ECMASCRIPT MODULES

introduction
enabling
package.json "type" field
package scope & file extensions.
--input-type flag.

packages.
package entry points.
  main entry point export
  subpath exports
  package exports fallbacks
  exports sugar
  conditional exports
  nested conditions
  self-referencing a package using its name

dual commonjs / es module packages
  dual package hazard
  writing dual packages while avoiding or minimizing hazards.
    approach #1 use an ES module wrapper
    approach #2 isolate state. → SLG

import specifiers.
terminology
data: imports
import.meta.

differences between ES modules and commonJS.
- mandatory file extensions
- no NODE_PATH
- no require, exports, module.exports, __filename, __dirname
- no require.resolve
- no require.extensions
- no require.cache
- URL-based paths.

interoperability with commonJS.
- require
- import statements
- import expressions

CommonJS, JSON, and native modules.
builtin modules
experimental JSON modules
experimental WASM modules.
experimental top-level await
experimental loaders

hooks.
- resolve hook
- getFormat hook
- getSource hook
- transformSource hook
- getGlobalPreloadCode hook
- dynamicInstantiate hook

example
- HTTPS loader
- transpiler loader

resolution algorithm
features
resolver algorithm
customizing ESM specifier resolution algorithm.

{ import tinyurl like unpkg, SHA sum }.