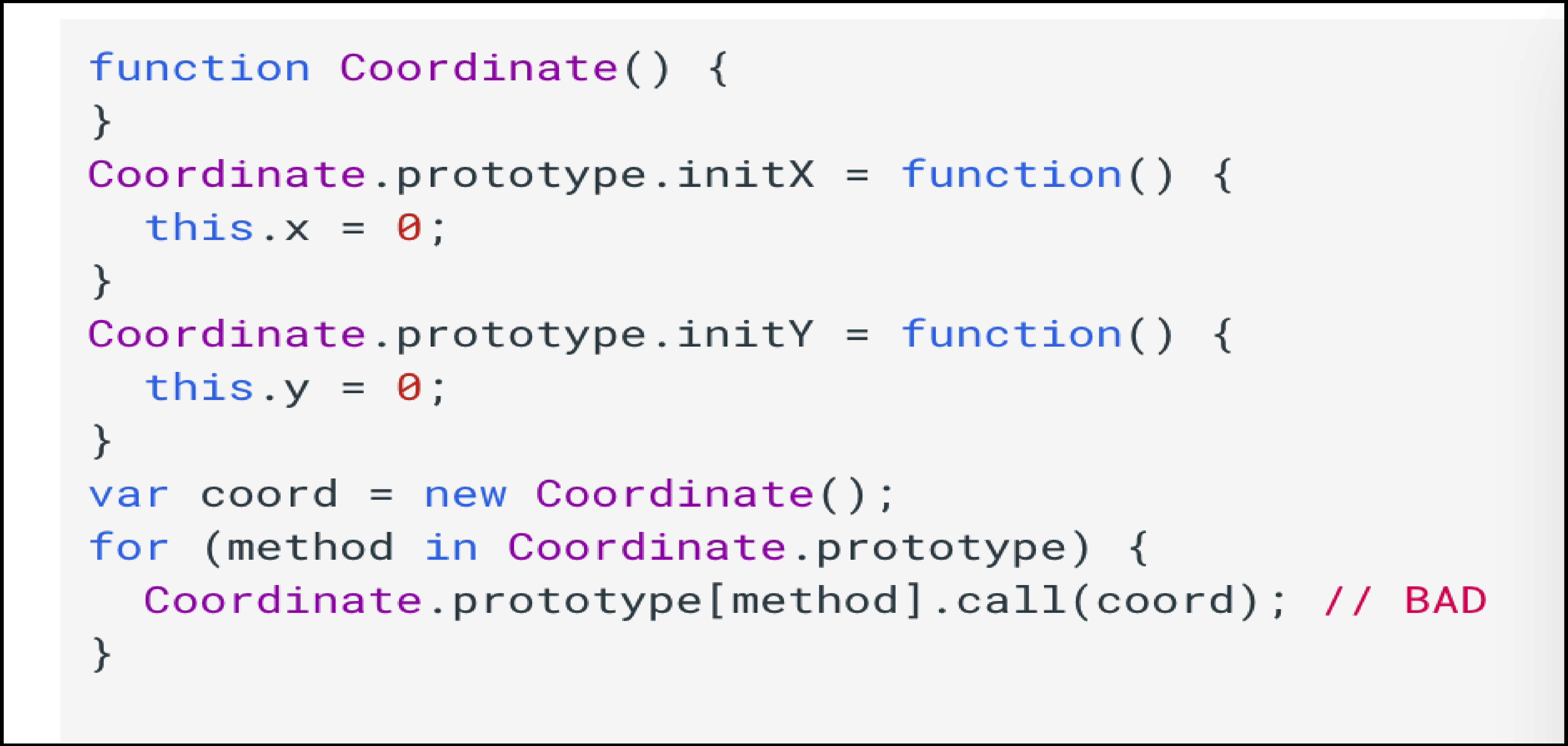
- dead code removal



```
function alertF(f) {
  alert(f());
}
function getHello() {
  return 'hello';
}
alertF(getHello);
```

