```
// 對外擴充方法
const _$ImportModules = (function (G) {
    // //////////////////////////////////////////////////////////////////////////
    //
    // extension_1
    //
    // //////////////////////////////////////////////////////////////////////////
    // 資訊
    const _extension1 = {
        callback_uid: 1,
        callback_key: "$_callbackGuid",
        "_": null,
        info: {
            sourceScriptPath: null,
            extension1ScriptPath: null,
            environment: null,
        },
    };

    const ImportModuleList = [];
    // ==========================================================================
    // 工廠函式
    class Factory {
        // ----------------------------
        // 取得 _, _extention 的路徑
        static _getPath(_) {
            // debugger;

            const info = _extension1.info;
            const environment = info.environment;

            if (/nodejs/.test(environment)) {

                info.extension1ScriptPath = __dirname;

            } else if (/browser/.test(environment)) {

                if (typeof document == 'undefined') {
                    return;
                }

                let scripts = Array.from(document.querySelectorAll('script'));
                let script = scripts.pop();

                info.extension1ScriptPath = script.src;
                // ----------------------------
                // find scriptPath
                let reg = /(\\|\/)?([^\/]*?(underscore|lodash)[^\/]*?)$/i;

                while ((script = scripts.pop()) != null) {
                    let src = script.src;
                    if (src && reg.test(src)) {
                        info.sourceScriptPath = src;
                        break;
                    }
                }
            }
        }
        // ----------------------------
        static _link(_) {
            Object.defineProperty(_, '$extension1', {
                value: _extension1,
                enumerable: false,
                writable: false,
                configurable: true
            });
        }
        // ----------------------------
        // m: 注入函式
        static importModule(m) {
            // debugger
            const environment = _extension1.info.environment;
            const _ = _extension1._;

            switch (environment) {
                case 'nodejs':
                    if (_ == null) {
                        // _尚未引入，先將模組放置等候區
                        ImportModuleList.push(m);
                    } else {
                        m(_);
                    }
                    break;
                default:
                    if (_ == null) {
                        throw new Error('_extension1 need import _');
                    } else {
                        m(_);
                    }
                    break;
            }
        }
        // ----------------------------
        // nodejs
        static injectModules() {
            const _ = _extension1._;

            if (_ == null) {
                throw new Error('no import _');
            }
            let m;

            while ((m = ImportModuleList.shift()) != null) {
                m(_);
            }
        }
        // ----------------------------
        static main() {

            const _ = _extension1._;

            if (_ == null) {
                throw new Error('no import _');
            }

            if (_.$extension1 != null) {
                // 避免重複
                return;
            }

            Factory._link(_);

            Factory._getPath(_);
        }
```

```
    }
    // ==========================================================================

    // 對外引入模組的函式
    function ImportModule(m) {
        Factory.importModule(m);
    };

    // 引入 _ 的窗口
    // arg: [_.path: _的路徑， _obj: _本身]
    function nodeJs_extension(arg) {
        let _;
        let _path;

        if (typeof arg == 'string') {
            _ = require(path);
        } else {
            _ = arg;
        }
        // 取得 _ 在本機的位置
        _path = require.resolve(path);

        _extension1.info.sourceScriptPath = _path;

        Factory.main();

        Factory.injectModules();

        return _;
    }
    // ==========================================================================

    (function () {
        // debugger
        // 環境檢測

        if (typeof window != 'undefined' && typeof document != 'undefined') {
            // browser

            let environment = 'browser';
            let _ = window._;

            _extension1._ = _;
            _extension1.info.environment = environment;

            Factory.main();
        } else if (typeof (module) != 'undefined' && typeof (module.exports) != 'undefined') {

            _extension1.info.environment = 'nodejs';

            // nodejs 對外窗口
            module.exports = {
                extension: nodeJs_extension,
                importModule: ImportModule,
            };

        } else if (typeof (window) == 'undefined' && self != 'undefined' && typeof (importScripts) == 'function') {
            // webWorker 環境

            let _ = self._;
            _extension1.info.environment = 'worker';
            _extension1._ = _;

            Factory.main();
        } else {
            throw new Error('no support current system');
        }
    }());

    // ==========================================================================
    return ImportModule;
})(this);
```