lastRun()

Retrieves the last time a task was successfully completed during the current running process. Most useful on subsequent task runs while a watcher is running.

When combined with src(), enables incremental builds to speed up execution times by skipping files that haven't changed since the last successful task completion.

Usage

```
const { src, dest, lastRun, watch } = require('gulp');
const imagemin = require('gulp-imagemin');

function images() {
  return src('src/images/**/*.jpg', { since: lastRun(images) })
    .pipe(imagemin())
    .pipe(dest('build/img/'));
}

exports.default = function() {
  watch('src/images/**/*.jpg', images);
};
```

Signature

```
lastRun(task, [precision])
```

Parameters

parameter	type	note
task	function	The task function or the string alias of a
(required)	string	registered task.
precision	number	Default: 1000 on Node v0.10, 0 on Node v0.12+. Detailed in Timestamp precision section below.

Returns

A timestamp (in milliseconds), matching the last completion time of the task. If the task has not been run or has failed, returns undefined.

To avoid an invalid state being cached, the returned value will be undefined if a task errors.

Errors

When called with a value other than a string or function, throws an error with the message, "Only functions can check lastRun".
When called on a non-extensible function and Node is missing WeakMap, throws an error with the message, "Only extensible functions can check lastRun".

Timestamp precision

While there are sensible defaults for the precision of timestamps, they can be rounded using the precision parameter. Useful if your file system or Node version has a lossy precision on file time attributes.
lastRun(someTask) returns 1426000001111
lastRun(someTask, 100) returns 1426000001100
lastRun(someTask, 1000) returns 1426000001000
A file's mtime stat precision may vary depending on the node version and/or the file system used.

platform	precision
Node v0.10	1000ms
Node v0.12+	1ms
FAT32 file system	2000ms
HFS+ or Ext3 file systems	1000ms
NTFS using Node v0.10	1s
NTFS using Node 0.12+	100ms
Ext4 using Node v0.10	1000ms
Ext4 using Node 0.12+	1ms