一些可能是比较特殊的技术

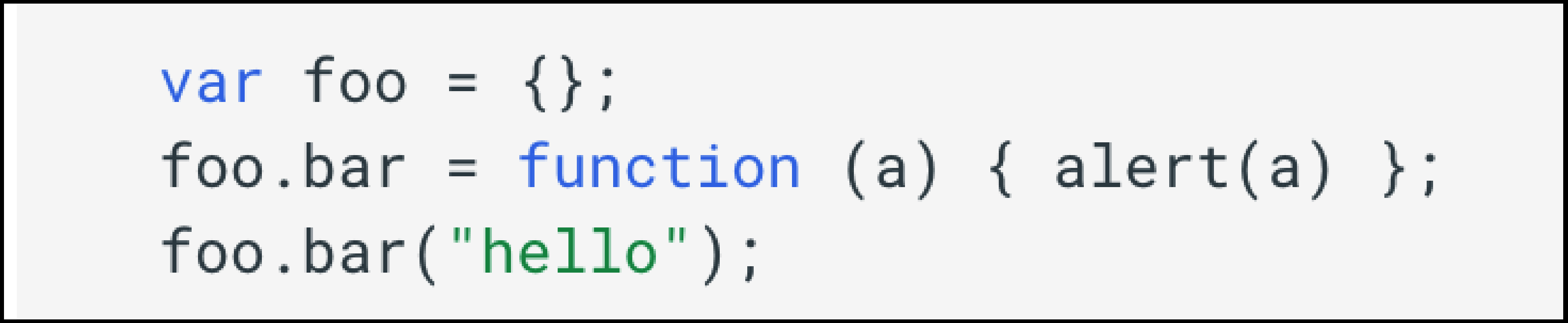
- dead code removal (bad)



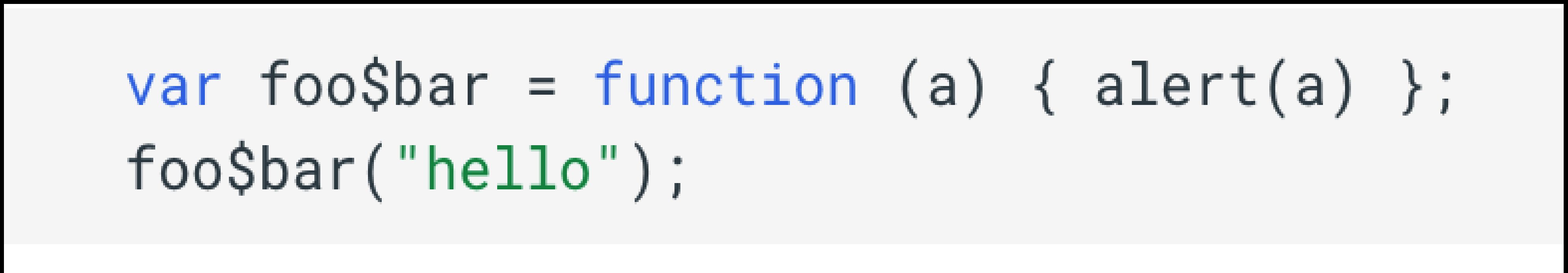
```
function Coordinate() {
}
Coordinate.prototype.initX = function() {
  this.x = 0;
}
Coordinate.prototype.initY = function() {
  this.y = 0;
}
var coord = new Coordinate();
for (method in Coordinate.prototype) {
  Coordinate.prototype[method].call(coord); // BAD
}
```

一些可能是比较特殊的技术

- collapse object properties



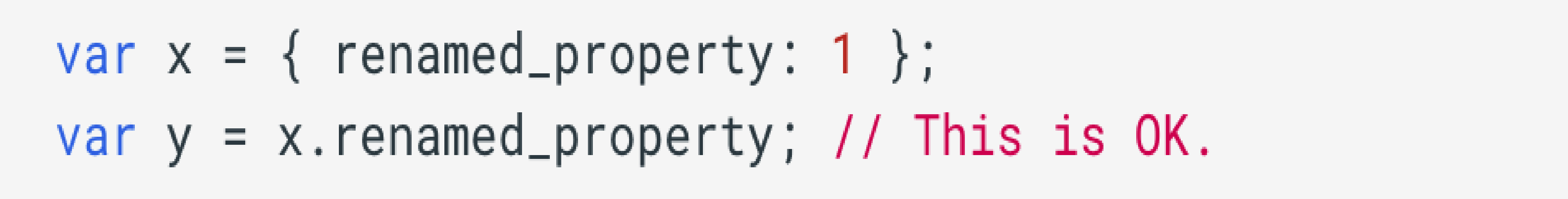
```
var foo = {};
foo.bar = function (a) { alert(a) };
foo.bar("hello");
```



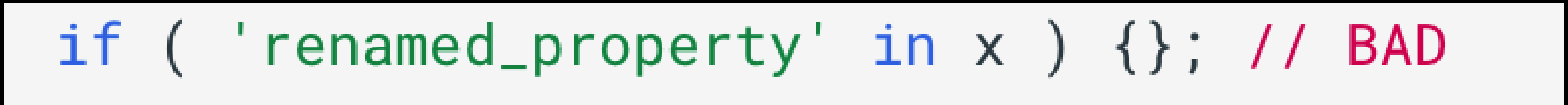
```
var foo$bar = function (a) { alert(a) };
foo$bar("hello");
```

