This example demonstrates how to build a library with webpack that has dependencies on other libraries which should not be included in the compiled version.

We use the `libraryTarget: "umd"` option to build a UMD module that is consumable in CommonJS, AMD and with script tags. We don't specify the `library` option so the library is exported to the root namespace.

We use the `externals` option to define dependencies that should be resolved in the target environment.

In the simple case we just need to specify a string (`"add"`). Then it's resolved as `"add"` module in CommonJS and AMD, and as global `add` when used with the script tag.

In the complex case we specify different values for each environment:

environment	config value	resolved as
CommonJS (strict)	`["./math", "subtract"]`	`require("./math").subtract`
CommonJS (node.js)	`"./subtract"`	`require("./subtract")`
AMD	`"subtract"`	`define(["subtract"], ...)`
script tag	`"subtract"`	`this.subtract`

example.js

```
var add = require("add");
var subtract = require("subtract");

exports.exampleValue = subtract(add(42, 2), 2);
```

webpack.config.js

```
module.exports = {
    // mode: "development || "production",
    output: {
        libraryTarget: "umd"
    },
    externals: [
        "add",
        {
            subtract: {
                root: "subtract",
                commonjs2: "./subtract",
                commonjs: ["./math", "subtract"],
                amd: "subtract"
            }
        }
    ]
};
```

dist/output.js

```
(function webpackUniversalModuleDefinition(root, factory) {
    if(typeof exports === 'object' && typeof module === 'object')
        module.exports = factory(require("add"), require("./subtract"));
    else if(typeof define === 'function' && define.amd)
        define(["add", "subtract"], factory);
    else {
        var a = typeof exports === 'object' ? factory(require("add"),
require("./math")["subtract"]) : factory(root["add"], root["subtract"]);
        for(var i in a) (typeof exports === 'object' ? exports : root)[i] = a[i];
    }
})(self, function(__WEBPACK_EXTERNAL_MODULE__1__, __WEBPACK_EXTERNAL_MODULE__2__) {
return /******/ (() => { // webpackBootstrap
/******/        var __webpack_modules__ = ([
/* 0 */,
/* 1 */
/*!**********************!*\
  !*** external "add" ***!
  \**********************/
/*! dynamic exports */
/*! exports [maybe provided (runtime-defined)] [no usage info] */
/*! runtime requirements: module */
/***/ ((module) => {

"use strict";
module.exports = __WEBPACK_EXTERNAL_MODULE__1__;

/***/ }),
/* 2 */
/*!*****************************************************************************************

  !*** external {"root":"subtract","commonjs2":"./subtract","commonjs":
["./math","subtract"],"amd":"subtract"} ***!

\*****************************************************************************************

/*! dynamic exports */
/*! exports [maybe provided (runtime-defined)] [no usage info] */
/*! runtime requirements: module */
/***/ ((module) => {

"use strict";
module.exports = __WEBPACK_EXTERNAL_MODULE__2__;

/***/ })
/******/        ]);
```

▶ /* webpack runtime code */

```
var __webpack_exports__ = {};
// This entry need to be wrapped in an IIFE because it need to be isolated against
other modules in the chunk.
(() => {
var exports = __webpack_exports__;
/*!*********************!*\
  !*** ./example.js ***!
  \*********************/
/*! default exports */
/*! export exampleValue [provided] [maybe used in main (runtime-defined)] [usage
prevents renaming] */
/*! other exports [not provided] [maybe used in main (runtime-defined)] */
/*! runtime requirements: __webpack_exports__, __webpack_require__ */
var add = __webpack_require__(/*! add */ 1);
var subtract = __webpack_require__(/*! subtract */ 2);

exports.exampleValue = subtract(add(42, 2), 2);
})();

/******/ 	return __webpack_exports__;
/******/ })()
;
});
```

Info

Unoptimized

```
asset output.js 3.28 KiB [emitted] (name: main)
chunk (runtime: main) output.js (main) 194 bytes [entry] [rendered]
  > ./example.js main
  dependent modules 84 bytes [dependent] 2 modules
  ./example.js 110 bytes [built] [code generated]
    [exports: exampleValue]
    [used exports unknown]
    entry ./example.js main
    used as library export
webpack 5.51.1 compiled successfully
```

Production mode

```
asset output.js 679 bytes [emitted] [minimized] (name: main)
chunk (runtime: main) output.js (main) 194 bytes [entry] [rendered]
  > ./example.js main
  dependent modules 84 bytes [dependent] 2 modules
  ./example.js 110 bytes [built] [code generated]
    [exports: exampleValue]
    entry ./example.js main
```

```
    used as library export
webpack 5.51.1 compiled successfully
```