

04 /

자바스크립트 모듈 시스템

CJS (CommonJS)

AMD (Asynchronous Module Definition)

UMD (Universal Module Definition)

ESM (ECMAScript Module)

CJS(CommonJS) - 특징: 동기

```
// 아래와 같이 require을 통해 package/lib 모듈을 변수에 담을 수 있습니다.  
var lib = require('package/lib');  
  
// 가져온 모듈을 아래와 같이 사용할 수 있습니다.  
function foo () {  
  lib.log('hello world!');  
}  
  
// foo 함수를 다른 파일에서 사용할 수 있도록, 다른 모듈로 추출될 수 있습니다.  
exports.foobar = foo;
```

브라우저

curl.js, SproutCore, PINF

서버사이드

Node.js, Narwhal, Persevere, Wakanda

AMD(Asynchronous Module Definition) - 특징: 비동기

```
// 종속성을 갖는 모듈인 'package/lib'를 모듈 선언부의 첫 번째 파라미터에 넣으면,  
// 'package/lib'은 콜백 함수의 lib 파라미터 안에 담깁니다.  
define(['package/lib'], function (lib) {  
  
    // 로드된 종속 모듈을 아래와 같이 사용할 수 있습니다.  
    function foo () {  
        lib.log('hello world!');  
    }  
  
    // 생성된 foo 함수는 리턴을 통해 foobar라는 이름의 다른 모듈로 추출될 수 있습니다.  
    return {  
        foobar: foo  
    };  
});
```

브라우저

RequireJS, curl.js, bdLoad, Yabble, PINF

서버사이드

RequireJS, PINF

UMD(Universal Module Definition) - 특징: CJS, AMD 모두 호환

```
(function (root, factory) {  
  if (typeof define === 'function' && define.amd) {  
    // AMD  
    define(['exports', 'b'], factory);  
  } else if (typeof exports === 'object' && typeof exports.nodeName !== 'string') {  
    // CommonJS  
    factory(exports, require('b'));  
  } else {  
    // Browser globals  
    factory((root.commonJsStrict = {}), root.b);  
  }  
})(this, function (exports, b) {  
  //use b in some fashion.  
  
  // attach properties to the exports object to define  
  // the exported module properties.  
  exports.action = function () {};  
}));
```

04 / 자바스크립트 모듈 시스템 - ESM

ESM(ECMAScript Module) - 특징: ES 공식 모듈

```
import lib from 'package/lib';

function foo () {
  lib.log('hello world!');
}

export {
  foobar: foo
};
```

**THANK
YOU**