一些可能是比较特殊的技术

- collapse object properties



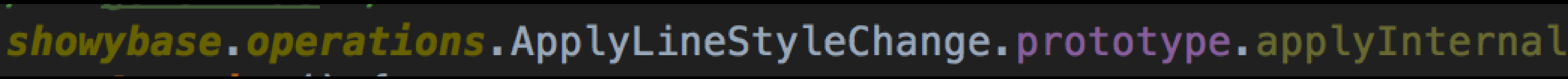
```
var x = { renamed_property: 1 };
var y = x.renamed_property; // This is OK.
```



```
if ( 'renamed_property' in x ) {}; // BAD
```

一些可能是比较特殊的技术

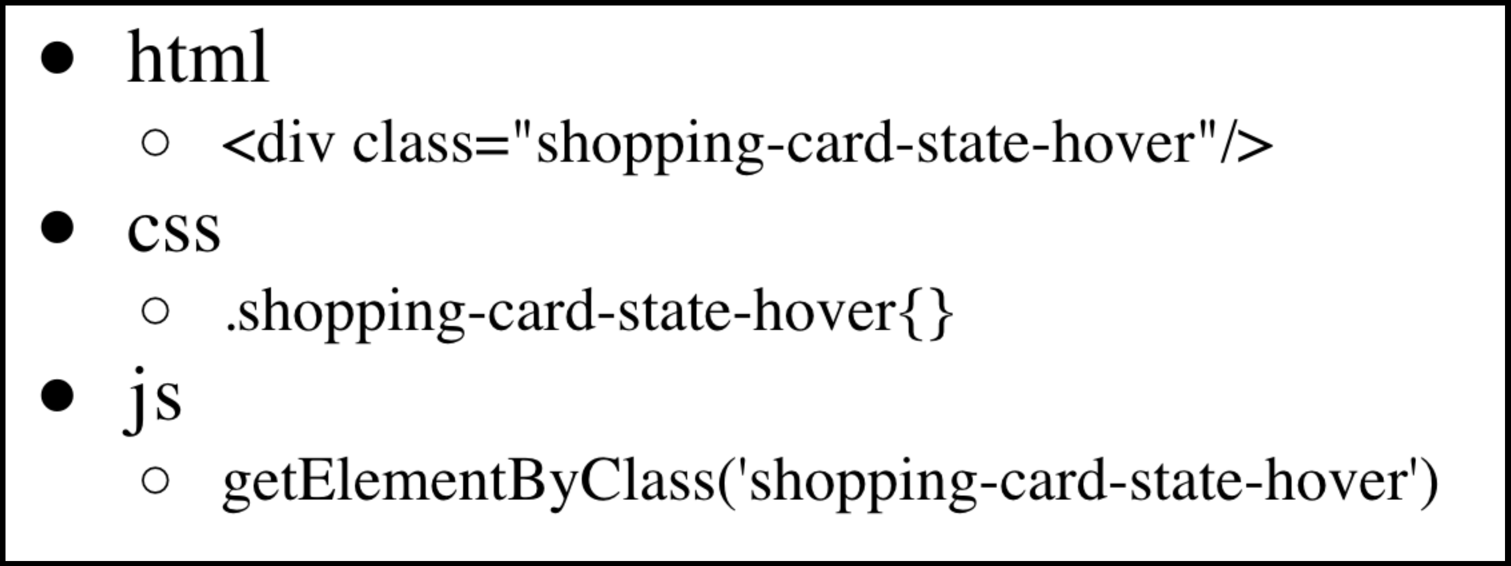
- collapse object properties



```
showybase.operations.ApplyLineStyleChange.prototype.applyInternal
```

