

# Chapter 1

- ① JS three pillars { scope/closure 作用域/闭包  
prototypes/objects 原型/对象  
types/coercion 类型/强制

② JS 的称呼, 因为 JavaScript 版权问题, 官方建议用 JS 或 ECMAScript 2019 (ES2019) 来指代该语言。

③ Typical paradigm-level code categories include procedural (过程式), object-oriented (面向对象), (OO/classes), and functional (FP) (函数式)。

④ backwards compatibility 向后兼容。  
Once ~~src~~ is accepted as valid JS, there will not be a future change to the language that causes that code to become invalid JS.

"我们不破坏网络" JS 是向后兼容, 而不向前兼容。  
HTML 和 CSS 相反! 无法识别的会被跳过。

⑤ 对于新的和不兼容的语法, 解决方案是使用 babel 等 transpiler 来解决, 从较新的 JS 语法版本转换为等效的旧语法。

⑥ 对于最近添加的 API 缺少, 解决方案是 polyfill, 为缺少的方法提供一个定义 (~~或~~判断)

⑦ interpreted script or a compiled program  
解释性脚本 或 编译过的程序

majority  
⑧ ES6 默认为 strict mode ? 严格模式的定义。