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

{{example.js}}

webpack.config.js

{{webpack.config.js}}

dist/output.js

{{dist/output.js}}

Info

Unoptimized

{{stdout}}

Production mode

{{production:stdout}}