一些可能是比较特殊的技术

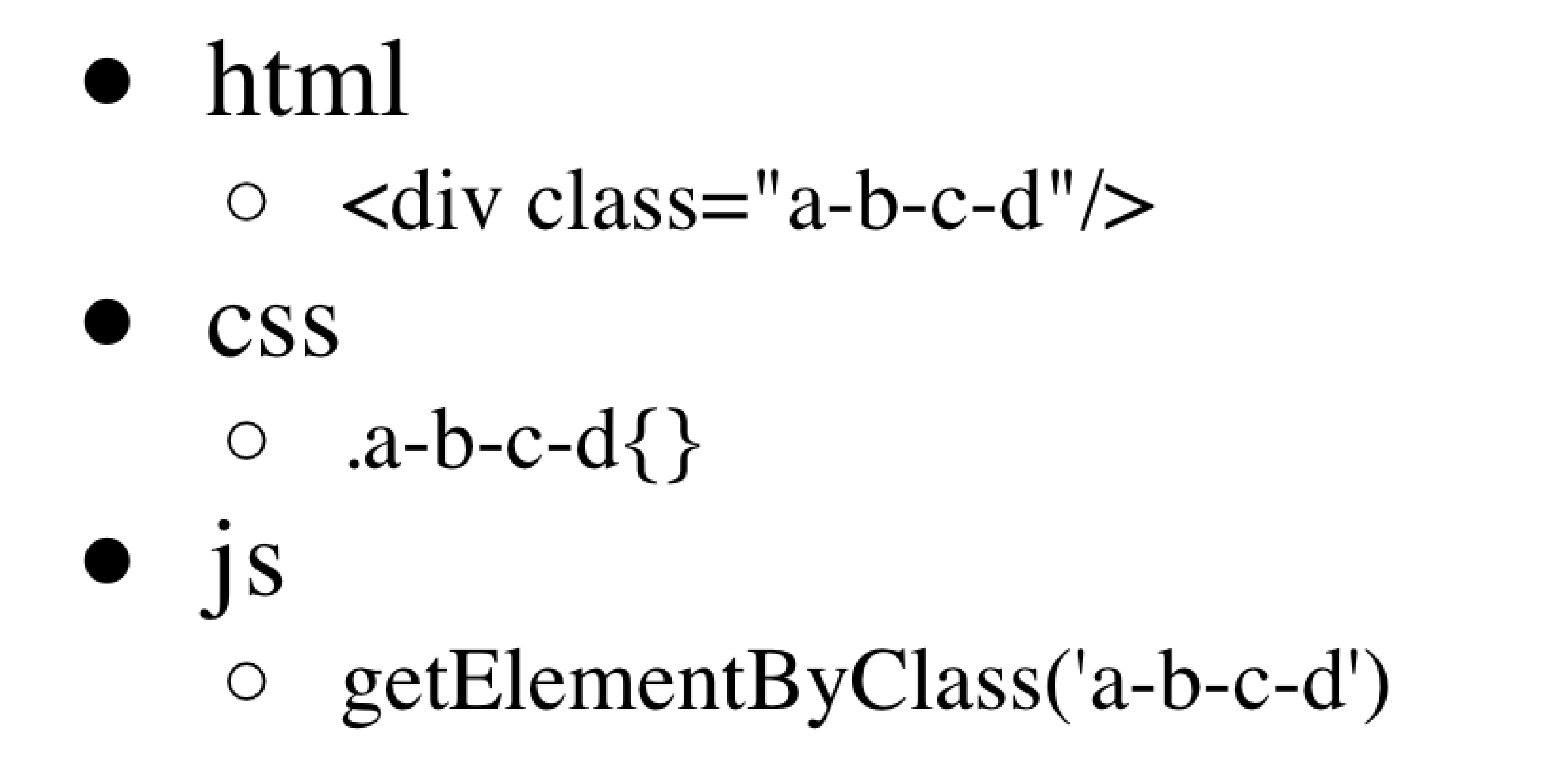
- consistent css renaming across js/css/template



- html
  - <div class="shopping-card-state-hover"/>
- css
  - .shopping-card-state-hover{}
- js
  - getElementByClass('shopping-card-state-hover')

一些可能是比较特殊的技术

- consistent css renaming across js/css/template



- html
  - <div class="a-b-c-d"/>
- css
  - .a-b-c-d{}
- js
  - getElementByClass('a-b-c-d')

一些可能是比较特殊的技术

- Google’ s react native
  - Strictly model - view - controller
  - Embed a V8 in native app
  - Reuse Model and Controller code in native app across android and ios
  - Model and Controller are much more complex than view

工具层面

Protobuf to JS

gJs lint

Js automation test

- Unit test
- Integration test
- Continuous run

JS stack trace deobfuscation

- Client catches all exceptions
- Send back to server
- Server deobfuscate the stack, automatically generates bug and assign to engineers

组织架构层面

基础设施团队

- different framework team
  - closure team
  - closure compiler team
  - gwt team
  - angular team

产品相关的前端团队

- google+ frontend infrastructure team
- news feed team
- profile team
- …

很多技术都来源于产品的实践

- Gmail —-> Closure
- Google+ —> Wiz(react/angular like js framework)

结尾

- 严格的执行这些规则需要一个非常强的JS基础架构团队
- 实际实施的过程中，可能会打折再打折
- 过度的实施这些技巧反而会降低团队的生产力