

```

1 // 對外擴充方法
2 const _$ImportModules = (function (G) {
3     // //////////////////////////////////////
4     //
5     // extension_1
6     //
7     // //////////////////////////////////////
8     // 資訊
9     const _extension1 = {
10         callback_uid: 1,
11         callback_key: "$_callbackGuid",
12         "_": null,
13         info: {
14             sourceScriptPath: null,
15             extension1ScriptPath: null,
16             environment: null,
17         },
18     };
19
20     const ImportModuleList = [];
21     // =====
22     // 工廠函式
23     class Factory {
24         // -----
25         // 取得 _, _extension 的路徑
26         static _getPath(_) {
27             // debugger;
28
29             const info = _extension1.info;
30             const environment = info.environment;
31
32             if (/nodejs/.test(environment)) {
33
34                 info.extension1ScriptPath = __dirname;
35
36             } else if (/browser/.test(environment)) {
37
38                 if (typeof document == 'undefined') {
39                     return;
40                 }
41
42                 let scripts = Array.from(document.querySelectorAll('script'));
43                 let script = scripts.pop();
44
45                 info.extension1ScriptPath = script.src;
46                 // -----
47                 // find scriptPath
48                 let reg = /(\\|\/)?(?:[^\w]*?(underscore|lodash)[^\w]*?)$/i;
49
50                 while ((script = scripts.pop()) != null) {
51                     let src = script.src;
52                     if (src && reg.test(src)) {
53                         info.sourceScriptPath = src;
54                         break;
55                     }
56                 }
57             }
58         }
59         // -----
60         static _link(_) {
61             Object.defineProperty(_, '$extension1', {
62                 value: _extension1,

```

```

63         enumerable: false,
64         writable: false,
65         configurable: true
66     });
67 }
68 // -----
69 // m: 注入函式
70 static importModule(m) {
71     // debugger
72     const environment = _extension1.info.environment;
73     const _ = _extension1._;
74
75     switch (environment) {
76         case 'nodejs':
77             if (_ == null) {
78                 // _尚未引入, 先將模組放置等候區
79                 ImportModuleList.push(m);
80             } else {
81                 m(_);
82             }
83             break;
84         default:
85             if (_ == null) {
86                 throw new Error('_extension1 need import _');
87             } else {
88                 m(_);
89             }
90             break;
91     }
92 }
93 // -----
94 // nodejs
95 static injectModules() {
96     const _ = _extension1._;
97
98     if (_ == null) {
99         throw new Error('no import _');
100     }
101     let m;
102
103     while ((m = ImportModuleList.shift()) != null) {
104         m(_);
105     }
106 }
107 // -----
108 static main() {
109
110     const _ = _extension1._;
111
112     if (_ == null) {
113         throw new Error('no import _');
114     }
115
116     if (._.$extension1 != null) {
117         // 避免重複
118         return;
119     }
120
121     Factory._link(_);
122
123     Factory._getPath(_);
124 }

```

```

125     }
126     // =====
127
128     // 對外引入模組的函式
129     function ImportModule(m) {
130         Factory.importModule(m);
131     };
132
133     // 引入 _ 的窗口
134     // arg: [_.path: _的路徑, _obj: _本身]
135     function nodeJs_extension(arg) {
136         let _;
137         let _path;
138
139         if (typeof arg == 'string') {
140             _ = require(path);
141         } else {
142             _ = arg;
143         }
144         // 取得 _ 在本機的位置
145         _path = require.resolve(path);
146
147         _extension1.info.sourceScriptPath = _path;
148
149         Factory.main();
150
151         Factory.injectModules();
152
153         return _;
154     }
155     // =====
156
157     (function () {
158         // debugger
159         // 環境檢測
160
161         if (typeof window != 'undefined' && typeof document != 'undefined') {
162             // browser
163
164             let environment = 'browser';
165             let _ = window._;
166
167             _extension1._ = _;
168             _extension1.info.environment = environment;
169
170             Factory.main();
171         } else if (typeof (module) != 'undefined' && typeof (module.exports) != 'undefined') {
172
173             _extension1.info.environment = 'nodejs';
174
175             // nodejs 對外窗口
176             module.exports = {
177                 extension: nodeJs_extension,
178                 importModule: ImportModule,
179             };
180
181         } else if (typeof (window) == 'undefined' && self != 'undefined' && typeof
(importScripts) == 'function') {
182             // webWorker 環境
183
184             let _ = self._;
185             _extension1.info.environment = 'worker';

```

```
186         _extension1._ = _;
187
188         Factory.main();
189     } else {
190         throw new Error('no support current system');
191     }
192 }());
193
194 // =====
195 return ImportModule;
196 })(this);
```