npm-init

create a package.json file

SYNOPSIS

```
npm init [--force|-f|--yes|-y|--scope]
npm init <@scope> (same as `npx <@scope>/create`)
npm init [<@scope>/]<name> (same as `npx [<@scope>/]create-<name>`)
```

EXAMPLES

Create a new React-based project using **`create-react-app`**:

```
$ npm init react-app ./my-react-app
```

Create a new **`esm`**-compatible package using **`create-esm`**:

```
$ mkdir my-esm-lib && cd my-esm-lib
$ npm init esm --yes
```

Generate a plain old package.json using legacy init:

```
$ mkdir my-npm-pkg && cd my-npm-pkg
$ git init
$ npm init
```

Generate it without having it ask any questions:

```
$ npm init -y
```

DESCRIPTION

`npm init <initializer>` can be used to set up a new or existing npm package.

`initializer` in this case is an npm package named **`create-<initializer>`**, which will be installed by **`npx`**, and then have its main bin executed – presumably creating or updating **`package.json`** and running any other initialization-related operations.

The init command is transformed to a corresponding `npx` operation as follows:

- `npm init foo -> npx create-foo`
- `npm init @usr/foo -> npx @usr/create-foo`
- `npm init @usr -> npx @usr/create`

Any additional options will be passed directly to the command, so `npm init foo --hello` will map to `npx create-foo --hello`.

If the initializer is omitted (by just calling `npm init`), init will fall back to legacy init behavior. It will ask you a bunch of questions, and then write a package.json for you. It will attempt to make reasonable guesses based on existing fields, dependencies, and options selected. It is strictly additive, so it will keep any fields and values that were already set. You can also use `-y` / `--yes` to skip the questionnaire altogether. If you pass `--scope`, it will create a scoped package.